EECE 5554 Lab_2 Report

Shaoshu Xu

Part 1: IMU stationary data

1. **Plot time series of 3 accelerometers, 3 angular rate gyros, 3- axis magnetometers**
2. **Figure out the noise characteristics [*lab2_part1.m*]**

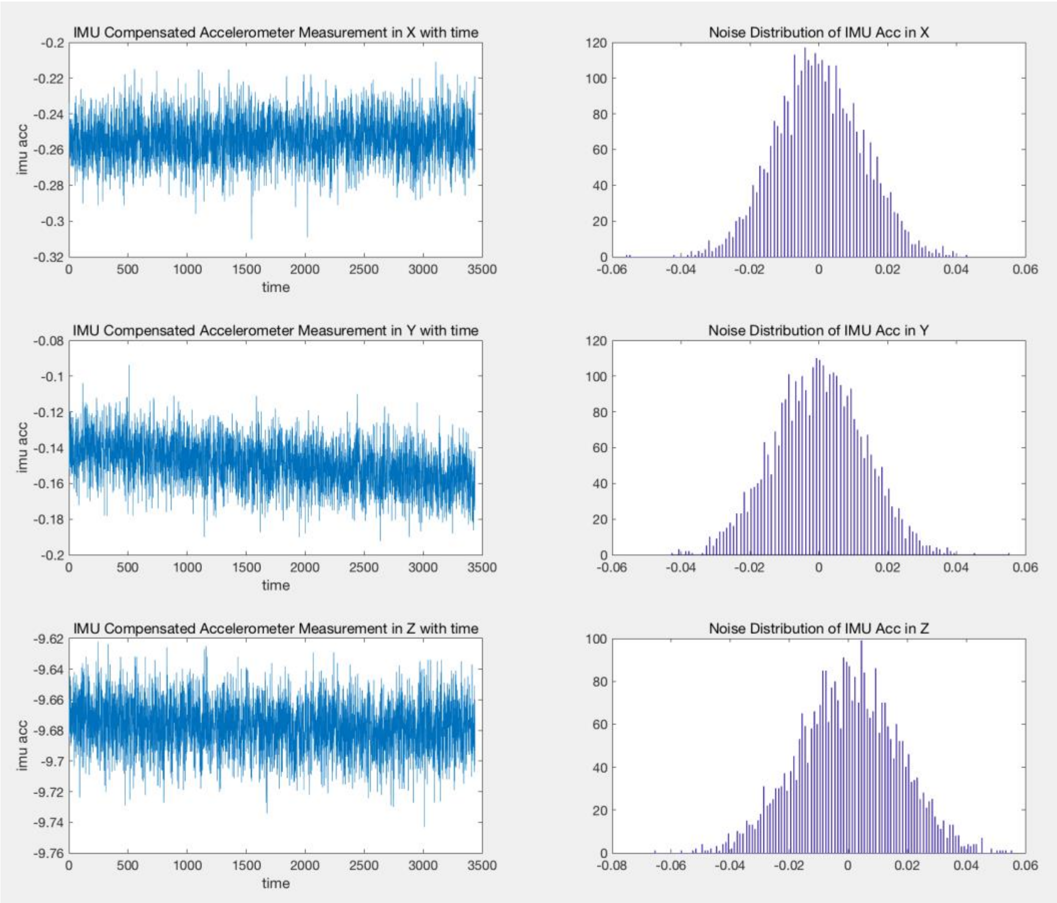

Figure 1. stationary IMU accelerometer measurement in x, y and z-axis

Figure 1. shows the stationary IMU accelerometer measurement and noise distribution in x, y and z-axis. The noise was calculated by subtracting the mean value of each measurement and we can clearly see that **the noise is a shape of Gaussian distribution**.

