Fall 2021 ME424 Modern Control and Estimation

Lecture Note 7: Kalman Filter
- Extended Kalman Filter

Prof. Wei Zhang
Department of Mechanical and Energy Engineering
SUSTech Institute of Robotics
Southern University of Science and Technology

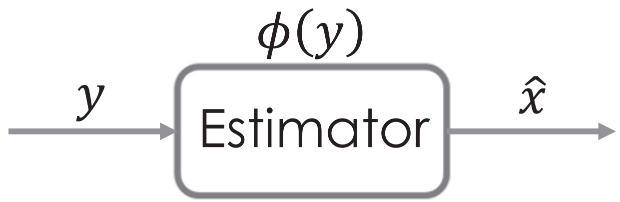
zhangw3@sustech.edu.cn
https://www.wzhanglab.site/

Recall:

- Suppose we want to estimate the value of a hidden random vector $X \in \mathrm{R}^n$ based on observations of a related vector $Y \in \mathrm{R}^m$.

- We have to know the relationship between X and Y. Suppose we take probabilistic viewpoint of their relations, namely, $(X, Y) \sim f_{XY}(x, y)$

- An estimator $\phi(y)$ is a function that maps each measurement $\mathrm{Y} = y$ to an estimate $\hat{x}$



- **MMSE Theorem**: The Minimum Mean-Squared Estimator for X given $Y = y$, that minimizes $E\left(||\phi(Y) - X||^2\right)$ is given by

$$\hat{X}_{MMSE} = \phi_{MMSE}(y) = E(X|Y = y)$$

Recall:

- Kalman filter is a recursive way to compute $E(x_k|Y_k)$ for linear Guassian system

- For nonlinear systems, we can use Extended Kalman Filter (EKF)
  - System setup: $\begin{cases} x_{k+1} = f(x_k, u_k) + w_k \\ y_k = h(x_k, u_k) + v_k \end{cases}$
  - $x_k \in R^n$--- system state at time k
  - $y_k \in R^m$ --- measurement vector at time k
  - $Y_k \triangleq \begin{bmatrix} y_0^T & y_1^T & \ldots & y_k^T \end{bmatrix}^T$ --- collection of measurements up to time k
  - $u_k \in \mathrm{R}^p$--- system input at time k (deterministic input)
  - $w_k \in R^n \sim N(0, Q_k)$, $v_k \in R^p \sim N(0, R_k)$
  - Assume $x_0 \sim N(\mu_0, \Phi_0)$, $x_0 \perp w_k, x_0 \perp v_k, w_k \perp v_j, \forall k, j, and$

$$w_k \perp w_j, \qquad v_k \perp v_j, \qquad \forall k \neq j$$

EKF

Ensemble KF

Uncented KF

Particle Filter

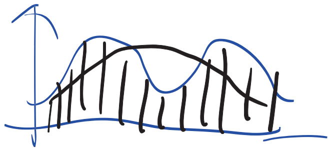
Preview of Extended Kalman Filter

Mean Squared Error

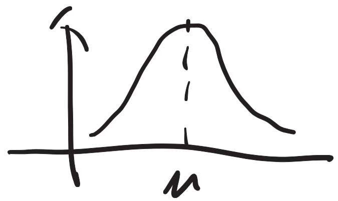
- By fundamental theorem of estimation, we know that the MMSE is given by $E(x_k|Y_k)$ ← we need $f_{x_k|Y_k}(x_k|Y_k)$
  - So we again needs to compute $E(x_k|Y_k)$?

- With nonlinear dynamics, x_k is a random variable that **may not be Gaussian.**

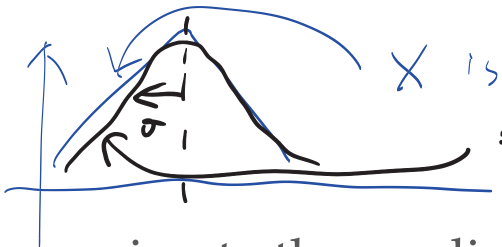
e.g.:



- Extended Kalman Filter tries to
  - Approximate x_k as a Gaussian



what characterizes Gaussian: Mean μ, Σ $\Longrightarrow$



X is not Gaussian

$\hat{X}$ approximate X as Gaussian $\sim \mathcal{N}_{\hat{X}}(\hat{\mu}, \hat{\Sigma})$

