

CS 3630 - Assignment 3

Alexander Huynh - James Liu

Create an Environment

We created our hallway-like environment out of various cardboard boxes. It contains three 90-degree turns. We left the surfaces brown because when testing the robot with the boxes covered in white paper, the IR sensor became too sensitive.



Figure 1. A picture of our physical environment. The robot's actual starting position is near the left entrance.



Figure 2. A top-down view of our hallway setup.

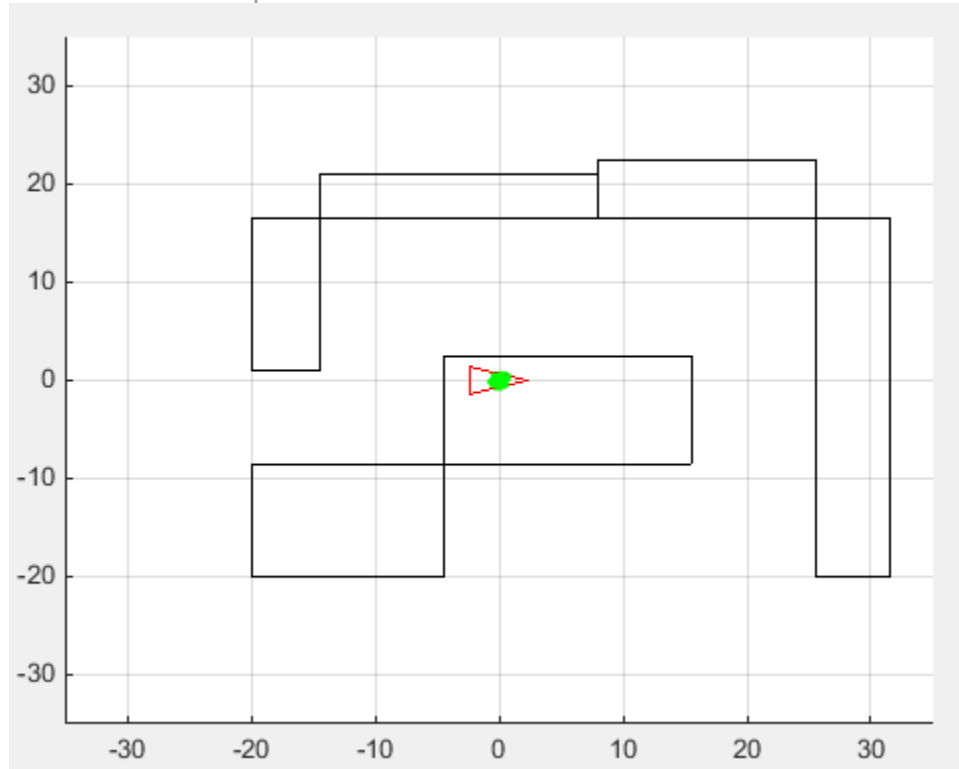
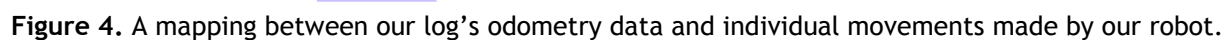


Figure 3. The physical setup imported as a map in MATLAB. Our units were in inches. The robot's actual starting pose was, $x = -20, y = -3, \theta = 0$.

Create a Motion Plan

We used dead-reckoning to move our robot through the world, starting at the left entrance and ending at the right exit. We added a brief pause between move commands to let the robot 'settle'. We gathered sensor data at a rate of 10Hz.

We verified our motion plan by comparing the output logs to our MatLab map, as seen in Figure 4 below. The output log contained reasonable values.



Run Particle Filter without Sensor Data

The end state of us simulating our robot with the motion model only filter is shown below in Figure 5.

Due to the lack of accuracy in the actual odometry data (modeled by the Gaussian noise added to the change in pose in the motion update step), the covariance of the distribution will naturally increase as more movements are made. We experimented with the values of Q and L . When the variance for the robot's speed was increased, the particle cloud grew along the axis the robot was moving. When the variance for the robot's rotational velocity was increased, the particle cloud became crescent-shaped with a fixed distance from the robot.

