Written Report - Lab Assignment 2

Quirkybot

Date: Oct. 16, 2023

4. Output

4.1 How does each of the model variance parameters affect the performance of the localizer? Give <u>possible reasons</u> for the behavior you see. The parameters are defined at the top of the file.

Answer:

1. '**LINEAR_MODEL_VAR_X**' and '**LINEAR_MODEL_VAR_Y**' represent the variance in x- and y-directions in the linear motion model. Decreasing or increasing these values will lead to less or more noise in the robot's motion estimation. Specifically, increasing these parameters will result in an elongated, elliptical shape of the distribution of particles, with one side being stretched due to the increased variance. That is to say, higher values will result in more significant uncertainty in the robot's orientation. However, if the value is set too low, it can lead to difficulties in accurately estimating the robot's orientation, as the filter may become overly concentrated and have difficulties of being conditioned on measurements.
2. **'ANGULAR_MOTION_VAR'** plays a crucial role in determining the impact of noise in the robot's rotational motion estimation. Lower values of **'ANGULAR_MODEL_VAR'** result in less noise in the estimated rotational motion. As a result, the particles are prone to follow more predictable motion trajectories. Besides, the distribution of particles will have a lower variance in their rotation angles, leading to a more concentrated distribution. Higher values of '**ANGULAR_MODEL_VAR**' might be beneficial when the robot motion involves sharp change in rotations, because a broader range of orientations will be considered given a higher variance.
3. **'SENSOR_MODEL_VAR'** represents the variance in the sensor model that is to model the errors in measurements. Increasing this parameter will allow for larger discrepancies between the predicted and actual sensor measurements. Setting it too high can lead to a decreased accuracy of the sensor model caused by the fact that the importance assignment to particles becomes less accurate.

In summary, a moderate value of variance results in a faster convergence. If the variance is extremely low, the particles will become more sensitive to the motion and measurements. On the other hand, given an extremely large variance, the particles will become less accurate. Results including the distribution of particles as well as the error profiles are presented below, along with their captions and explanations.

a. LINEAR_MODEL_VAR_X = 0.05, 0.5, 5

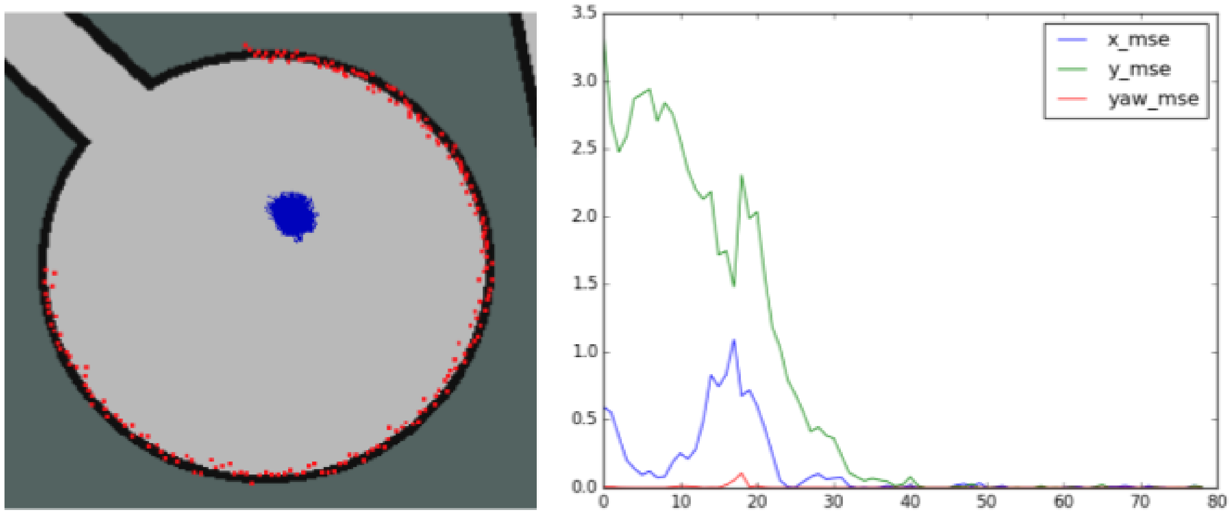

Fig. 1.1 Distribution of particles and error profiles for x-direction variance of 0.05.

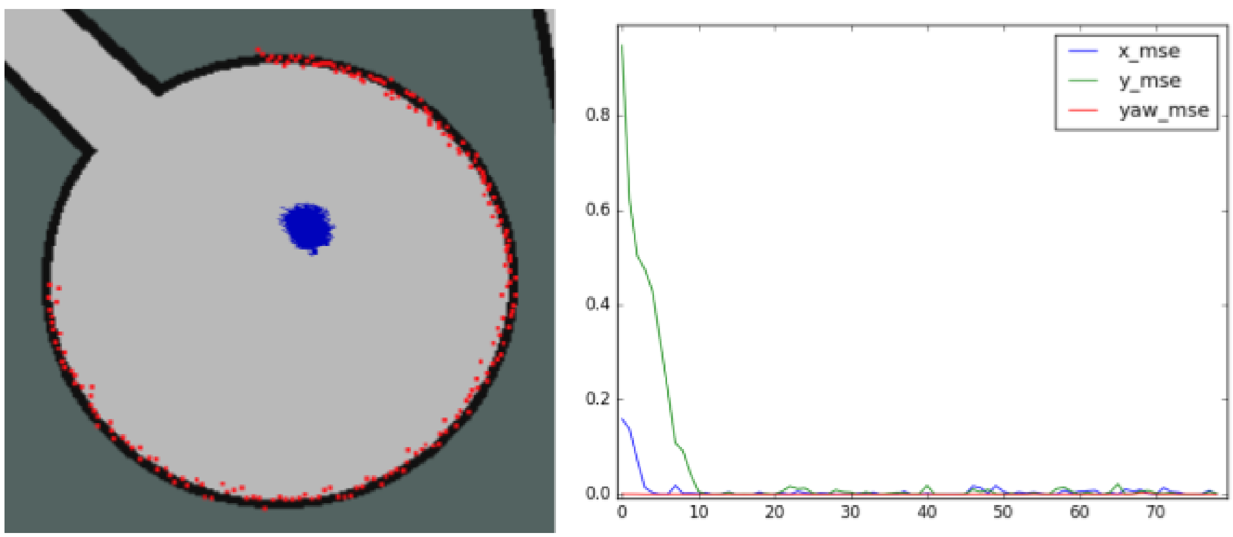

Fig. 1.2 Distribution of particles and error profiles for x-direction variance of 0.5.

