EL2320: Applied estimation

Diego González Morín {diegogm@kth.se}

Lab 2: Particle filters

1. Part I: Preparatory questions

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

The particles represent the different possible state hypotheses taken using a probabilistic distribution. In other words, they represent all the possible states of the system with the information given.

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

The importance weights are used to indicate how good each of the particles can represent the actual state given a measurement. Once a measurement is taken, we compare this to the measurement we would get if the actual state were the current particle. Then the particles that are more likely to get a similar measurement will get a higher weight than the ones that aren´t.

The proposal distribution, on the contrary, is built before the measurements. In other words, the distribution of the particles that were sampled during the predict step of the algorithm.

The target distribution, as it is explained in the lecture notes, is the posterior true distribution that represents the true state of the system.

They are related as we use the weights of the particles to resample the proposal distribution so it gets more similar to our target distribution.

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

Particle deprivation happens when there are no particles close to the correct state, and then the main danger is that the actual state will have a low probability to be reached. This happens in the random resampling step of the algorithm and it is more likely to happen when we have a low number of particles.

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

Resampling is an important step that allows the algorithm to converge. This step changes the distribution of the particles so the clouds of particles are around the hypothesis with higher weights so in the following time steps, these particles are more likely to represent the actual state of the system and the variance decreases with each step. Then, if we didn't resample, the particles would be more spread and the probability of any of them to represent the actual state of the system will be lower and we would need many more particles to reach the truth state of the system.

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

The problem with this method would be that if we have to different clouds of particles, or more, with similar possibilities (density of points), the average would represent some state between all these possible states and this would not be a correct representation.

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

As it was proposed in the lecture notes, we have several possibilities:

1. We can fit a Gaussian to the mean and variance of the particle set so we get a continuous distribution
2. We can create bins and count how many particles are in each bin to form a histogram and then compute the interpolation between bins.
3. We can place a so called kernel around each particle, again, to get a continuous distribution.

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

The sample variance represents the difference between the particle distribution and the target one. The main problem would be that if this value increases to much, our representation of the true state would not be good enough. We need then to make sure that this variance decreases on each iteration. We can achieve this using the following two methods:

1. Delayed resampling: in every iteration it keeps the weight of each particle as part of the description for the next iteration so the re-sampling step is done less frequently.
2. Low variance sampling: samples selection depends on one random number for every iteration step. Then the samples are not independent from each other.

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 the pose uncertainty is high, the number of possible hypothesis in the next step would be higher than if the pose is precisely known. Then we would need more particles to make sure that we avoid deprivation and all the possible hypotheses are taken into account. This said, the higher the pose uncertainty is, more number of particles we need.

2. Part II: Matlab exercises

2.1. Warm up problem with the Particle filter

2.1.1. Predict

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

If we compare both equation (6) and (8) we can see that in the (6) equation, the angle θo is prefixed and is always the same in every time step. On the contrary, the (8) equation's angles, changes at every time step. This means that we can change the value of this last angle with the noise (R matrix) that we are adding. This way we can represent more complex dynamics that just using (6).

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)?**

This model is expressed by the next formula:

$$\bar{u}_t = \begin{bmatrix} dx_t \\ dy_t \\ d\theta_0 \end{bmatrix} = dt \begin{bmatrix} v_0 \cos x_{t-1,\theta} \\ v_0 \sin x_{t-1,\theta} \\ \omega_0 \end{bmatrix}$$

We can see that v_o and w_o are fixed and constant. This means that the radius remains constant as the linear velocity is constant. We can only model then perfect circular movements with constant radius and angular speed.

2.1.2. Sensor model

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

The constant part in the denominator of the likelihood function is used to normalize this function.

2.1.3. 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?**

If we observe the pseudo code of the algorithms we realize that in the Multinomial method the generation of the random number is done inside a loop, while in the Systematic method it is generate outside the loop. Therefore, the Multinomial method generates M random numbers while the Systematic only one.

