Particle Filter Programming Assignment

Assignment

For this assignment, the objective was to program a particle filter algorithm on a simulated two-wheeled robot. The robot was equipped with sensors to sense its range from each specific landmark.

To implement the particle filter, two functions needed to be completed. The first one implements the measurement update step of a particle filter, using a range-only sensor. It takes input landmark positions and landmark observations and returns a list of weights for the particle set. The second function takes a set of particles and their corresponding weights as input and returns a sampled set of particles.

Solution

For the first method, I had a loop within a loop. The outer loop went through all the particles, and the inner would go through all of the landmarks. For each particle, it will get the measurement of each landmark, compare the measurement with the known position of the landmark, and then calculate the weight and error to add on. From there, it will normalize the sum of the weights and save it to be returned.

For the second method, I used Monte Carlo resampling. For Monte Carlo resampling, first I make a cumulative distribution of the weights for the particles. Then, loop through all of the particles and if the cumulative distribution is greater than a random value picked uniformly between 0 and 1, then a new particle will be generated.

Findings

To test the solution, two different sampling methods were used with varied amounts of measurement noise. This would show the impact that measurement noise has on the system and also show which resampling method is more tolerant to measurement noise.

As expected, low measurement noise was very important regardless of the resampling method being used. For example, the convergence of localization error with Monte Carlo resampling grew by 660% when increasing the measurement noise by 600%. These results are shown in *Table 1.* The results were similar when using systematic resampling.

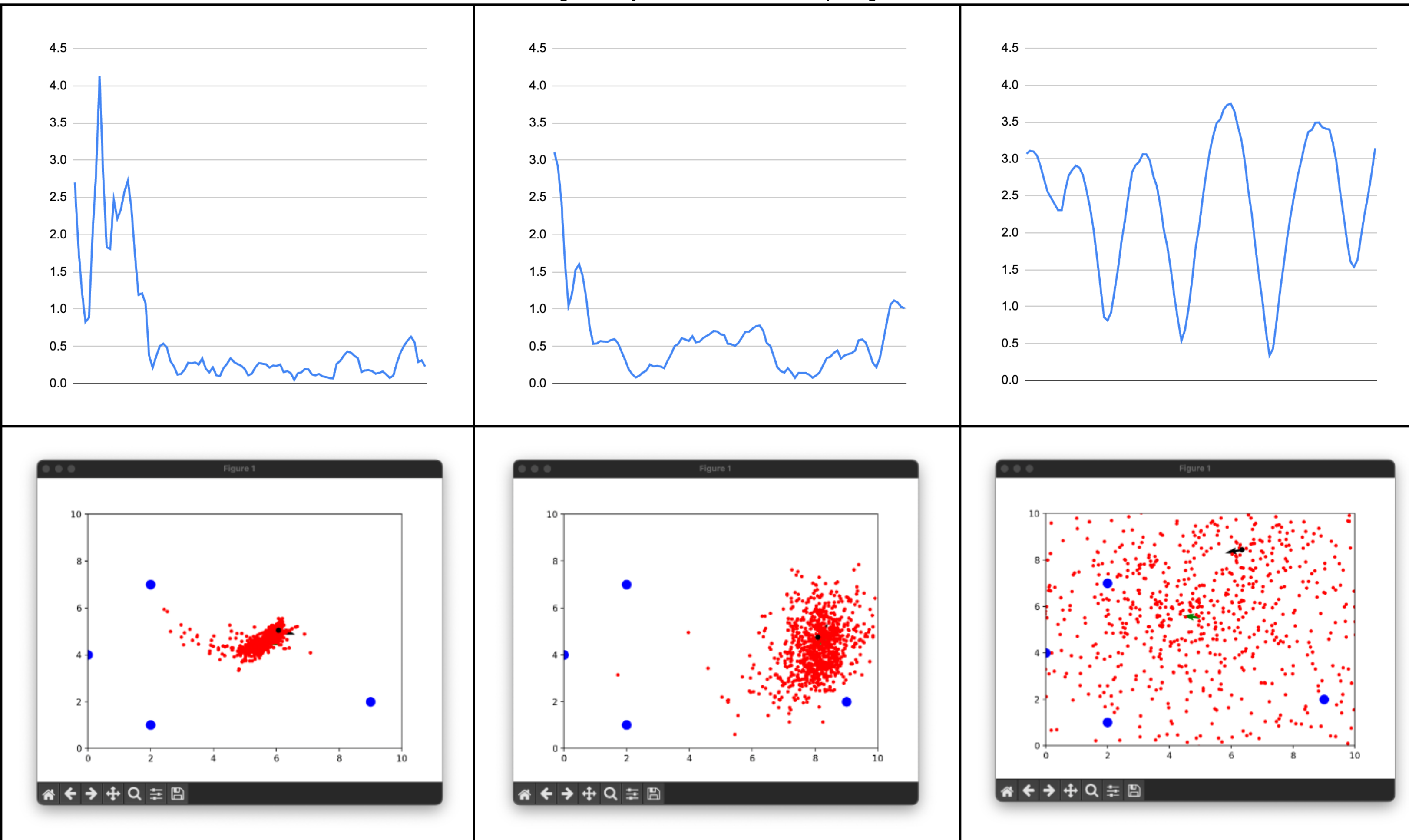
Table 1: Comparison of Monte Carlo and Systematic Resampling

sigma_r value (measurement noise)	Convergence of Localization Error with Monte Carlo Resampling	Convergence of Localization Error with Systematic Resampling
0.2	0.11	0.229
1.2	0.729	1.011
10.2	2.493	3.147

