Name: Tianxiao Zhao
Email: tzh@kth.se

EL2320 Applied Estimation - Lab 2

Part I – Preparatory Questions

Particle Filter:

1. What are the particles of the particle filter?

The particles are the samples of a posterior distribution, which is used to approximately represent this distribution.

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

Importance weight $\omega_t^{[m]}$ is the probability of the measurement z_t given a particle $x_t^{[m]}$, which can be expressed as $\omega_t^{[m]} = p(z_t|x_t^{[m]})$; Target distribution is the probability distribution from which we want to obtain a sample; Proposal distribution is the related distribution from which we can generate particles.

In practice, sampling from the target distribution directly is hard to achieve. We could indirectly sample it by sampling from the proposal distribution. The probability of each sample's being drawn from the proposal distribution is equal to the importance weight.

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

Particle deprivation occurs because the number of particles is too small to cover all relevant regions with high likelihood. When this happens, there may be no particles around the true state when estimation is performed in a high-dimensional space. In general, particle deprivation will always happen after it is run long enough, because random resampling may unluckily generate some random particles which all stay away from the true state.

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

We execute the resampling steps in order to change the distribution of the particles from the proposal distribution $\overline{bel}(x_t)$ to the target distribution $bel(x_t)$. In this way, the particle set will gather in regions of high posterior probability. Contrary to this, maintaining a weight for each particle would cause a bad distribution of particles over regions with low posterior probability.

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

Suppose we have a multimodal distribution which has two peaks. The particles will be located around two peaks in a relatively high density. Then, the average of the particle set will sort of end up in the middle position of the two peaks, and turns out to be a bad representation of the particle set.

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

To obtain inferences about states that lie between particles, we could interpolate the probability of the adjacent particles around the states and estimate it approximately.

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

After resampling, the variance of the particle set decreases, which makes the error between the particle distribution (as an estimator of true state) and the target distribution increased. The two remedies consist of reducing the frequency at which resampling occurs, and using low variance sampling.

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?

The higher the pose uncertainty, the wider the spread of the true posteriori, which means an increased number of particles are needed to represent this distribution.

Part II – Matlab Exercises

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

In 2D state space, equation (6) shows that the angle θ is set to a constant value θ_0 in this case. When noise is taken into consideration, the trace of the robot will turn out to be a zigzag line around the true value. On the other hand, in 3D state space, the angle θ is a state which will be controlled by combining information from the previous time-step during the process of estimation. This would give a smoother trace than before. The advantages of using (6) may be its compact form of model and less computational storage.

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

According to equation (9), it models a circular motion with constant linear velocity v_0 and constant angular velocity ω_0, which need to be known in advance.

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

The constant part in the denominator equals to the integration over the whole distribution. The purpose of keeping it in the denominator is to normalize the likelihood function for an observation.

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?

For the Multinomial re-sampling method, we need to generate M random numbers. For Systematic re-sampling method, we only need one random number.

Question 5: With what probability does a particle with weight $\boldsymbol{\omega = \frac{1}{M} + \epsilon}$ survive the re-sampling step in each type of re-sampling (vanilla and systematic)? What is this probability for a particle with $\boldsymbol{0 \leq \omega < \frac{1}{M}}$? 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 each particle, the probability of selection equals to the weight ω. Then we could know the probability of not surviving for each particle is $1 - \omega$. In vanilla re-sampling, the selection process repeats M times and each time the random number generated is within the interval $[0,1]$, which means the probability of not surviving for a particle becomes $(1 - \omega)^M$ after these M rounds. Inversely, the probability of survival for a particle should be $1 - (1 - \omega)^M$.

In systematic re-sampling, if the weight $\omega = \frac{1}{M} + \epsilon$, the particle will always be selected since the weight ω is larger than the selection step $\frac{1}{M}$. In this way, the cumulative distribution function of this particle will always be above the generated random number $r_0 + \frac{m-1}{M}$ no matter what index the sample possesses. If the weight $0 \leq \omega < \frac{1}{M}$, the minimum probability for a particle to be chosen is proportional to the value of weight, and the minimum occurs when the cumulative distribution function of the previous selected particle equals to $\frac{k}{M}, k = 0, 1, \dots, M-1$. Then, this minimum probability could be computed by the following equation: $p_{min} = \frac{\omega - 0}{1/M - 0} = \omega M$.

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

In the warmup code, variable `params.Sigma_Q` models the measurement noise model, and variable `params.Sigma_R` models the process noise model.

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

The re-sampling process ends up with duplicating the same particle M times. All other particles converge to the one with the highest weight.

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

If re-sampling is not used, all the particles will still exist in the state space, and they will move according to the prediction step but they will not converge to the true state. The weights will somehow be updated but not be used for re-sampling, which keeps the uniform distribution of particles unchanged.

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)

When we increase the diagonal elements of the matrix Q, the particles could converge in a quick manner but the uncertainty around the true state increases as well; when we decrease the matrix Q, the particles will have a difficulty in convergence because a lot of measurements are beyond the likelihood threshold and regarded as outliers.

