

Artificial Intelligence in Control Engineering exercise

Lecturer: Dr. Pham Viet Cuong

October 08th, 2018

Group 9:

Nguyen Chinh Thuy	1513372
Nguyen Tan Phu	1512489
Le Van Hoang Phuong	1512579
Do Tieu Thien	1513172
Nguyen Tan Sy	1512872
Nguyen Van Qui	1512702

1 Problem

Description:

A robot operates in an environment where static points (landmarks) are known (coordinates). Using Particle Filter and Extended Kalman Filter, we control the robot to follow a defined path. Actually, the robot has a laser that can measure its relative position to landmarks. Some parameters are given as the following:

* Number of particles: Unlimited and denote as M .

* Number of steps: $N = 625$.

* Control signal frequency: 40Hz.

* Defined path: $XTRUE = [xtrue_1, xtrue_2, \dots, xtrue_N]$ of size $[3 \times N]$, where $xtrue_i = [x_n, y_n, \phi_n]$.

* Landmark: $lm = [coord_1, coord_2, \dots, coord_L]$ of size $[2 \times L]$, where $coord_l = [x_l, y_l]$.

* Measurement: Z of size $[3 \times 5 \times N]$, where $z_n = [z_1, z_2, z_3, z_4, z_5]$ of size $[3 \times 5]$ represents measured range at the time step n , in which $z_i = [r_i, b_i, id_{x_i}]$.

* Control signal: $VG = [vg_1, vg_2, \dots, vg_N]$ of size $[2 \times N]$, where $vg_n = [v_n, g_n]$.

Due to thermal noise, measured values of velocity (v), steering angle (g), range (r), bearing angle (b) have errors. These errors have zero means and standard deviations of $\sigma_v = 0.5m/s$, $\sigma_g = 3^\circ$, $\sigma_r = 0.2m$, $\sigma_b = 2^\circ$, respectively. Data of $XTRUE$, Z , lm , and VG is given in the file "data20171107.m".

Questions:

- Plot the defined path and controlled paths with 3 particles, namely maximum, median, and minimum important factor, using Particle Filter.
- Compute the Root Mean Square error of the controlled paths to the defined path.
- Repeat a and b using Extended Kalman Filter.

2 Implementation

2.1 Particle Filter

2.1.1 Algorithm

To solve the Particle Filter problem, we implement follow below steps:

- Prediction

Process model:

$$x_t = x_{t-1} + V_t \Delta t \cos(\theta_t + \varphi_{t-1}) \quad (1a)$$

$$y_t = y_{t-1} + V_t \Delta t \sin(\theta_t + \varphi_{t-1}) \quad (1b)$$

$$\varphi_t = \varphi_{t-1} + \frac{V_t \Delta t \sin \theta_t}{WB} \quad (1c)$$

Measurement model:

$$r_t = \sqrt{(x_t - x_L)^2 + (y_t - y_L)^2} \quad (2a)$$

$$b_t = \arctan \frac{y_t - y_L}{x_t - x_L} + \varphi_t \quad (2b)$$

- Measurement model
- Implementing a loop with M step which is numbers of particles
- Using process model and control signals u_t in **VG** which are affected by thermal noise to calculate coordinate $x_t^{[m]}$ of robot.
- Combining coordinate of robot from above step and coordinate of landmarks in **lm** to calculate range r_t and bearing angle b_t .
- Calculating importance factor $w_t^{[m]}$ depend on probability density function fomula with μ is matrix of expected range and bearing which are from **Z**.

$$f_x(x_1, x_2, \dots, x_N) = \frac{1}{N} \frac{1}{(2\pi)^{\frac{1}{2}} \|\Sigma\|^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right) \quad (3)$$

