CMPE245 Project 1

Chien-Pin Chen
06/07/2016

In Extend Kalman Filter, the system model (equation 1 and 2) of the project 1 could refer to:

$$\underline{x}_{k+1} = \underline{f}_k(\underline{x}_k, \underline{w}_k) \tag{1}$$

then, for computing, I could set:

$$\begin{aligned} \underline{x}_{k+1} &= \underline{f}_k(\hat{\underline{x}}_k, 0) \\ \underline{x}_{k+1} &= \begin{bmatrix} \underline{x}_1(k+1) \\ \underline{x}_2(k+1) \\ \underline{x}_3(k+1) \\ \underline{x}_4(k+1) \end{bmatrix} = \begin{bmatrix} \underline{x}_1(k) + \Delta t \cdot x_3 \cos x_4 \\ \underline{x}_2(k) + \Delta t \cdot x_3 \sin x_4 \\ \underline{x}_3(k) \\ \underline{x}_4(k) \end{bmatrix} \end{aligned} \tag{2}$$

$$\underline{\Phi}_k = \frac{\partial \underline{f}_k(\hat{\underline{x}}_k, 0)}{\partial \underline{x}_k} = \begin{bmatrix} 0 & 0 & \Delta t \cdot \cos x_4 & -\Delta t \cdot x_3 \sin x_4 \\ 0 & 0 & \Delta t \cdot \sin x_4 & \Delta t \cdot x_3 \cos x_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$\underline{\Gamma}_k = \frac{\partial \underline{f}_k(\hat{\underline{x}}_k, 0)}{\partial \underline{w}_k} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \delta_v \sqrt{dt} & 0 \\ 0 & \delta_w \sqrt{dt} \end{bmatrix} \tag{4}$$

$$\underline{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

In equation (2), (3), and (4), because the measure data are took every 10 frames (30 frames per second), δt and dt are set as $\frac{1}{3}$. In addition, in equation (2) of the project 1, because $\xi_v(t)$ and $\xi_\theta(t)$ are zero-mean white noises, I set δ_v and δ_θ (in my equation 4) as one at first. Then, I will tune those values according to the result of extended Kalman filter later.

Next, the measurement model (the equation 3) of the project 1 could refer to:

$$\underline{y}_k = \underline{h}_k(\underline{x}_k) + \underline{v}_k \tag{6}$$

then, for computing, I could set:

$$\underline{\text{H}}_k = \frac{\partial \underline{h}_k(\hat{\underline{x}}_k(-))}{\partial x_k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{7}$$

$$R = \begin{bmatrix} \text{var}\{x_m\} & 0 \\ 0 & \text{var}\{y_m\} \end{bmatrix} \tag{8}$$

In equation 8, $var\{x_m\}$ and $var\{y_m\}$ are the variance of the measurement of robot position. Because the provided data are measured from the video of moving epuck, the variance of robot position will depend on the quality of the video. The following steps describe how to compute the variance of any position in the video. First, I use the movie player in MATLAB to load 'Project1Movie.AVI' and exported the first frame to Image tool (also a toolbox in MATLAB). In the first frame, I measure how many pixels between two back lights, and this number (28.75 in figure 1) divide the real distance between two back lights, which is 46 mm, is 1.6 mm per pixel. Because the position of LED light in the video is blurred, it is the noise for the measure data. Then, from figure 1, I get the range of blurred LED light is 7.1 pixel (11.36 mm), and I assume it is 3 times std. of the Gaussian distribution the noise of LED light. Therefore, I get the standard deviation of the noise any position in the video is 0.3786, and the variance is 0.1434.

