Report: HW2 Particle Filter for Indoor Positioning using IMU Measurements

Name: Wenqi Xu ID: 400156647

**Weight in the complementary filter:**

In my implementation, I set the weight to be 1, which means the result is totally computed by gyro data. The result is as figure 1 shows:

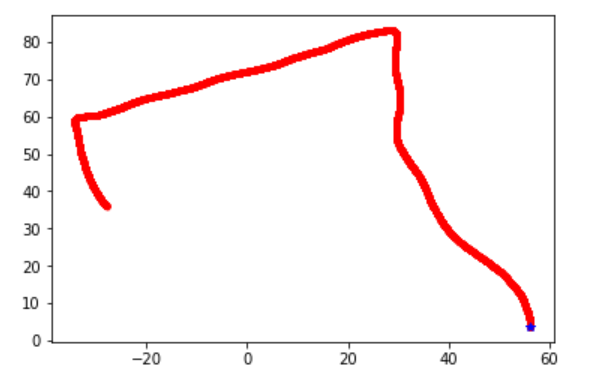
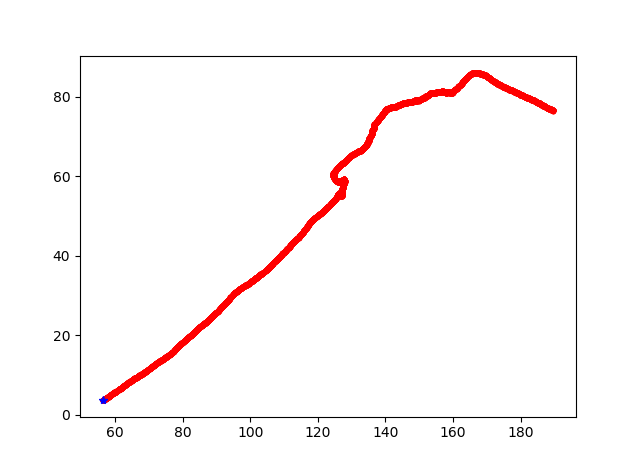
 

Figure 1. Illustrate the headings (weight = 1) Figure 2. Illustrate the headings (weight < 1)

If setting weight less than 1, all of the results are like figure 2, which is not the correct heading. The result generates by mag and acc is not accurate because the magnetometer may not be calibrated.

**The max number of particles:**

To evaluate the effect of the max number of particles, I set *num\_particle = 500, 2000, 8000,* respectively. The other parameters are as follows:

*mean\_step\_length = 1*

*heading\_error = np.pi/4*

*thres = 0.66*

Evaluate the results by mean location errors:

When N = 500, mean location errors = 4.55

When N = 2000, mean location errors = 4.54

When N = 8000, mean location errors = 6.56

The results show that too many particles may cause the error to be bigger. I also found that when N is bigger, the particle cloud is more dispersed. N = 2000 is reasonable.

**Heading error parameter:**

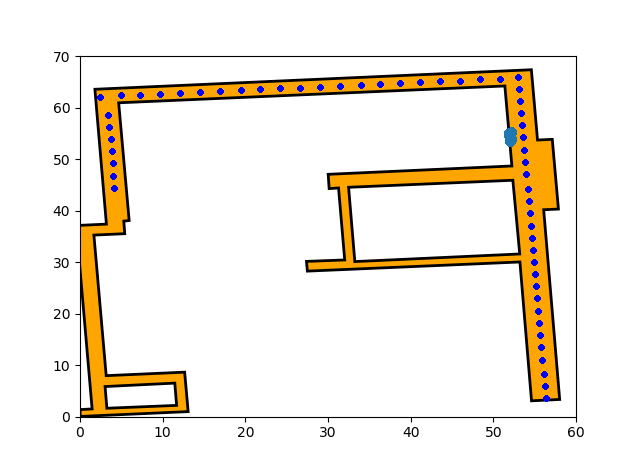
To evaluate the effect of the heading error parameter, I set *heading\_error = pi/8, pi/4, pi/2.* The other parameters are as follows:

*mean\_step\_length = 1*

*num\_particle = 2000*

*thres = 0.66*

When heading\_error = pi/8, the particle cloud got stuck at the first corridor, like figure 3. after a fewer second, it comes back to the original place. The location error is 22.01. The tracking algorithm does not work.

  
Figure 3. The particles cloud got stuck at this place when heading error is pi/8

When heading error is pi/4, the tracking algorithms works well. The mean location error is 4.54. The particle cloud is dispersed, and finally, it will get to the right destination as figure 4.