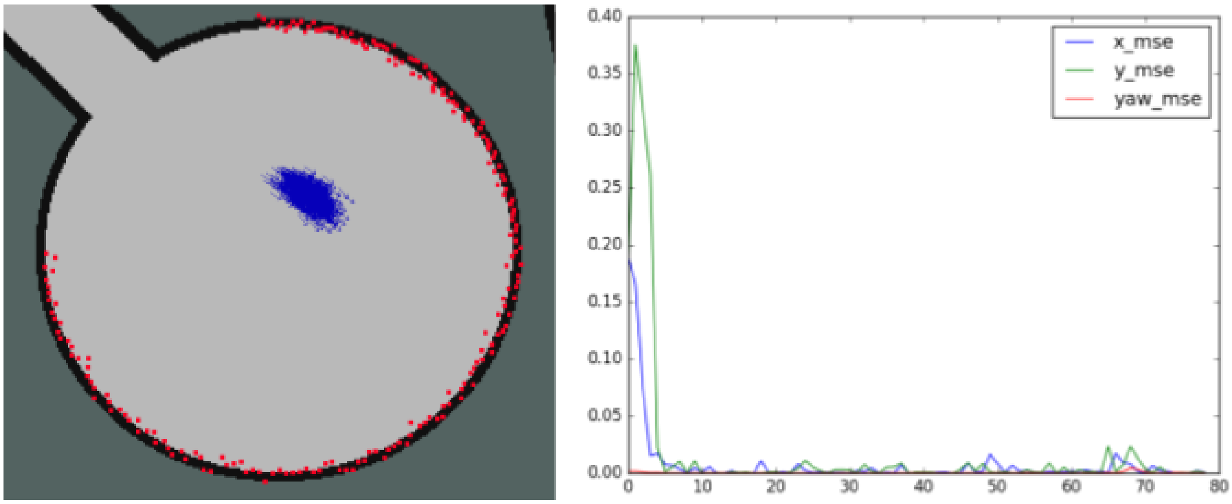

Fig. 1.3 Distribution of particles and error profiles for x-direction variance of 5.

b. LINEAR_MODEL_VAR_Y = 0.05, 0.5, 5

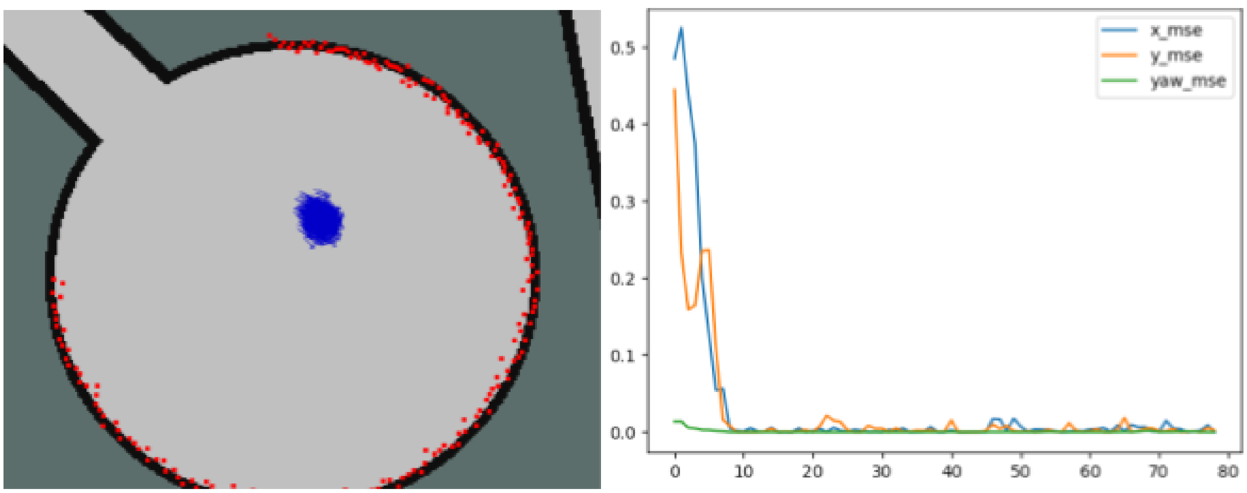

Fig. 1.4 Distribution of particles and error profiles for y-direction variance of 0.5.

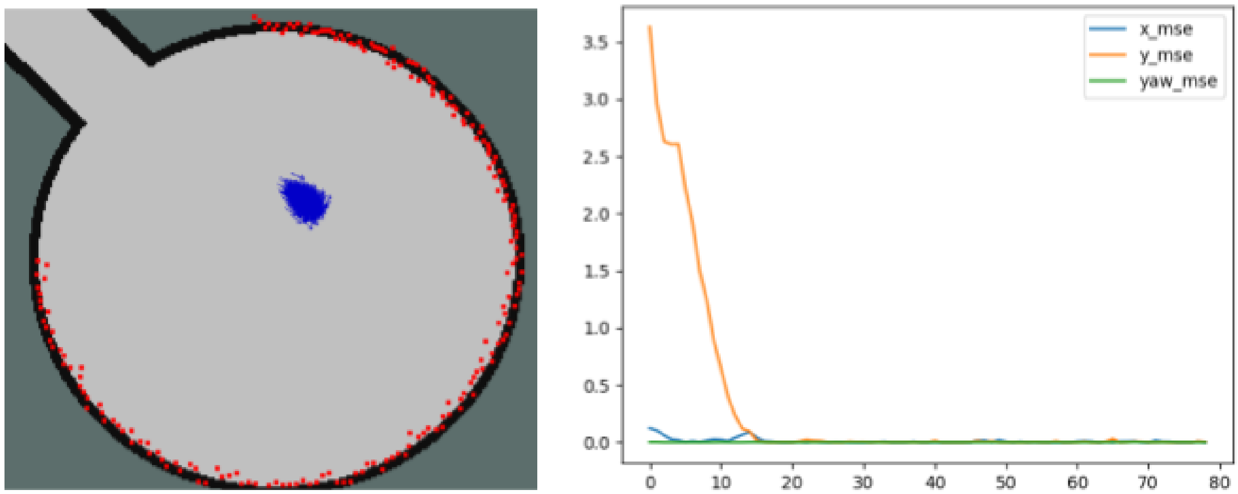

Fig. 1.5 Distribution of particles and error profiles for y-direction variance of 0.05.

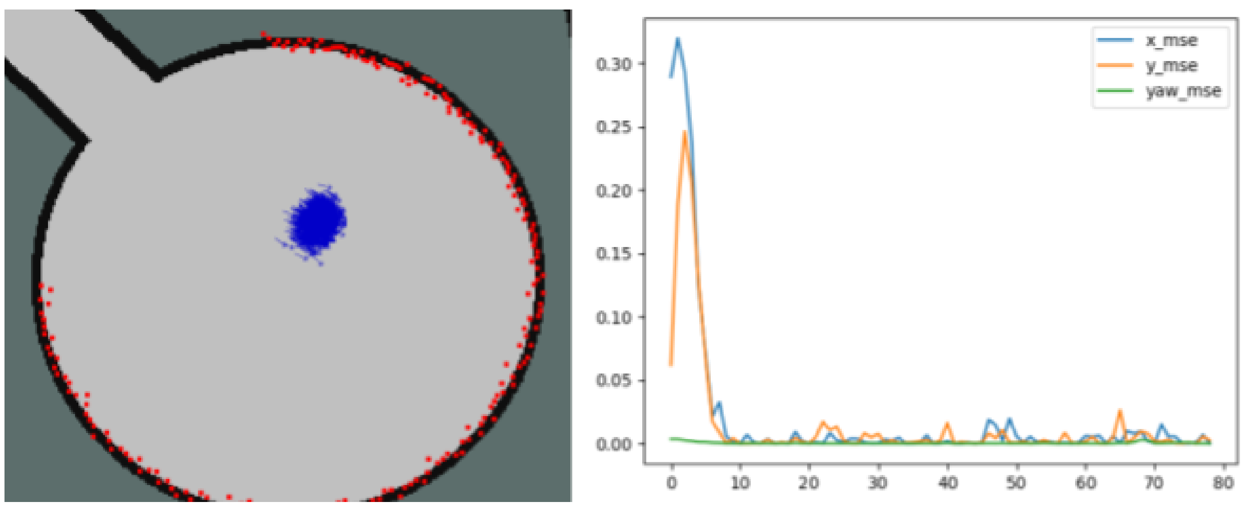

Fig. 1.6 Distribution of particles and error profiles for y-direction variance of 5.

c. ANGULAR_MODEL_VAR = 0.3, 0.03, 3, 30

'ANGULAR_MODEL_VAR' affects the orientation of the robot. As the variance gets larger, more noise gets added to the rotational velocity. As a result, the performance of the localizer on the robot's orientation drops. When the variance is 0.03, the orientations of the particles are about the same. When the variance is 30, the orientations of the particles are noticeably different from each other.