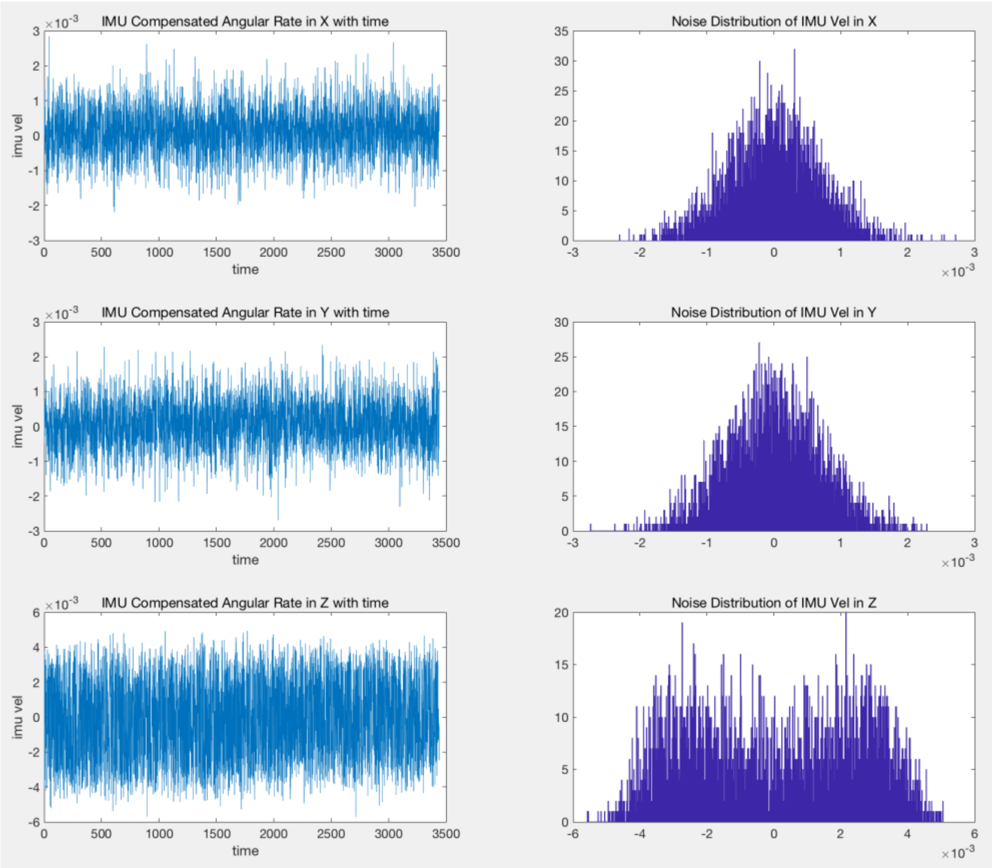

Figure 2. stationary IMU angular rate measurement in x, y and z-axis

Figure 2. shows the stationary IMU gyroscope measurement (angular rate) and noise distribution in x, y and z-axis. The noise was calculated by subtracting the mean value of each measurement. We can clearly see that **in x and y-axis the noise is a Gaussian distribution**. For **the noise in z-axis, it appears a bimodal distribution** [1].

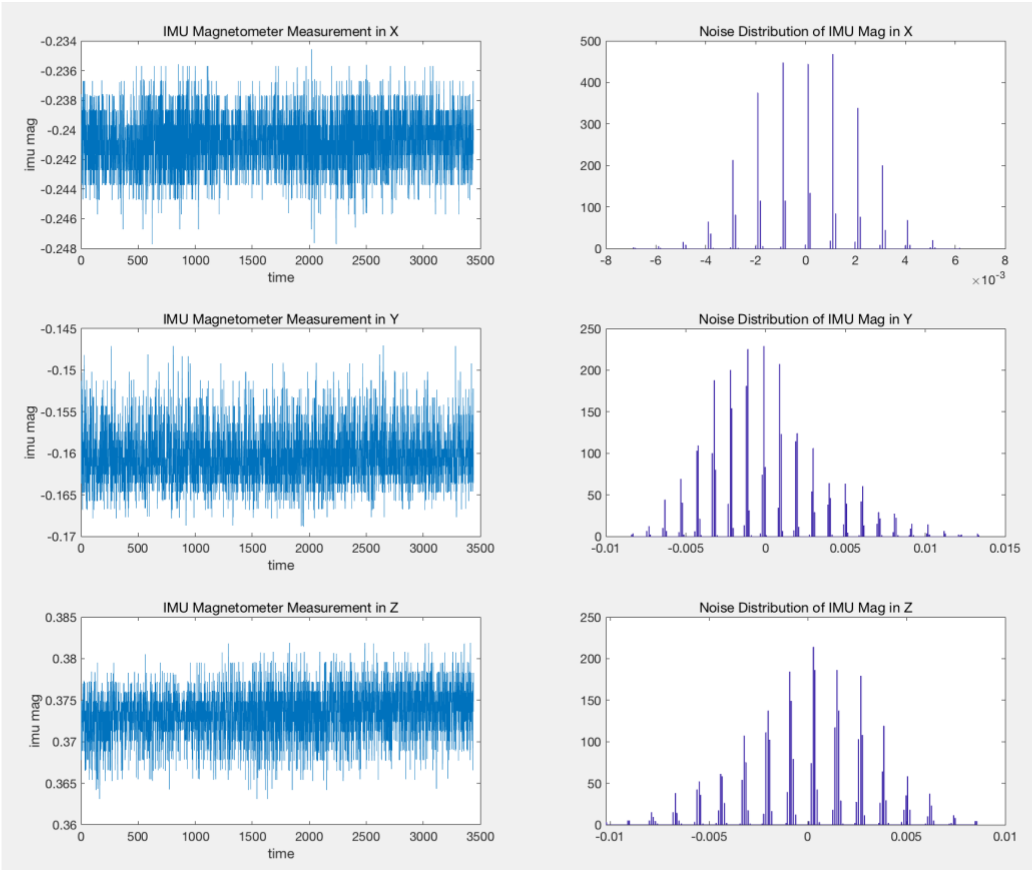

Figure 3. stationary IMU magnetometer measurement in x, y and z-axis

Figure 3. shows the stationary IMU magnetometer measurement and noise distribution in x, y and z-axis. The noise was calculated by subtracting the mean value of each measurement. We can clearly see that **in x and z-axis the noise is a Gaussian distribution**. For the noise **in y-axis, it appears skewness and could be a log-normal distribution** [2].