Question 5: **With what probability does a particle with weight w = 1/M + e survive the re–sampling step in each type of re–sampling (vanilla 6 and systematic)? What is this probability for a particle with 0 ≤ w < 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 subtracts that amount from 1.0 to find the probability of surviving.**

In the case of the vanilla re–sampling, the probability of a particle being sampled in an iteration is w_i, then the probability of not being sampled is $1-w_i$. The loop is iterated M times, then the probability of not being sampled during the entire loop iteration is $(1 - w_i)^M$. Then, I we subtract this to one, we get the probability of surviving:

$$1 - (1 - w)^M$$

This is true for both

$$w = \frac{1}{M} + \epsilon \; and \; 0 \leq w \leq \frac{1}{M}$$

If we analyze the Systematic re–sampling method, in the case that w = 1/M + e we can conclude that the particle will always survive. This is because the CDF function will always be bigger than the generated random number, as this number is

$$\mathrm{ro} = \mathrm{rand}\{0 \leq \mathrm{ro} \leq \frac{1}{\mathrm{M}}\}$$

and of course is smaller than w = 1/M + e.

In the case of $0 \leq w < 1/M$, it now depends of both the weight and the size of M. It depends on the size of M as r0 grows in each iteration of the loop as it is expressed:

$$r0 + \frac{m-1}{M}$$

Then, the bigger M is, the bigger the probability of CDF being greater than this number and then, the particle of surviving. It is also proportional to the weight w, of course, so the probability of surviving would be proportional to M*w.

2.1.4. Experiments

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

`params.Sigma_R` models the process noise and `params.Sigma_Q` the measurement noise.

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

If we don't perform the diffusion step, which means adding noise to the particles in each time step, the particles that survived the previous step would make the exact same movement than the one they were sampled from. Then, all the particles that were subsampled will be at the exact same position as the particle they were subsampled from. Due to this, after a several subsampling, we will only have one useful particle, and the rest of the particles will be in the same position as this particle. This useful particle will be the one with higher weight in the first subsampling.

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

Logically, we can observe that the particles will never converge as we are not using the weights of the particles to re−sample them. This is why, in every time step, we get a new set of particles with the same random distribution that was chosen for the first time set. However, these particles will move accordingly to the motion model plus the added noise.

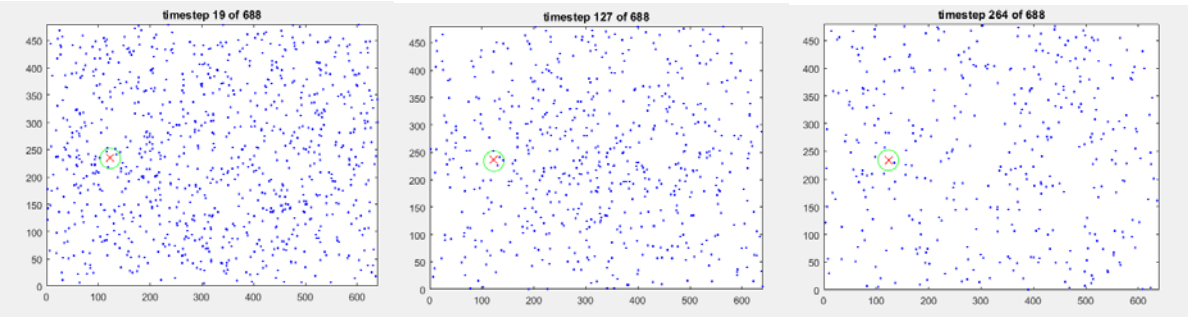


Figure 1: particles distribution on time step 19, 127 and 264 respectively.

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)**

Observing the figure 2 we can observe that for lower values of the measurement covariance, the particles won't converge. For higher values than 1 for the covariance the particles will convert, but can also observe that the particles get more spread if we increase this value.

This is because, if we have low values of the measurement covariance it would mean that out measurements are really accurate and the outliers will not be classified properly, making the particles to never converge. On the other side, if the covariance is set too high, the value of noise distribution will be bigger. Due to this, more particles are likely to have the same weights making the cloud of particles be more spread.