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)

When the standard deviations of the process noise model are increased, the diversity of particles is increased as well. The convergence will be fast but the uncertainty of the estimation is quite high, with particles spreading and moving around the true state. When the standard deviation decreases, the particles will concentrate on few points and these points will possibly converge to the true state slowly .

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

If the motion model is accurate enough to model the true state, then we can choose a process noise model with low values. On the other hand, if the motion model lacks of accuracy, then a process noise model with a high covariance will be needed since the diversity of the particles will compensate for that inaccuracy to some degree.

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?

An accurate motion model will result in a higher accuracy of the results, and fewer particles will be needed in this case. Conversely, an inaccurate model will cause a lower accuracy of the results and will need a higher process noise model, which means more particles will be spread around the true state.

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?

When the observation is far away from what the filter has predicted, its likelihood will take a very small value. To detect and rule out these outliers, we could set a threshold to discard those whose average likelihoods are below it.

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 results are listed in Table 1. The linear and circular motion model give similar optimal results. And fixed motion model is relatively more sensitive to the choice of parameters. It seems that all motion models are more sensitive to the change of R compared to Q.

Motion model	R	Q	Estimate error
Fixed	$\begin{bmatrix} 30 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 0.15 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 0 & 300 \end{bmatrix}$	11.6 ± 4.9
Linear	$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0.025 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 0 & 300 \end{bmatrix}$	8.4 ± 3.2
Circular	$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}$	$\begin{bmatrix} 300 & 0 \\ 0 & 300 \end{bmatrix}$	7.9 ± 2.6

Table 1: best results for different motion model

Part III– Main problem

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 $|Q| \to 0$?

The likelihood threshold λ_ψ and the measurement noise model will affect the outlier detection approach. If the measurement noise model is very weak, the low value of the covariance matrix will generate a "narrow and high" likelihood distribution. The peak is so narrow that particles are easily be classified as outliers since their likelihoods tend to fall beyond the threshold with ease.

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

If outliers are not detected, then all the measurements are used to calculate the weights. The weights will be erroneous after re-sampling process since the likelihoods of more particles are taken into account.

Result

1. map_sym2.txt + so_sym2_nk

In this map, the simulated tracking task works well because the initial position and pose of the robot is known, which means the filter could spread particles near the true state at the beginning. Figure 1 and Figure 2 show the results of the tracking tasks with different numbers of particles.