Part 3: Analysis the driving data collected in Part 2

3.1 Estimate the heading (yaw) - Magnetometer Calibration [*lab2_part3_1.m*]

1. Correct magnetometer readings for "hard-iron" and "soft-iron" effects

Hard-iron correction: subtract the x and y offsets;

Soft-iron correction: rotating to align the major axis of the ellipse with the reference frame X axis;

scaling the major axis such that the ellipse is converted to a circle;

rotating the circle back to the original position;

2. Plots the magnetometer data before and after the correction

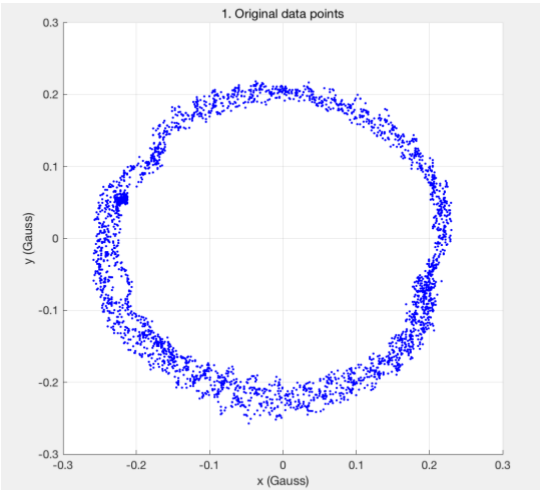

Figure 4. original magnetometer data

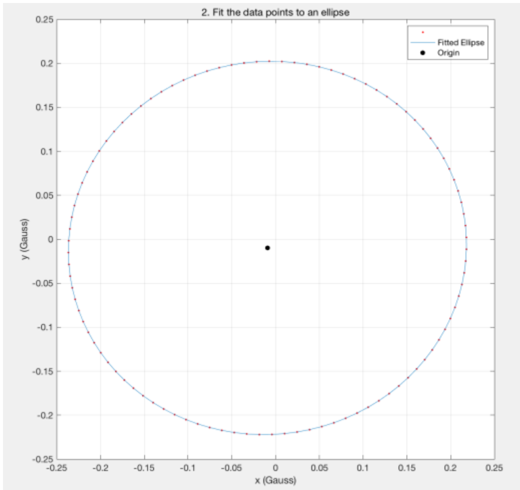

Figure 5. fit the data to an ellipse

From figure 4, we can observe that there are slight hard-iron and soft-iron effects in the original magnetometer data. **The data is fitted to an ellipse** [3] to get a more accurate offset value, major axis, minor axis, rotation degree and scaling factor. The fitted ellipse was shown in figure 5.

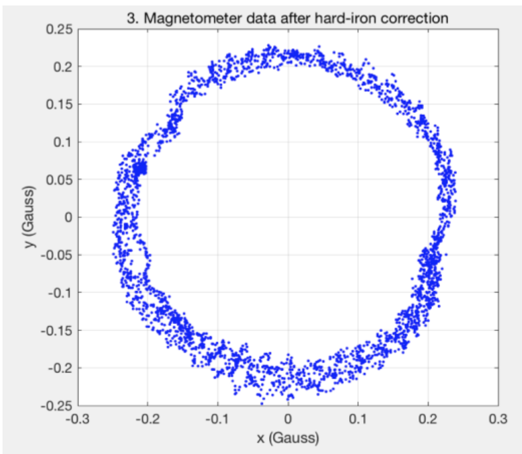

Figure 6. the magnetometer data after hard-iron correction

Figure 6 shows the magnetometer data after hard-iron correction, we can see that the x and y-axis offsets have been removed.

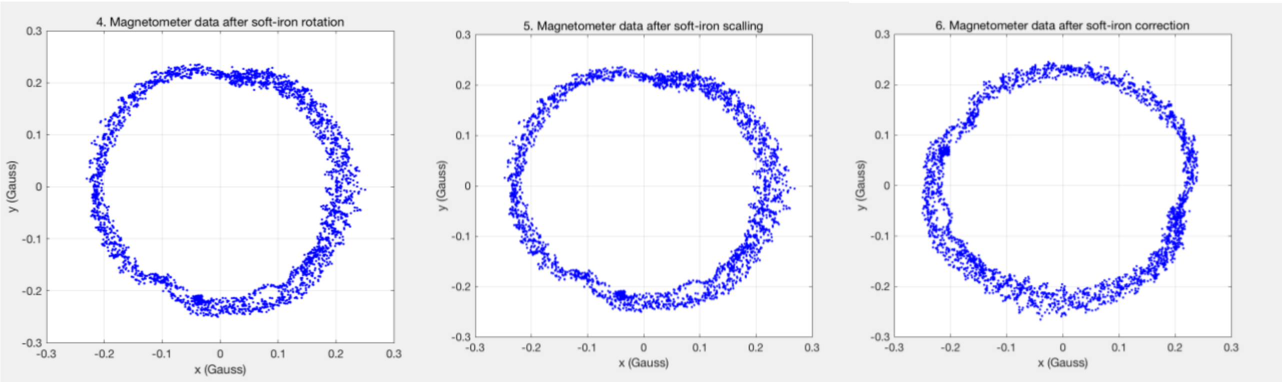

Figure 7. the magnetometer data after soft-iron correction

Figure 7 shows the results after soft-iron correction process: rotation, scaling and final correction result. In the third plot of figure 7, we can see that the magnetometer data has been well corrected.