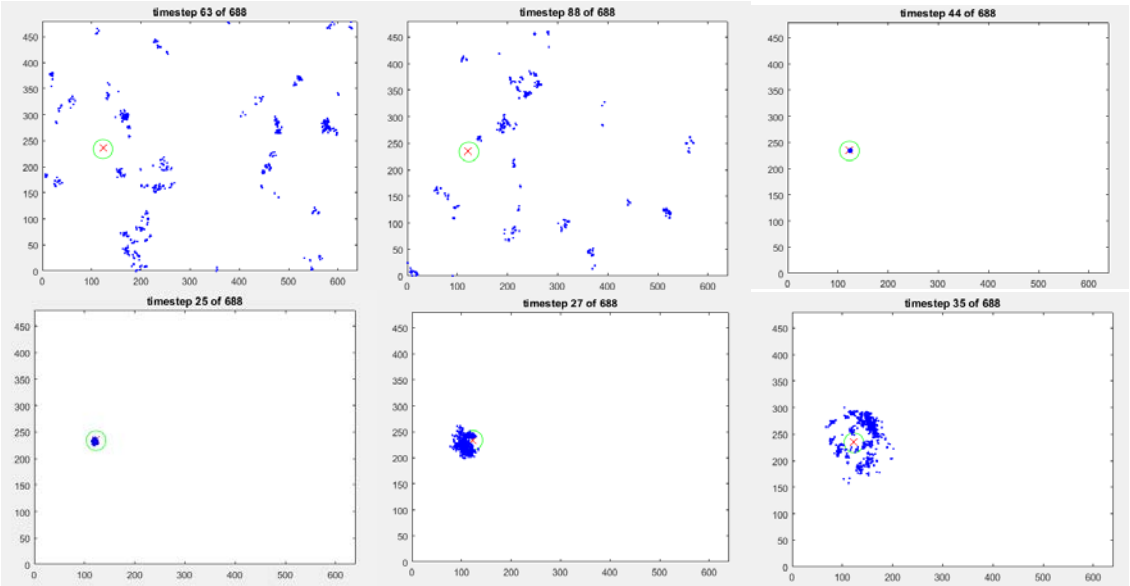


Figure 2: Particle distribution for measurement noise covariance 0.0001, 0.1, 1, 10, 100, 1000 respectively

