EECE 5550 Mobile Robotics - Section2 -Lab #2

Zhexin Xu , xu.zhex@northeastern.edu

October 27, 2023

Abstract

This is the solution of HW2. The code is implemented using c++ and mainly based on Eigen , Sophus and PCL libraries. Matplotlib are also used for visualization. All the code can be found in:https://github.com/zhexin1904/EECE5550/tree/mian/HW2

1 Question1: Extended Kalman Filter

Question1 is a simplified case of EKF, where linear velocity is constant, angular velocity is zero, and measurement model only consists of the Euclidean distance. All the noise incorporated above is addictive mean-zero Gaussian noise. We can formulated the EKF as following :

(a) Motion model

$$x_{t+1} = g_t\left(x_t, u_t\right) + \epsilon_t, \quad \epsilon_t \sim N\left(0, R_t\right) \tag{1}$$

where

$$\begin{aligned} x_t &= \left[p_x, v_x, p_y, v_y\right]^{\intercal} \in R^{4\times 1} \\ u_t &= v = \left[v_x, v_y\right]^{\intercal} \in R^{2\times 1} \\ \epsilon_t &= \left[\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4\right]^{\intercal} \in R^{4\times 1} \end{aligned}$$

discrete time formulation:

$$p_x\left(t+1\right) = p_x\left(t\right) + v_x\left(t\right)\Delta t \ + \sigma_1 \tag{2}$$

$$v_x\left(t+1\right) = v_x\left(t\right) + \sigma_2 \tag{3}$$

$$p_y\left(t+1\right) = p_y\left(t\right) + v_y\left(t\right)\Delta t \ + \sigma_3 \tag{4}$$

$$v_x\left(t+1\right) = v_y\left(t\right) + \sigma_4 \tag{5}$$

Therefore, we can derive the Jacobian of motion model with respect to the state variables:

$$
\boldsymbol{G}_t = \frac{\partial g_t\left(x_t, u_t\right)}{\partial x} = \begin{pmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{pmatrix} \in R^{4\times 4} \tag{6}
$$

Based on the assumption of addictive, independent noise, we can derive the covariance matrix:

$$
\boldsymbol{R}_t = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix} \in R^{4\times 4} \tag{7}
$$

(b) Observation model

$$
z_t = h_t\left(x_t\right) + \delta_t, \quad \delta_t \sim N\left(0, Q_t\right) \tag{8}
$$

where

$$
\begin{aligned}
&x_t = [p_x, v_x, p_y, v_y]^\intercal \in R^{4\times 1} \\
&z_t \in R^{2\times 1} \\
&\delta_t = \delta_1 \in R
\end{aligned}
$$

Specifically, measurement of Euclidean distance in this case can be given as:

$$
z_t = \sqrt{\left(p_x - l_x^i\right)^2 + \left(p_y - l_y^i\right)^2} \tag{9}
$$

where $i-th$ 2D landmark observed by robot in position (p_x, p_y) is:

$$
\left[l_x^i, l_y^i\right]^\intercal \in R^{2\times 1}, for\ i = 1, 2, ...
$$

We can also derive the Jacobain of observation model with respect to the state variables:

$$
\boldsymbol{H}_t = \frac{\partial h_t\left(x_t\right)}{\partial x} = \begin{pmatrix} \frac{p_x - l_x^i}{\sqrt{\left(p_x - l_x^i\right)^2 + \left(p_y - l_y^i\right)^2}} & 0 & \frac{p_x - l_x^i}{\sqrt{\left(p_y - l_y^i\right)^2 + \left(p_y - l_y^i\right)^2}} & 0 \end{pmatrix} \in R^{1\times 4} \tag{10}
$$