3. Calculate the yaw angle from the corrected magnetometer readings

4. Integrate the yaw rate sensor to get yaw angle

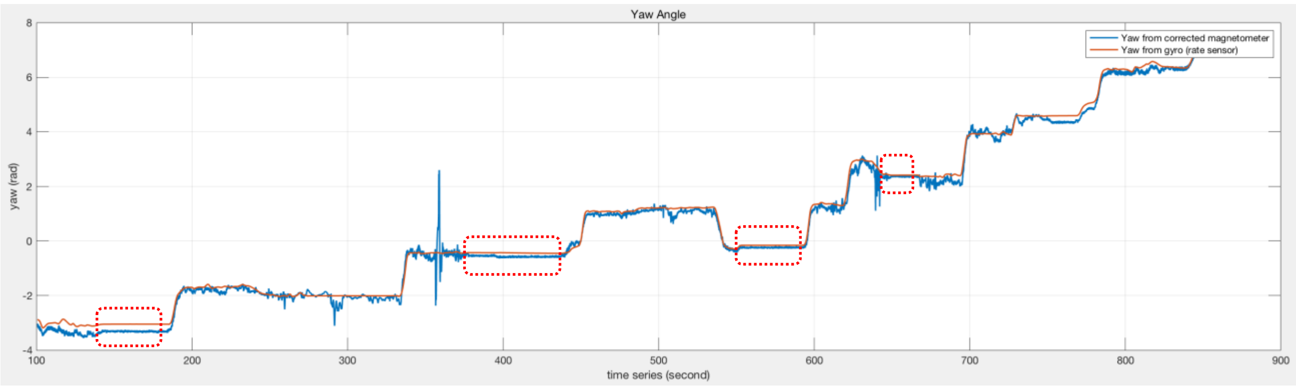

Figure 8. Yaw angle from magnetometer and gyroscope

In figure 8, we can notice that **the overall yaw angle trend calculated from the above two methods are consistent**.

However, **the yaw angle from corrected magnetometer readings (blue) is still very noisy**. There is much high-frequency component noise during the whole data. Also, at period 150-180 sec, 380-440 sec, 550-590 sec and 640-660 sec (as shown in the red box) when my car was stopped to waiting for the red light, we can see that the yaw from **magnetometer readings gives a good indicator of orientation in this static conditions and almost no noises**.

In comparison, **the yaw angle from integrated gyroscope measurement (orange) is smoother and provides a good indicator of yaw angle in dynamic conditions**.

5. Complementary Filter

Passing the magnetometer signals through a low-pass filter and the gyroscope signals through a high-pass filter and combine them to get the final yaw angle. The frequency response of the high-pass and low-pass filters should add up to 1 at all frequencies. Because *852sec / (852+0.025) = 0.99997*, and after several tests I'm using *hpf = 0.999* and *lpf = 1 – hpf = 0.001*.

6. Compare the result to the yaw angle computed by the IMU

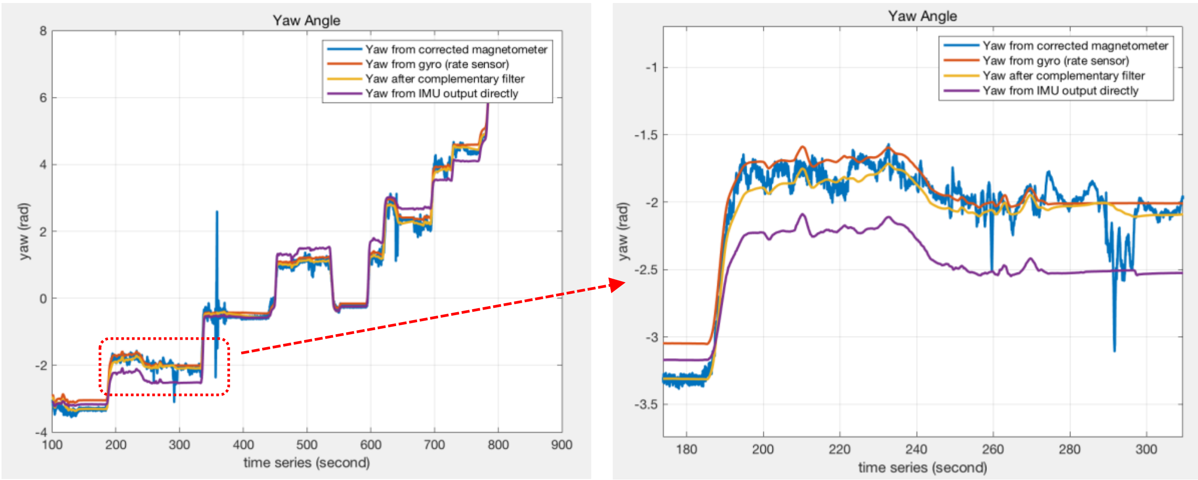

Figure 9. Yaw angle from all sources: magnetometer, gyroscope, filter, and IMU

From figure 9, it's easily seen that **the filtered yaw (yellow) follows the gyroscope (orange) for fast changes, but keeps following the mean value of the magnetometer (blue) for slower changes**. **The filtered yaw angle is not noisy like magnetometer's and does not drift away either**.