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)**

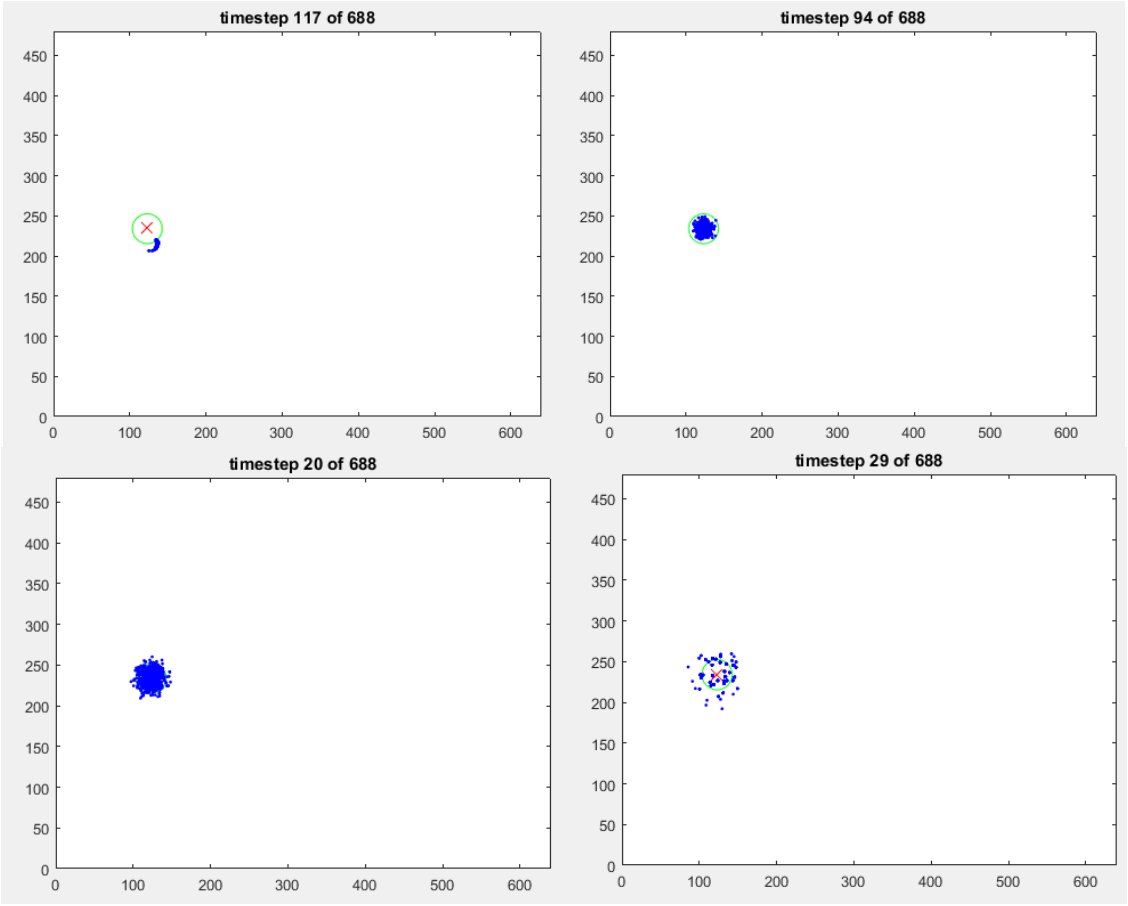


Figure 3: Particle distribution for process noise covariance 0.01, 1, 10, 1000 respectively

In this case, as we can see in the figure 3, for low values of process covariance, the particles are really condensed around the particle with greater weight and it will need a lot of time steps to converge. For high values the particle cloud will converge much faster but will be more spread as this value increases.
The reason why the cloud is more spread for higher values of the covariance is the same as in the question 9, the noise distribution and consequently the diffusion is bigger and the particles are able to accomplish greater movements in the space during the diffusion step. On the contrary, if the diffusion is small, the particles will only be able to do small motions, and they will be much closer together, around the particles with the higher weight. Similar as it was happening in the question 7.

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

The choice of the process noise depends on how good the motion model matches the simulated model. If the motion model is not good enough we will need to add

more noise to the process to spread the particles cloud. This way it will be more likely to have the particles cloud around the truth state. In conclusion, the better our motion model is, the less noise amount of noise we need to add to the process.

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?**

As a consequence of the previous answer, if our motion model is not accurate enough we will need to increase the diffusion, spreading the particles cloud. We will need then more particles, to make sure that we have enough particles around the truth state.
And of course, a good choice of the motion model will lead to a higher accuracy of the filter.

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?**

We can detect the outliers by setting a threshold for the likelihoods of the observations. If this likelihood values is to low we can reject it for being 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?**

After playing around with the different parameters and motion models I conclude that the fixed motion get the worst results. Being the circular and the linear the best options. Both of them gave similar output errors being the circular one a little better. We also need to add more noise to reach to the lowest error possible for the fixed motion model thant for the circular and linear.

2.2. Main problem: Monte Carlo Localization

2.2.1. Outlier detection

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 and the measurement noise both affect the mentioned outlier detection approach. A weak measurement noise means that the measurements are supposed to be extremely precise. Then, a lot of measurements will be mistaken as outliers and wrongly rejected.