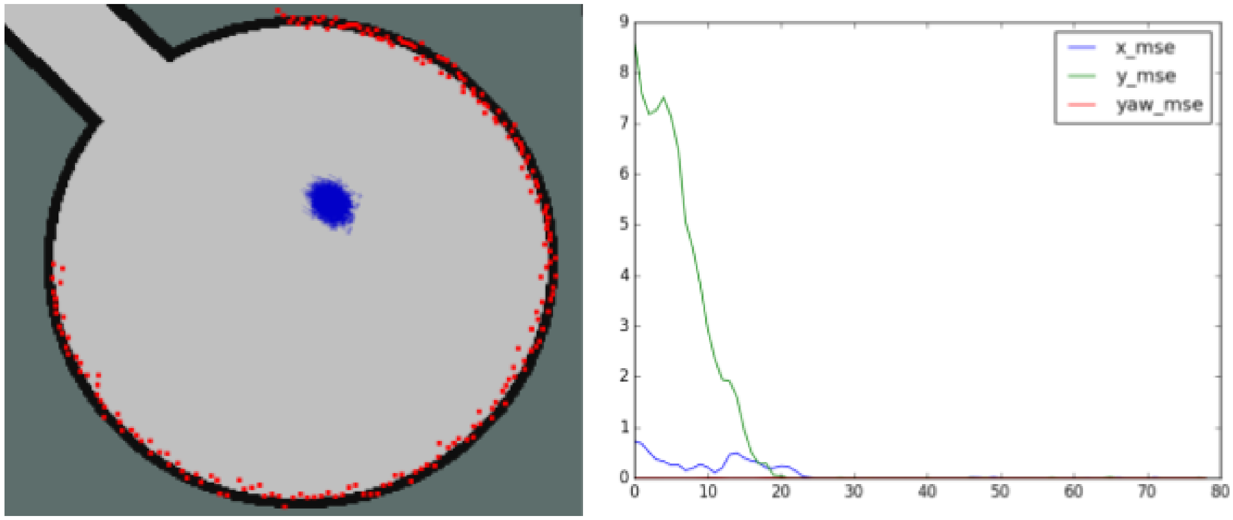

Fig. 2.1 Distribution of particles and error profiles for angular motion variance of 0.03.

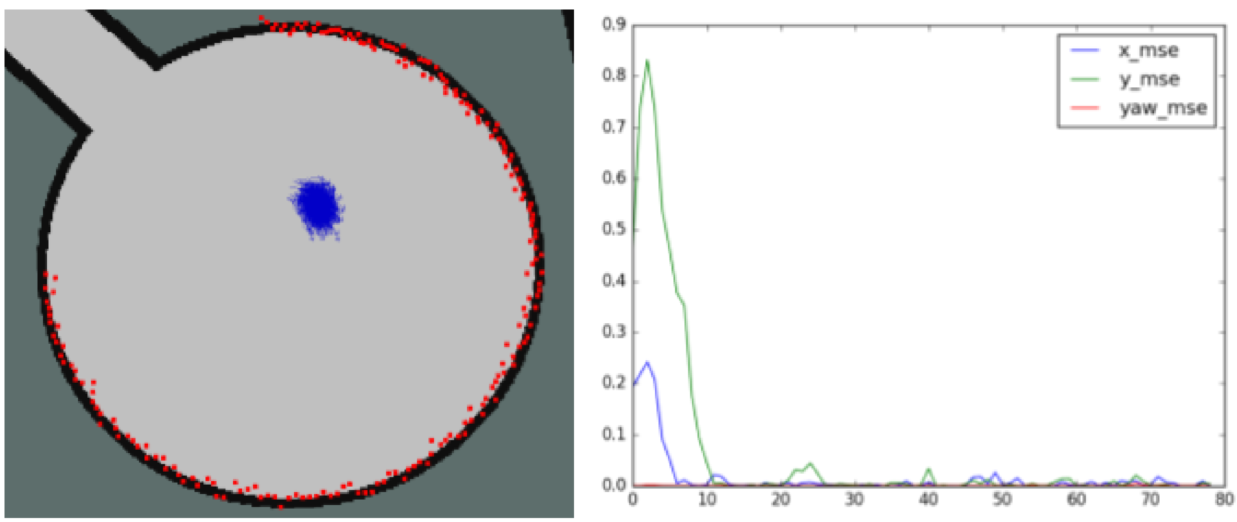

Fig. 2.2 Distribution of particles and error profiles for angular motion variance of 30.

d. SENSOR_MODEL_VAR = 1.0, 15.0, 225.0

The sensor model variance affects the spread of the particle cloud and the uncertainty of the orientation. The first image Figure 3.1 below shows the default sensor variance = 15.0.

Low Sensor Variance will run the result with a smaller particle cloud spread and more uniform orientation. As we can see in Figure 3.2 below, when sensor variance is 1.0, the spread of the particles will be more narrow, and the orientations of them are more united.

High Sensor Variance will produce the result with a larger particle cloud spread and more messy orientation. As we can see in Figure 3.3 below, when sensor variance is 225.0, the spread of the particles will be wider, the orientations of them are more deviated.

In short, low sensor variance generates a convergent result in terms of particle spread and orientation, while high sensor variance generates a robust result in terms of particle spread and orientation.

