Extended Kalman Filter

★ Kalman Filter

⇒ It is a Bayes filter

⇒ Estimator for the linear Gaussian case.

⇒ Optimal solution for linear models and Gaussian distributions.

★ Properties : Marginalization and Conditioning

■ Given $x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}$ $\quad p(x) = \mathcal{N}$ { Gaussian distribution }

■ The marginals are Gaussians

$$p(x_a) = \mathcal{N} \qquad p(x_b) = \mathcal{N}$$

■ As well as the Conditionals

$$p(x_a | x_b) = \mathcal{N} \qquad p(x_b | x_a) = \mathcal{N}$$

★ Marginalization

■ Given $p(x) = p(x_a, x_b) = \mathcal{N}(\mu, \Sigma)$

With $\mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}$, $\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$

■ The marginal distribution is

$$\boxed{p(x_a) = \int_{x_b} p(x_a, x_b)\, dx_b} = \mathcal{N}(\mu, \Sigma)$$

So $\mu = \mu_a \quad \Sigma = \Sigma_{aa}$

★ Conditioning

- Given $P(x) = P(x_a, x_b) = N(\mu, \Sigma)$

$$\text{with } \mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix} \quad \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

- The Conditional distribution is

$$P(x_a | x_b) = \frac{P(x_a, x_b)}{P(x_b)} = N(\mu, \Sigma)$$

$$\text{with } \mu = \mu_a + \Sigma_{ab} \Sigma_{bb}^{-1} (b - \mu_b)$$

$$\Sigma = \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba}$$

★ Linear Model

⇒ The Kalman filter assumes a linear motion and observation model.

⇒ Zero mean Gaussian noise

$$\boxed{x_t = A_t x_{t-1} + B_t u_t + \epsilon_t} \quad \{\text{Linear motion model}\}$$

$$\boxed{z_t = C_t x_t + \delta_t} \quad \{\text{Linear observation model}\}$$

Noise

A_t ⇒ Matrix that describe how the state evolves from $t-1$ to t without control or noise

B_t ⇒ Matrix that describe how the control u_t, changes the state from $t-1$ to t

$C_t \Rightarrow$ Matrix describes how to map the state x_t to an observation z_t.

ε_t, δ_t $\Rightarrow$ Random variables representing the process and measurement noise that are assumed be independent and normally distributed with Covariance R_t and Q_t respectively.

★ **Linear Motion Model**

$$\boxed{\begin{aligned} P(x_t \mid u_t, x_{t-1}) &= \det(2\pi R_t)^{-\frac{1}{2}} \\ &\exp\left(-\tfrac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right) \end{aligned}}$$

★ **Linear Observation Model**

$$\boxed{\begin{aligned} P(z_t \mid x_t) &= \det(2\pi Q_t)^{-\frac{1}{2}} \\ &\exp\left(-\tfrac{1}{2}(z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right) \end{aligned}}$$

★ **Kalman filter Algorithm**

Kalman-filter$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$

$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$ (Kalman gain)

$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$

$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$

return μ_t, Σ_t

★ Non linear Dynamic Systems

⇒ Most realistic problems (in robotics) involve nonlinear functions.

$$\boxed{x_t = g(u_t, x_{t-1}) + \epsilon_t} \qquad \boxed{Z_t = h(x_t) + \delta_t}$$

⇒ The non-linear functions leads to non-Gaussian distribution.

⇒ Kalman filter is not applicable anymore.

⇒ One way to solve the problem is to Perform Local linearization on motion and observation model.

Extended Kalman Filter

★ EKF Linearization: First order taylor Expansion

- **Prediction:**

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \underbrace{\frac{\delta g(u_t, \mu_{t-1})}{\delta x_{t-1}}}_{G_t} (x_{t-1} - \mu_{t-1})$$

- **Correction:**

$$h(x_t) \approx h(\bar{\mu}_t) + \underbrace{\frac{\delta h(\bar{\mu}_t)}{\delta x_t}}_{H_t} (x_t - \bar{\mu}_t)$$

Jacobian matrices

★ Reminder: Jacobian Matrix

⇒ G_t is a non-square matrix $m \times n$ in general.

⇒ Given a vector valued function.

$$g(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{pmatrix} \qquad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

⇒ The Jacobian matrix is defined as:

$$G_x = \frac{\delta g(x)}{\delta x} = \begin{pmatrix} \frac{\delta g_1}{\delta x_1} & \frac{\delta g_1}{\delta x_1} & \cdots & \frac{\delta g_1}{\delta x_n} \\ \frac{\delta g_2}{\delta x_1} & \frac{\delta g_2}{\delta x_n} & \cdots & \frac{\delta g_2}{\delta x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\delta g_m}{\delta x_1} & \frac{\delta g_m}{\delta x_n} & \cdots & \frac{\delta g_m}{\delta x_n} \end{pmatrix}_{m \times n}$$

★ Extended Kalman Filter Algorithm

1 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 μ_t, Σ_t

Complexity: $O(\underbrace{K^{2.4}}_{\text{Size of observation}} + \underbrace{n^2}_{\text{Size of state}})$ {for best inversion algo till now}