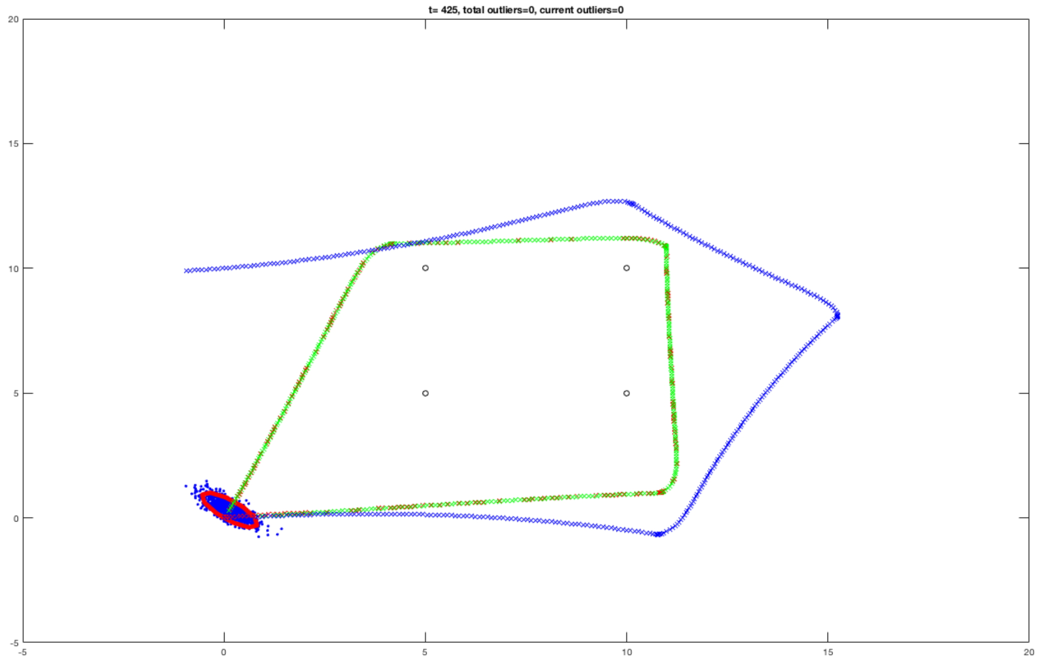

Figure 1: tracking task with 1000 particles

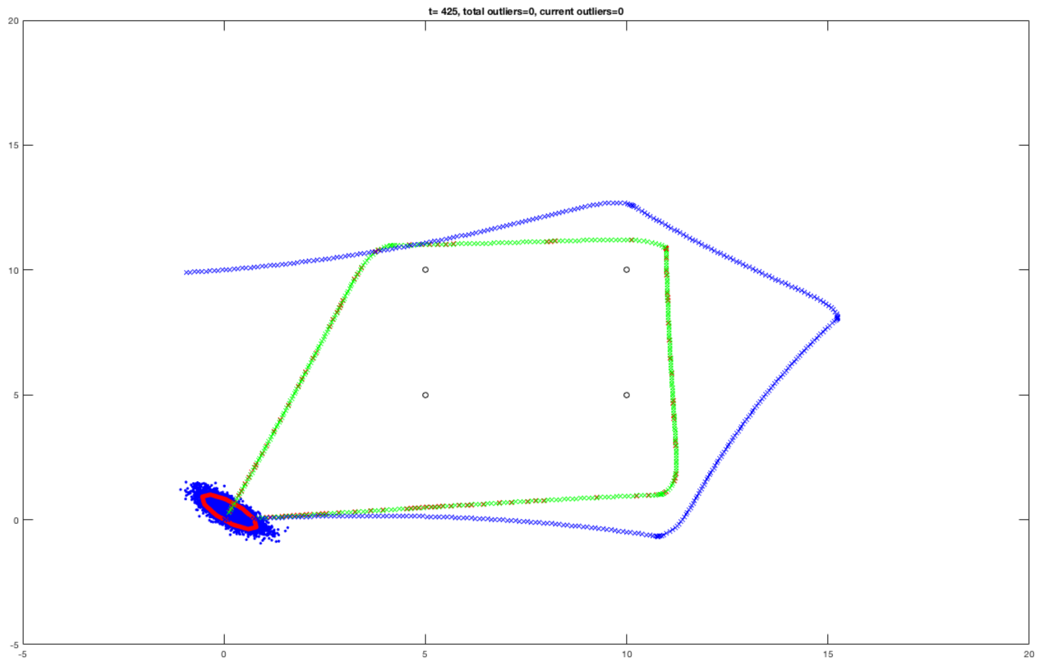

Figure 2: tracking task with 10000 particles

In order to keep all valid hypotheses, we need to increase the value of the `part_bound` variable because this variable is responsible for setting the bounds of area where particles are spread randomly in the beginning. Figure 3-Figure 5 show the results of the localization task with 1000 particles at different time instant. We could notice that the multiple valid hypotheses concentrate on three major ones and then converge to the correct one. When we increase the amount of particles, we could notice that there should be four major hypotheses existing during the process of convergence, see Figure 6-Figure 8. This phenomenon proves that increasing particles could effectively keep the main hypotheses because more particles are able to cover more areas in the state space and prevent particle deprivation.