  - Approximate the nonlinear dynamics as linear dynamics

matching two moments

$$\begin{cases} \hat{\mu} = E(X) \\ \hat{\Sigma} = cov(X) \end{cases}$$

- Notations: $\hat{x}_{k|k} = E(x_k|Y_k)$, $\quad \hat{x}_{k|k-1} = E(x_k|Y_{k-1})$

$$P_{k|k} = E\left(\left(x_k - \hat{x}_{k|k}\right)\left(x_k - \hat{x}_{k|k}\right)^T \middle| \mathrm{Y_k}\right)$$

$$P_{k|k-1} = E\left(\left(x_k - \hat{x}_{k|k-1}\right)\left(x_k - \hat{x}_{k|k-1}\right)^T \middle| Y_{k-1}\right)$$

Simplified notation: $\hat{x}_k \triangleq \hat{x}_{k|k}$, $\quad P_k = P_{k|k}$

- Goal: recursively compute:

$$\begin{Bmatrix} \hat{x}_k \\ P_k \end{Bmatrix} \xrightarrow{\text{prediction}} \begin{Bmatrix} \hat{x}_{k+1|k} \\ P_{k+1|k} \end{Bmatrix} \xrightarrow{\text{meas. update}} \begin{Bmatrix} \hat{x}_{k+1} \\ P_{k+1} \end{Bmatrix}$$

Extended Kalman Filter Derivation:

- **Step 1: Prediction (via linearization):**
  - Given $\hat{x}_k = E(x_k|Y_k), P_k = E\big((x_k - \hat{x}_k)(x_k - \hat{x}_k)^T|Y_k\big),$
  - Need: $\hat{x}_{k+1|k} = E(x_{k+1}|Y_k),\ P_{k+1|k} = E\Big(\big(x_{k+1} - \hat{x}_{k+1|k}\big)\big(x_{k+1} - \hat{x}_{k+1|k}\big)^T|Y_k\Big)$
  - Recall the linear Gaussian case: $x_{k+1} = Ax_k + Bu_k + w_k,$

    the prediction step: $\hat{x}_{k+1|k} = A_k\hat{x}_k + B_k u_k,\ P_{k+1|k} = A_k P_k A_k^T + Q_k$
  - EKF: Linearize $f(x,u)$ around the current state estimate $\hat{x}_k$ and input u_k

$x_{k+1} = f(x_k, u_k) + w_k$, $f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

$f(x_k, u_k) \approx f(\hat{x}_k, u_k) + \underbrace{\left(\frac{\partial f}{\partial x_k}\Big|_{\hat{x}_k, u_k}\right)}_{\triangleq F_k \in \mathbb{R}^{n\times n}} \cdot (x_k - \hat{x}_k) + \text{H.O.T.}$

$$\Rightarrow \boxed{x_{k+1} \approx F_k x_k + \underbrace{f(\hat{x}_k, u_k) - F_k \hat{x}_k}_{\triangleq \tilde{u}_k} + w_k}$$

$\hat{x}_{k+1|k} = E(x_{k+1}|Y_k) = E(F_k x_k + \tilde{u}_k + w_k|Y_k) = F_k \cdot \hat{x}_k + \tilde{u}_k + 0$

$$= F_k \cancel{\hat{x}_k} + f(\hat{x}_k, u_k) - F_k \cancel{\hat{x}_k}$$

$$= f(\hat{x}_k, u_k) \quad \cdots \text{①}$$

Without linearization

$$E(x_{k+1} | Y_k)$$

$$= E(f(x_k, u_k) + w_k | Y_k) = E(f(x_k, u_k) | Y_k) \overset{?}{\neq} f(\underbrace{E(x_k | Y_k)}_{\hat{x}_k}, u_k)$$

$\Downarrow$

$$E(x^2) \neq (E(x))^2$$

$\downarrow$

$$f(x) = x^2$$

Summary of EKF Prediction Step:

Linearization: $F_k \triangleq \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_k, u_k}$

① $\leftarrow$ $\hat{x}_{k+1|k} = f(\hat{x}_k, u_k), \quad P_{k+1|k} = F_k P_k F_k^T + Q_k$

Extended Kalman Filter Derivation:

- Step 2: Measurement update through linearization:

Recall the linear case:

$$x_{k+1} = Ax_k + Bu_k + w_k$$
$$y_k = Cx_k + Du_k + v_k$$

$$K_{k+1} = P_{k+1|k}C^T\left(CP_{k+1|k}C^T + R_{k+1}\right)^{-1}$$
$$\hat{x}_{k+1} = \hat{x}_{k+1|k} + K_{k+1}\left(y_{k+1} - C\hat{x}_{k+1|k} + Du_{k+1}\right)$$
$$P_{k+1} = (I - K_{k+1}C)P_{k+1|k}$$

we have $\hat{x}_{k+1|k}$ $P_{k+1|k}$

$$y_{k+1} = h(x_{k+1}, u_{k+1}) + v_{k+1} \approx h(\hat{x}_{k+1|k}, u_{k+1}) + \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k+1|k}, u_{k+1}} (x_{k+1} - \hat{x}_{k+1|k}) + H.O.T$$

