

EE267 Project Report

Zixi Xu

Stanford University

zixixu@stanford.edu

Abstract

The project consists of two main parts. In the first part, we incorporated the magnetometer into the complimentary filter for orientation tracking in flatland, together with gyroscope. We compared the results with the complimentary filter using gyro and accelerometer.

In the second part, we used the Levenberg-Marquardt algorithm to estimate the pose position and examined its performance.

Both ideas are discussed in the textbook [2].

1. Orientation Tracking in Flatland using Complimentary Filter with Gyro and Magnetometer

We used the magnetometer in our complimentary filter for flatland tracking in roll angle (page 44, lecture 9 [1]). As did in the homework, we first measure the bias and variance of the magnetometer for a reference:

MAG-BIAS: 10.15896 77.31941 2.31309

MAG-VAR: 0.58133 0.51194 0.58675

The tricky part of the magnetometer is the calibration process. Following the method given in [5], we use a hard-iron method to calibrate the magnetometer. The method is to rotate the magnetometer in all possible angles and find the maximum and minimum values in each axis. Since here we only need to measure the *roll angle*, we just give the max and min in x and y axis:

magx-max = 67, magx-min = -20

magy-max = 88, magy-min = -8

Then we subtract the magnetometer values by the mean of the max and min:

$$mag_i = mag_i - 0.5(mag_{i_{max}} + mag_{i_{min}})$$

where i can be x, y, z.

Then we need to rotate the local coordinates into the world coordinates. We can do this by doing cross product with the north position vector (instead of the up-vector as given in course notes [3]), or we can simply rescale the

angle by removing the offset. We use the latter method in this project, and the screenshots of the results are shown in Figure 2. As a comparison, the screenshot of using accelerometer is also given in Figure 1.

It is easy to see that both gyro (blue one) and accelerometer or magnetometer (red one) have some disadvantages. Gyro has a large offset, although it is smooth and has correct trend, while accelerometer or magnetometer is very noise but does not have bias. After we use a complimentary filter (green one), we can easily see that the result has no bias and much less fluctuation.

We used $\alpha = 0.99$ as the parameter in the complimentary filter, the result is satisfying. The readings from magnetometer is more "step-like" in each time step, and has less upheavals in local measurements, compared with the accelerometer readings. The accelerometers has more intensive fluctuation and the curve is more "ramp-like". Both of them eliminated the bias of gyros and give relatively accurate estimation of the angles.

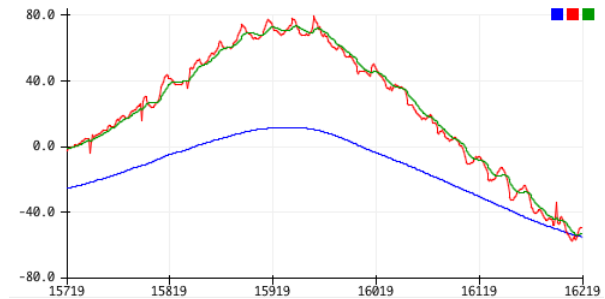


Figure 1. Screenshot of orientation tracking in Flatland using gyro (blue), accelerometer (red) and complimentary filter (green).

2. Using Levenberg-Marquardt algorithm for pose estimation

2.1. Algorithm

We followed the algorithm 5 in section 6.2 of course notes [4].

Then we compute the residual in each iteration by calcu-

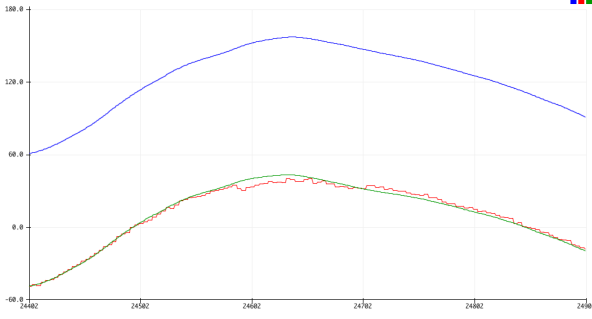


Figure 2. Screenshot of orientation tracking in Flatland using gyro (blue), magnetometer (red) and complimentary filter (green).

lating:

$$Residual_{LM} = \|pos_{measured} - pos_{estimated}\|_2$$

We use the homography estimation as a start, and compare the residuals of the homography method with those of LM method.