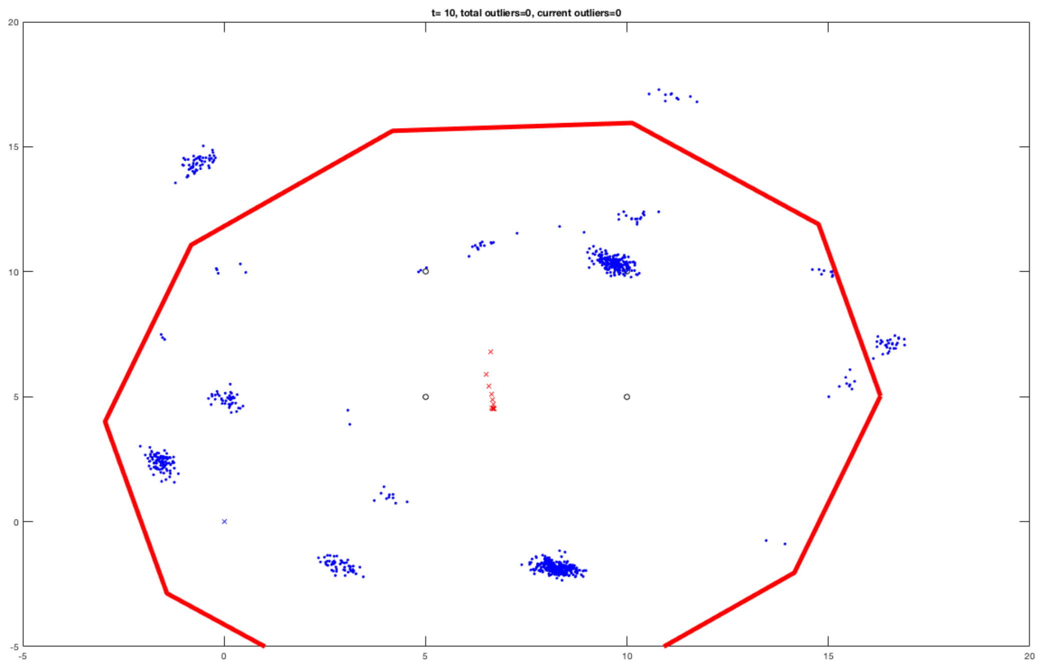

Figure 3: localization with 1000 particles at t = 10

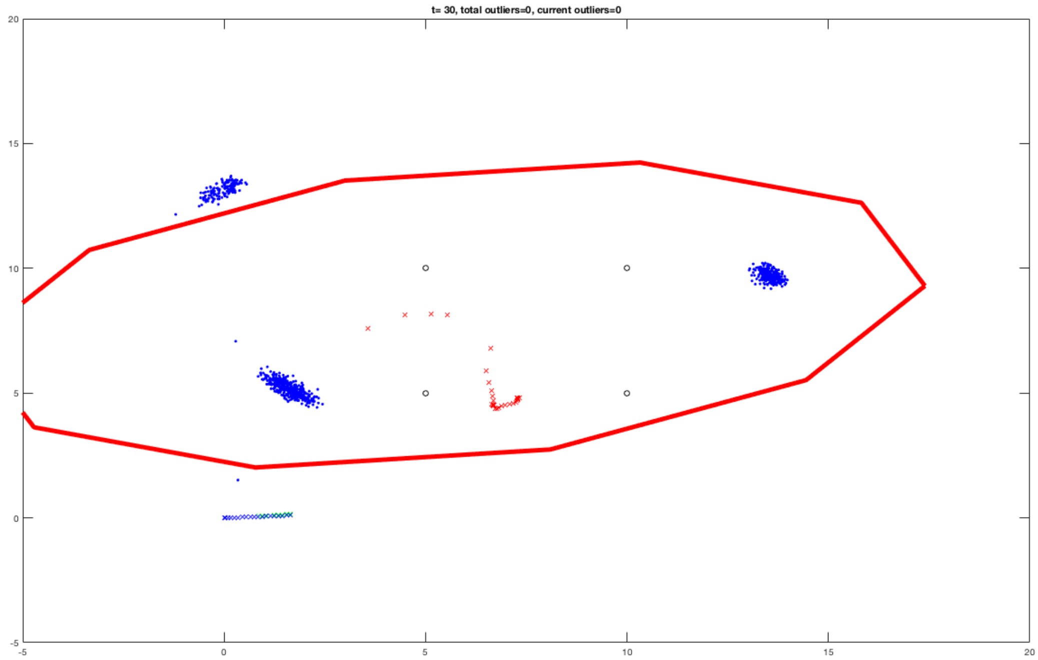

Figure 4: localization with 1000 particles at t = 30

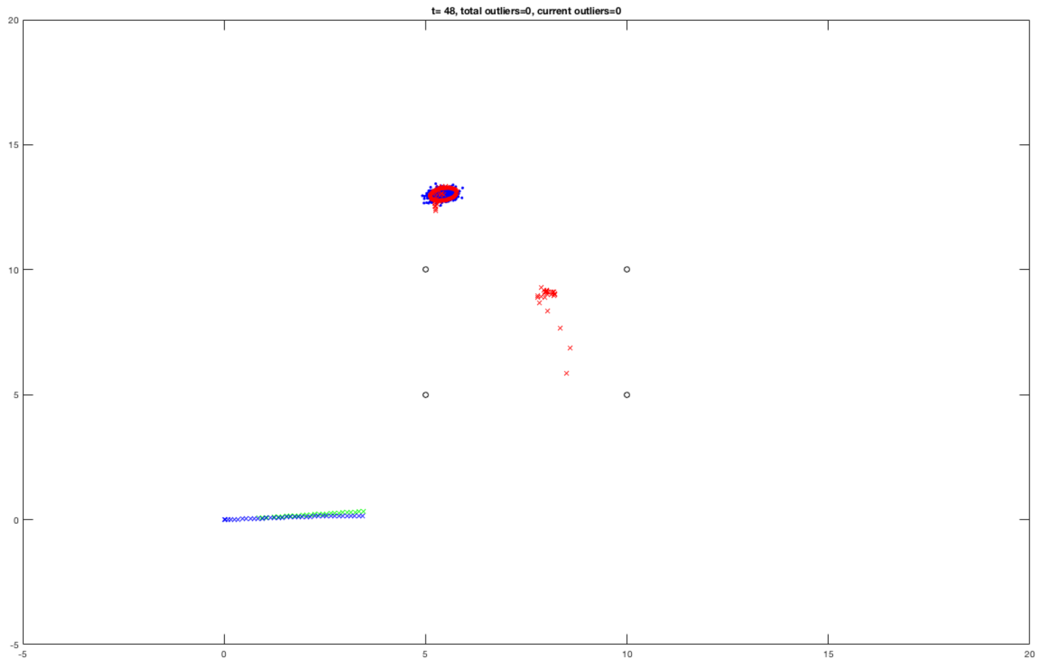

Figure 5: localization with 1000 particles at t = 48

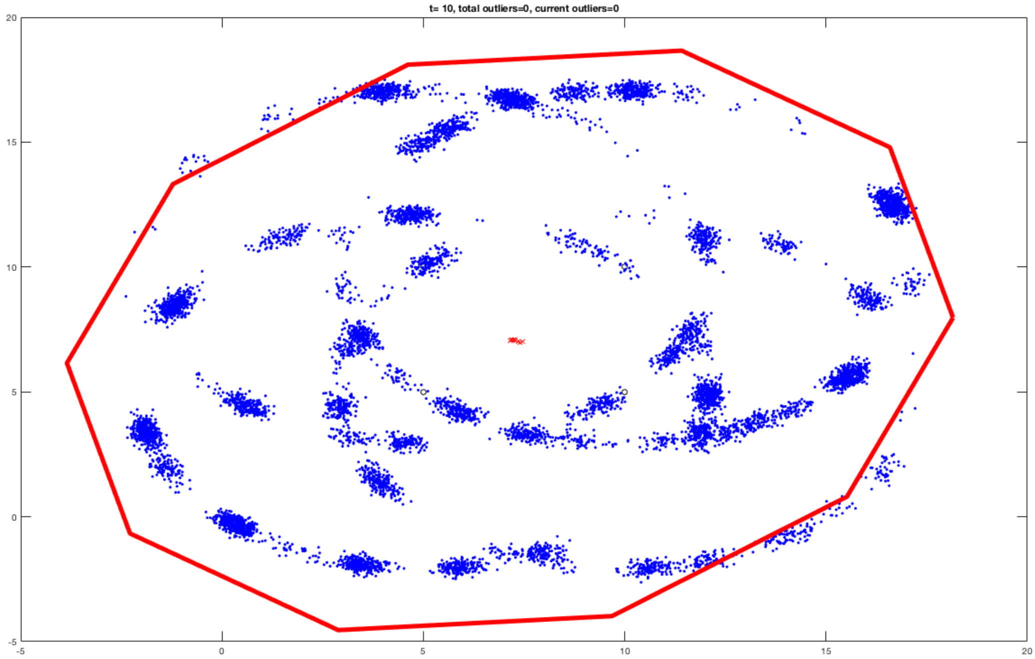

Figure 6: localization with 10000 particles at t = 10

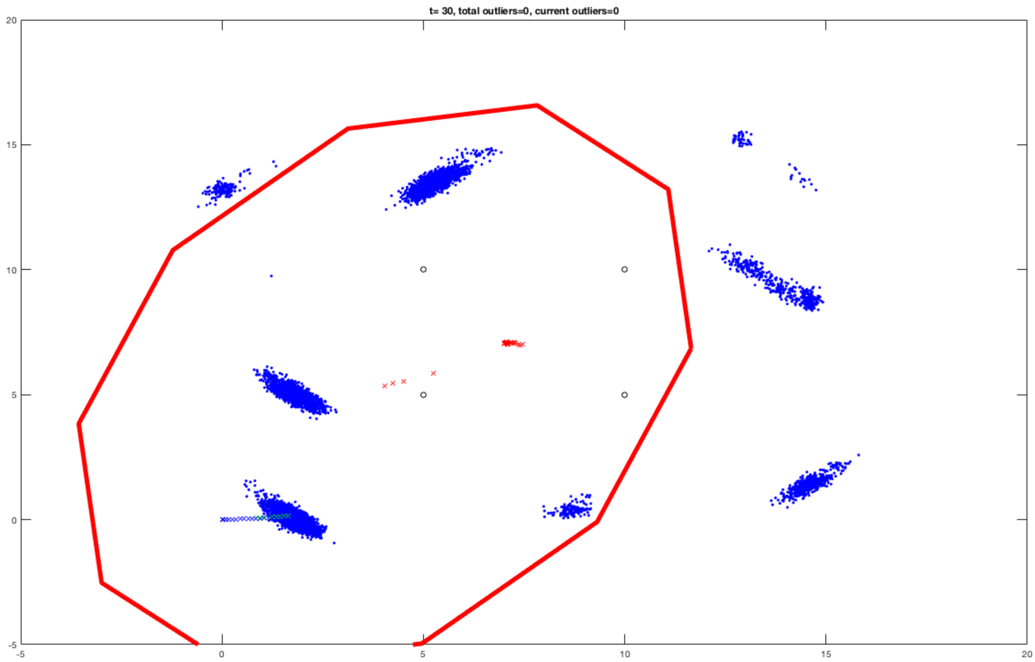

Figure 7: localization with 10000 particles at t = 30

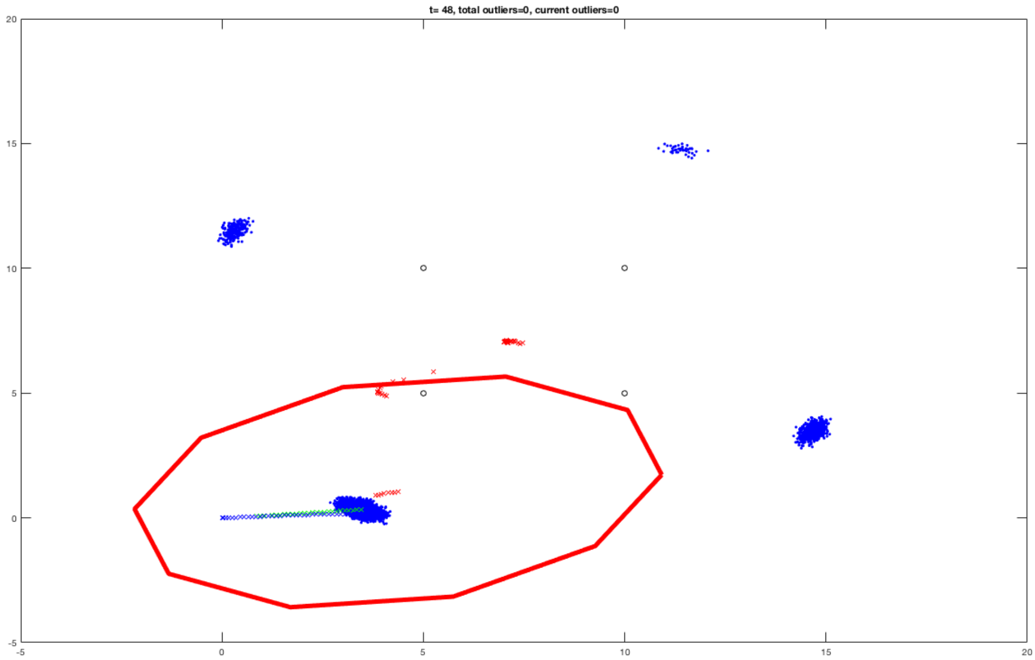

Figure 8: localization with 10000 particles at t = 48

When vanilla re-sampling is used in particle filter, particles concentrate on the main hypotheses in a quicker manner since vanilla re-sampling tends to choose those samples with higher weights. And the convergence of the particles becomes more rapid, which shows a weak ability to preserve multiple hypotheses for vanilla re-sampling. The same phenomenon will also occur when we model weaker measurement noises. On the contrary, hypotheses will be better preserved when a stronger measurement noise model is used.

2. map_sym3.txt + so_sym3_nk

Figure 9-Figure 13 depict how the particles evolve with the count t increases. In the beginning, five main hypotheses exist near the five landmarks. As the robot precedes before 180 time steps, five hypotheses are reduced to four and converge to the correct one when the robot sees the landmark which breaks the symmetry and helps the filter recover to the true state. Figure 14 shows the result of a finished localization task.