$\triangleq H_{k+1}$

v_{k+1}

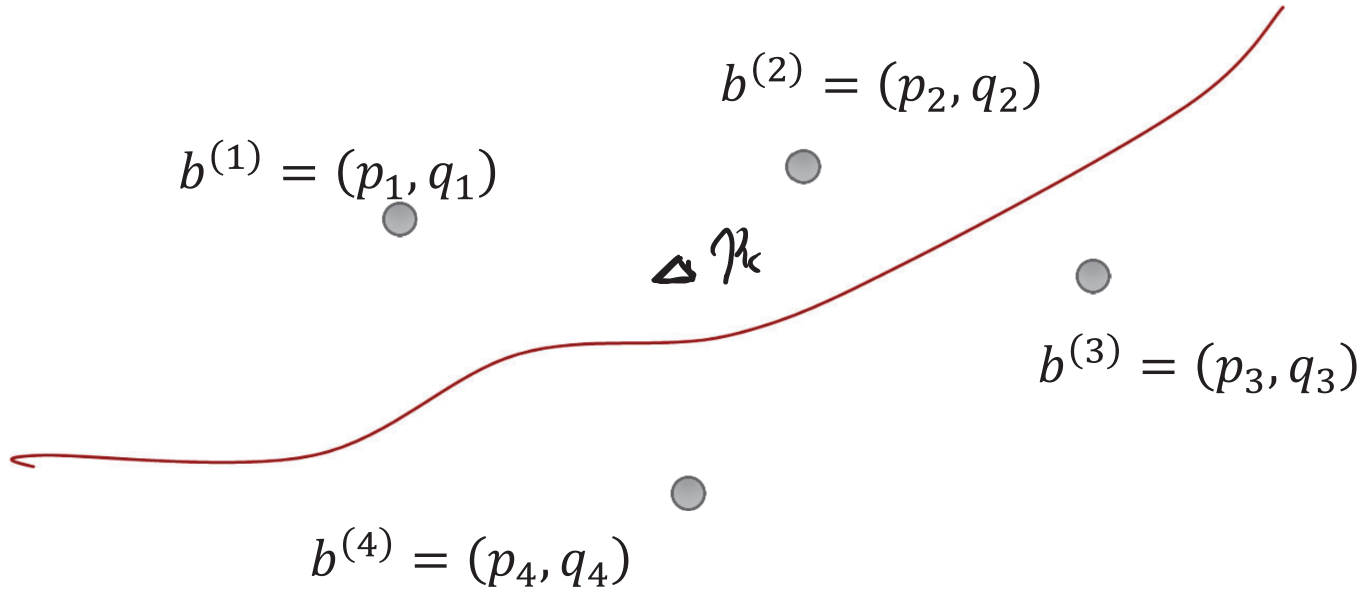
$$\approx \underline{H_{k+1}} \cdot x_{k+1} + \underbrace{h(\hat{x}_{k+1|k}, u_{k+1}) - H_{k+1}\hat{x}_{k+1|k}}_{d_{k+1}} + v_{k+1} + H.O.T.$$

Let $C = H_{k+1}$

plugin to the linear case, (with $D \cdot u_{k+1} = d_{k+1}$)

$$\Rightarrow \boxed{\begin{aligned} \hat{x}_{k+1} &= \hat{x}_{k+1|k} + K_{k+1}\left(y_{k+1} - h(\hat{x}_{k+1|k}, u_{k+1})\right) \\ K_{k+1} &= P_{k+1|k} \cdot H_{k+1}^T \left(H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}\right)^{-1} \\ P_{k+1} &= (I - K_{k+1} H_{k+1}) P_{k+1|k} \end{aligned}}$$

▪ Application Example I for EKF



- beacons with known positions $b^{(i)} = \left(b_1^{(i)}, b_2^{(i)}\right)$
- p_k: robot location at time k
- $y_{k,i}$: range measurement from beacon i at time k.
  - Typical measurement model: $y_{k,i} = \|b^{(i)} - p_k\| + v_i$
- Goal: find the best estimate of p_k given measurement $\{y_0, y_1, \dots, y_k\}$

- Derivation of the system model under constant speed assumption

Here, we want to use dynamics information in addition to the beacon measurement. we assume constant speed motion model:

Model assumption : $\dot{p}_t = \text{constant} \triangleq s \in \mathbb{R}^2$

$$\Rightarrow \boxed{\ddot{p}_t = 0} \leftarrow \text{model}$$

Let's define : $x_t = \begin{bmatrix} p_t \\ \dot{p}_t \end{bmatrix} \in \mathbb{R}^4 \Rightarrow \dot{x}_t = \begin{bmatrix} \dot{p}_t \\ \ddot{p}_t \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}}_{A_c} \begin{bmatrix} p_t \\ \dot{p}_t \end{bmatrix}$

