Zeinab Ebrahimi
ID: 501074656
zeinab.ebrahimi@ryerson.ca

Assignment 1
ME 8135: Robot Estimation

1 Particle Filter

Let the classic motion model and range observations made from a landmark located at L=[10,10] as follows:

$$\dot{x} = \frac{r}{2}(u_r + u_l) + w_x \tag{1}$$
$$\dot{y} = \frac{r}{2}(u_r + u_l) + w_y \tag{2}$$

where the radius of the wheel is $r = 0.1m$, u_r and u_l are the control signal employed to the right and left wheels, respectively. $w_x = N(0, 0.1)$ and $w_y = N(0, 0.15)$. The speed of each wheel is assumed as $0.1m/s$. In order to turn, angle is incorporated into the linear speed equation as:

$$\dot{\theta} = \frac{r}{L}(u_r + u_l) + w_\psi \tag{3}$$

where $w_\psi = N(0, 0.01)$. It is assumed that the motion model is computed 8 times a second, and every second a measurement is provided as:

$$z = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} r_x & 0 \\ 0 & r_y \end{bmatrix} \tag{4}$$

Here $r_x = N(0, 0.05)$ and $r_y = N(0, 0.75)$. The system need to discretize as:$\dot{x} = \frac{x_k - x_{k-1}}{T}$, where T is 1/8 sec. According to equation 1, the process noise covariance is provided by:

$$Q = \begin{bmatrix} w_x & 0 \\ 0 & w_y \end{bmatrix} \tag{5}$$

and the measurement noise covariance is given by:

$$R = \begin{bmatrix} r_x & 0 \\ 0 & r_y \end{bmatrix} \tag{6}$$

The objective of a particle filter is to estimate the posterior density of the state variables given the observation variables. Particle filters have the main feature of approximating a target posterior probability distribution function by a set of weighted 'particles'. Hence, under the particle filtering methodology, the target posterior PDF of the state x_t is approximated by samples (particles) that are recursively generated from the prediction and filtering distributions as new information becomes available. Once the filtering PDF is approximated, any characteristic of the state can be estimated from the set of weighted particles. The particle filter algorithm is defined as follows [1]:

1. Draw M samples from the joint density comprising the prior and the motion noise:

$$\begin{bmatrix} \hat{x}_{k-1}, m \\ w_{k,m} \end{bmatrix} \leftarrow p(x_{k-1} | \acute{x}_0, v_{1:k-1}, y_{1:k-1}) p(w_k) \tag{7}$$

where m is the unique particle index.

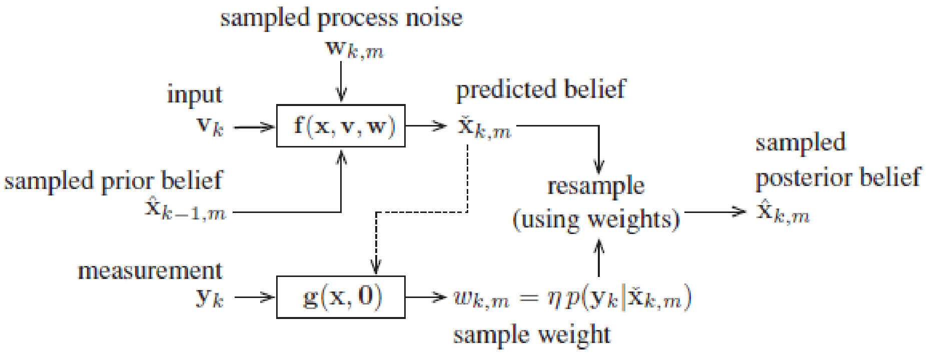

Figure 1: Particle filter algorithm [1]