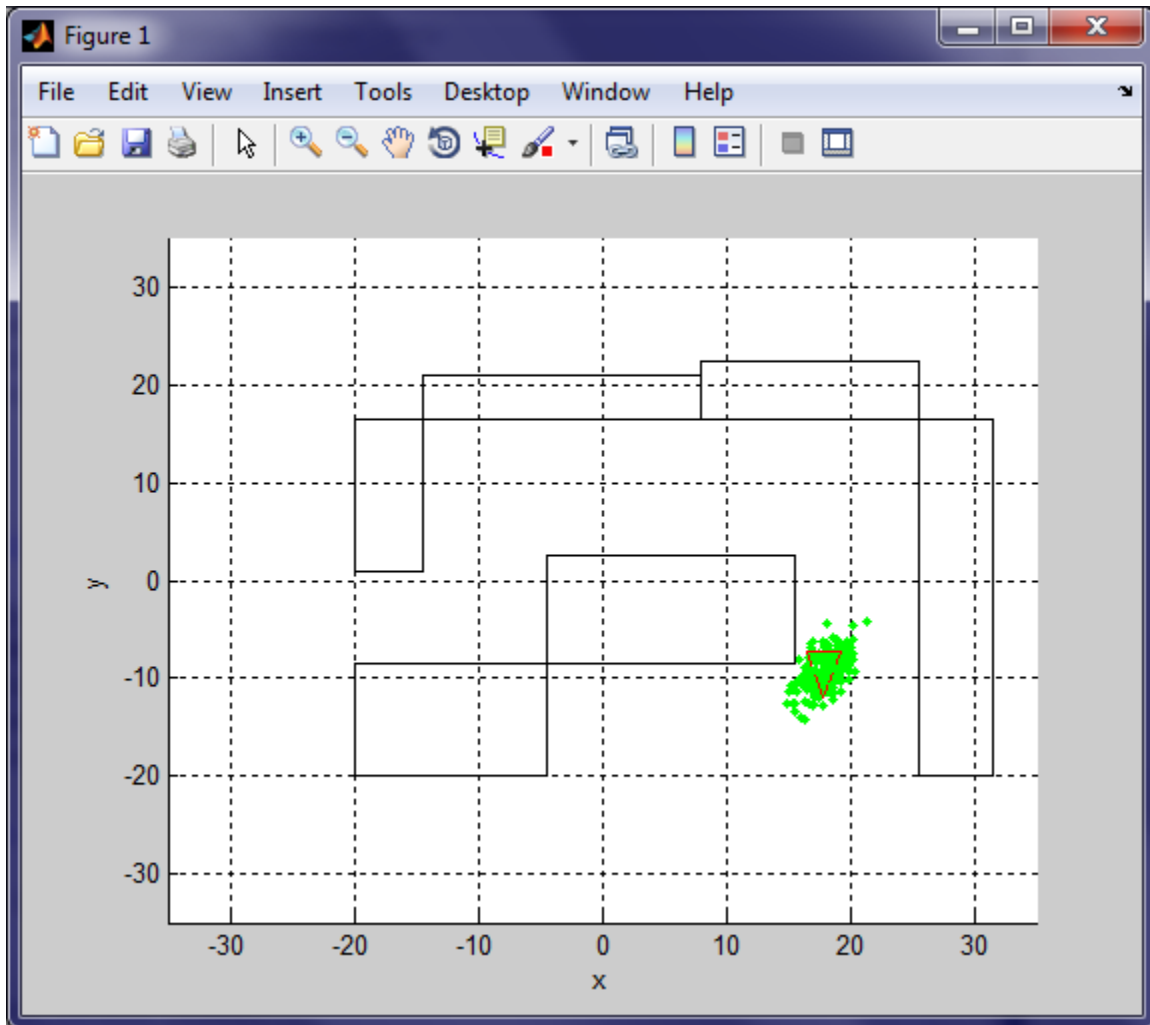


Figure 5. The final pose of our simulation using the motion model only filter.

Run Particle Filter with Sensor Data

The end state of us simulating our robot with the full particle filter is shown below in Figure 6.

Instead of one large particle cloud, the resultant set of particles seems to be focused on small, tightly-clustered groups. Particles that didn't 'match up' well with the sensor data were less likely to be chosen when resampling. Each small cluster's centroid is a candidate pose. Compared to when using only the motion model, these candidate poses are far more likely to be the robot's actual pose than any individual.

