EL2320 Applied Estimation Lab2

Chia-Hsuan Chou

chchou@kth.se

Part 1: Preparitory Questions

Question 1: What are the particles of the particle filter?

In particle filter, the particles are the samples of a posterior distribution. They are denoted as :

$$X_t := x_t^{[1]}, x_t^{[2]}, x_t^{[3]} \dots., x_t^{[M]}$$

In the other word, a particle is a hypothesis as to what the state might be in the real world.

Question 2: What are importance weights, target distribution, and proposal distribution and what is the relation between them?

Importance weight is the probability of the measurement z_t under the particle $x_t^{[m]}$, it is used to incorporate the measurement z_t into the particle set, we denote it as $\omega_t^{[m]}$.Target distribution is the probability density function that we expect to have, we denoted it as f. On the other hand, The proposal distribution is the distribution which we are given samples generated from, we denoted it as g. In particle filters, f corresponds to the belief $bel(x_t)$ and g corresponds to the belief $\overline{bel}(x_t)$.

The relation between them are denoted as:

$$\omega_t^{[m]} = \frac{f(x^{[m]})}{g(x^{[m]})}$$

Question 3: What is the cause of particle deprivation and what is the danger?

Particle deprivation is a problem that there are no particles in vicinity of correct state. It occurs when the number of samples is too small to cover all the relevant region with high likelihood. If particle deprivation happens, we will generate an estimate that is totally wrong.

Question 4: Why do we resample instead of simply maintaining a weight for each particle always.

Because the new state might not include in the present particle set. When processing particle filter, we want to have higher resolution instead of particles distribute on only some regions so that we can get more accurate estimate.

Question 5: Give some examples of the situations which the average of the particle set is not a good representation of the particle set.

When we have a mixture of Gaussian distribution, the average of particle set might not have a good representation of the particle set. The distribution has more than one peak, and the particle will converge to those peaks and we don't exactly know where the robot locates.

Question 6: How can we make inferences about states that lie between particles.

We can use Density extraction to convert a particle set into a continuous density. They are several ways to extract a density from such particles, for example, Gaussian approximation to a unimodal distribution on density. If we have multimodal sample distributions we can use k-mean clustering. Also, we can use Kernel density estimation as well. Which technique is used in practice is depend on the problem at hand.

Question 7: How can sample variance cause problems and what are two remedies?

Sample variance can cause Particle deprivation so we need to keep it as low as possible. There are two remedies on it. First of all, we can reduce the frequency at which resampling take place. When the state is known to be static, we should never resample. Second remedy is that we can use low variance sampling.

Question 8: For robot localization for a given quality of posterior approximation, how are the pose uncertainty (spread of the true posteriori) and number of particles we chose to use related.

If we have higher uncertainty, we should have larger number of particles so that we can make sure all the region is covered. And In contrast, if we have lower uncertainty, we can choose smaller number of particles.

Part 2: Matlab Exercises

2.1 Prediction

2.1.2 3D State Space

Question 1: What are the advantage/drawbacks of using (6) compared to (8)? Motivate.

The advantage of using (6) :

1. Reduce the computational time because it is only two dimensional.
2. No need to compute new dx and dy each time t.

The drawbacks of using (6):

1. It can't be used to predict a more complexed model, only one with constant linear velocity and constant angular velocity.

Question 2: What types of circular motions can we model using (9) ? What are the limitation (what do we need to know/fix in advance?)

We can model uniform circular motion using (9), because the magnitude of velocity is constant, so does the rotational velocity. In equation (9), we need to fix ω_0 and θ_0.

2.2 Sensor Model

Question 3: What is the purpose of keeping the constant part in the denominator of (10)?

Without the denominator, the right side of (10) is merely a likelihood instead of a probability. A probability should be in an interval [0,1], but a likelihood can be larger than 1. Therefore, the constant part in denominator allows us to get the probability of the event, rather than a relative likelihood compared to other events.

2.3 Re-Sampling

Question 4: How many random numbers do you need to generate for the Multinomial re-sampling method? How many do you need for the Systematic re-sampling method?

In Multinomial resampling method, M numbers of random numbers are needed to generate, on the other hand, only 1 random number is needed to generate in Systematic re-sampling method.