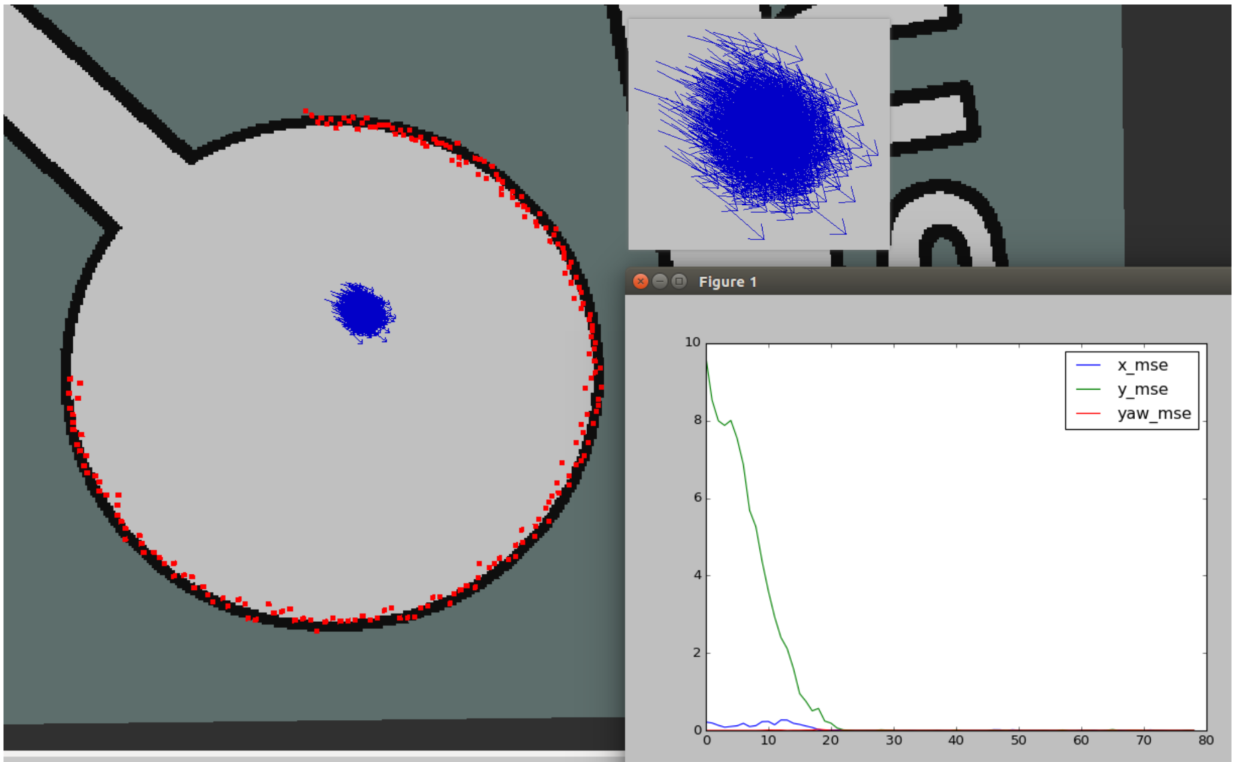

Fig. 3.1 Distribution of particles and error profiles for sensor model variance of 15.

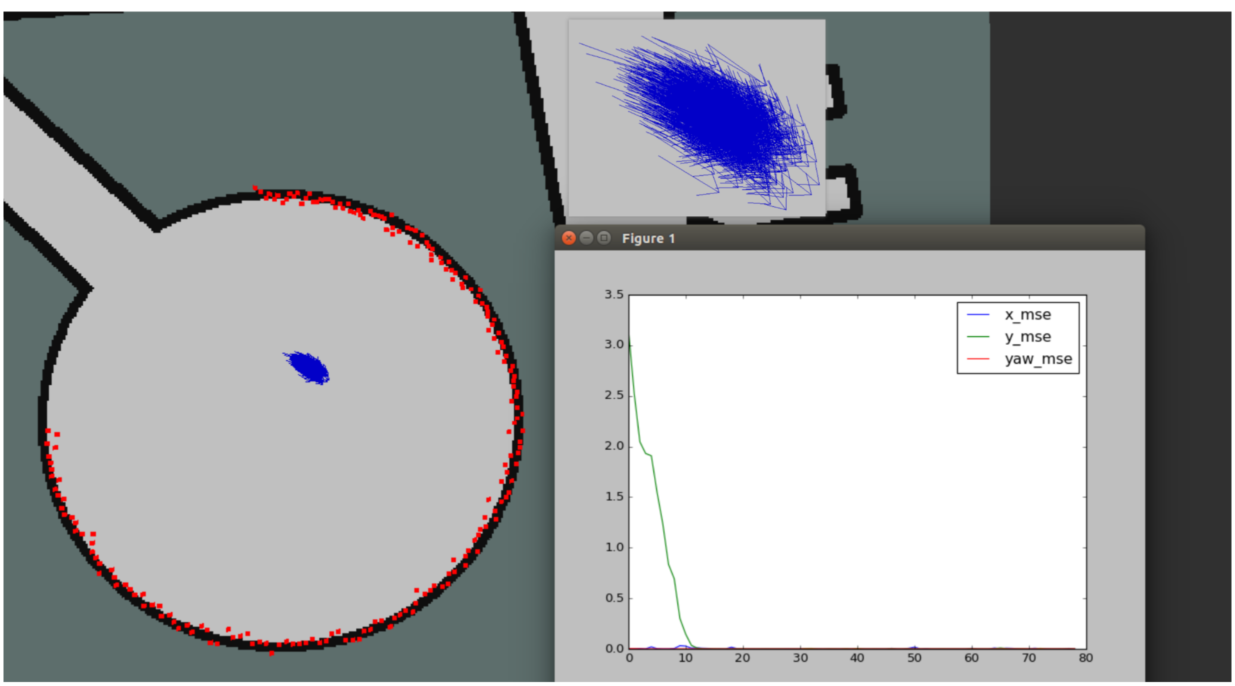

Fig. 3.2 Distribution of particles and error profiles for sensor model variance of 1.

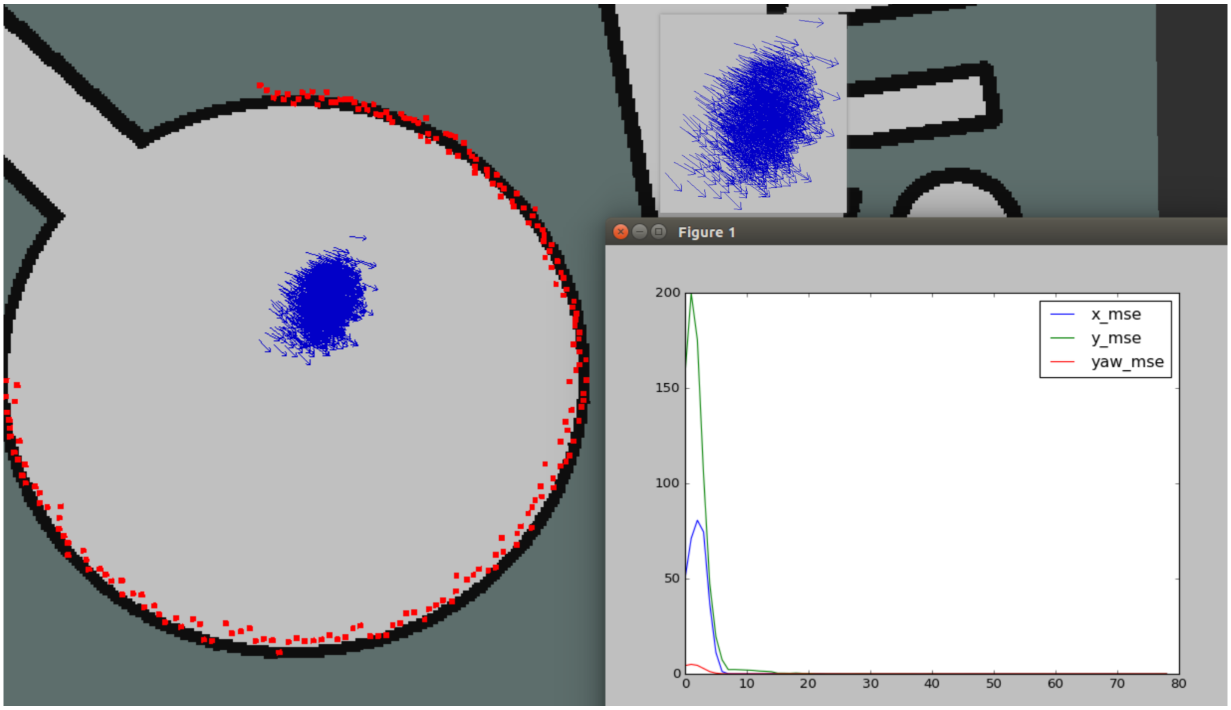

Fig. 3.3 Distribution of particles and error profiles for sensor model variance of 225.

4.2 How does the number of particles affect the behavior of the localizer? The number of particles can be modified at the top of the file.

For this Assignment we are using Monte Carlo Localizer which is based on a probabilistic filter called particle filter. Number of particles has effect on the localization:

- With the increase in the number of particles the accuracy of the approximation is easily determined and filters converge more quickly to the true robot pose. But we also need to have computational efficiency when we increase the number of particles.
- The increase in the number of particles may also lead to redundant data which doesn't add much to the accuracy and also require high computational speed.

