$\mathcal{N}$ = normal distr. (gaussian)

Lecture 4 - Extended Kalman Filter

- most frequently used implementation of Bayes Filter
- for gaussian distributions & linear models - it is the most optimal estimator
  - ↳ in reality nothing is perfectly gaussian or linear

Kalman Filter Distribution

- everything is gaussian

$$P(x) = \det(2\pi \Sigma)^{-1/2} \exp\left(-\tfrac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$ → Multivariate Gaussian distr.

↳ Σ Covariance matrix ↳ μ mean

- Given $x = \begin{bmatrix} x_a \\ x_b \end{bmatrix}$ $P(x) = \mathcal{N}$
  - ↳ the marginals are gaussian
    $P(x_a) = \mathcal{N}$ $P(x_b) = \mathcal{N}$
  - ↳ the conditionals are gaussian
    $p(x_a | x_b) = \mathcal{N}$ $p(x_b | x_a) = \mathcal{N}$

Marginalization

- Given $P(X) = P(x_a, x_b) = \mathcal{N}(\mu, \Sigma)$ where $\mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}$, $\Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}$

the marginal distribution is:

$$P(x_a) = \int p(x_a, x_b)\, dx_b = \mathcal{N}(\mu, \Sigma) \quad \text{where } \mu = \mu_a,\ \Sigma = \Sigma_{aa}$$

Conditioning

- Given $p(x) = P(x_a, x_b) = \mathcal{N}(\mu, \Sigma)$ where $\mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}$, $\Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}$

the conditional distr. is:

$$p(x_a | x_b) = \frac{p(x_a, x_b)}{P(x_b)} = \mathcal{N}(\mu, \Sigma) \quad \text{where } \mu = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(b - \mu_b),$$
$$\Sigma = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

k; dimensionality of observation
n; dimensionality of our state

Linear Model

- Kalman Filter assumes linear motion & observation models
- Zero mean Gaussian noise

noise term

motion $x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$
observation $z_t = C_t x_t + \delta_t$

get put into the multivariate gaussian distr.

↳ Design A, B, C depending on application

Components

- A_t Matrix (n,n) that describes how state evolves from $t-1$ to t w/o controls or noise
- B_t Matrix (n,l) that describes how the control u_t changes the state from $t-1$ to t
- C_t Matrix (k,n) that describes how to map the state x_t to an observation z_t (expected observation)

- ε_t, δ_t Random variables representing the process & measurement noise that are assumed to be independent & normally distributed with covariance R_t & Q_t respectfully.

Defining the Linear Motion Model - Distribution

- we have everything we need now

now plug these into Kalman Filter

$$p(x_t \mid x_{t-1}, u_t) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - \underbrace{A_t x_{t-1} - B_t u_t}_{\text{linear model assumption of Kalman Filter}})^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right)$$

↳ R_t describes noise of motion

Defining the Linear Observation Model - Distribution

$$p(z_t \mid x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_t - \underbrace{C_t x_t}_{\text{linear model assumption of Kalman Filter}})^T Q_t^{-1} (z_t - C_t x_t)\right)$$

↳ Q_t describes measurement noise

- Kalman filter is basically a weighted mean between observation & motion

Everything Stays Gaussian

- Given an initial gaussian belief, the belief will always be gaussian

Pred. $\overline{bel}(x_t) = \int \underbrace{p(x_t | u_t, x_{t-1})} bel(x_{t-1})\, dx_{t-1}$ → we have these now ... Done!