Question 5: With what probability does a particle with weight $\boldsymbol{w = \frac{1}{M} + \varepsilon}$ survive the re-sampling step in each type of re-sampling (vanilla and systematic)? What is this probability for a particle with $\boldsymbol{0 \leq w < \frac{1}{M}}$

What does this tell you? (Hint: it is easier to reason about the probability of not surviving, that is M failed binary selections for vanilla, and then subtract that amount from 1.0 to find the probability of surviving.

For weight $\omega = \frac{1}{M} + \varepsilon$, the probability to survive in each method is:

1. Vanilla resampling Method:

$$P(paticle\ survive) = 1 - [1 - \left(\frac{1}{M} + \varepsilon\right)]^M$$

2. Systematic resampling Method:

$$P(particle\ survive) = 1$$

For weight $0 \leq \omega < \frac{1}{M}$, the probability to survive in each method is:

1. Vanilla resampling Method:

$$P(paticle\ survive) = 1 - [1-\omega]^M$$

2. Systematic resampling Method:
   In this situation, only if $r_0 \leq \omega$ can the particle survive, therefore, the probability to survive can be derived as :
   $P(particle\ survive) = \frac{w}{\frac{1}{M} - 0} = \omega M$

If a particle has its weight larger than $\frac{1}{M}$, then no matter how large the weight is, this particle will definitely be chosen into the new particle set in systematic resampling method. Therefore, systematic resampling method assures that the particles who's weight is larger than $\frac{1}{M}$ will be evenly sampled.

If a particle has its weight smaller than $\frac{1}{M}$, then there is still chance to be picked with relatively small probability. If the weight of particle is smaller, the probability being picked will be smaller, and vice versa.

2.4 Experiments

Question 6: Which variables model the measurement noise/process noise models?

The variable rams.Sigma_Q is the measurement noise covariance matrix, which models the measurement noise; The variable params.Sigma_R is the process noise covariance matrix, which models the process noise.

Question 7: What happens when you do not perform the diffusion step? (You can set the process noise to 0)

After resampling few times, most of the particles will be removed and only one of the particle will survives. Then all other particles will converge to this one, which is the closest particle that we apply the motion without noise in the beginning. After several times of re-sampling, this particle will gradually be distant from the target.

Question 8: What happens when you do not re-sample? (set RESAMPLE MODE=0)

If we do not do the re-sample, the particles will not converge into the target and stay in the initial place, and the error will thus become very large and the robot can hardly find where it is.

Question 9: What happens when you increase/decrease the standard deviations (diagonal elements of the covariance matrix) of the observation noise model? (try values between 0.0001 and 10000)

If we increase the standard deviations, which means that we trust our observation more, the particles will converge into the vicinity of correct state much more quickly and tightly(small variance of particles). However, if we choose a too large number of standard deviation, the error would increase as well because the particle will disperse into a larger estimated correct state (large variance) and thus the estimated states might be far away from the real states.

In the same time, if we decrease the standard deviations, the particles might not converge into an area and the particle deprivation could occur. This is because the covariance is too small that the initial particles can hardly lie on such high probability range. Therefore, the error would increase as well. The following figures compares the error under different standard deviations:

Question 10: What happens when you increase/decrease the standard deviations (diagonal elements of the covariance matrix) of the process noise model? (try values between 0.0001 and 10000)

The standard deviation of the process noise controls the variance of the particles. If we increase the standard deviation, the variance of the particles will increase as well, if we decrease the standard deviation, the variance of the particles will decrease.

Question 11: How does the choice of the motion model affect a reasonable choice of process noise model?

The process noise model is chosen based on the choice of the motion model. If the chosen motion model is not the true motion, then we will get a larger error thus we will need a larger process noise model.

Question 12: How does the choice of the motion model affect the precision/ accuracy of the results? How does it change the number of particles you need?

If we choose a wrong motion model, then the accuracy of the results will be bad, then we will need a larger process noise and more particles so that we can cover the target and keep track of it.

Question 13: What do you think you can do to detect the outliers in third type of measurements? Hint: what happens to the likelihoods of the observation when it is far away from what the filter has predicted?

Setting a threshold for an average likelihood to the measurements, if the likelihood of one measurement is smaller than the threshold, then we consider this measurement as an outlier.

Question 14: Using 1000 particles, what is the best precision you get for the second type of measurements of the object moving on the circle when modeling a fixed, a linear or a circular motion (using the best parameter setting)? How sensitive is the filter to the correct choice of the parameters for each type of motion?