While the results show that both resampling methods struggle when presented with high measurement noise, it is clear from the same results that Monte Carlo resampling fares better than systematic resampling at both high and low measurement noise levels. This is shown in Table 1, where with a low sigma_r value, Monte Carlo resampling was 200% better.

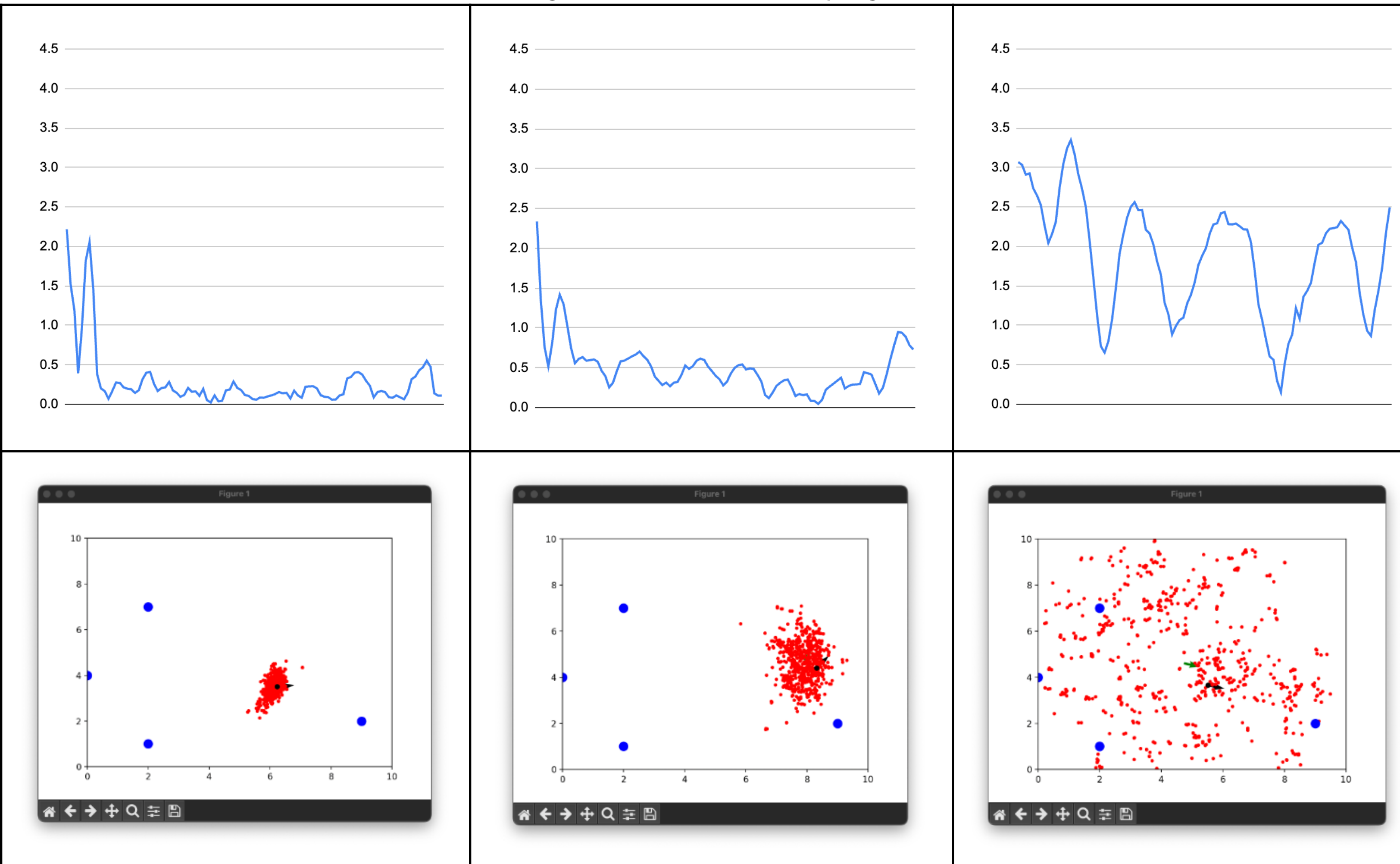
The plots for both resampling methods and varied measurement noise values have been included in *Fig. 1* and *Fig. 2*.

Fig. 1: Systematic Resampling



sigma_r = 0.2	sigma_r = 1.2	sigma_r = 10.2

Fig. 2: Monte Carlo Resampling



sigma_r = 0.2	sigma_r = 1.2	sigma_r = 10.2