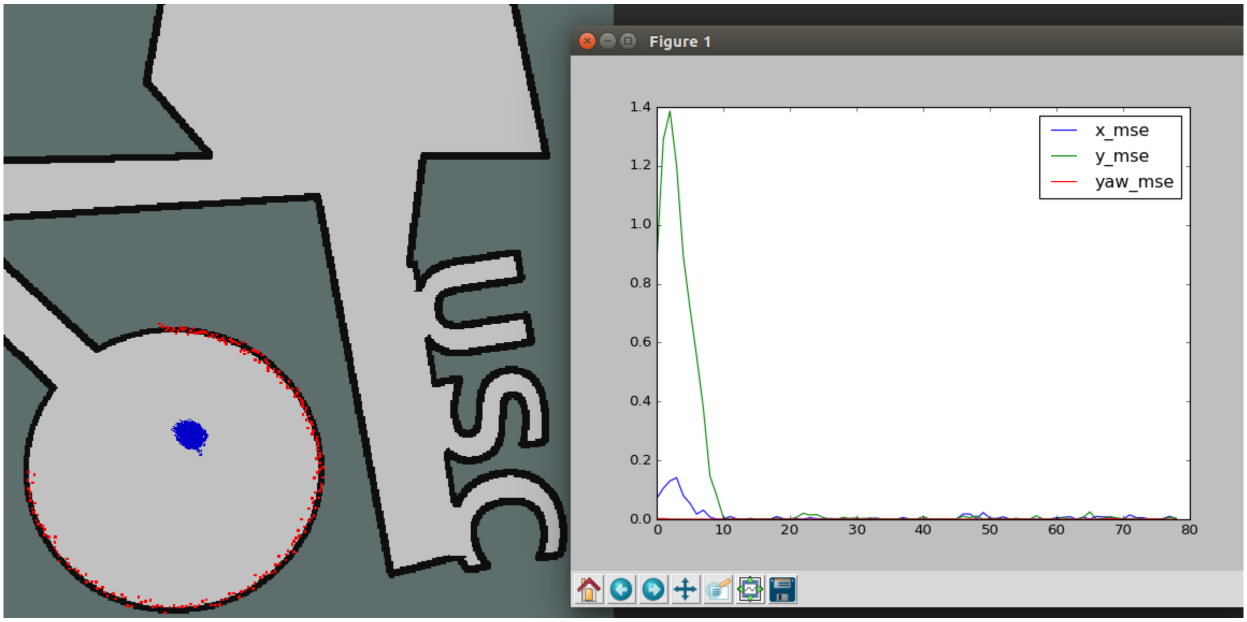

Fig. 4.1 Distribution of particles and error profiles for the case with 2000 particles.

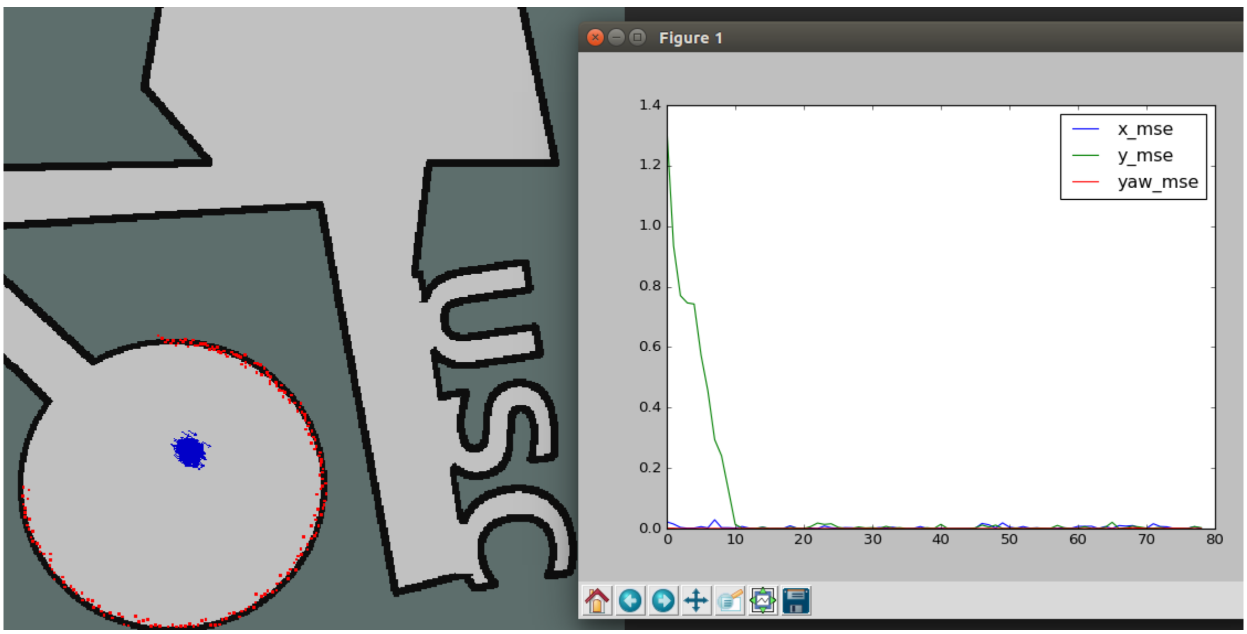

Fig. 4.2 Distribution of particles and error profiles for the case with 1000 particles.

- As seen below when NUM_PARTICLES=100 the algorithm discards all the particles near the correct pose during subsampling step and the algorithm might not recover causing the robot to end up in a incorrect location.This happens when there are fewer particles which are spread over large area of volume.

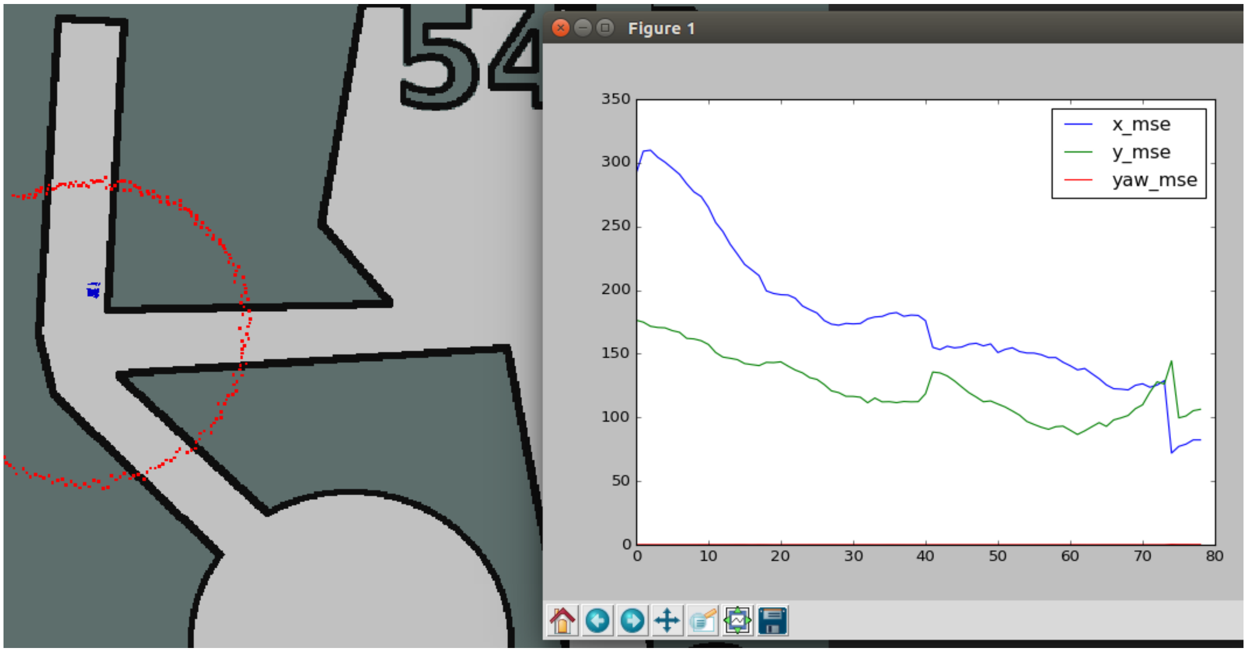

Fig. 4.3 Distribution of particles and error profiles for the case with 100 particles.