The best parameter of three motion:

Motion Type	Sigma_Q	Sigma_R	error
Fixed motion	diag([900 900])	diag([50,50,0.5])	11.2 ± 5.5
Linear Motion	diag([600,600])	diag([2,2,0.01])	7.5 ± 3.4
Circular Motion	diag([300,300])	diag([2,2,0.01])	7.0 ± 3.5

In the Linear Motion and Circular motion, only measurement noise is sensitive to find a smaller error. In Fixed motion, the process noise is sensitive, because in our case, the target in moving on the circle, so we need to set a larger noise on process to compensate. In the table above, the circular motion has the least error and the fixed motion has the most error as our expected.

3.5 Outlier Detector

Question 15: What parameters affect the mentioned outlier detection approach? What will be the result of the mentioned method if you model a very weak measurement noise |Q| -> 0?

The threshold affects the outlier detection, if we have a higher threshold, then less outliers will be detected, on the other hand, if we have a smaller threshold, then more outliers will be detected.

If we have a very week measurement noise Q, which means that we have high confidence on our measurement, we will have a much narrow shape of measurement prediction and a higher average likelihood of the particles, so the measurement is less likely to be detected as an outlier.

Question 16: What happens to the weight of the particles if you do not detect outliers?

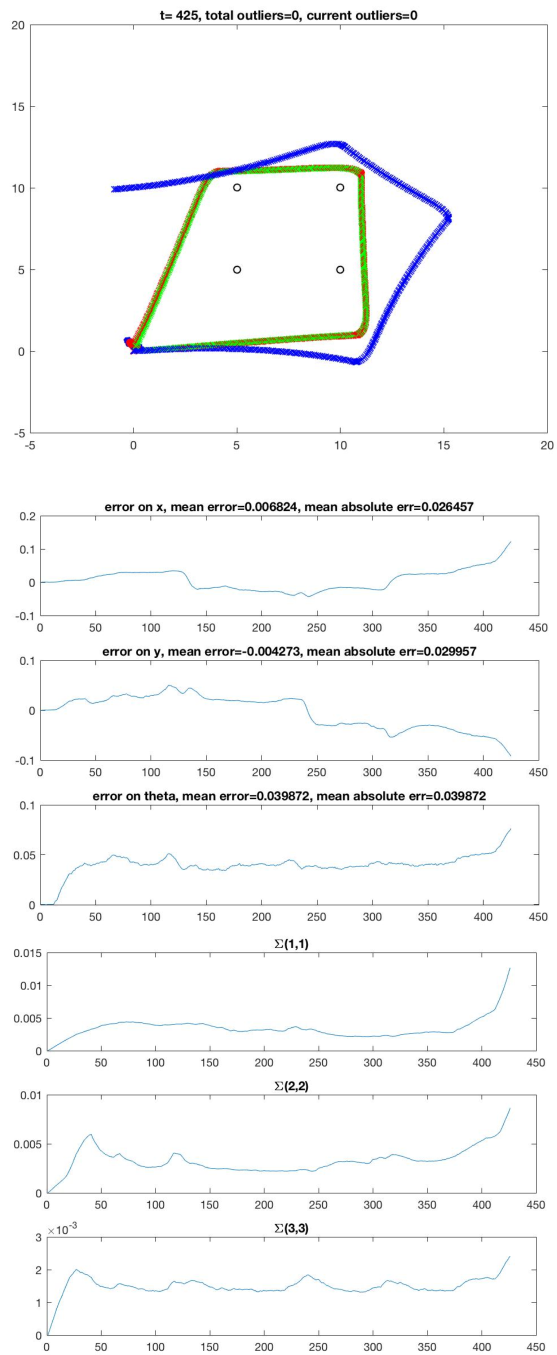
If we don't detect outliers, then the particles near an outlier will be given to a higher weight, and we might get a wrong particle set on the next re-sampling and it won't be easier for particles to converge into the correct states.

