Autonomous Mobile Robots
Homework 4

Ahmed Mohsen ALi

November 2022

Task Description

The robot needs to navigate from point A (0,0) to point B (D,S) using **particle filter with extended kalman ekf** as state estimation. The control strategy and motion model have to be determined.

Problem solution

Motion Model

The model used in this solution is **bicycle model** on a circular path. The motion model is defined as follows:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + \frac{v}{\omega}\sin(\theta_t + \omega\delta t) \\ y_t - \frac{v}{\omega}\cos(\theta_t + \omega\delta t) \\ \theta_t + \omega\delta t \end{bmatrix} \tag{1}$$

The right hand side of the above equation is used as $f(x,t)$ in the code implementation.

Measurement Model

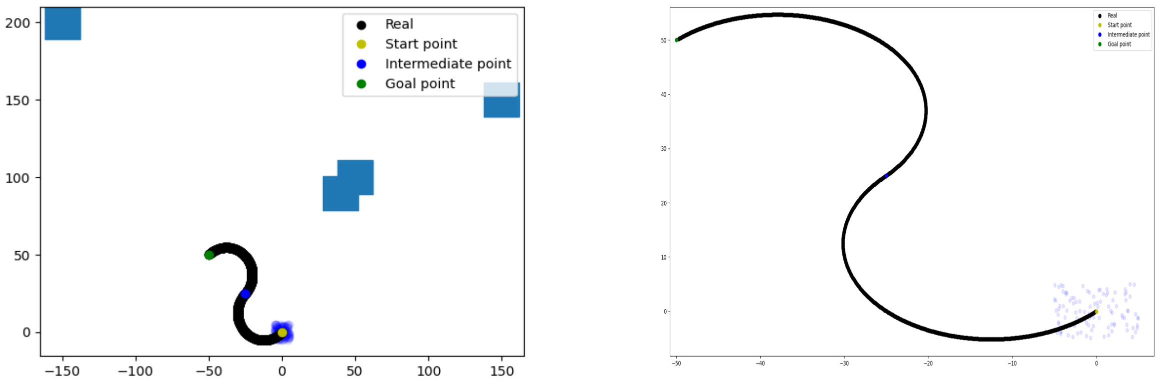
There are three landmarks in the environment: L1 (5,30), L2 (5,-30), L3 (-5,0). The sensor readings are defined as follows:

$$\begin{bmatrix} x_t^i \\ \theta_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{i,x} - x)^2 + (m_{i,y} - y)^2} \\ \text{atan2}(m_{i,y} - y, m_{i,x} - x) - \theta \end{bmatrix} + \mathcal{N}(0, R) \tag{2}$$

The right hand side of the above equation is used as $h(x,t)$ in the code implementation.

Control Strategy

The point A is at (D,S) which are assumed to be at -50 and 50 respectively. The planned path is computed analytically to be two half circles as shown in the figure below:



(a) Desired path (zoomed out)

(b) Desired path (zoomed in)

Figure 1: Desired Trajectory

Each circle has a radius r which can be calculated as $\sqrt{(D - x_{initial})^2 + (S - y_{initial})^2}/4$. By doing so, the intermediate point is located at exactly at the midpoint between initial and final poses. Thus, the control strategy will be:

Before intermediate point:

$$v = 1$$
$$w = -1/r$$

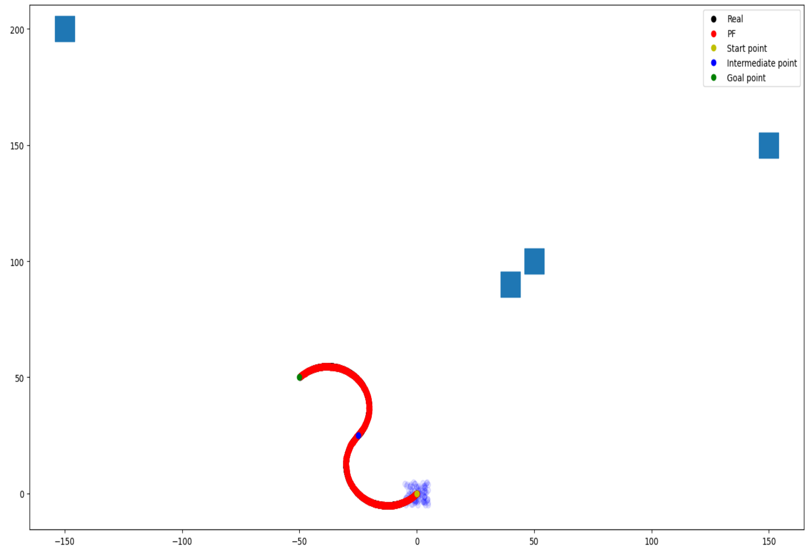
After intermediate point:

$$v = 1$$
$$w = 1/r$$

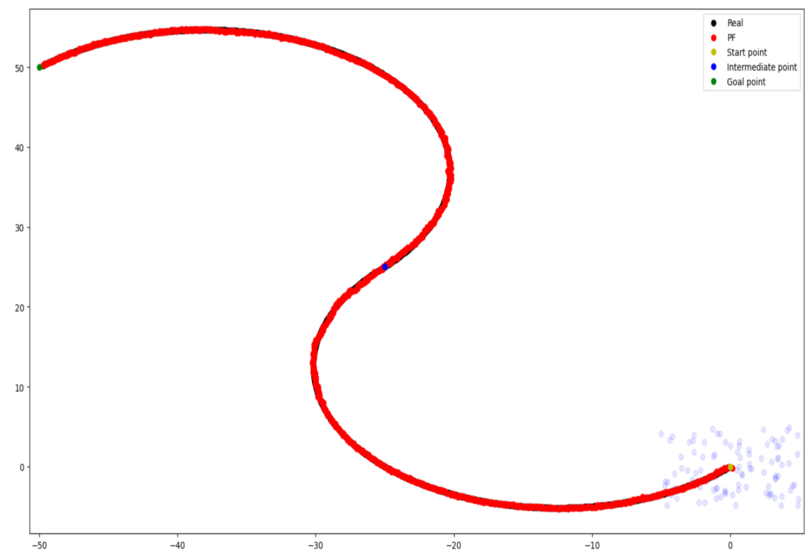
Also, there is a tolerance `self.dmin` to determine whether the robot reaches either the intermediate point or the final goal.

Results and Discussion

The following graph shows the robot trajectory using particle filter with ekf localization. The Desired data are obtained analytically while Estimated data are generated from the filter:



(a) Particle filter with ekf (zoomed out)



(b) Particle filter with ekf (zoomed in)

Figure 2: Particle filter with ekf

It is clear that both desired and estimated trajectories are identical and the estimated trajectory is more accurate than the one using particle filter only. However, this high precision comes at cost. The code for ekf particle filter took around 15 minutes while using particle filter alone took less than 30 seconds.