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

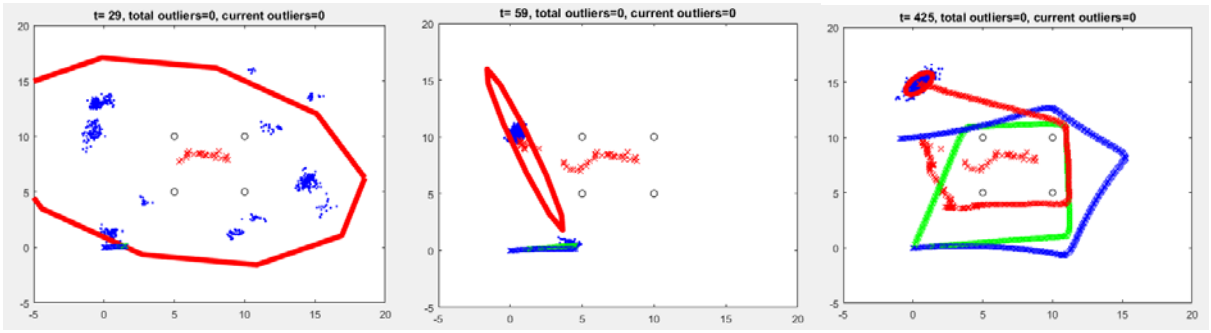
The main problem that appears as a consequence of a not detected outlier is that the filter will assign wrongly the weights to the particles, given more weight to the outliers. This will have consequences in the next time step, and the algorithm will need more time to converge.

2.2.2. Data sets

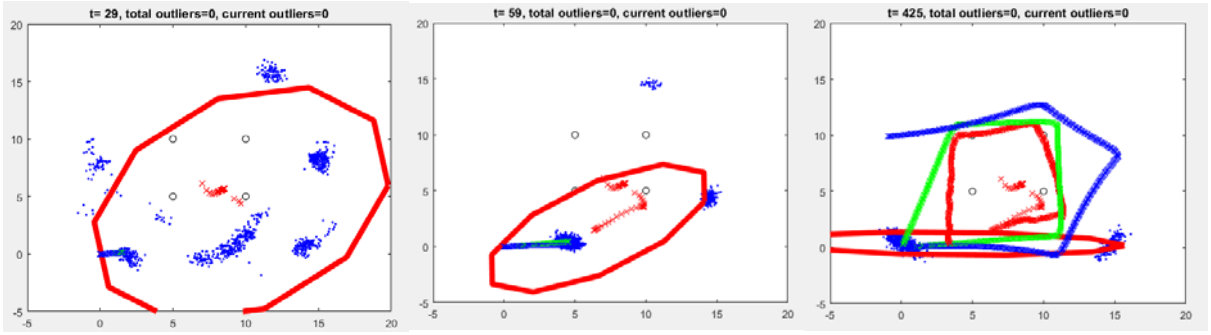
Data set 1**: map _sym2.txt + so_sym2_nk.txt**

The map used for this first test is a perfectly symmetric scenario with 4 landmarks. Due to this, we have 4 valid hypotheses. To guarantee that all of these hypotheses survive, we need to increase the particles bound parameter as if this boundary is not great enough some hypotheses can lay out of this boundary and will not survive.

The first experiments with these map and simulation files were done with 1000 particles. The problem was that with this amount of particles not all the hypotheses are surviving (deprivation). In the figure 4 we can see how the algorithm converges both for the multinomial and systematic method.



(a)

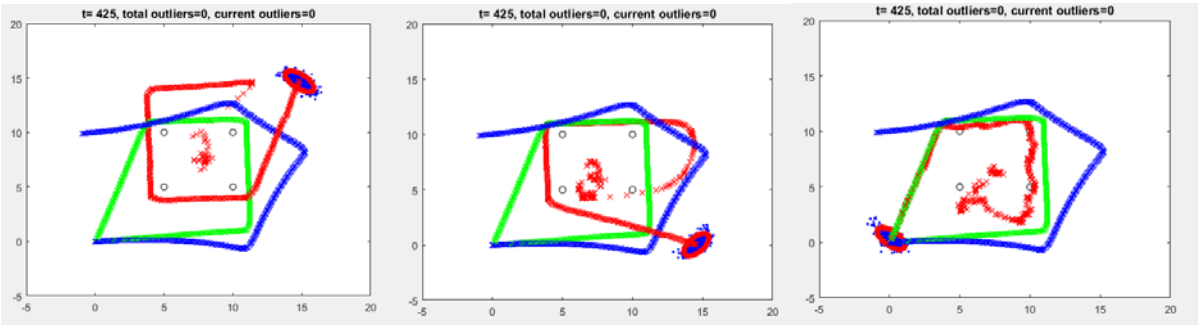


(b)