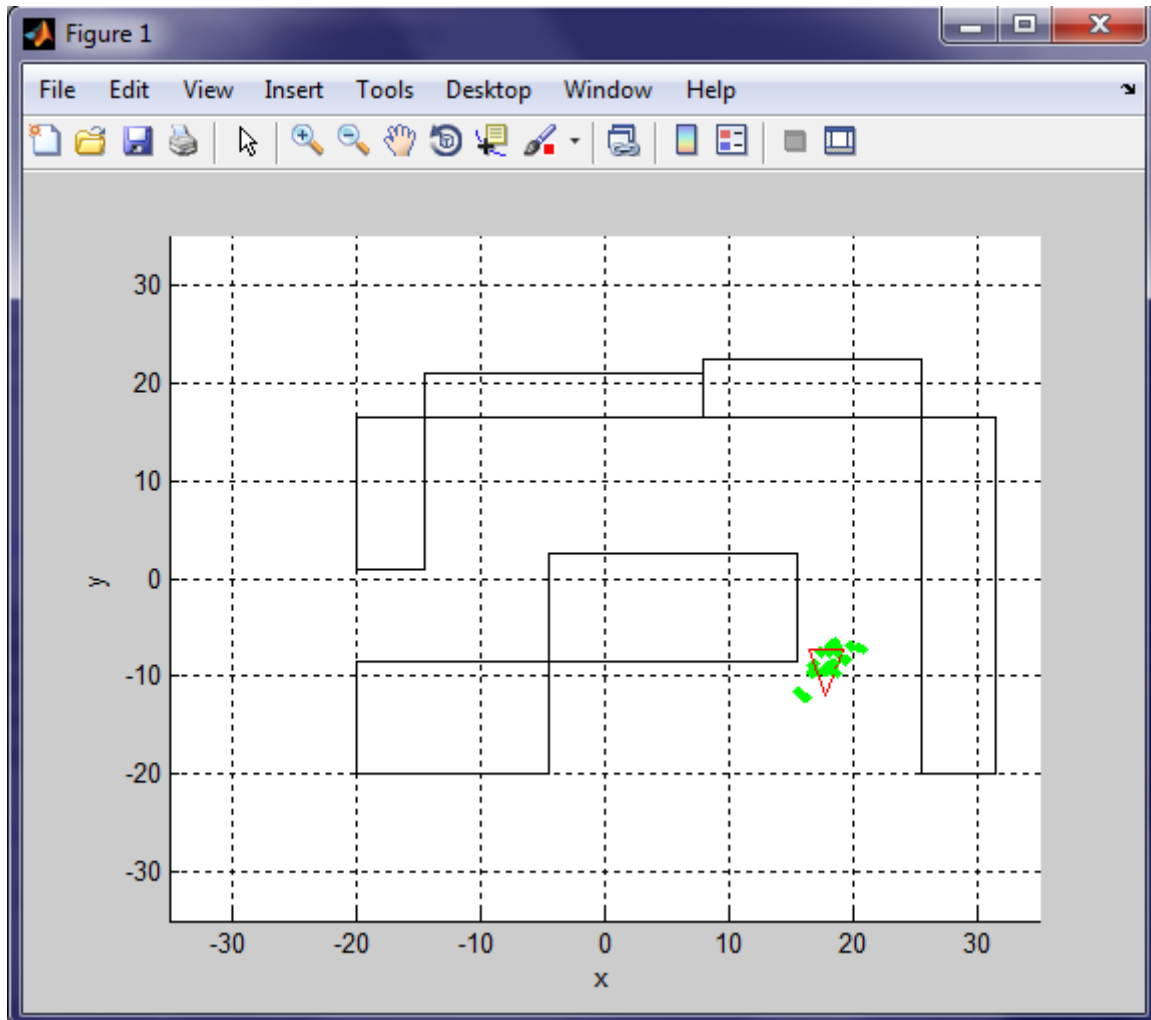


Figure 6. The final pose of our simulation using the full particle filter. Note how the size of the particle clouds are significantly smaller and more focused than when running only with the