The covariance matrix:

$$
\boldsymbol{Q}_t = \delta_1^2 \in R \tag{11}
$$

(c d)
Finally, we can derive the formulation of EKF for this simplified case:

$$
\begin{aligned}
&\check{\boldsymbol{\Sigma}}_t = \boldsymbol{G}_{t-1}\hat{\boldsymbol{\Sigma}}_{t-1}\boldsymbol{G}_{t-1}^{\mathrm{T}} + \boldsymbol{R}_t \\
&\check{\boldsymbol{\mu}}_t = \boldsymbol{g}\left(\hat{\boldsymbol{\mu}}_{t-1}, \boldsymbol{u}_t, \boldsymbol{0}\right) \\
&\boldsymbol{K}_t = \check{\boldsymbol{\Sigma}}_t\boldsymbol{H}_t^{\mathrm{T}}\left(\boldsymbol{H}_t\check{\boldsymbol{\Sigma}}_t\boldsymbol{H}_t^{\mathrm{T}} + \boldsymbol{Q}_t\right)^{-1} \\
&\hat{\boldsymbol{\Sigma}}_t = \left(\boldsymbol{1} - \boldsymbol{K}_t\boldsymbol{H}_t\right)\check{\boldsymbol{\Sigma}}_t \\
&\hat{\boldsymbol{\mu}}_t = \check{\boldsymbol{\mu}}_t + \boldsymbol{K}_t\left(\boldsymbol{z}_t - \boldsymbol{h}\left(\check{\boldsymbol{\mu}}_t, \boldsymbol{0}\right)\right)
\end{aligned} \tag{12}
$$

Where $\boldsymbol{\mu}$ is the mean of Gaussian distribution of the state variables, and $\boldsymbol{\Sigma}$ is the covariance of the distribution of the state variables.

2 Question2: Scan matching using Iterative Closet Point

Here, two methods of data association are used, one is Brute force nearest serach, the other is kdtree. The comparison will be given in Table1. The code can be found in:https://github.com/zhexin1904/EECE5550/tree/mian/HW2

Notice PCL in the solution is only used for the data presentation and kdtree, Brute force matching and all the workflow of ICP is achieved manually.

Result of ICP in $SE3$:

$$T = \begin{pmatrix} 0.947817 & -0.125278 & -0.293168 & 0.483643 \\ 0.196485 & 0.953703 & 0.227703 & -0.248299 \\ 0.251068 & -0.273425 & 0.92855 & 0.255736 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

	Brute force matching	pcl::search::KdTree
RMSE	0.0202175	0.0220894
Time	58.7837s	0.761015s

Table 1: Comparison of two data association methods in SVD-ICP

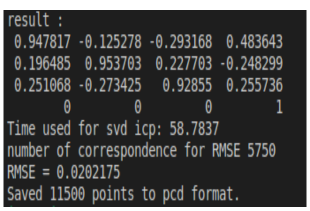

Figure 1: Basic ICP result

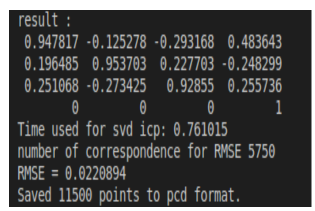

Figure 2: Kdtree ICP result

pcl viewer is used to visualize, the source pointcloud is blue, target pointcloud is green and result pointcloud is red, as Figure 3 and Figure 4 showed.

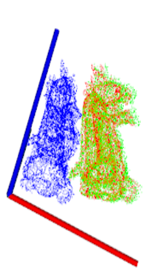

Figure 3: Visulization result 1

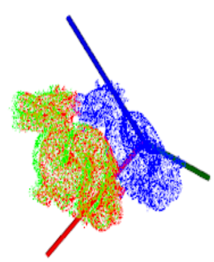

Figure 4: visulization result 2

3 Question 3: Particle Filter

The code is in :https://github.com/zhexin1904/EECE5550/tree/mian/HW2
The pose of robot at time $t = 10$:

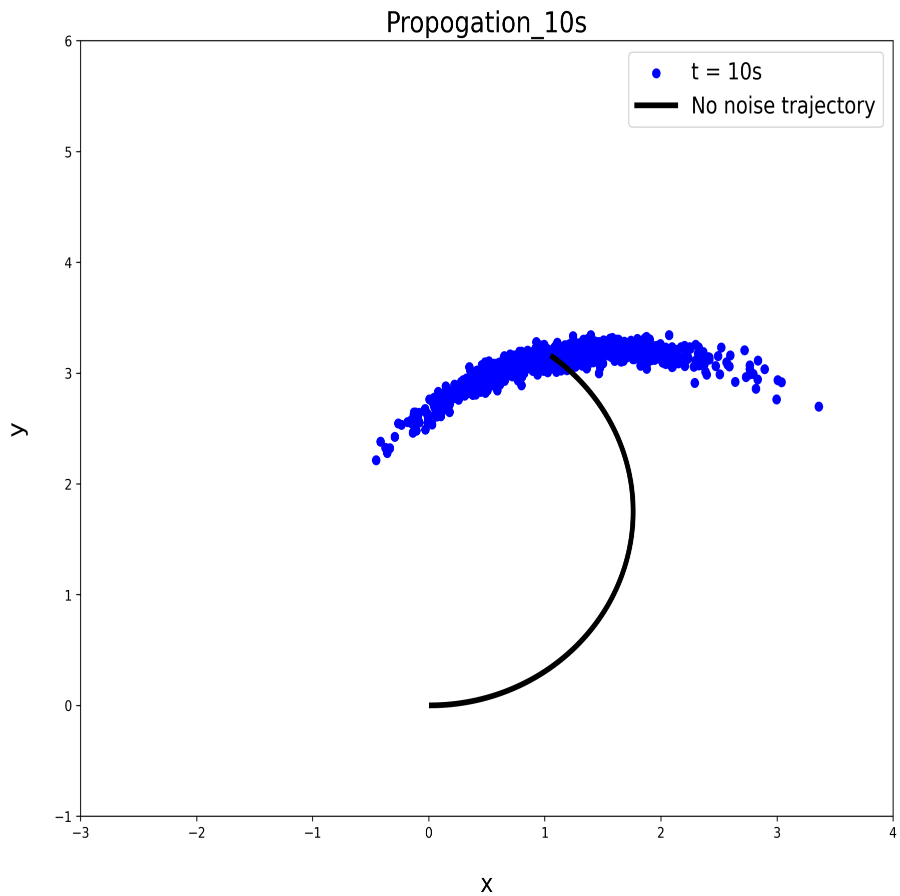

Figure 5: The pose distribution of robot at time $t = 10$

where

$$x_{mean} = 1.064206761941$$
$$y_{mean} = 3.06920707$$
$$cov_{xy} = \begin{pmatrix} 0.38947009 & 0.06859593 \\ 0.06859593 & 0.02674491 \end{pmatrix}$$

The pose of trajectory:

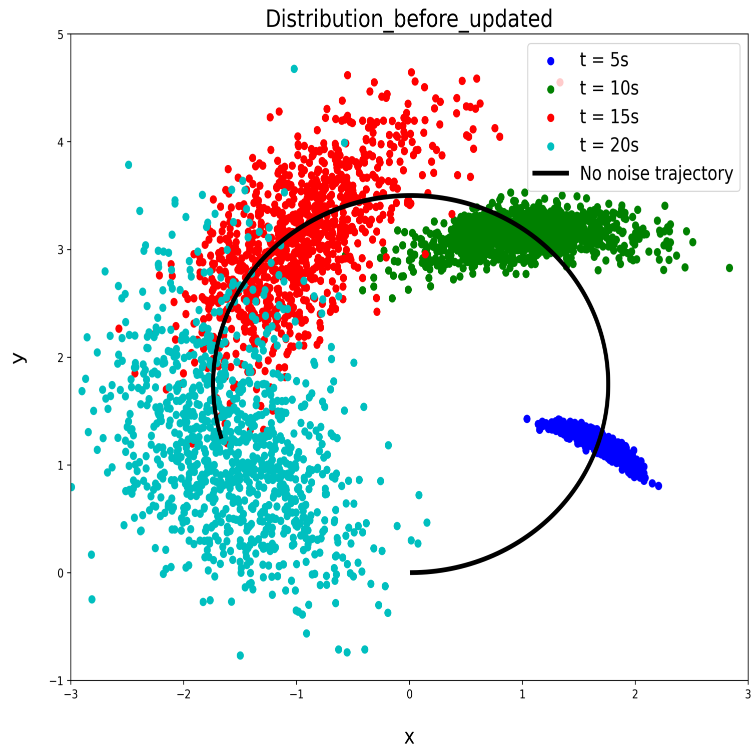

Figure 6: The pose distribution only use propagation

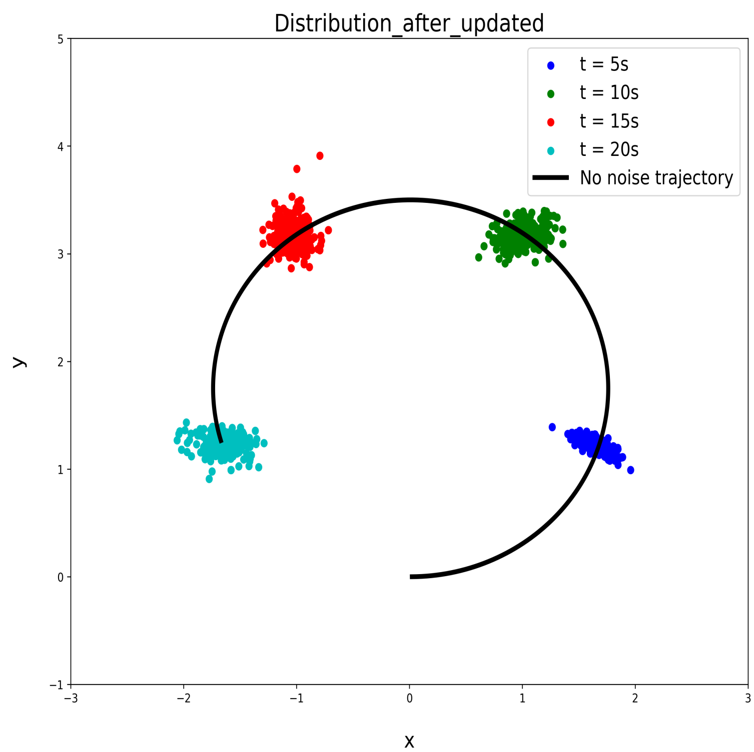

Figure 7: The pose distribution with updating procedure

The mean and covariance in each timestamp after updating is given:
Time at $t = 5s$:

$$x_{mean} = 1.64045907$$
$$y_{mean} = 1.243317277$$
$$cov_{xy} = \begin{pmatrix} 0.00487942 & -0.00190896 \\ 0.00142453 & 0.00555451 \end{pmatrix}$$

Time at $t = 10s$:

$$x_{mean} = 1.007671594$$
$$y_{mean} = 3.1470947299$$
$$cov_{xy} = \begin{pmatrix} 0.00907724 & 0.00142453 \\ 0.00142453 & 0.0055545 \end{pmatrix}$$

Time at $t = 15s$:

$$x_{mean} = -1.007761016$$
$$y_{mean} = 3.2044884599$$
$$cov_{xy} = \begin{pmatrix} 0.00474982 & -0.00019635 \\ -0.00019635 & 0.00702789 \end{pmatrix}$$

Time at $t = 20s$:

$$x_{mean} = -1.6408427499999998$$
$$y_{mean} = 1.215501501$$
$$cov_{xy} = \begin{pmatrix} 0.00673011 & -0.00020435 \\ -0.00020435 & 0.00292774 \end{pmatrix}$$