$\Rightarrow$ transform to DT model (Δt)

$$x_{k+1} = x_k + \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x_t \cdot \Delta t$$

$$= \underbrace{\begin{bmatrix} I & I\Delta t \\ 0 & I \end{bmatrix}}_{A} x_k$$

2×2 identity matrix

- **EKF derivation and implementation**

$$y_k = \begin{bmatrix} y_{k,1} \\ y_{k,2} \\ y_{k,3} \\ y_{k,4} \end{bmatrix} = \begin{bmatrix} h_1(x_k) \\ h_2(x_k) \\ h_3(x_k) \\ h_4(x_k) \end{bmatrix} + \begin{bmatrix} v_{k,1} \\ v_{k,2} \\ v_{k,3} \\ v_{k,4} \end{bmatrix}$$

$$h_i(x_k) \triangleq \sqrt{(x_{k,1} - b_1^{(i)})^2 + (x_{k,2} - b_2^{(i)})^2}$$

For EKF: we can use the following model

$$\begin{cases} x_{k+1} = A x_k + w_k \leftarrow \\ y_k = h(x_k) + v_k \end{cases}$$

noise terms are added to account for deviation from the constant speed assumption

$$\Rightarrow \quad F_k = A \qquad \underset{H_k}{C_k} = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k|k-1}} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots \\ \frac{\partial h_2}{\partial x_1} & \vdots & \\ \vdots & \vdots & \\ \frac{\partial h_4}{\partial x_1} & \vdots & \end{bmatrix}$$

Application Example II : Joint State and Parameter Estimation

- Consider a 2^{nd}-order continuous time system:

$$\ddot{y}(t) + 2\xi\omega_n\dot{y}(t) + \omega_n^2 y(t) = \omega_n^2 u(t) \Leftarrow H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

- System state: $x = [y, \dot{y}]^T$, system parameter $\theta = [\xi, \omega_n]^2$
- System input-output u, y
- Question: How to use (u, y) data to jointly estimate x and θ?

Let's

$$\begin{cases} \dot{x} = f(x, u, \theta) + w_x(t) \\ y = h(x, u, \theta) + v(t) \end{cases}$$

parameter

Goal: use $\{u(t), y(t)\}$ to estimate θ, and $\{x(t)\}$

Step 1: view parameter θ as a state

$$\dot{\theta} = 0 + w_\theta(t) \leftarrow$$ artificially imposed to allow estimate to modify

the θ value

Now we have $\tilde{x}(t) \triangleq \begin{bmatrix} x(t) \\ \hline \theta(t) \end{bmatrix} \in \mathbb{R}^n$

$$\dot{\tilde{x}}(t) = \underbrace{\begin{bmatrix} f(x, u, \theta) \\ \hline 0 \end{bmatrix}}_{\tilde{f}(\tilde{x}, u)} + \underbrace{\begin{bmatrix} w_x(t) \\ w_\theta(t) \end{bmatrix}}_{\tilde{w}(t)}$$

$$\tilde{x} = \begin{bmatrix} x \\ \theta \end{bmatrix}$$

$$y(t) = h(\tilde{x}, u) + v(t)$$

Step 2: discretization Δt

$$\begin{cases} \tilde{x}_{k+1} = \boxed{\tilde{x}_k + \tilde{f}(\tilde{x}_k, u_k)\Delta t} + \tilde{w}_k \quad \rightarrow \tilde{f}_d(\tilde{x}_k, u_k) \\ y_k = h(\tilde{x}_k, u_k) + v_k \end{cases}$$

For simplicity, let's assume $w_n = 1$ is known

$$\tilde{x} = \begin{bmatrix} x_1 \\ x_2 \\ z \end{bmatrix} \in \mathbb{R}^3 \qquad x_1 \to y, \; x_2 \to \dot{y}$$

$$\tilde{x}_{k+1} = \overbrace{\begin{bmatrix} \tilde{x}_{k,1} + \tilde{x}_{k,2} \cdot \Delta t \\ \tilde{x}_{k,2} + (-\tilde{x}_{k,1} - 2\tilde{x}_{k,3}\tilde{x}_{k,2} + u_k)\cdot\Delta t \\ \tilde{x}_{k,3} \end{bmatrix}}^{f(\tilde{x}_k,\, u_k)} + w$$

$$F_k = \frac{\partial f}{\partial x}\Big| = \begin{bmatrix} 1 & \Delta t & 0 \\ -\Delta t & 1 - 2x_{k,3}\Delta t & -2x_{k,2}\Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

$$y_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tilde{x}_k$$