Particle Filter localization

David FILLIAT - Goran FREHSE
ENSTA Paris

January 7, 2021

1 Introduction

In this practical work, we will work on robot localization using a Particle Filter. For this, we will use the python code available on the course Moodle. The provided code makes it possible to simulate a robot moving on a given trajectory in an environment made up of point landmarks. It implements a simple particle filtering method using the perception of the direction and distance of these point landmarks. The provided code requires the installation of the `numpy`[1] and `matplotlib`[2] python packages.

Upload your report as a pdf file that includes your answers to the questions and the code you wrote on the Moodle.

2 Models

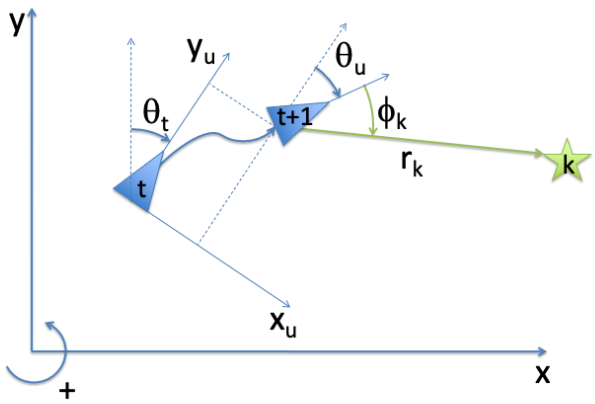

Figure 1: Notations for the motion and observation models used in the code.

The modeled robot (Fig. 1) moves on a plane and perceives the direction and the distance of point landmarks located on this same plane. Its state is represented by a column vector containing its position and its orientation in a global coordinate system:

$$X_t = [x_t, y_t, \theta_t]^T$$

The motion of the robot between t et $t+1$ is measured through odometry and given by its position at time $t+1$ in the robot frame at time t (Fig. 1):

$$U_t = [x_u, y_u, \theta_u]^T$$

With a gaussian noise given by the covariance matrix Q, the evolution model is then :

$$X_{t+1} = f(X_t, U_t) = \begin{bmatrix} x_t + x_u cos(\theta_t) - y_u sin(\theta_t) \\ y_t + x_u sin(\theta_t) + y_u cos(\theta_t) \\ \theta_t + \theta_u \end{bmatrix}$$

The perceptions are the distance and direction of a landmark k supposed to be perfectly identifiable $Y_t = \left[r_t^k, \phi_t^k\right]^T$ (Fig. 1). The observation model is therefore:

$$Y_t^* = h^k(X_t^*) = \begin{bmatrix} \sqrt{(x_k - x_t)^2 + (y_k - y_t)^2} \\ atan2(\frac{y_k - y_t}{x_k - x_t}) - \theta_t \end{bmatrix}$$

where x_k et y_k are the known coordinates of the landmark in the global coordinate system. This model is corrupted by a Gaussian noise of covariance matrix P_Y.

[1] https://numpy.org/
[2] https://matplotlib.org/

3 Questions

Question 1 : Parameters Change the different parameters of the particle filter to evaluate their influence on the performance of the filter (speed, precision, stability, performance of recovery after sensor failure at time 400 ....). In particular, study the effect of:

- The number of particles (`nParticles`)
- The initial variance around the true position for particle creation (`np.diag([1,1,0.1])` around line 160)
- The noise of the perception and motion models used for the particle filter (`UEst` and `REst`) with respect to the simulation noise (smaller, equal or bigger than `UTrue` and `RTrue`).

Question 2 : Global Localization Use the particles initialization without any knowledge of the initial position (uncomment `xParticles = 120 * np.random.rand(3, nParticles)-60` around line 163). Choose the filter parameters to reach a correct performance. How do you compare the performances with the situation in the previous question ?

Question 3 : Sensor Resetting Localization Modify the resampling function to use the "Sensor Resetting Localization" method [1] (see the article). You must replace a certain number of particles with particles drawn according to the likelihood of the observations.

The observation model of the direction and the distance of a point landmark allows to generate new particles on a circle around the perceived landmark (Fig. 2). Note that this is not necessarily possible or practical with all perception models:

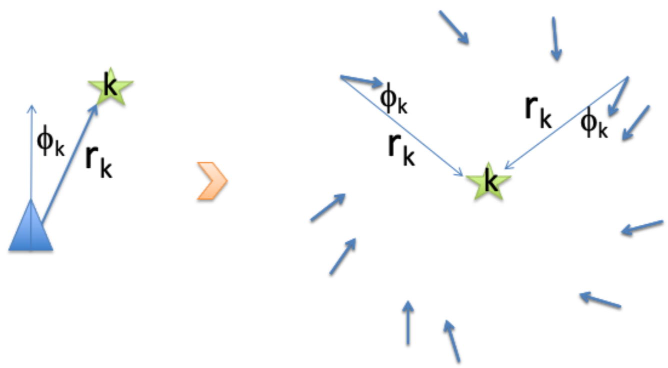

Figure 2: For a given perception (on the left), it is possible to generate a set of positions around the landmark that can lead to this perception (on the right).

These new particles should only be generated in the map area ($-60 < x < 60, -60 < y < 60$) and only when the localization is poorly estimated, ie when the sum of the likelihoods of the particles is below a threshold.

Test this method on the global localization problem and show how it can improve the performances.

References

[1] Scott Lenser and Manuela Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proceedings of ICRA-2000, the International Conference on Robotics and Automation*, April 2000.