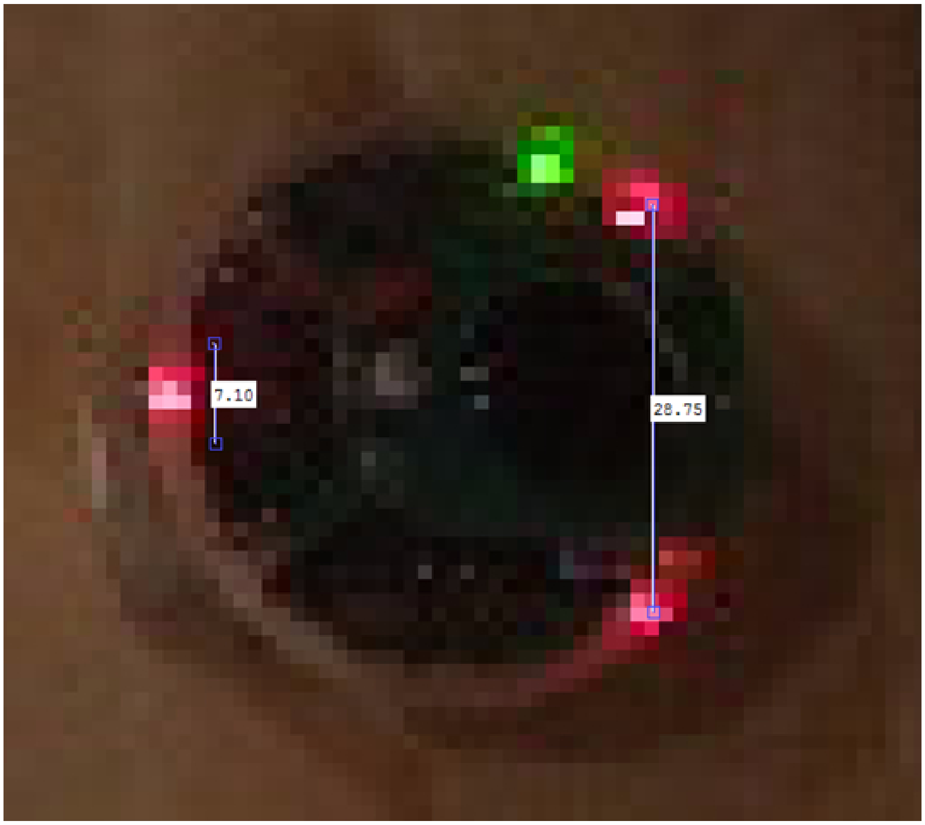

Figure 1: Compute the noise of the video.

In the prediction step of Extended Kalman filter (EKF), I use following equations for discrete time model:

$$\underline{\hat{x}}_{k+1}(-) = \underline{\hat{f}}_k(\underline{\hat{x}}_k, 0) \tag{9}$$

$$\underline{P}_{k+1}(-) = \underline{\Phi}_k \underline{P}_k \underline{\Phi}_k^T + \underline{\Gamma}_k Q \underline{\Gamma}_k^T \tag{10}$$

For initial guess of $\hat{\underline{x}}_0$, I use the first element of the measurement data as $E\{x(0)\}$ and $E\{y(0)\}$, and also use the second element with the first to compute initial velocity ($E\{v(0)\}$) and heading angle ($E\{\theta(0)\}$):

$$\hat{\underline{x}}_0 = \begin{bmatrix} x_m(0) \\ y_m(0) \\ \sqrt{(x_m(1) - x_m(0))^2 + (y_m(1) - y_m(0))^2} \\ \tan^{-1}(\frac{(y_m(1) - y_m(0))}{(x_m(1) - x_m(0))}) \end{bmatrix} \tag{11}$$

For initial guess of $\hat{\underline{P}}_0$, because the robot position is measured by the average of the position of three LED light, which are affected by the variance of the noise. Also, my initial expected value of velocity and heading angle is acquired from initial robot position, so I set the covariance of v_0 and θ_0 according to the variance of the robot position. The following is how I compute the state covariance:

$$\begin{aligned} &\text{cov}\{x, x\} = P_{11} = P_{22} = 3 \times (light\_noise^2)/9 \\ &\text{cov}\{\theta, \theta\} = P_{44} = (10^{pi}/_{180})^2 \\ &P_{13} = P_{14} = P_{23} = P_{24} = P_{11} \\ &\hat{\underline{P}}_0 = \begin{bmatrix} P_{11} & 0 & P_{13} & P_{14} \\ 0 & P_{22} & P_{23} & P_{24} \\ P_{13} & P_{23} & 2^{P_{11}}/_{\Delta t^2} & 0 \\ P_{14} & P_{24} & 0 & P_{44} \end{bmatrix} \end{aligned} \tag{12}$$