2.2. Parameter tuning

We searched among different pairs of λ and iteration number to fully understand how the algorithm performs. Since the residual in each case is similar, we take a rough average among about 30 estimations and give the result of the LM estimation in the table below.

Table 1. Best residuals of different λ and iteration number.

IterNum	$\lambda = 1$	0.1	0.001
5	0.000520	0.000538	0.000364
10	0.000476	0.000465	0.000315
20	0.000426	0.000312	0.000277

The average residual of using homography method is 0.001305. So the LM method further decreases the residual by about 50%.

The higher the λ and the iteration number are, the lower the best residual. To keep the latency not too high, in the demonstration we used 10 iterations and $\lambda = 0.1$.

2.3. Demonstration

The positions of pose are the simulated data given as the data used for homework 6. The snapshots of the results shown in visualize.html is given in Figure 3 and 4.

During the simulation, we observed that at some certain points, the pose movement not very smooth when moving fast or drastically. This shows more clearly how the actual pose moved. Also, the algorithm is sensitive to the initial

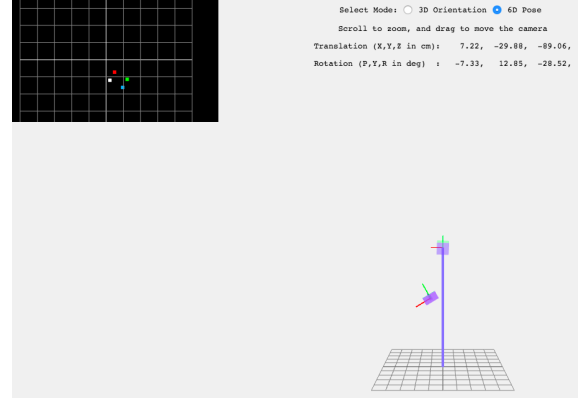


Figure 3. Screenshot of the result of Levenberg-Marquardt algorithm in pose estimation, the simulated data is given in class. $\lambda = 0.1$, 10 iterations each time.

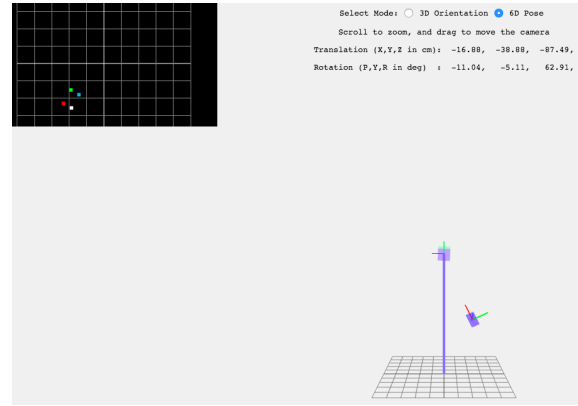


Figure 4. Screenshot at another time of the result of Levenberg-Marquardt algorithm in pose estimation, the simulated data is given in class. $\lambda = 0.1$, 10 iterations each time.

condition. If we use some fixed initial guess as the start of the algorithm, the result can be inaccurate.

References

- [1] EE267 virtual reality: Slides of lecture 9 and 10, 2018.
- [2] S. M. LaValle. *Virtual Reality*. Cambridge University Press, 2016.
- [3] G. Wetzstein. Course notes: 3-DOF orientation tracking with IMUs, 2017.
- [4] G. Wetzstein. Course notes: 6-DOF pose tracking with the VRduino, 2017.
- [5] K. Winer. Simple and effective magnetometer calibration. <https://github.com/kriswiner/MPU6050/wiki/Simple-and-Effective-Magnetometer-Calibration>. Accessed: 2018-06-06.