- When the NUM_PARTICLES=1 the algorithm will place the particle in a random position on the map. The filter will get overconfident with the solution and misrepresents the true probability distribution resulting in converging at the wrong pose.

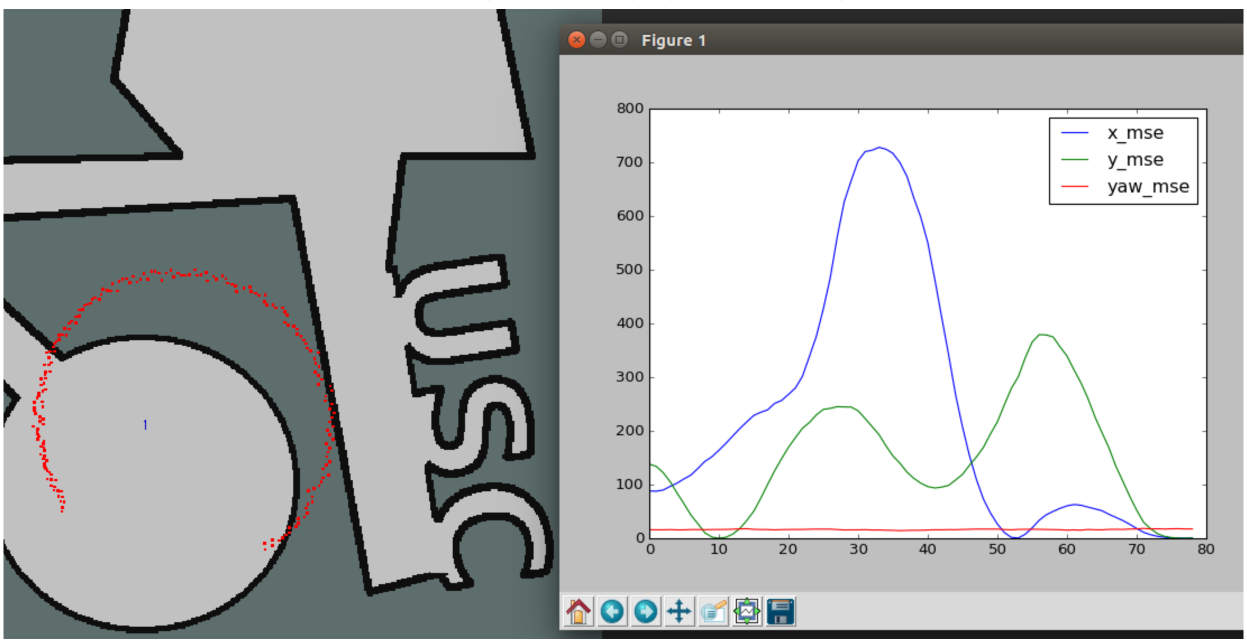

Fig. 4.4 Distribution of particles and error profiles for the case with one particle.

In Summary the number of particles is a parameter that enables the user to trade off the accuracy of the computation and the computational resources necessary. So the number of particles may vary based on application and computational efficiency. The optimal particle range can be determined by trial and error.

4.3 Can a particle filter with a single particle perform well? Why or why not? What if it starts in the correct position?

Below shows two trials using a single particle with random position. When there is only one single particle, the particle filter algorithm has difficulty with moving its position efficiently towards the correct position. Instead, the states of the particle (i.e., positions) will be updated randomly, as reflected by the error profile below.

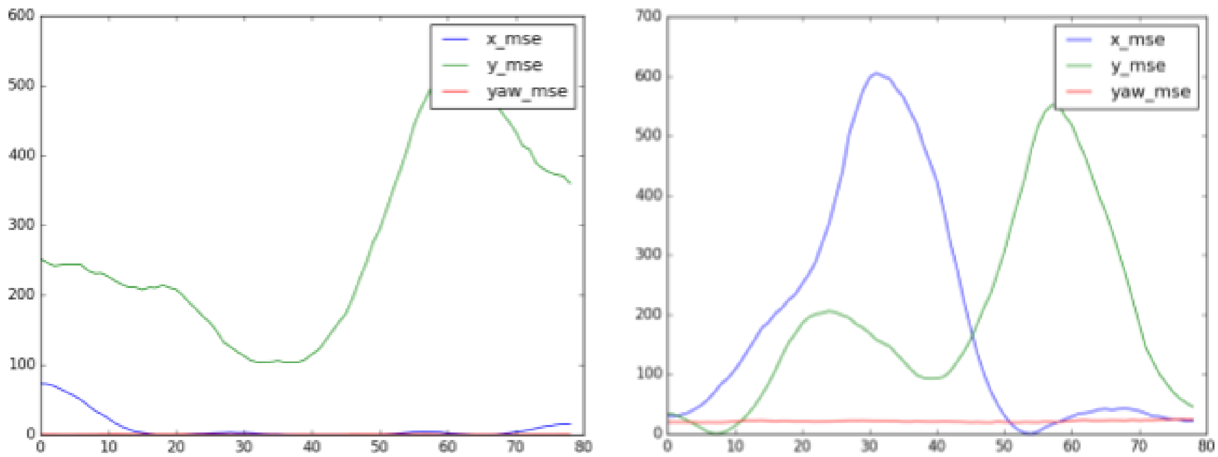

Fig. 5.1 The error profiles of positions and rotation of two runs of particle filter algorithms using a single particle with random starting positions.

If the particle starts with the correct position, the algorithm will still fail to correctly update the state of the particle. As reflected in the error profiles shown below, the position of the particle is not updated correctly despite the correct starting position. This is probably because, with a single particle, the algorithm will fail to resample conditioned on measurements and therefore is mistakenly sensitive to the measurements. On the contrary, the rotation is maintained within a lower range.

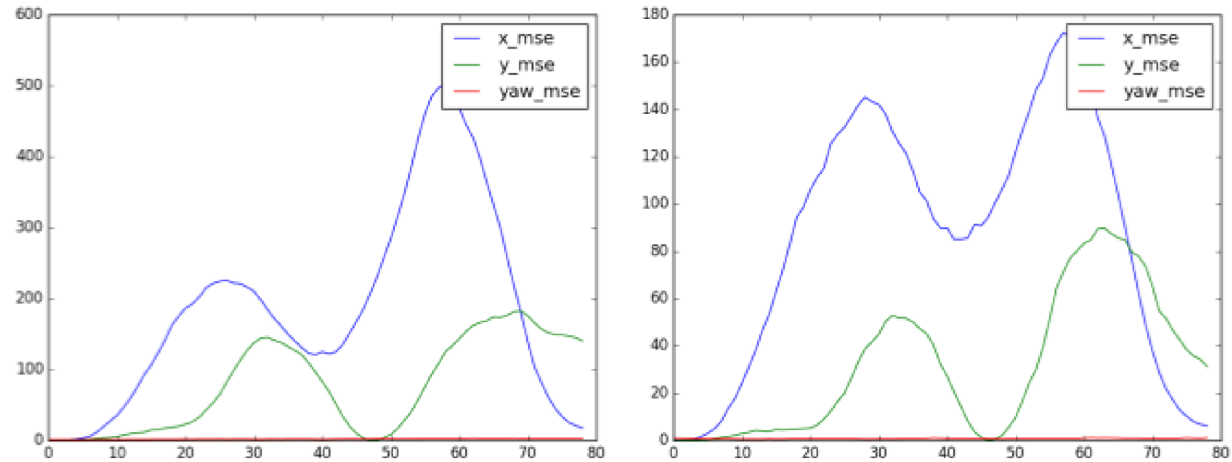

Fig. 5.2 The error profiles of positions and rotation of two runs of particle filter algorithms using a single particle with correct positions.