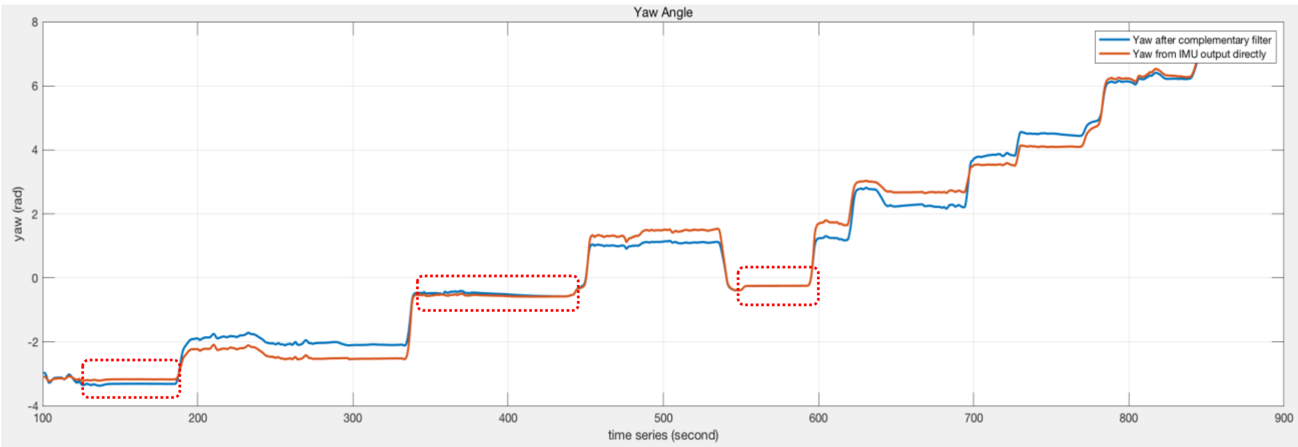

Figure 10. Yaw angle from complementary filter and IMU direct output

Figure 10 shows the yaw angle from the complementary filter and the angle computed by IMU, **the overall trend is very similar, except for some offsets**. Looking back to the period 150-180 sec, 380-440 sec and 550-590 sec (as shown in the red box) **when my car was stopped to waiting for the**

red light, we can observe that the values are very consistent. It may because the measurement values for both magnetometer and gyroscope are both very accurate during these moments.

3.2 Estimate the forward velocity [*lab2_part3_2.m*]

1. Integrate the forward acceleration to estimate the forward velocity

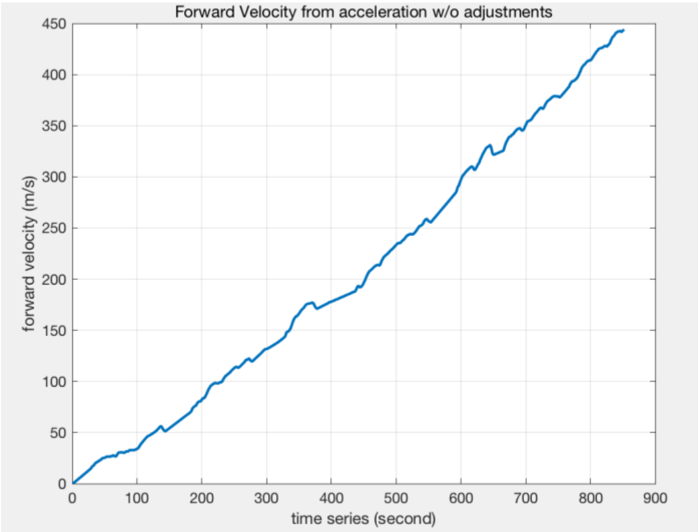

Figure 11. estimated velocity by integrating the acceleration

2. An estimate velocity from GPS

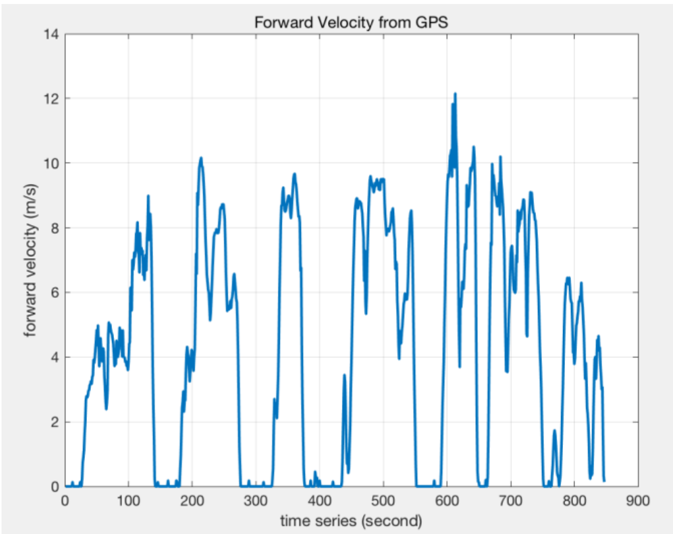

Figure 12. estimated velocity from GPS

3. Plot both velocity estimates and make observations

Figure 11 shows the forward velocity by integrating the forward acceleration. **Due to the drift and noises, there is a huge accumulative error between the estimated velocity and the true velocity**. Further adjustments for the accelerometer measurement needed.

Figure 12 shows the forward velocity calculated by GPS latitude & longitude information. We will use this GPS velocity as ground truth in the following analysis. Notice that:

Length in meters of 1° of latitude = 111.32 km

*Length in meters of 1° of longitude = 111.32 km * cos(latitude) = 82.2965 km*

4. Adjust the acceleration measurements and provide the rationale

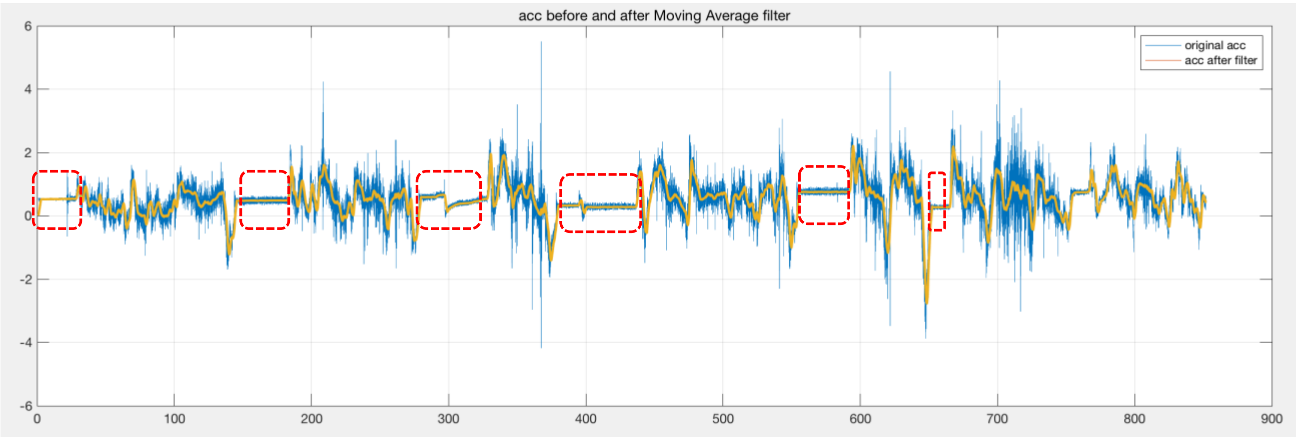

Figure 13. original acceleration and filtered acceleration

In figure 13, the blue line shows the original acceleration from IMU without any adjustments. It's clearly seen that there are many high-frequency components and **the original measurement is very noisy. After applying a low-pass filter through the moving average method, the values get smoother as the blue line shows.**

After closer observation of figure 13, I notice that there are some small constant value periods in the plots, e.g. at period 140-180 sec, 180-300 sec, 380-430 sec, 550-580 sec and so on (the red box). By comparing with the estimated velocity from GPS in figure 12, **the velocity and acceleration during these periods should both be zero.** However, due to some errors from the accelerometer, **it doesn't return zero readings and provides the wrong measurements continuously.** It probably because **the road is pitched**, so the accelerometer still has readings even though when the car stops.

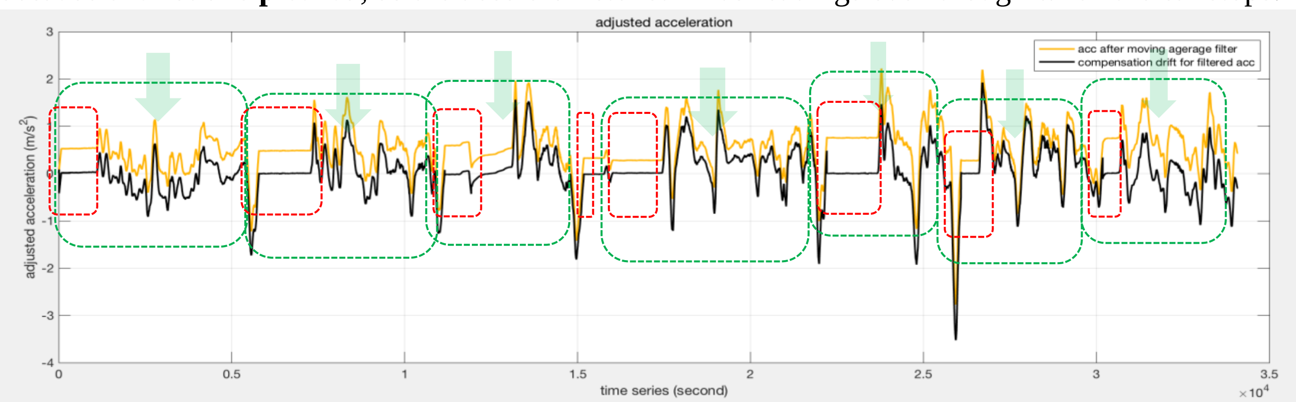

Figure 14. filtered acceleration and the acceleration after adjustment

As shown in figure 14, the drifts for each previous constant value period (red box) were subtracted for the following measurements value. Then the velocity was obtained through segmented integration. Details about this process are shown below:

First step: find the beginning and end time of the constant value period (red box). Here I defined two parameters to find the region:

delta_vel: the difference between the current acceleration value and the next value;

len: the length of constant value period;

If all the *delta_vel <= 0.75* and *len > 500*, it counts a valid "constant value period (red box)".

Second step: Calculate the mean value of each "constant value period (red box)" as the drift for its corresponding "green box" period.

Third step: For each period (green box) that follows the "constant value period (red box)", individually adjust these values by only subtracting the corresponding drift calculated from its previous "constant value period (red box)". The result is plotted as the black line in figure 14.

Fourth step: Segmented integrate the acceleration to obtain the velocity for each period (green box). And manually set the velocity of the beginning of each period to zero.

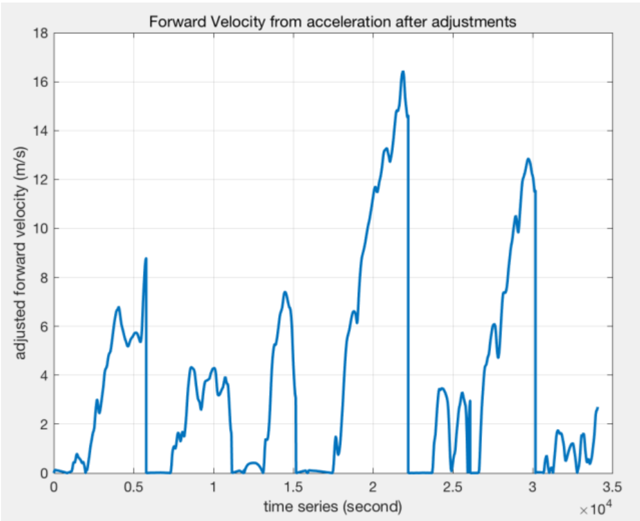

Figure 15. velocity from adjusted acceleration

Figure 15 shows the integrated velocity using the acceleration after adjustments. Compared with the original estimated velocity shown in figure 11, **these values are more reasonable and closer to the true value. Although there are still big errors with the true value, the overall trend is more similar to the GPS's measurement.**