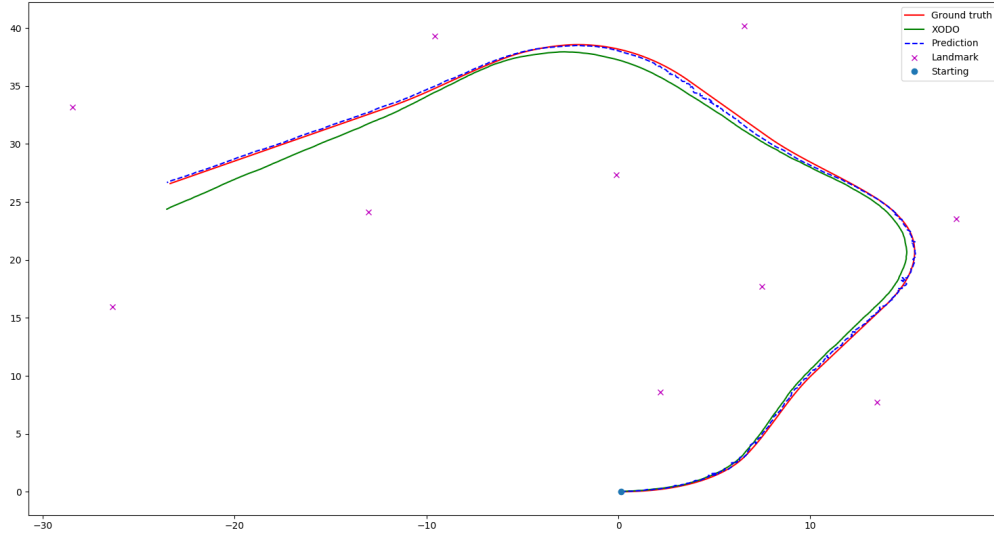
- Selection

- Implementing a loop with M step.
- Choosing a index in range $[1, M]$ for $x_t^{[m]}$ with probabilities $w_t^{[m]}$.

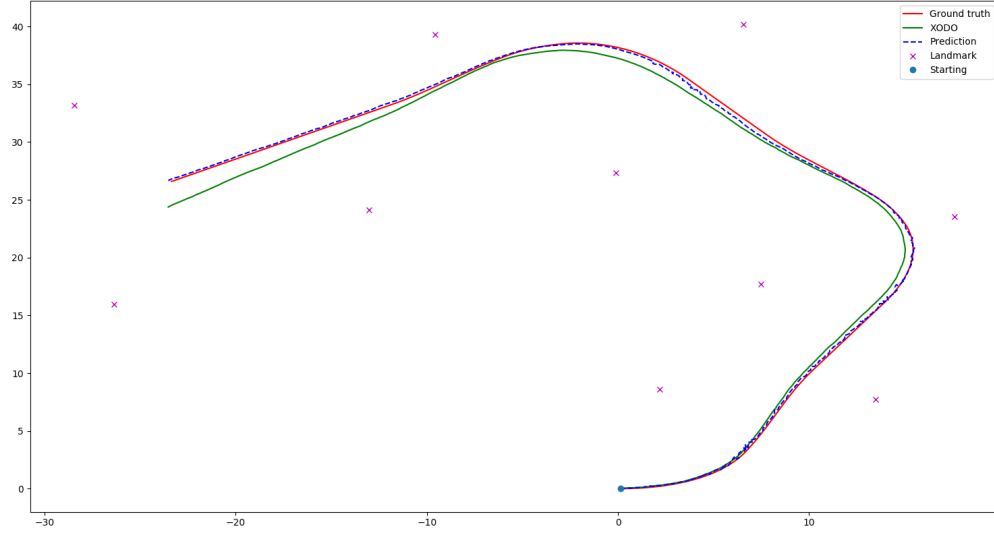
2.1.2 Result

Comments:

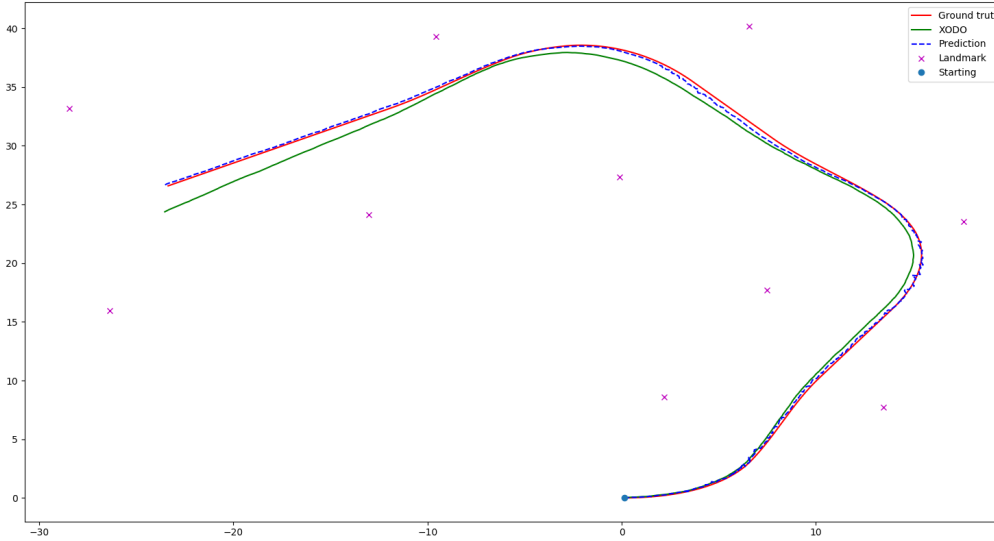
- Overall, three trajectories of Prediction fits Ground truth with the same patterns. However, the best fit is belong to trajctoty with choosing max w_t , so the root mean square is smallest.
- Line of XODO which is calculated from process model is different from Ground truth because of effect on range and bearing angle from thermal noise, while Prediction is calculated and chosen with importance factor w_t . That is the reason why line of Prediction is better than XODO.



(a) Trajectories with max w_t . $RMS = 11.289769$



(b) Trajectories with median w_t . $RMS = 11.428931$



(c) Trajectories with min w_t . $RMS = 11.475249$

Figure 1: Trajectories of Ground truth (XTRUE), XODO, Prediction and root mean square (RMS) of Prediction compared to Ground truth.

2.2 Extended Kalman Filter

2.2.1 Algorithm

- Extended Kalman Filter Overview:

The extended Kalman filter (EKF) calculates an approximation to the true belief. It represents this approximation by a Gaussian distribution. The belief is merely an approximated form, not exact as the case in Kalman filters. The next-state probability and the measurement probability are governed by nonlinear functions.

```

1: Algorithm Extended_Kalman_filter( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):
2:    $\bar{\mu}_t = g(u_t, \mu_{t-1})$ 
3:    $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$ 
4:    $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ 
5:    $\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$ 
6:    $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$ 
7:   return  $\mu_t, \Sigma_t$ 

```

Figure 2: Extended Kalman Filter Algorithm

- Design the system model:

Process model:

$$x_t = x_{t-1} + V_t \Delta t \cos(\theta_t + \varphi_{t-1}) \quad (4a)$$

$$y_t = y_{t-1} + V_t \Delta t \sin(\theta_t + \varphi_{t-1}) \quad (4b)$$

$$\varphi_t = \varphi_{t-1} + \frac{V_t \Delta t \sin \theta_t}{WB} \quad (4c)$$

We model system as a nonlinear model plus noise:

$$x_t = g(u_t, x_{t-1}) + \epsilon_t \quad (5a)$$

Calculate G by taking the Jacobian of g (nonlinear function):

$$G = \begin{bmatrix} 1 & 0 & -R \cos(\theta) + R \cos(\theta + \varphi) \\ 0 & 1 & -R \sin(\theta) + R \sin(\theta + \varphi) \\ 0 & 0 & 1 \end{bmatrix}$$

- Design the measurement model:

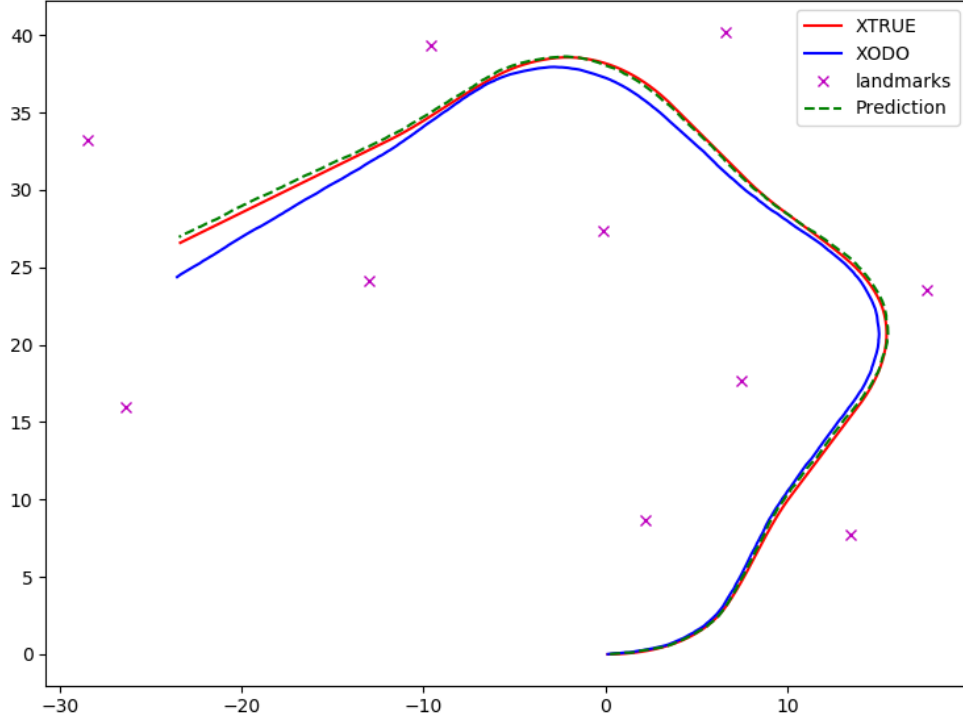
Measurement model: $r(t)$ is range between state of robot and position of landmark. The sensor provides bearing relative to the orientation of the robot, we subtract the robot's orientation from the bearing to get the sensor reading $b(t)$

$$z_t = h(x_t) + \delta_t \quad (6a)$$

$$r_t = \sqrt{(x_t - x_L)^2 + (y_t - y_L)^2} \quad (6b)$$

$$b_t = \arctan \frac{y_t - y_L}{x_t - x_L} + \varphi_t \quad (6c)$$

2.2.2 Result



(a) Root Mean Square (RMS) = 26.201

Figure 3: Position of landmarks, path of Ground truth (XTRUE), XODO, Prediction(using EKF) and RMS.

Comments:

- The prediction result (the green dash line) is quite tight to the ground truth (the red line). This reveals that the Extended Kalman Filter is capable enough for controlling the robot to follow the defined path, despite of the approximation of the belief.
- The value of RMS in the case of using Extended Kalman Filter is worse than using Particle Filter. This can be explained that the Extended Kalman Filter try to approximate the belief by a Gaussian distribution, this may lead to error which depends on how the belief form is closed to a Gaussian distribution. Meanwhile, the Particle Filter does not assume the belief can be approximated by any kind of normal distribution. It directly figure the belief regardless its form of distribution. Therefore, the result of the Particle Filter is more correct than one of the Extended Kalman Filter.
- Nevertheless, when carrying out an experiment to measure execution time between the Particle Filter and Extended Kalman Filter, we recognize that the runtime of the Extended Kalman Filter is much faster than of the Particle Filter (367ms comparing to 4905ms). This is due to the linear approximation of the Extended Kalman Filter. In fact, the process and measurement model is non-linear, which requires a complex computation. The Extended Kalman Filter, in this assignment, proves that its approximation of the non-linear models can reduce the computation time, but only endure a small error. Any way, its result can be accepted.