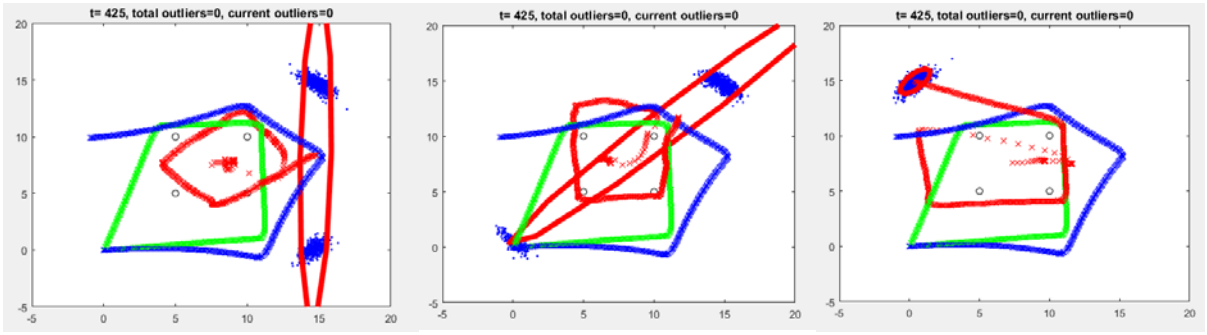
Figure 4: Output for (a) multinomial resample method and (b) systematic resample method for n = 1000, R = 0.1 and Q = 1

We can also see in the systematic method, more hypotheses survive. This is because the multinomial method, as it updates the random number on every loop step, gives more importance to the particles with higher weights while the systematic method, as it uses the same random number for every particle, the distribution of surviving particles is wider, and then there are more possibilities that the more valid hypotheses survive.

In the figure 5 we can observe how the algorithm converges to one (multinomial) or two (systematic) different hypotheses every time we run the algorithm. This makes sense as all the 4 hypothesis are equally valid and can be chosen with the same probability. We are not using enough particles to avoid the particle deprivation effect; this is why the algorithm is converging to one or two of the hypotheses instead of preserving the 4 of them.



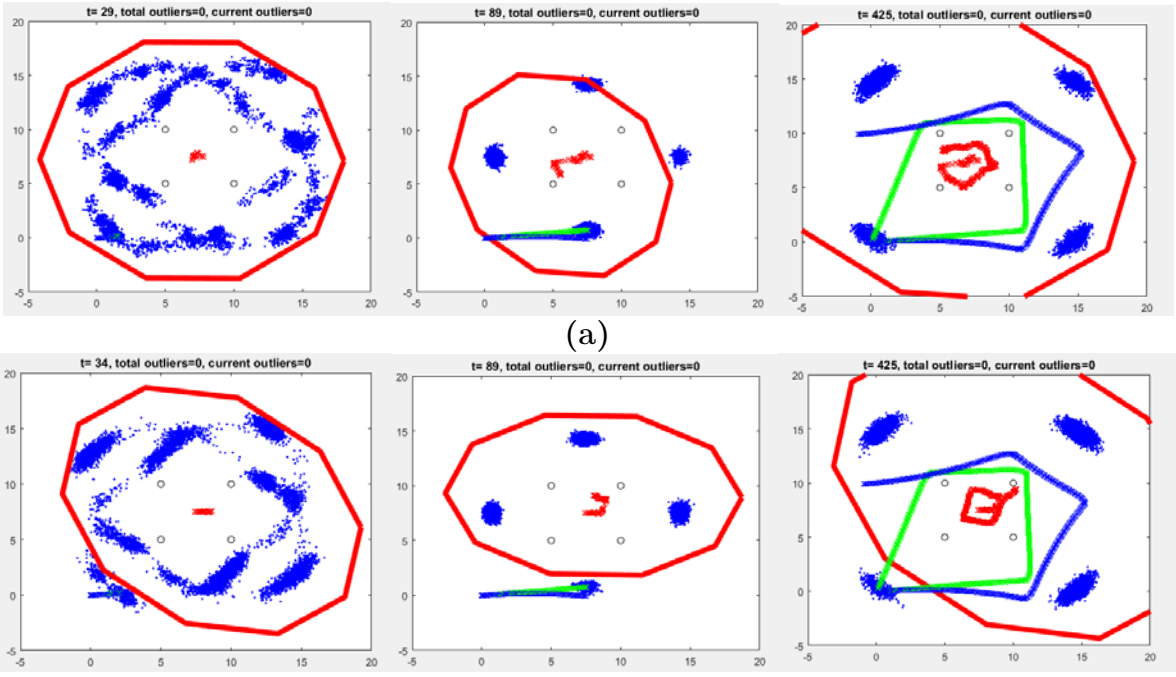
(a)