After further observation of the black line in figure 14, we can also notice that there are more accelerations than deceleration. Compared with the estimated velocity from GPS, we can see that the speeds get faster and faster in each region, but without any effective deceleration. I don't think this error can be solved by simply adjusting the mean value of the acceleration. **There may exist a linear or nonlinear relation between the error and input. We need a scaling method to scale the measurements and reduce errors.**

3.3 Dead Reckoning - Integrate IMU data to obtain displacement and compare with GPS [*lab2_part3_3.m*]

1. Integrate $\ddot{x}_{obs}$ to obtain $\dot{X}$, then compute $w\dot{X}$ and compare with $\ddot{y}_{obs}$

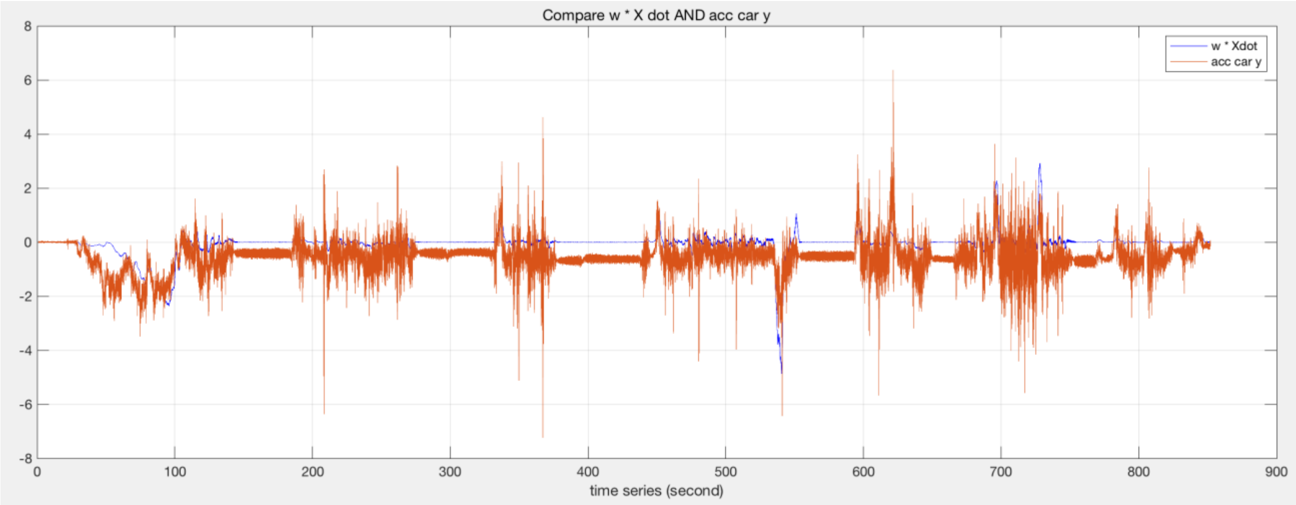

Figure 16. Compare w * X_dot AND acc_car_y

From figure 16, we can see that the overall trend between $w\dot{X}$ and $\dot{y}_{obs}$ is consistent. However, **the y-axis acceleration measured by the inertial sensor is very noisy** and exists many high-frequency components. Also, we can observe that **there are different drifts for each period which is similar characteristics as shown in figure 13**.

It may because when doing integration, the low-frequency white noise will yield a random walk, and the errors accumulate more and more. We can use a low pass filter with a specific cutoff frequency to remove the low-frequency part.

2 Use the heading from the magnetometer to rotate into a fixed (East, North) reference frame

- **Vector $(v_e, \ v_n)$**
- **Integrate it to estimate the trajectory of the vehicle (x_e, x_n)**
- **Compare the estimated trajectory with the GPS track (adjust the starting point)**

