MEAM 620 : Project 3

Localization and navigation of a mobile robot through a known environment using a noisy on-board laser sensor.

Submitted by: Subhrajit Bhattacharya

Aim

To navigate a mobile robot through a **known** environment with

i. An erroneous information about the initial position of the robot
ii. An on board noisy laser sensor.

No GPS, Odeometry or Ground Truth information were used for motion planning, localization or navigation! (Ground truth was used only to note the actual trajectory followed by the robot for comparison with the trajectory estimated by the particle filter.)

Motion Planning

The same motion planning scheme as in Projects 1 and 2 was used. It made use of *visibility graph* and the concept of *active obstacles* as mentioned in the report of project 2. However in project 3 we had a known environment without any unknown obstacle.

Localization

A particle filter was implemented for the purpose of localization. It has the following important features:

1. *Initialization and initial localization:*
   We started with 2000 particles, $\boldsymbol{p_i}^0 = [x_i, y_i, \theta_i]^0$, $i = 1$ to 2000. All the weights are initiated to a value of 1. If an estimate of the initial position is available the particles are generated according to it, otherwise they are scattered randomly all over the workspace.
   Consequently we made the robot turn at 3 different directions while staying at the same x and y coordinates, and hence update and re-sample the particles for 3 times (the procedures for that described later). This resulted in a pretty good localization of the initial position of the robot.

2. *Propagation / Prediction:*
   Given a particle at the k-th time step and the velocity commands (with added noise) issued between the k and k+1 -th time steps, a non-linear exact model of the non-holonomic robot was used to determine the position of the particle at the (k+1)-th timestep.

3. *Updating particle weights:*
   The challenging and most expensive part of the particle filter was to determine the weights corresponding to each particle, given a history laser scan data. Given a set laser scan points $L^k = [l_1, l_2, \ldots , l_n]^k$ in the local coordinate of the robot, we used the following steps to determine the weight associated with the i-th particle $\boldsymbol{p_i}^k = [x_i, y_i, \theta_i]^k$:
   i. Transform (rotate and translate) the laser scan points from the local coordinate frame of

the i-th particle to the global frame to obtain $\underline{L}_i^k = [\underline{l}_1, \underline{l}_2, \dots, \underline{l}_n]_i^k$

ii. Each point in $\underline{L}_i^k$ is checked for the smallest distance form the known world obstacles (i.e. smallest distance from the nearest obstacle). Let those distances be $\underline{D}_i^k = [\underline{d}_1, \underline{d}_2, \dots, \underline{d}_n]_i^k$

iii. The updated weight associated with the i-th particle is hence computed as

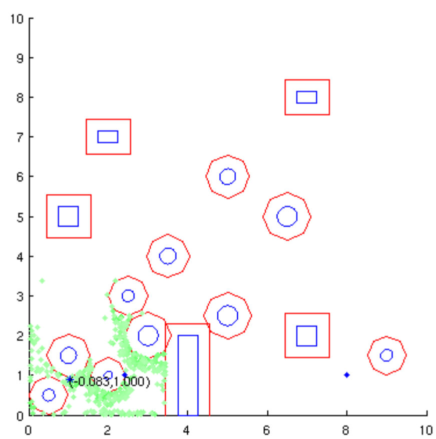
$$w_i^{k+1} = w_i^k \;/\; |\underline{D}_i^k| ,$$

where $|.|$ is the 2-norm of a vector.

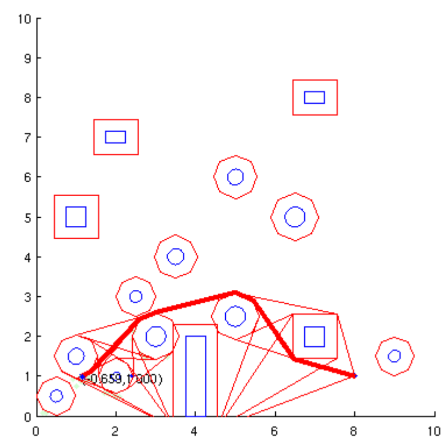
iv. The weights are then rescaled by dividing by the maximum value of weight.

4. *Resampling:*
After the initialization step we down-sample the particles and selected 100 particles for tracking and estimating position of the robot. Thereafter we perform the re-sampling at every time step by selecting just 10 best particles and replicating them to obtain 100 particles.

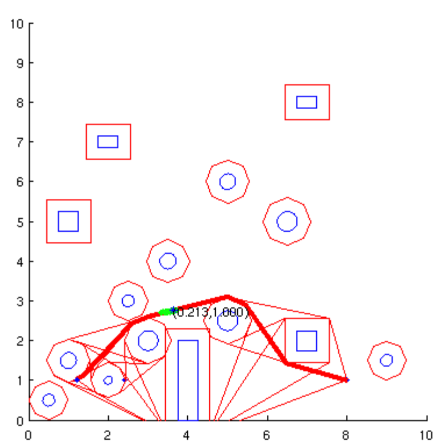
Figures and results:



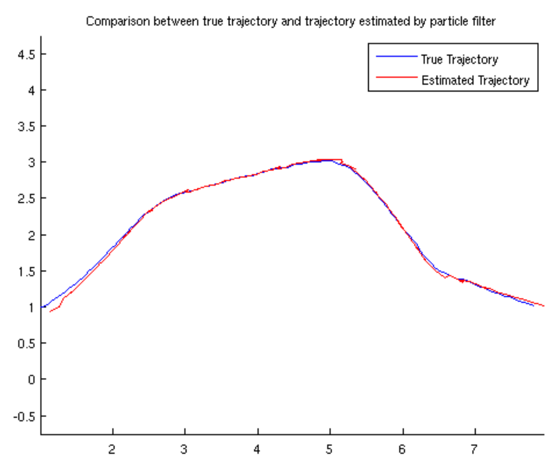
Particles (in light green dots) after the first pass of initial localization (blue dot – erroneous initial knowledge about start position, blue star – localized start position)



Path planned from estimated initial point to goal



Particles (100 of them) during the run



Comparison between the actual trajectory followed by the robot and the trajectory as estimated by the particle filter

Discussions and Conclusions:

1. The initial localization of the robot was most challenging. That's why we had to start off with more particles at the beginning. Once the robot has been localized initially, it's position was tracked quite reliably with just 100 particles. The number 100 was chosen based on experiments. Having too less particles would result in inaccurate tracking, while having too many particles would be computationally expensive and may actually increase noise in the estimation!
2. The most computationally expensive part at each time step was matching the global coordinates of the laser scan points corresponding to each particle with the known obstacle boundaries. Presently a very crude way of computing the distances of the scan points from the obstacle boundaries is being used. More efficient methods that make use of the matchings made at previous time-steps would make the code run more efficiently.
3. Some very conservative value for the speeds were used that made the robot move very slowly. There are lots of scope of making improvement on this by adaptively changing the speeds depending on the confidence level from the particle filter.