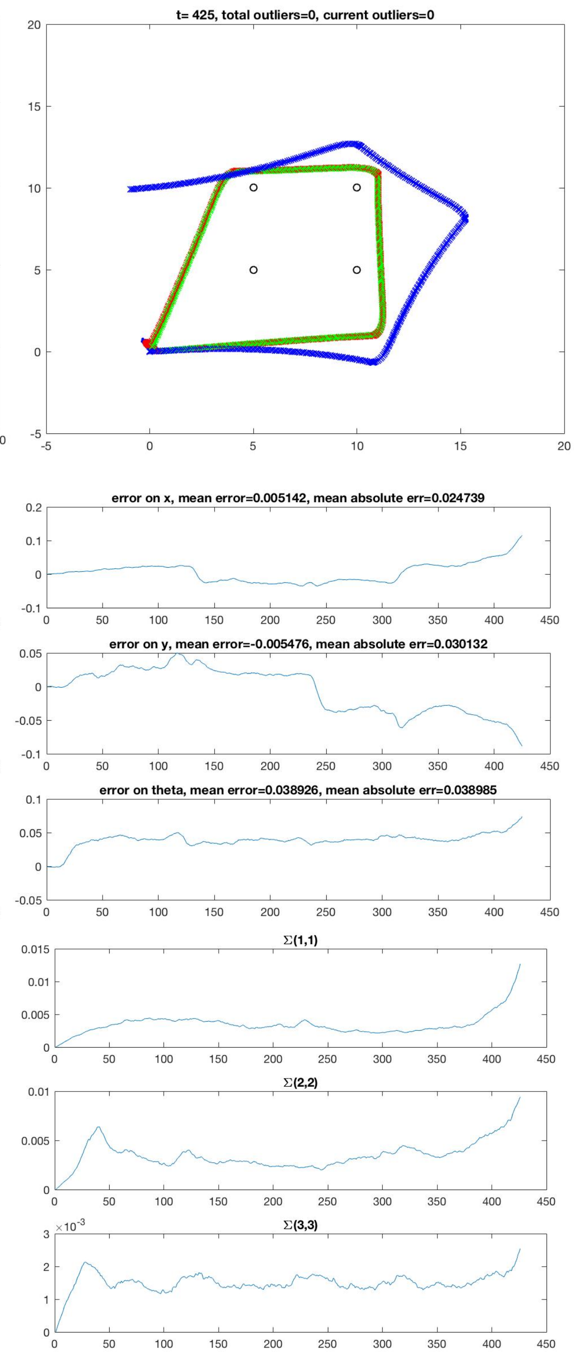
Part 3: Results

3.1 map_sym2.txt + so_sym2_nk.txt

When performing the tracking and localization on this case, we found that the results of tracking is better than the results of localization, because the robots know where is the initial position so it knows where to initialize the particles. On the other hand, when doing the localization, there is no initial position, and it is hard for the robot to know where he is due to the symmetry of the map.

The following figures shows the tracking on this dataset with different numbers of particles:



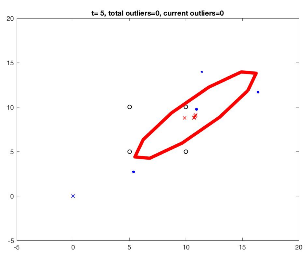


fig(1)(3)(5) 10000 particles *fig(2)(4)(6) 1000 particles*

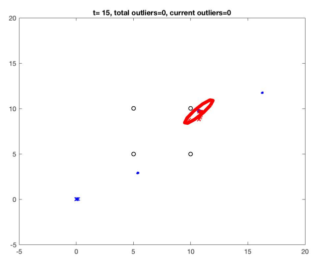
There are four hypotheses, which are around the four landmarks. When the number of the particles increase, the particles will stay in the four hypotheses longer. This is because when we have only 1000 particles, there are not enough particles to cover all the hypotheses (particle deprivation), if we increase the numbers of particles into 10000, then the particles covering the area with high likelihood will be enough.

The following figure shows that with 1000 and 10000particles, when t = 5,15,25,35,45,55. From the following figure we can see that with1000 particles, the particles converge really fast, but they converge into a wrong place, while with 10000 particles, though they converge late, but they converge into a right place.

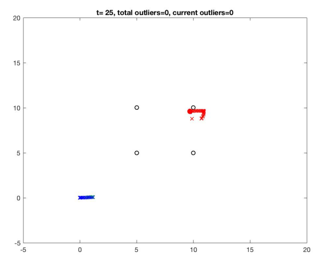
1. 1000 particles in t = 5,15,25,35,45,55