Correc. $bel(x_t) = \eta \underbrace{p(z_t | x_t)} \overline{bel}(x_t)$

Kalman Filter Algorithm

1. Kalman_Filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

Prediction

2. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
3. $\bar{\Sigma} = A_t \Sigma_{t-1} A_t^T + R_t$

Correction

4. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
5. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
6. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
7. return μ_t, Σ_t

derived from above equations

2. Project the state
3. Project the error
4. "Kalman gain" weighted mean of observation uncertainty & motion uncertainty
5./6. Correction steps

Sanity Check

- What if we have a perfect sensor? ($Q_t = 0$)

4. $K_t = \cancel{\bar{\Sigma}_t} \cancel{C_t^T} (C_t \cancel{\bar{\Sigma}_t} \cancel{C_t^T} + 0)^{-1} = C_t^{-1}$
5. $\mu_t = \cancel{\bar{\mu}} + C_t^{-1} z_t - \cancel{C_t^{-1} C_t \bar{\mu}_t} = C_t^{-1} z_t$ → C_t^{-1} maps observation to state (vice versa of C_t) so our new mean comes only from observation. Predictions from b4 erased

$\Sigma_t = (I - I)\bar{\Sigma}_t = 0$ predicted error

- What if sensor provides no info? ($Q_t = \infty$)

4. $K_t = \bar{\Sigma}_t C_t^T (\infty)^{-1} = 0$

$\mu_t = \bar{\mu}_t$ → only prediction information

$\Sigma_t = \bar{\Sigma}_t$

EKF: extended Kalman filter

Non-Linear Dynamic Systems - meeting reality

$$x_t = \cancel{A_t x_{t-1} + B_t u_t + \epsilon_t} \qquad z_t = \cancel{C_t x_t + \delta_t}$$

$$\downarrow \qquad\qquad \downarrow$$

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

↳ since these are not linear, we can just linearize them & then use standard Kalman filter alg.

EKF Linearization: 1st Order Taylor

★ the current mean determines the linearization point

Prediction:

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

Correction:

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

Jacobian matrices

Reminder: Jacobian Matrices

- given a vector-valued function $g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_m(x) \end{bmatrix}$

- the Jacobian mtx $(m \times n)$ defined as:

$$G_x = \begin{bmatrix} \frac{dg_1}{dx_1} & \cdots & \frac{dg_1}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{dg_m}{dx_1} & \cdots & \frac{dg_m}{dx_n} \end{bmatrix}$$

→ gives you a plane instead of a slope (line)
↳ in the high dimentional space

Defining New Linearized Motion Model

$$p(x_t \mid u_t, x_{t-1}) \approx \det(2\pi R_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})\right)^T R_t^{-1} \left(x_t - \underbrace{g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})}_{\text{Linearized model}}\right)\right)$$

☆ KF is a weighted mean between $\underbrace{\text{prediction step}}_{\text{from the motion model}}$ & the $\underbrace{\text{correction step}}_{\text{from the observation model}}$

Defining New Linearized Observation Model

$$P(z_t \mid x_t) = \det(2\pi Q_t)^{-1/2} \exp\Big(-\tfrac{1}{2}\underbrace{(z_t - h(\bar{\mu}_t) - H_t(x_t - \bar{\mu}_t))^T Q^{-1}(z_t - h(\bar{\mu}_t) - H_t(x_t - \bar{\mu}_t))}_{\text{Linearized Model}}\Big)$$

- by plugging $P(z_t \mid x_t)$ & $P(x_t \mid x_{t-1}, u_t)$ into bel and bel we get

(EKF) Extended Kalman Filter Algorithm

1. Extended_Kalman_Filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2. $\bar{\mu} = g(u_t, \mu_{t-1})$
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

$\boxed{A_t \rightarrow G_t \quad C_t \rightarrow H_t}$

☆ → we will need to re-build matrices G & H because the point of linearization will change each step.
↳ (first derivative changes)

→ if g & h happen to be linear we still get normal KF.

Summary

- Complexity: $O(k^{2.4} + n^2)$ (k - dimen. of observ. n - nxn covariance matrices)
- requires linear models. if not available:
  - linearize the func. around the current best estimate
    ↳ works well in practice with moderately non-linear functions
- for EKF, the larger the gaussian variance, the worse the linear approximation gets
- → if you have large # vars to estimate or large observation vector, things get costly.