2. Produce a prediction of the posterior PDF by using v_k. This is by passing each prior particle/noise sample through the nonlinear motion model:

$$\acute{x}_{k,m} = f(\hat{x}_{k-1,m}, v_k, w_{k,m}) \tag{8}$$

3. Correct the posterior PDF by incorporating y_k.First, assign a scalar weight to each predicted particle based on the divergence between the desired posterior and the predicted posterior for each particle:

$$w_{k,m} = \frac{p(\acute{x}_{k,m}|\acute{x}_0, v_{1:k}, y_{1:k})}{p(\acute{x}_{k,m}|\acute{x}_0, v_{1:k}, y_{1:k})} = \eta p(y_k|\acute{x}_{k,m}) \tag{9}$$

$$y_{k,m}^{'} = g(\acute{x}_{k,m}, 0) \tag{10}$$

4. Then, resample the posterior based on the weight assigned to each predicted posterior particle:

$$\hat{x}_{k,m} \xleftarrow{resample} \acute{x}_{k,m}, w_{k,m} \tag{11}$$

Madow provides a simple systematic technique to do resampling. M samples are assumed and each of them is assigned an unnormalized weight. So, from the weights, the bins with boundaries are created as:

$$\beta_m = \frac{\sum_{n=1}^{m} w_n}{\sum_{l=1}^{M} w_l} \tag{12}$$

The β_m define the boundaries of M bins on the interval [0, 1]. A random number ρ is selected, sampled from a uniform density on [0, 1). For M iterations we add to the new list of samples, the sample whose bin contains . At each iteration we step forward by 1/M. The algorithm guarantees that all bins whose size is greater than 1/M will have a sample in the new list. The structure of particle filter approach is illustrated in Figure 1 [1].

In this example, the initial values of state are set to (8,10). 100 particle is selected with standard deviation of 2. In addition, the initial angle is set to $\pi/2$. Hence, it is circled around the landmark with a 2 meter distance. The outcome of particle filter algorithm is shown in Figures 2 and 3. The landmark and the robot are defined as white and red square, respectively. As it is demonstrated, the particles converge to the movement of robot eventually as updating the model.

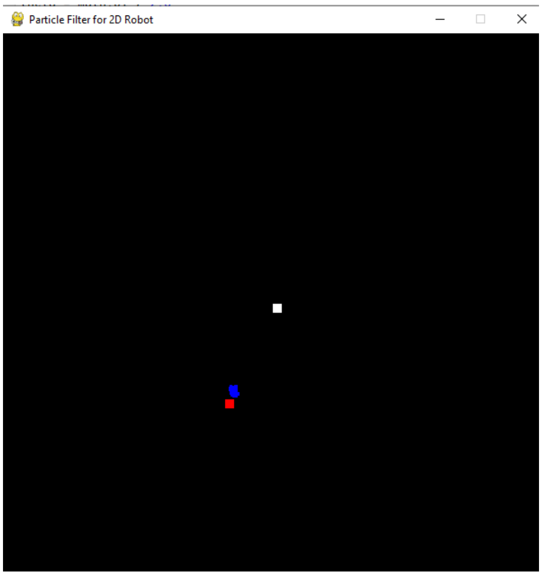

Figure 2: The outcome of Particle Filter algorithm

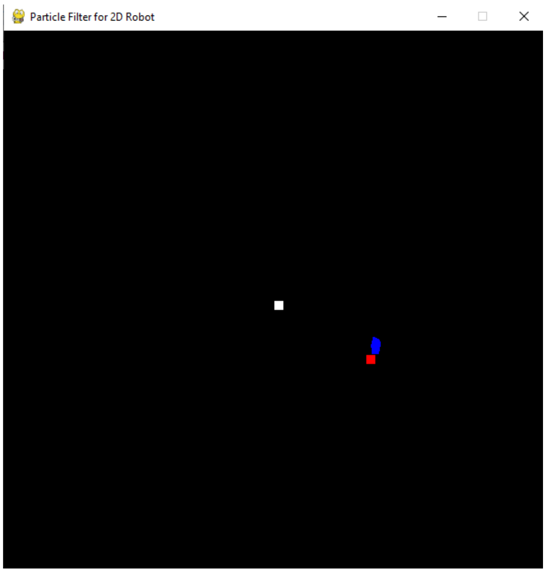

Figure 3: The outcome of Particle Filter algorithm

References

[1] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.