5. Extra credit

5.1 (Better noise model in sec 5.4) (Explain why simpler noise model has poor properties) (Implement)

5.1.1 Explain why simpler noise model has poor properties

The previous motion model is overly simplified in two senses: (1) the noise is directly added to the states, and (2) the motion is directly updated by taking into account the noisy action (translational and angular velocities). These two simplifications make the distribution of particles to behave always as a Gaussian distribution. In contrast, the more realistic motion model enables a non-Gaussian distribution of particles (See Fig. 6.1).

On the one hand, this motion model decomposes the motion into three components (i.e., rotation before translation, translation, and rotation after translation). This decomposition allows the motion model to integrate the uncertainties in a non-linear and coupled way. On the other hand, the variances of uncertainties are dependent on the two rotations and translations, rather than being fixed in the previous motion model. It seems that this dynamical setting of variances makes the particles capable of being adaptive to the environment. As reflected in Fig. 6.1, the distribution of particles is neither following the Gaussian distribution, nor fixed. In fact, the distribution shows adaptivity and can change via measurements. For example, as shown in the third time frame in Fig. 5.1.1, the front of the particle swamp becomes flat when it reaches the wall.

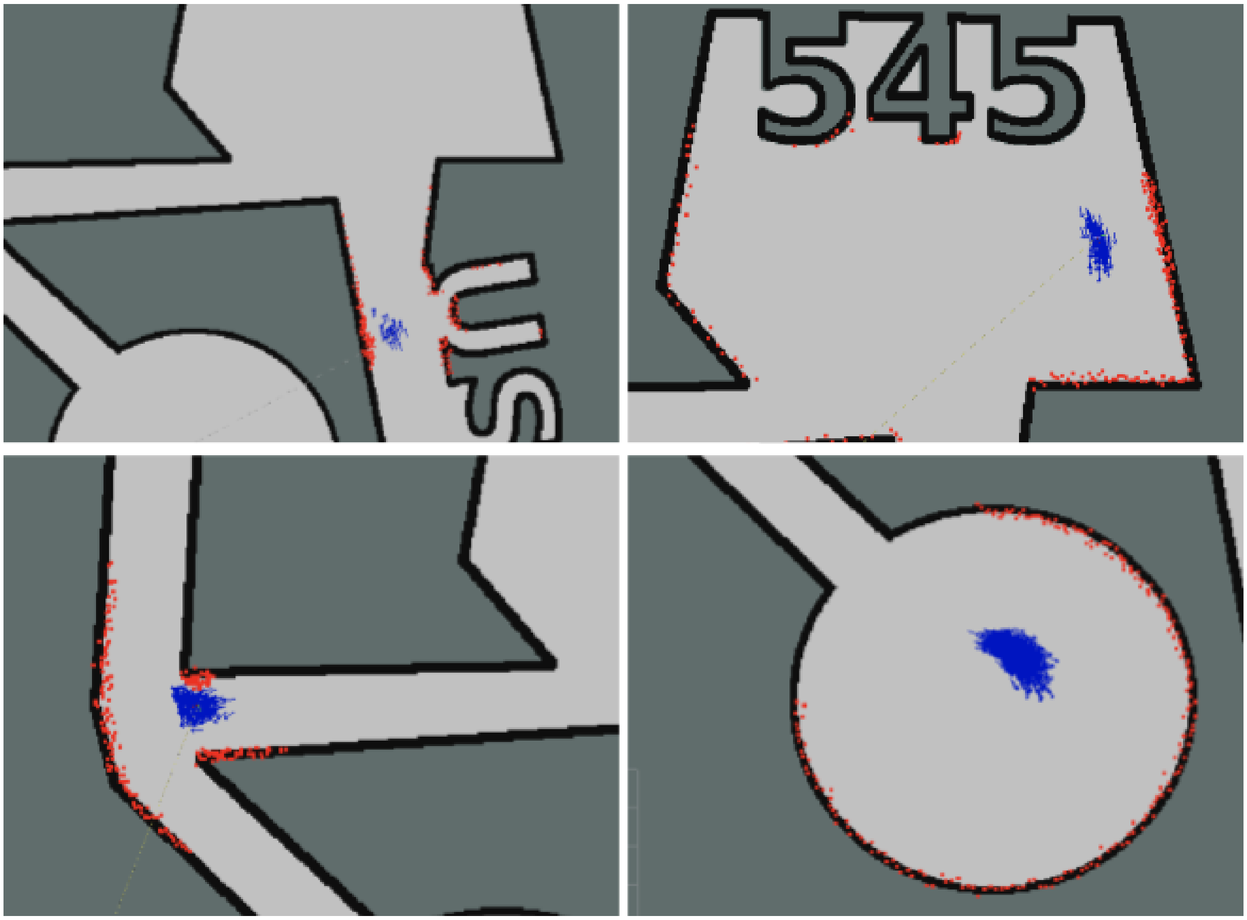

Fig. 6.1 Trajectories of particles over four different time steps for the more realistic motion model.

Another observation lies in the error profiles (Fig. 6.2), where there is always a spike in the rotation. This spike happens when the particles enter the T-shaped road as shown in the third frame in Fig. 6.1. The particles become divergent and less concentrated. This spike is probably caused by the way we get the pose which is averaged over all particles. This might cause significant error when the particles become divergent and spread over a large area on the map.

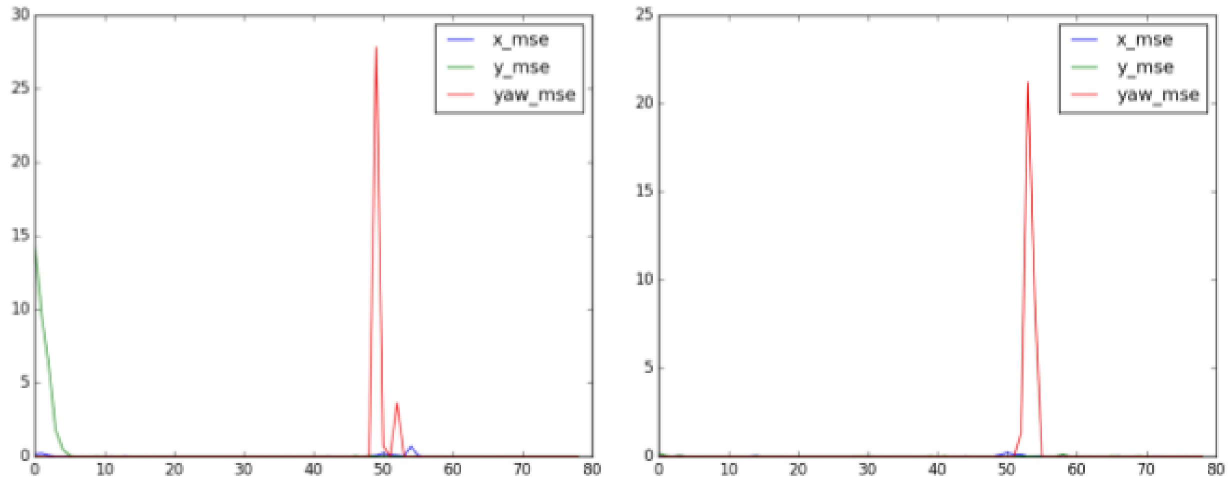

Fig. 6.2 The error profiles of positions and rotation of two runs using a more realistic motion model.