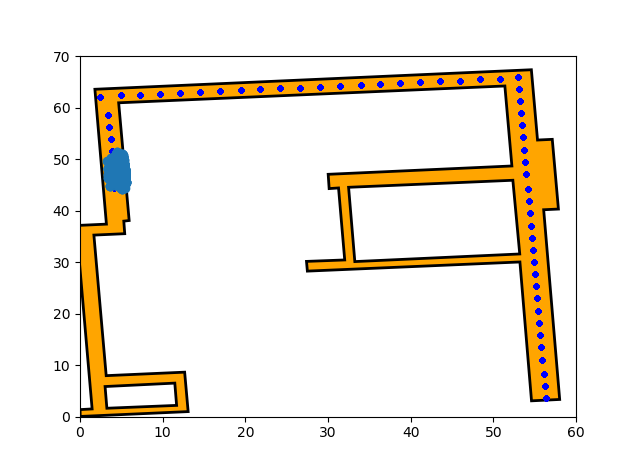


Figure 4. The particles cloud terminates at destination when heading error is pi/4

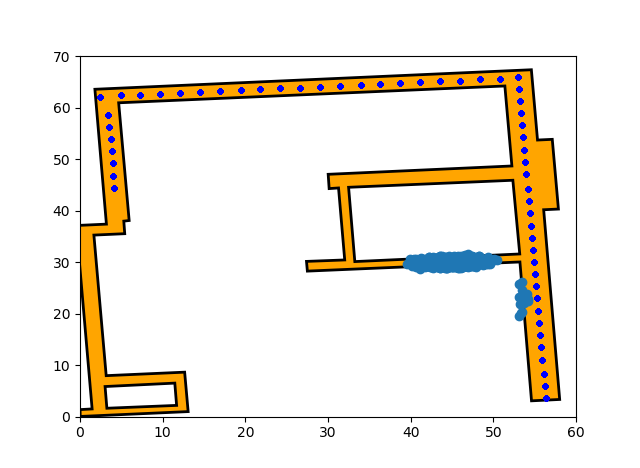
When heading error is pi/2. The tracking algorithm does not work well, the particle cloud will head for the second hallway and got stuck there like figure 5 shows. 

Figure 4. The particles cloud terminates at destination when heading error is pi/4

Overall, heading error = pi/4 is suitable.

**Random initialization vs known initial position and heading:**

I tested random initialization and set the parameter as follows:

*mean\_step\_length = 1*

*num\_particle = 2000*

*heading\_error = pi/4*

*thres = 0.66*

The tracking process is as figure 5 shows. At first stage, particle cloud cannot estimate the right location, after the cloud turn left, it works well and track to the destination. The mean location error is 14.17. It is big due to the bad performance at the first stage. For known initial position method, the error is 4.54, the result is as figure 4 shows. Overall, random initialization method is not as good as known position method. But when we don’t know the specific initial position. The performance can be accepted. It does not perform well at first stage and after a while, it can estimate location well.

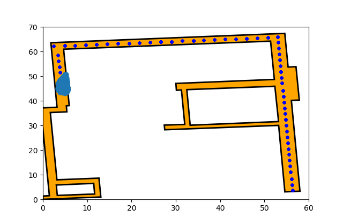
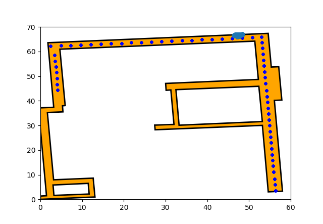
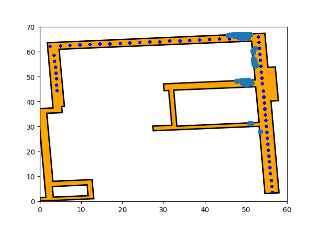
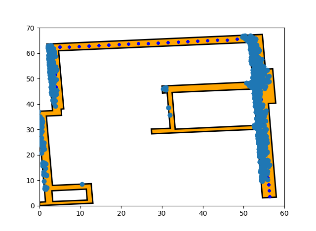


Figure 5. Tracking process by random initialization

**Discuss the pros and cons of the approach in “Run the Particle Filter”:**

Pros: The algorithms will be more efficient in this way. The update is time-consuming, if the update is performed for each sample, there will be much more computational overhead. It is also reasonable to update after a step. Because the heading direction is almost fixed for each step and observation may be similar in each sample for a step.

Cons: This method may not as accurate as update for every sample for a step. If the update is performed after each sample, more observation information can be used.