Next, I use following equations in the update step of EKF:

$$\underline{K}_{k+1} = \underline{P}_{k+1}(-) \underline{H}_{k+1}^T (\underline{H}_{k+1} \underline{P}_{k+1}(-) \underline{H}_{k+1}^T + \underline{R}_{k+1})^{-1} \tag{13}$$

$$\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_{k+1}(-) + \underline{K}_{k+1}(\underline{y}_{k+1} - \underline{h}_{k+1}(\hat{\underline{x}}_{k+1}(-))) \tag{14}$$

$$\underline{P}_{k+1} = (I - \underline{K}_{k+1} \underline{H}_{k+1}) \underline{P}_{k+1}(-) \tag{15}$$

After updating, do another extended Kalman filter with next measurement data $Z(k)$, which is provided in XYData_cm.csv file.

For second order Kalman filter, I use following equations for prediction step:

$$\hat{\underline{x}}_{k+1} = \underline{f}_k(\hat{\underline{x}}_k, 0) + \underline{b}_k \tag{16}$$

$$\underline{x}_{k+1} = \begin{bmatrix} \underline{x}_1(k+1) \\ \underline{x}_2(k+1) \\ \underline{x}_3(k+1) \\ \underline{x}_4(k+1) \end{bmatrix} = \begin{bmatrix} \underline{x}_1(k) + \Delta t \cdot x_3 \cos x_4 \\ \underline{x}_2(k) + \Delta t \cdot x_3 \sin x_4 \\ \underline{x}_3(k) \\ \underline{x}_4(k) \end{bmatrix} + \underline{b}_k \tag{17}$$

$$\underline{b}_k = \frac{1}{2} \sum_{i=1}^{n} e_i \left(\frac{1}{2} tr \left[F_k^i P_k \right] + \frac{1}{2} tr \left[G_k^i P_k \right] \right), e_i \in R^n \tag{18}$$

$$\underline{P}_{k+1}(-) = \underline{\Phi}_k \underline{P}_k \underline{\Phi}_k^T + \underline{\Gamma}_k Q \underline{\Gamma}_k^T \tag{19}$$

For $\underline{b}_k$ of equation 18, F_k^i is the element of hessian of $\underline{f}_k(\hat{\underline{x}}_k, 0)$ w.r.t $\underline{x}_k$. Therefore, I could compute as:

$$
\begin{gathered}
F_k^1 = \frac{\partial^2 f_k^1(\hat{x}_k,0)}{\partial^2 x_k} = \frac{\partial}{\partial x_k} \begin{bmatrix} 0 \\ 0 \\ \Delta t \cdot \cos x_4 \\ -\Delta t \cdot x_3 \sin x_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\Delta t \cdot \sin x_4 \\ 0 & 0 & -\Delta t \cdot \sin x_4 & -\Delta t \cdot x_3 \cos x_4 \end{bmatrix} \\
F_k^2 = \frac{\partial^2 f_k^2(\hat{x}_k,0)}{\partial^2 x_k} = \frac{\partial}{\partial x_k} \begin{bmatrix} 0 \\ 0 \\ \Delta t \cdot \sin x_4 \\ \Delta t \cdot x_3 \cos x_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta t \cdot \cos x_4 \\ 0 & 0 & \Delta t \cdot \cos x_4 & -\Delta t \cdot x_3 \sin x_4 \end{bmatrix} \\
F_k^3 = \frac{\partial^2 f_k^3(\hat{x}_k,0)}{\partial^2 x_k} = \frac{\partial}{\partial x_k} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \left[0_{4\times 4}\right], F_k^4 = \frac{\partial^2 f_k^4(\hat{x}_k,0)}{\partial^2 x_k} = \frac{\partial}{\partial x_k} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \left[0_{4\times 4}\right]
\end{gathered} \tag{20}
$$