(b)

Figure 5: Output for several runs of the code for (a) multinomial resample method and (b) systematic resample method for n = 1000, R = 0.1 and Q = 1

Now we want to test what happens if we increase the number of particles to 10000. By observing this figure we can conclude that with the correct choice of the parameters, and n = 10000 we can make all the hypotheses survive for both methods.



(a)

(b)

Figure 6: Output for (a) multinomial re–sampling method and (b) systematic resample method for n = 10000, R = 0.1 and Q = 1

The last step of our experimentation would be to see how our results change when we change the value of Q. We can observe, as we conclude in the previous questions, that for higher values of the noise covariance, the particle clouds are bigger. Due to this, the particle depravation effect is smaller and then, more hypotheses are preserved. We can observe that with the systematic method, all of the hypotheses survive now.
On the contrary, lower values of the variance mean smaller clouds and then, for both methods, only one hypothesis will survive.

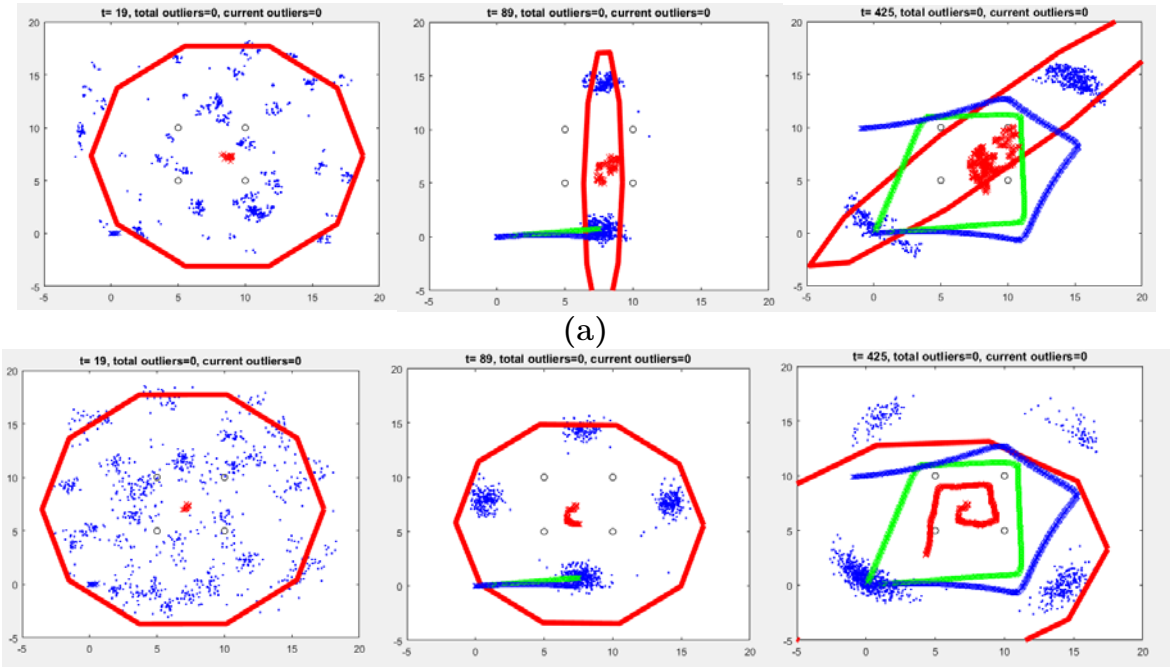


Figure 7: Output for (a) multinomial re−sampling method and (b) systematic resample method for n = 1000, R = 0.1 and Q = 10