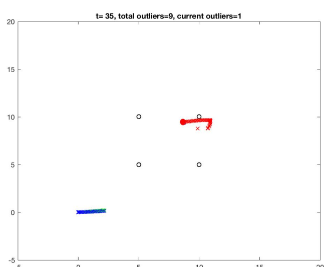
t = 5



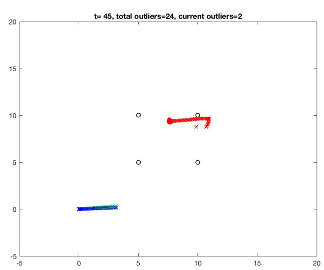
t = 15



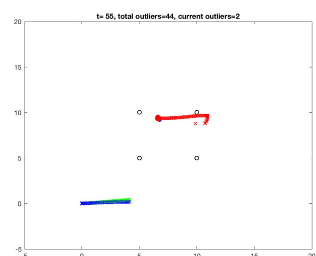
t = 25



t = 35

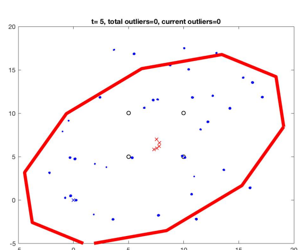


t = 45

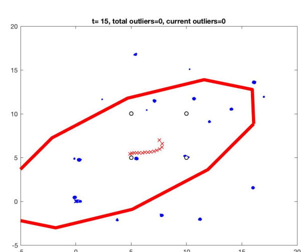


t = 55

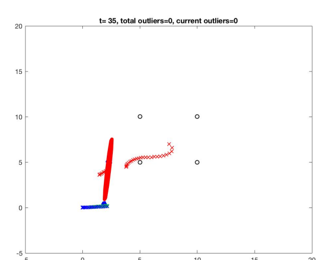
2. 10000 particles in t = 5,15,25,35,45,55



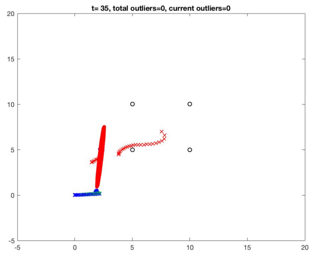
t = 5



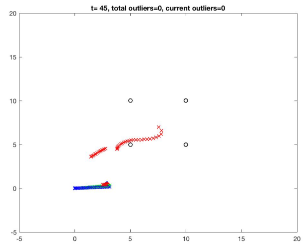
t = 15



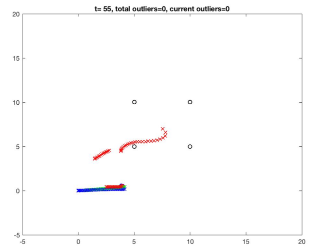
t = 25



t = 35

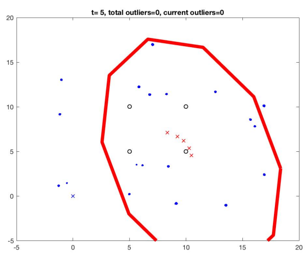


t = 45

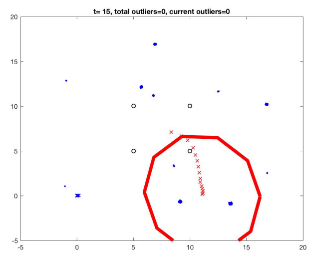


t = 55

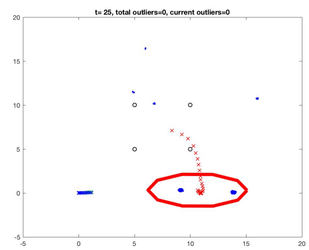
The following figures are localization with 10000 using multinomial resampling function:



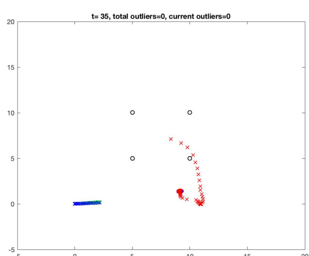
t = 5



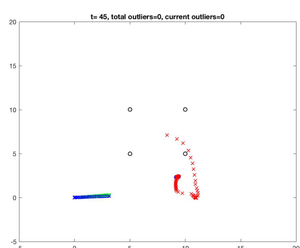
t = 15



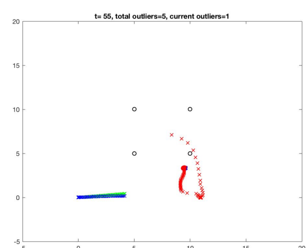
t = 25



t = 35



t = 45

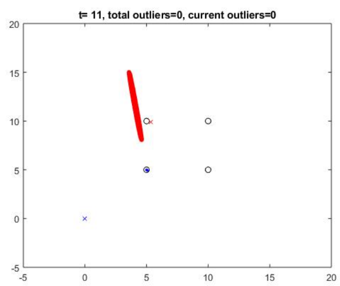


t = 55

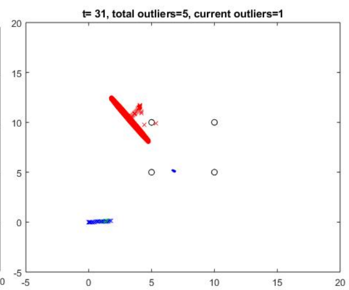
When using multinomial resampling function, the methods to pick up particles changes, the particles with higher weights are picked. Compared to two methods, the multinomial resampling function converge much fast so and the preservation is worse. (see the figure when t = 55)

The following set of figures show the difference with weak and strong noise:

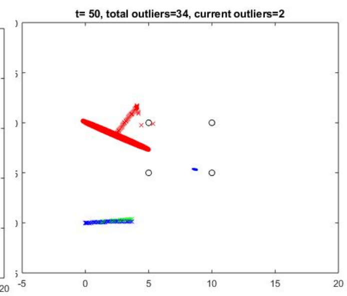
1. with Q 100 times smaller than the initial Q



t = 11

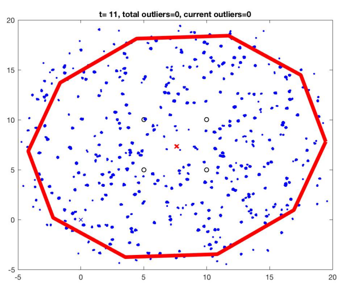


t = 31

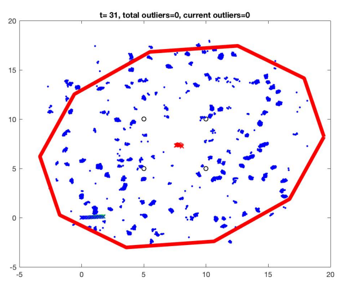


t = 50

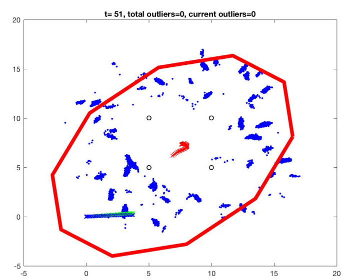
2. with Q 100 times larger than the initial Q



t = 11



t = 31



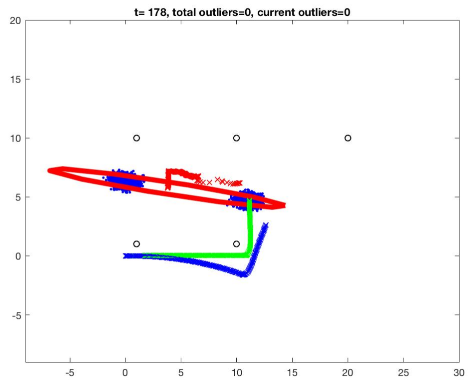
t = 51

When Q is larger, which means that we don't trust our measurement very much, thus the particles will be weighted more by the process motion, so the particles converges slowly. From the above figures we can see that when Q is 100 times larger, it converges very slowly, even when t = 51, the particles haven't converged yet. However, when Q is smaller, the particles disappears fast but here they converge into a wrong place.

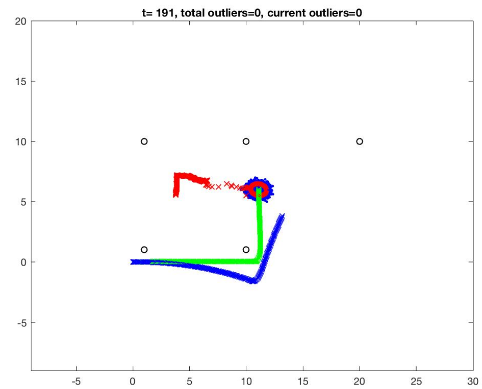
3.2 map_sym3.txt + so_sym3_nk.txt

In order to see the particles converge when the symmetry of the landmarks is broken, I set both measurement noise and process noise 10 times larger, so that we can see more clearly the particles converge into one hypotheses.

Here are the results:



t = 178



t = 191

when t = 178, the particles converge into two hypotheses, however, when t = 191, after detect a landmark in the right, the symmetry of the map is broken and the particles soon converge into the right area. From these two figures , we can see that how in the measurements correct the particles’ location when a new measurement is available.