And, G_k^i is the element of hessian of $\underline{f}_k(\hat{\underline{x}}_k, 0)$ w.r.t $\underline{w}_k$. Because it is second partial derivative to $\underline{w}_k$, all of G_k^i are matrices with all zero.
For update step of the second order Kalman filter, I use following eqauation:

$$P_{k+1} = P_{k+1}(-) - P_{k+1}(-)H^T_{k+1}\left[H_{k+1}P_{k+1}(-)H^T_{k+1} + R_{k+1} + S_{k+1}\right]^{-1} H_{k+1}P_{k+1}(-) \tag{21}$$

$$K_{k+1} = P_{K+1}H^T_{k+1}(R_{k+1} + S_{k+1})^{-1} \tag{22}$$

$$\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_{k+1}(-) + K_{k+1}(y_{k+1} - h_{k+1}(\hat{\underline{x}}_{k+1}(-))) + s_{k+1} \tag{23}$$

Also, in my kalman filter loop, I also use the position in previous loop to get the reasonable measurement velocity in order to compare with the expected velocity I get from the EKF. Now, with needed equations and initial values, I could start to run extended Kalman filter and second order filter, First, I run EKF with $\delta_v^2 = 1$ and $\delta_\theta^2 = 1$. The result is shown in figure 2. From the result of the robot position, it shows that the value of δ_v^2 and δ_θ^2 are too large, so my predicted position almost match with the measurement data. Although the result of robot velocity seems fine, the result heading angle is not match the α angle (from HeadingAngle_rad.csv) well. The expected heading angle is not smooth at 12-15 second.
Next, in order to get better result of the heading angle, I use smaller values, such as $\delta_v^2 = 0.05$ and $\delta_\theta^2 = 0.01$. From the result of figure 3, though the expected robot position is not follow the measurement data well, the expected heading angle is more smooth and follow the changing of α angle. Also, the expected velocity smoothly changing, but it do not diverge from the measurement velocity.

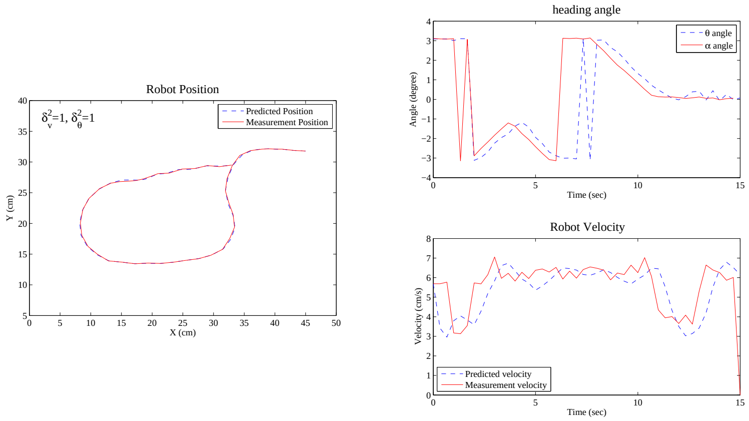

Figure 2: The result of EKF with $\delta_v^2 = 1$ and $\delta_\theta^2 = 1$.

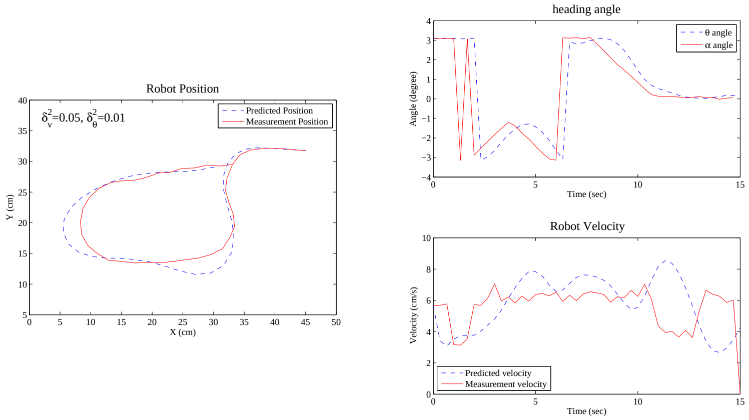

Figure 3: The result of EKF with $\delta_v^2 = 0.05$ and $\delta_\theta^2 = 0.01$.

After I have the suitable value ($\delta_v^2 = 0.05$ and $\delta_\theta^2 = 0.01$) for EKF, I use the same values to run second order filter, as figure 4. The result shows that second order filter seems return almost the same result as EKF (in figure 3). Next, I use larger values (let filter more fit with the measurement) in order to check if second order filter will present different results from the EKF.