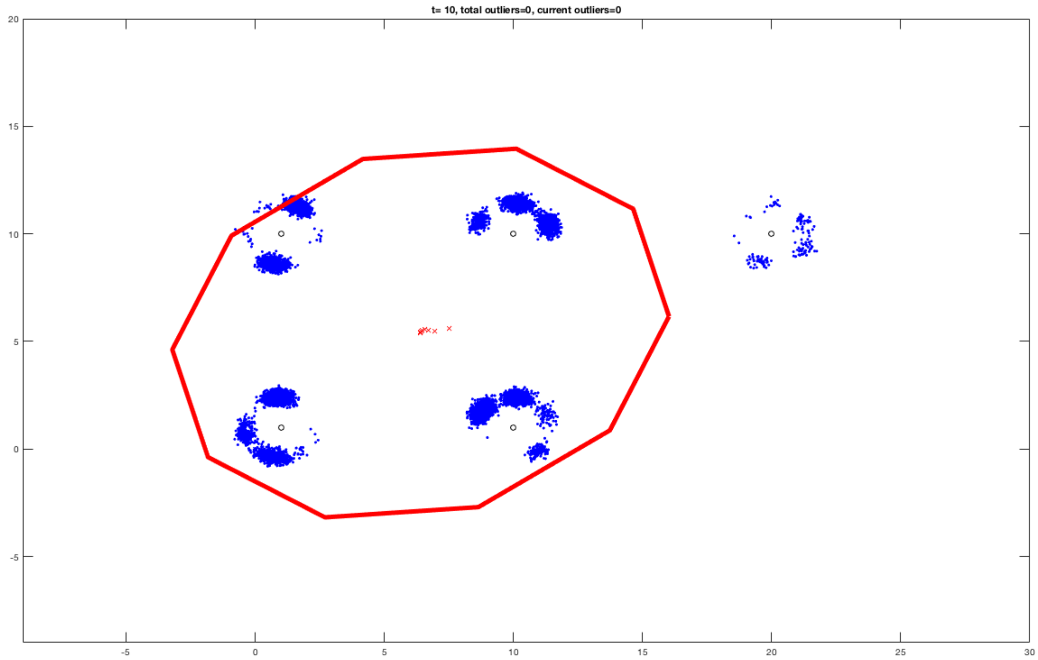

Figure 9: localization with 10000 particles at t = 10

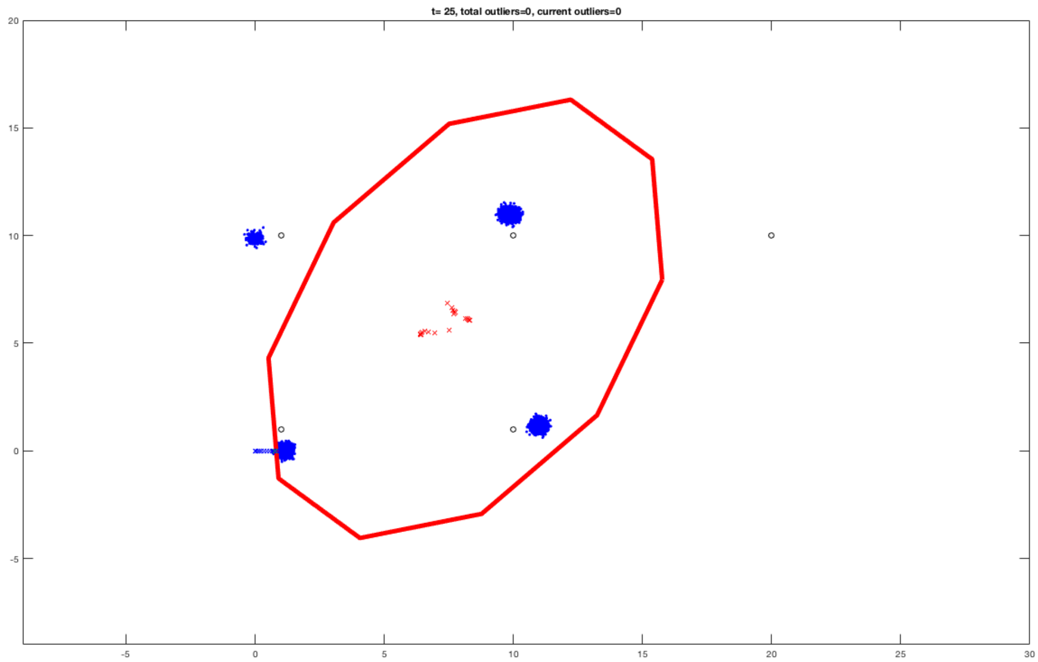

Figure 10: localization with 10000 particles at t = 25

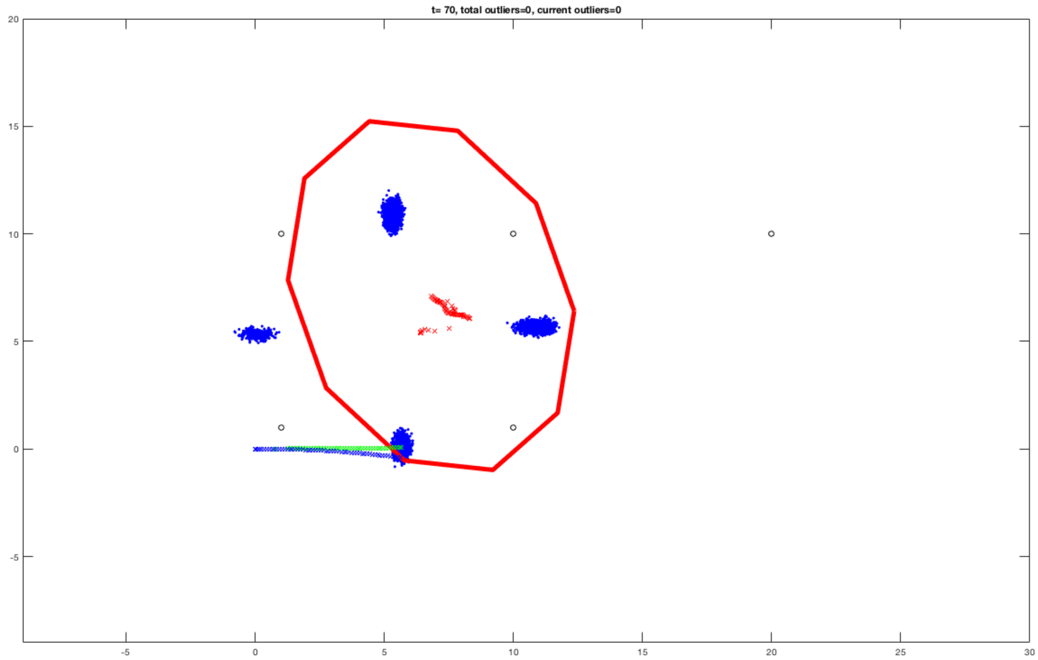

Figure 11: localization with 10000 particles at t = 70