**The unit in figure 17 and figure 18 is true distance "meter".*

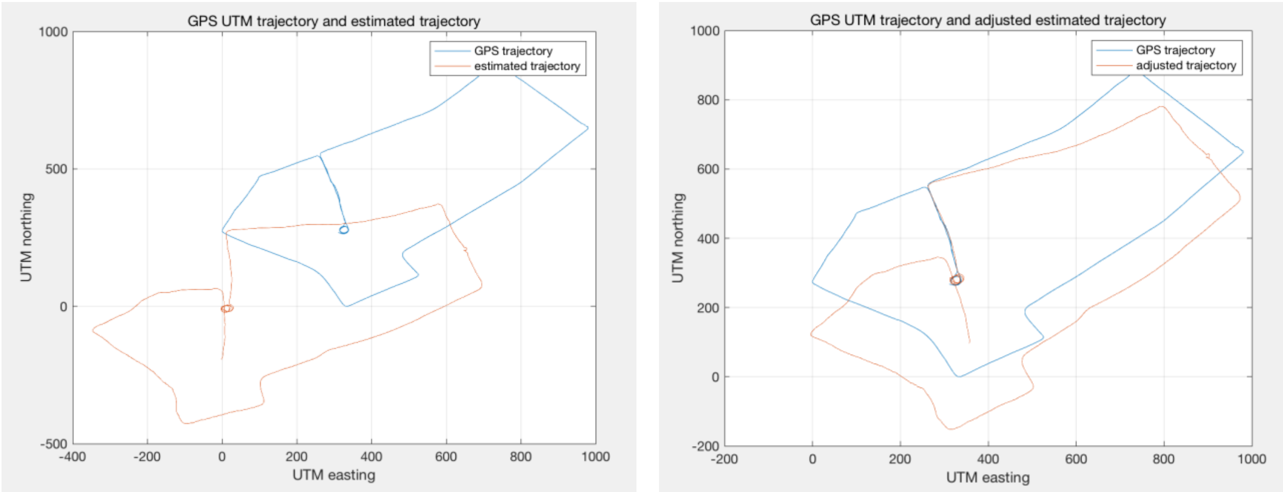

Figure 17. trajectory using **GPS velocity** before and after adjustment

At first, I plot the trajectory using GPS UTM information (blue line) and the trajectory using **integrated GPS velocity** (yellow line), as shown in figure 17. **We can see that the two trajectories are basically matching except for some skewness.** The GPS estimated velocity should be very close to the true value, so I conclude that **the mismatch mainly because of errors of the yaw angle from magnetometer.** From previous part, we also observe that the magnetometer measurement is very noisy.

Figure 18. trajectory using **estimated velocity** before and after adjustment

Figure 18 shows the trajectory using GPS UTM information (blue line) and the estimated trajectory using our **estimated velocity** $\boldsymbol{v_{estimated}}$:

$$v_{estimated} = \int acc_{adjusted}, (acc_{adjusted} \text{ was calculated in last part})$$

$$v_a = v_{estimated} * \sin(yaw\_angle_{magnetometer})$$

$$v_b = v_{estimated} * \cos(yaw\_angle_{magnetometer})$$

$$x_a = \int v_a$$

$$x_b = \int v_b$$

By using the estimated velocity to plot the trajectory, we can observe that the path is consistent at parts A, C and D. But for the path at parts B and after part E, the error gets huge. After comparing with the true GPS velocity shown in figure 12, I notice that the velocity for part A, C and D are more stable than other parts, and the changes of acceleration are also smaller. **Thus, I think the estimated trajectory would be more accurate if the sensor moves at a more uniform speed and a flatter road.**

3 Estimate x_c

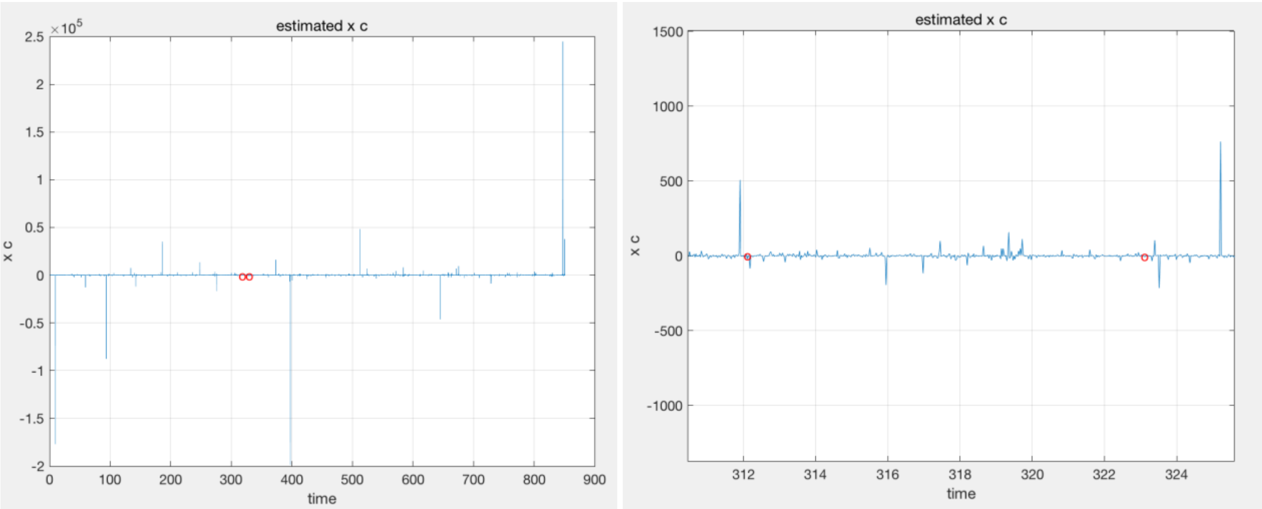

Figure 19. estimated x_c and constant velocity region

The estimated distance between inertial sensor and the CM point is shown in figure 19. To better estimated the x_c value, I calculate the mean value of x_c at time period 312 second to 323 second (shown as two red circle) that **the car is moving at a nearly constant velocity during this period.**

$$mean(x_{c_{312}} : x_{c_{323}}) = +0.276\,m = +27.6\,cm$$

The estimated distance I got is *+27.6 cm*, which is reasonable.

Reference:

[1] https://en.wikipedia.org/wiki/Multimodal_distribution

[2] https://en.wikipedia.org/wiki/Log-normal_distribution

[3] Richard Brown (2020). fitellipse.m, MATLAB Central File Exchange. (https://www.mathworks.com/matlabcentral/fileexchange/15125-fitellipse-m)