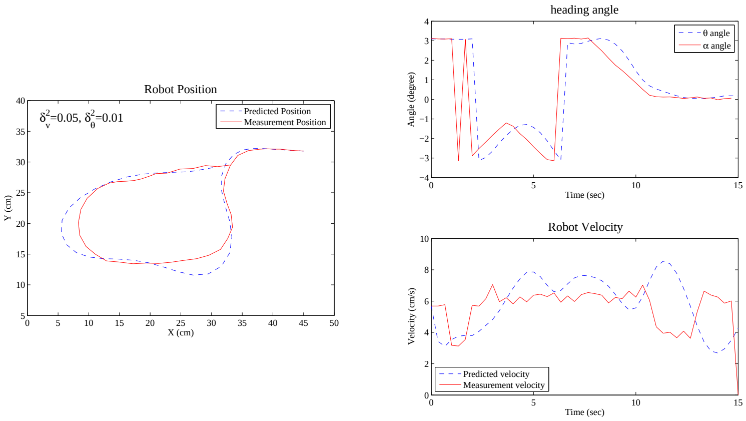

Figure 4: The result of second order filter with $\delta_v^2 = 0.05$ and $\delta_\theta^2 = 0.01$.

Here, I set $\delta_v^2 = 10$ and $\delta_\theta^2 = 10$ to compute EKF (figure 5) and second order filter (figure 6). The result of figure 5 (EKF with $\delta_v^2 = 1$ and $\delta_\theta^2 = 1$) is similar to the result of figure 3 (EKF with $\delta_v^2 = 1$ and $\delta_\theta^2 = 1$), and the expect velocity are more fit (but not smooth) to the measure velocity.

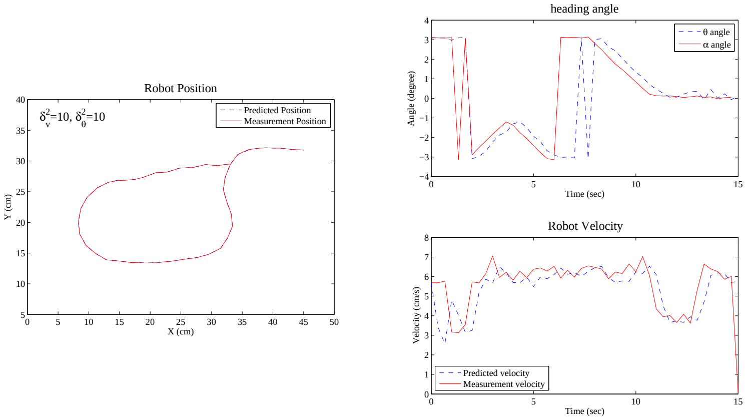

Figure 5: The result of EKF with $\delta_v^2 = 10$ and $\delta_\theta^2 = 10$.

While, in figure 6, it shows that the result of second order filter with larger variance of the process noise ($\delta_v^2 = 10$ and $\delta_\theta^2 = 10$) diverge from the measure data. From figure 5 and figure 6, it shows that second order filter is more sensitive to the change of the variance of the process noise.

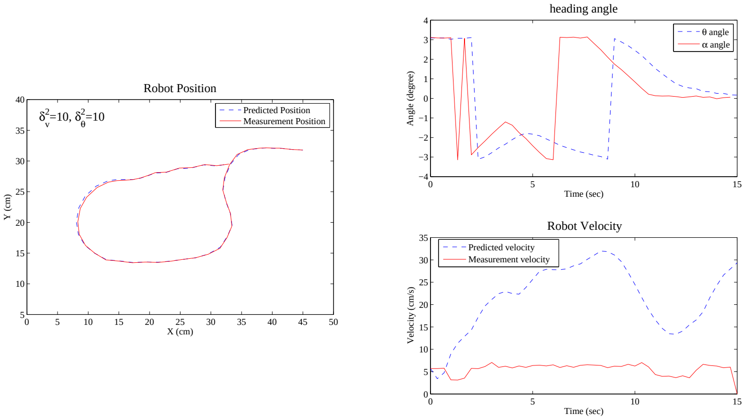

Figure 6: The result of second order filter with $\delta_v^2 = 10$ and $\delta_\theta^2 = 10$.