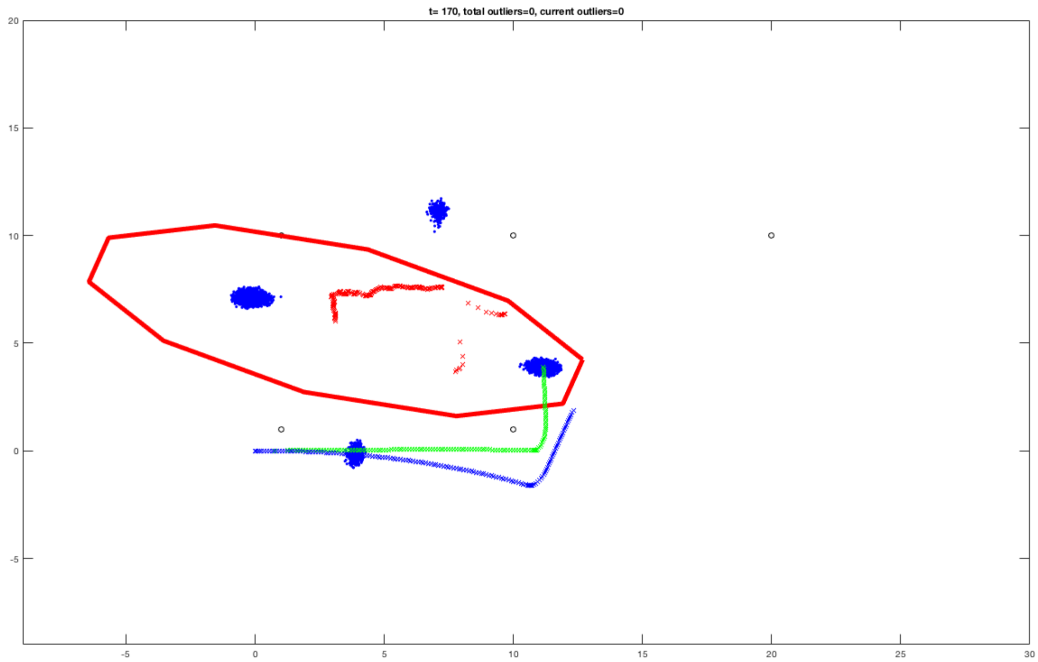

Figure 12: localization with 10000 particles at t = 170

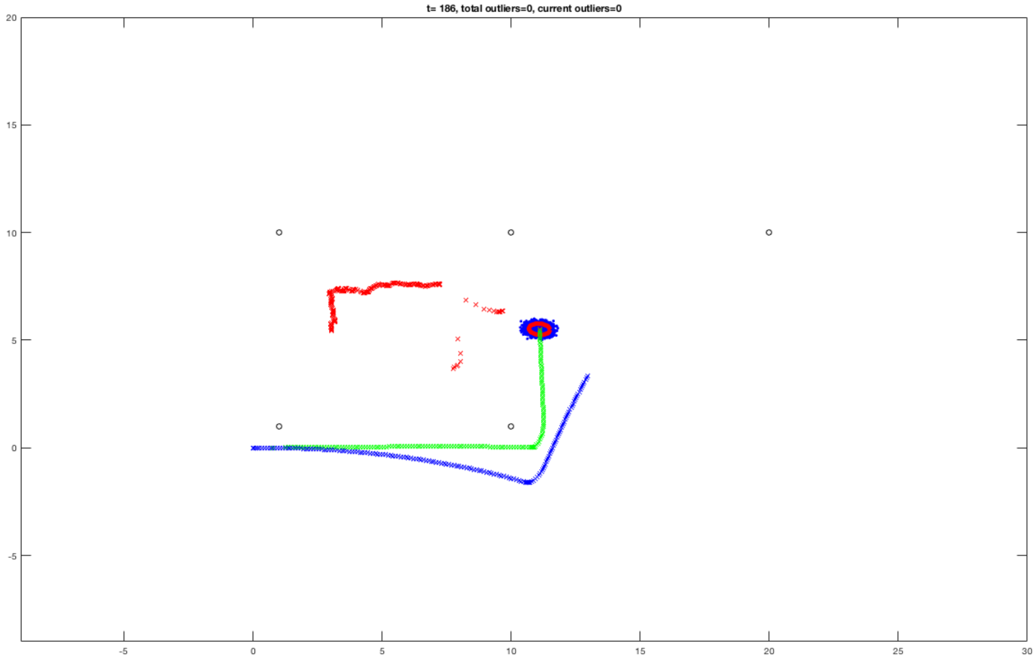

Figure 13: localization with 10000 particles at t = 186

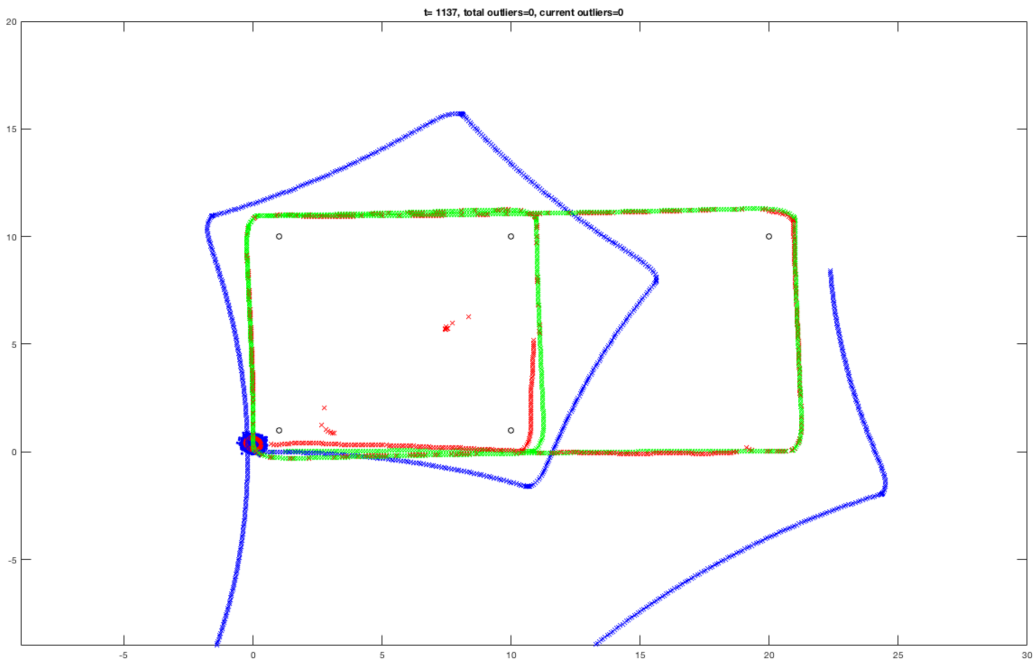

Figure 14: localization finished with 10000 particles