5.2 (Implement complete beam sensor model)

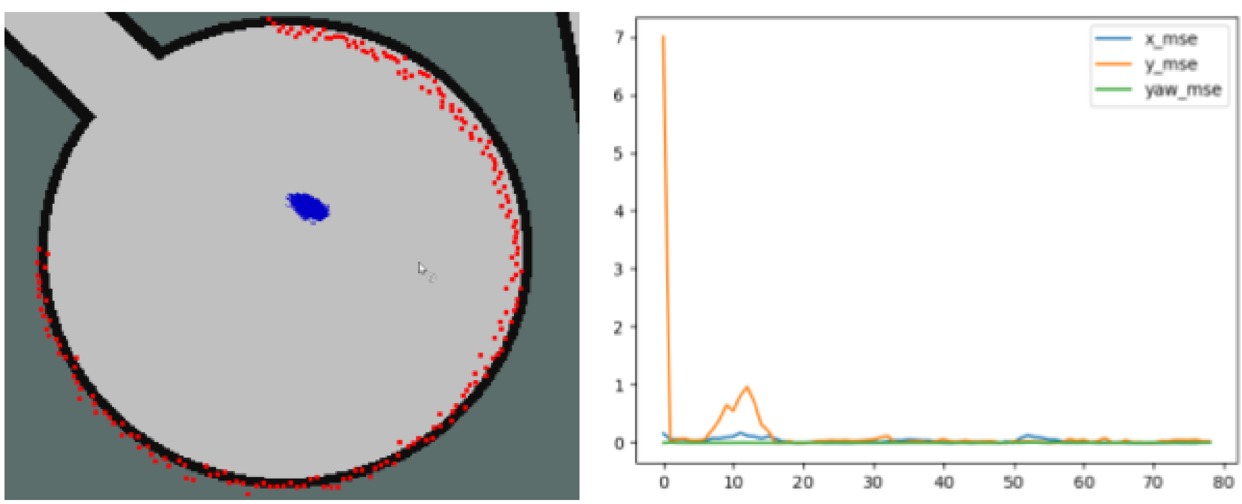

Fig. 7.1 Distribution of particles and error profiles for the case with a more realistic sensor model.

5.3 (Use a clustering algorithm to estimate the particles in the largest mode)

Without new motion model

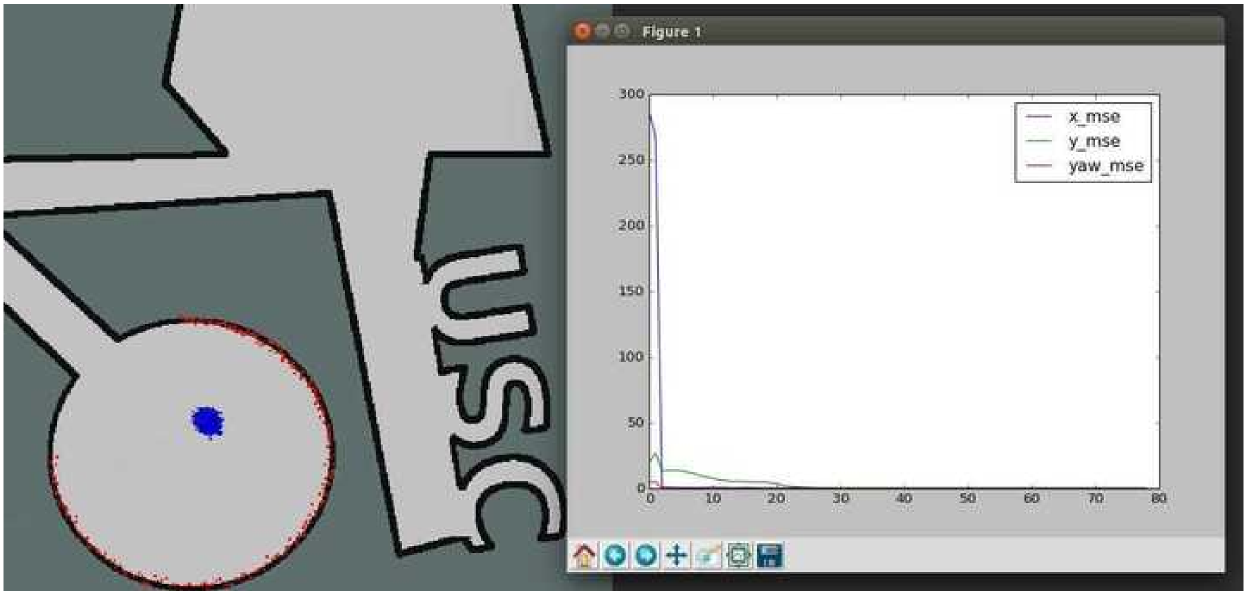

Fig. 8.1 Distribution of particles and error profiles for the case with a clustering algorithm and simplified motion model.

With new motion model:

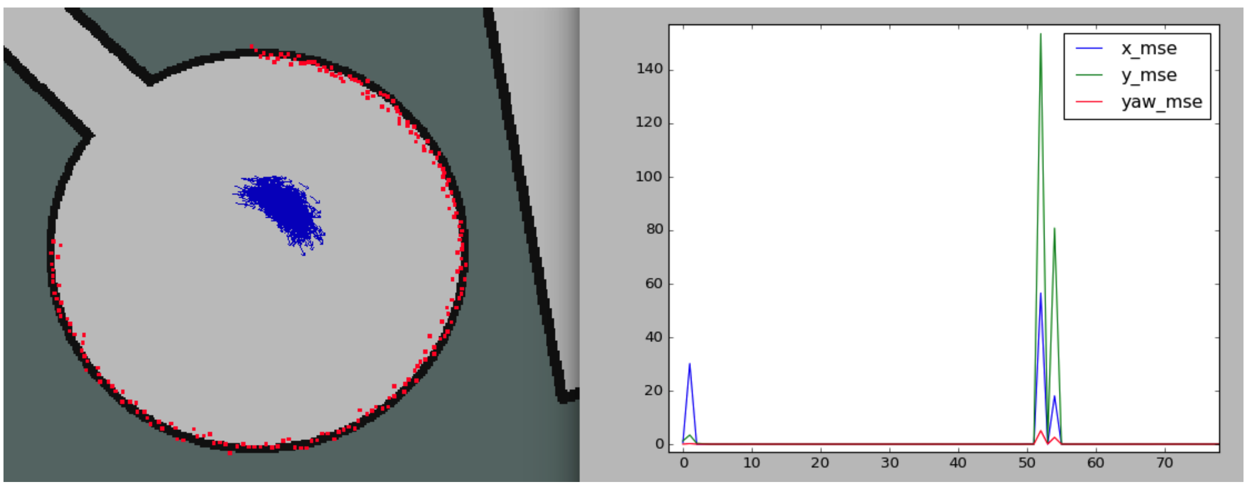

Fig. 8.2 Distribution of particles and error profiles for the case with a clustering algorithm and realistic motion model.