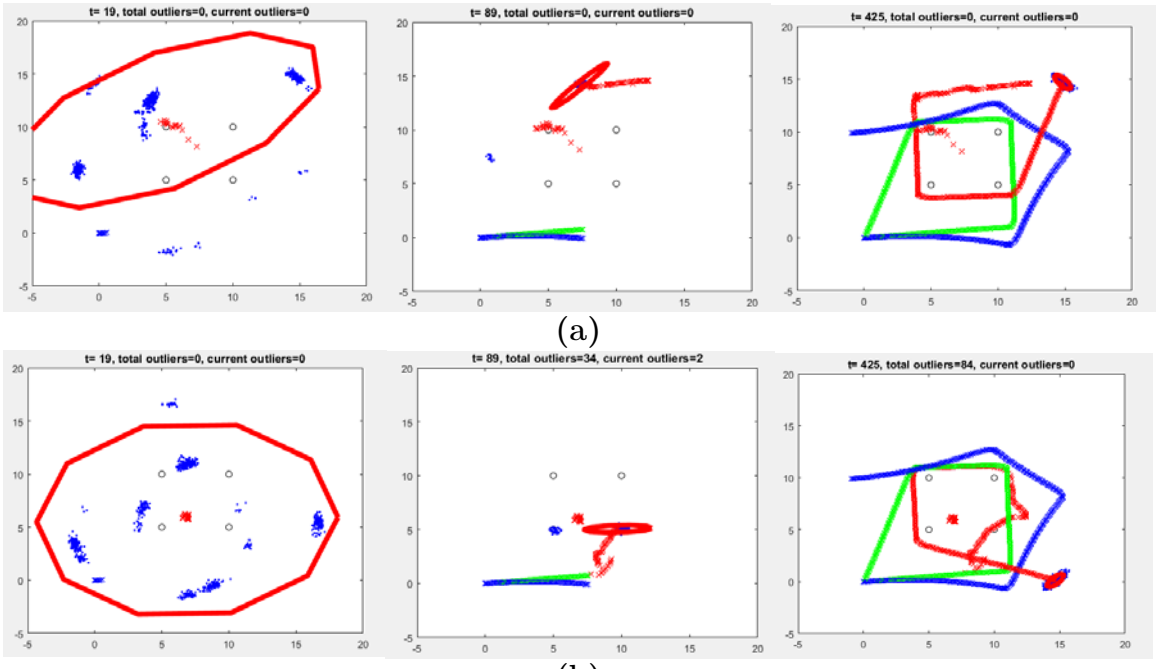


Figure 8: Output for (a) multinomial re−resampling method and (b) systematic resample method for n = 1000, R = 0.1 and Q = 10

Data set 2: **map_sym3.txt + so_sym3_nk.txt**

The map now is not symmetric, so when the 5th landmark is perceived we only have 1 valid hypothesis and the algorithm should converge to this valid solution. In the figure 9 we can see the output for n = 1000 using the systematic resampling method (the multinomial was not converging well). We can observe that, right after this 5th landmark is detected; the algorithm converges and remains converging until the 5th landmark is detected again. Due to the configuration of the map, some particles gain more weight and the algorithm starts to diverge. When the robot turns, and again due to the distribution of the landmarks, the algorithm converges again as we again only have one valid hypotheses now.
Is also remarkable that in the first steps of the algorithm, the main possible hypotheses are visible until the robot perceives the four first landmarks and the valid number of hypothesis decreases to 4. Then, as we explained before, the fifth landmark is detected and the algorithm converges. We don't need now to use more particles saving a lot of computational time this way.

Figure 9: Output for systematic re−sampling method for n = 1000, R = 0.1 and Q = 1