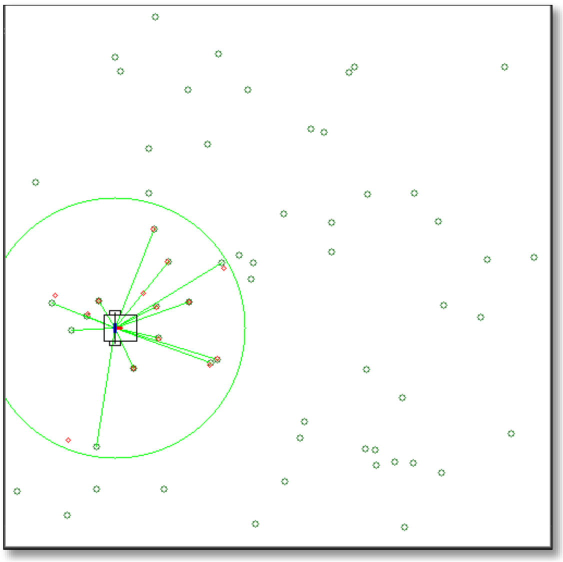
CS562000 Robotic Navigation and Exploration

Lab3 Particle Filter - Fast Slam

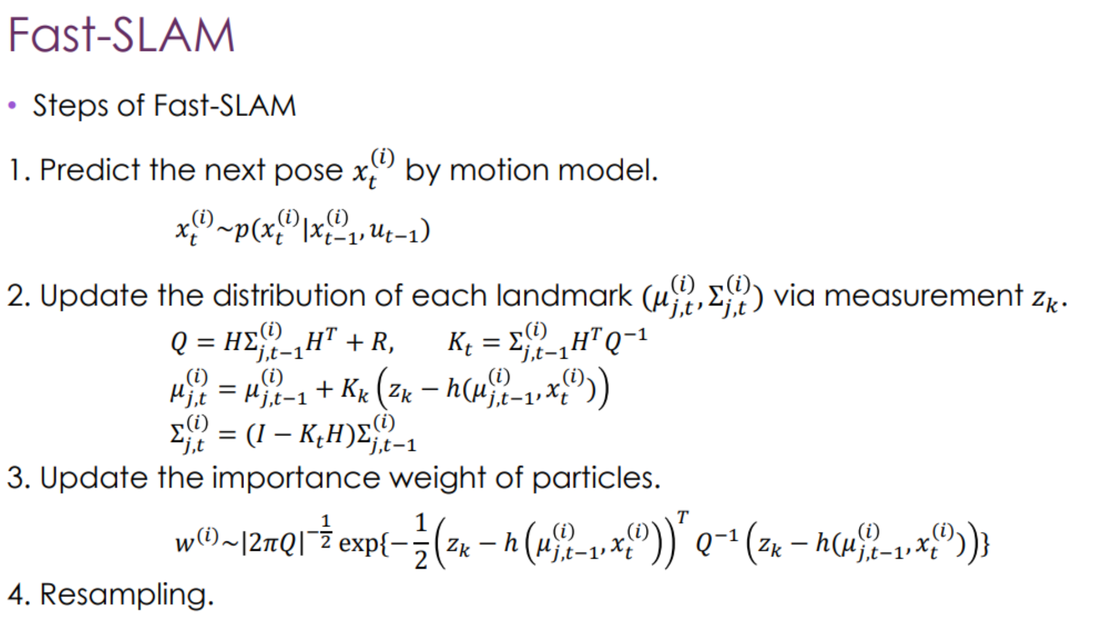


Introduction:

In class, we had taught some SLAM algorithms and for this lab we are going to practice particle filter.

Here is the algorithm.

==



Fast-SLAM

- Steps of Fast-SLAM

1. Predict the next pose $x_t^{(i)}$ by motion model.

$$x_t^{(i)} \sim p(x_t^{(i)} | x_{t-1}^{(i)}, u_{t-1})$$

2. Update the distribution of each landmark $(\mu_{j,t}^{(i)}, \Sigma_{j,t}^{(i)})$ via measurement z_k.

$$Q = H\Sigma_{j,t-1}^{(i)} H^T + R, \qquad K_t = \Sigma_{j,t-1}^{(i)} H^T Q^{-1}$$

$$\mu_{j,t}^{(i)} = \mu_{j,t-1}^{(i)} + K_k \left(z_k - h(\mu_{j,t-1}^{(i)}, x_t^{(i)}) \right)$$

$$\Sigma_{j,t}^{(i)} = (I - K_t H)\Sigma_{j,t-1}^{(i)}$$

3. Update the importance weight of particles.

$$w^{(i)} \sim |2\pi Q|^{-\frac{1}{2}} \exp\{-\frac{1}{2}\left(z_k - h\left(\mu_{j,t-1}^{(i)}, x_t^{(i)}\right)\right)^T Q^{-1}\left(z_k - h(\mu_{j,t-1}^{(i)}, x_t^{(i)})\right)\}$$

4. Resampling.

==

TODO:

Please finish the following functions in fast_slam.py:

Class Particle:

- **def update_landmark**

- **def update_obs**

Class ParticleFilter:

- **def resample**

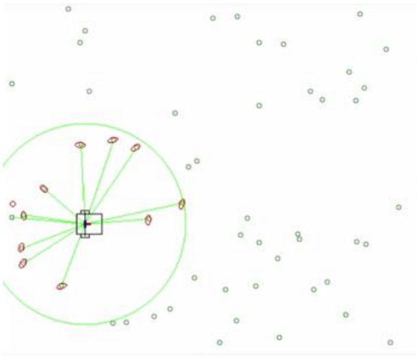
==

EXAMPLE:

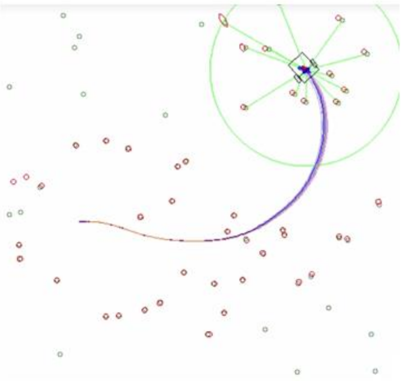
The red small circles are the landmarks you supposed, and the green small circles are ground truth.

If your code is right, you will find, as time passed by, the red circles will be close to green circles gradually.

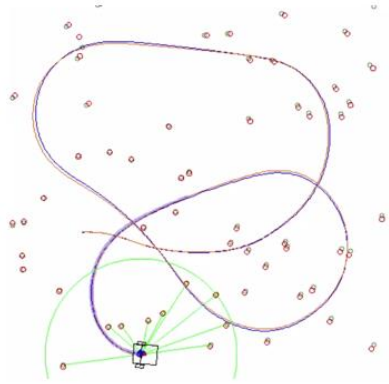
1.Start



2. As time passed by



3.



4.