Outline

0. Review

    0.1 EKF & implementation, any questions

    0.2 Unscented transform

    how to generate sigma points

1. UKF and implementation

2. Square-root UKF

    Augmented

3. Colored noise - only for discrete-time

4. Particle filter - brief intro

Ref needed

1) accuracy analysis for UKF

2) colored noise - continuous version

1. Unscented Kalman Filter

Truth model, $x_k \in \mathbb{R}^{n\times 1}$

$$\boxed{x_{k+1} = f(x_k, u_k, k) + w_k}$$

$$\tilde{y}_k = h(x_k, k) + v_k$$

$x_{k+1} = f(x_k, u_k, k, w_k)$

$\tilde{y}_k = h(x_k, k, v_k)$

$w_k \sim N(0, Q)$

$v_k \sim N(0, R)$

Initialize $\hat{x}_0^+ = x_0$ $E\{x_0\}$ (given condition, not the true value)

$\hat{P}_0^+ = P_0$

1) Generate sigma points around $\hat{x}_k^+ \in \mathbb{R}^{n\times 1}$

we have $\bar{x}_k = \hat{x}_k^+$ $P_{xx} = \hat{P}_k^+$ $\bar{x}_0 = \hat{x}_0^+$

$n=2$ $P_{xx} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$

In total, we generate $2n+1$ sigma points

$\mathcal{X}_k^{(1)} = \bar{x}_k + \begin{bmatrix} \sigma_1 \\ 0 \end{bmatrix}$

$\mathcal{X}_k^{(2)} = \bar{x}_k + \begin{bmatrix} 0 \\ \sigma_2 \end{bmatrix}$

$$\mathcal{X}_k^{(0)} = \bar{x}_k \qquad W_0 = \frac{\kappa}{n+\kappa}$$

$$\mathcal{X}_k^{(i)} = \underbrace{\bar{x}_k}_{n\times 1} + \overbrace{\left(\sqrt{\underbrace{(n+\kappa)}_{\text{scalar}}\underbrace{P_{xx}^+}_{n\times n}}\right)^{(i)}}^{n\times 1} \qquad W_i = \frac{1}{2(n+\kappa)} \qquad i = 1, 2, \cdots n$$

$$\mathcal{X}_k^{(i+n)} = \bar{x}_k - \left(\sqrt{(n+\kappa) P_{xx}^+}\right)^{(i)} \qquad W_{i+n} = \frac{1}{2(n+\kappa)}$$

$\mathcal{X}_k^{(i)}$ means i-th sigma point at time step k.

κ is a tuning parameter, can be positive or negative

$\left(\sqrt{(n+\kappa) P_{xx}^+}\right)$ is an $n\times n$ matrix

$\left(\sqrt{(n+\kappa) P_{xx}^+}\right)^{(i)}$ is i-th row or column of $\left(\sqrt{(n+\kappa) P_{xx}^+}\right)$

Note: The sigma points have the same mean and covariance with x_k (mean is $\hat{x}_k^+$, covariance is $\hat{P}_k^+$)

2) propagate sigma points using $f(x_k, u_k, k)$ $(2n+1)$

$$\mathcal{X}_{k+1}^{(i)} = f(\mathcal{X}_k^{(i)}, u_k, k)$$

3) Compute predict mean and covariance

$$\hat{x}_{k+1}^- = \sum_{i=0}^{2n} W_i \mathcal{X}_{k+1}^{(i)}$$

$$\hat{P}_{k+1}^- = \sum_{i=0}^{2n} W_i \{\mathcal{X}_{k+1}^{(i)} - \hat{x}_{k+1}^-\}\{\mathcal{X}_{k+1}^{(i)} - \hat{x}_{k+1}^-\}^T + Q_{k+1}$$

The prediction introduces errors in estimating the mean and covariance at the forth and higher orders in the Taylor series

4) predicted observation

$$\gamma_{k+1}^{(i)} = h(\mathcal{X}_{k+1}^{(i)}, k)$$

$$\hat{y}_{k+1} = \sum_{i=0}^{2n} W_i \gamma_{k+1}^{(i)} \qquad \tilde{y} = h(x_k, k) + v_k$$

$$P_{k+1}^{yy} = \sum_{i=0}^{2n} W_i \{\gamma_{k+1}^{(i)} - \hat{y}_{k+1}\}\{\gamma_{k+1}^{(i)} - \hat{y}_{k+1}\}^T$$

The covariance for $(\tilde{y}_{k+1} - \hat{y}_{k+1})$

$$P_{k+1}^{e_y e_y} = P_{k+1}^{yy} + R_{k+1}$$

$$P_{k+1}^{e_x e_y} = \sum_{i=0}^{2n} W_i \{\mathcal{X}_{k+1}^{(i)} - \hat{x}_{k+1}^{-}\}\{\gamma_{k+1}^{(i)} - \hat{y}_{k+1}\}^T$$

$$e_{k+1}^{-} = \tilde{y}_{k+1} - \hat{y}_{k+1}$$

$$P_{k+1}^{e_y e_y} = E\{e_{k+1}^{-} e_{k+1}^{-T}\}$$

$$P_{k+1}^{e_x e_y} = E\{(\hat{x}_{k+1}^{+} - \hat{x}_{k+1}^{-}) e_k^{-T}\}$$

5) Update

$$\left.\begin{aligned}
\hat{x}_{k+1}^{+} &= \hat{x}_{k+1}^{-} + K_{k+1} e_{k+1}^{-} \\
e_{k+1}^{-} &= \tilde{y}_{k+1} - \hat{y}_{k+1} \\
K_{k+1} &= P_{k+1}^{e_x e_y} (P_{k+1}^{e_y e_y})^{-1} \\
\hat{P}_{k+1}^{+} &= \hat{P}_{k+1}^{-} - K_{k+1} \hat{P}_{k+1}^{-} K_{k+1}^T
\end{aligned}\right\} \text{eq. 3.54 - 3.58}$$

Note: 1) evaluate accuracy (with Gaussian assumption)

x is a random variable with $\bar{x}$ and P_{xx}

$$y = f(x) = f(\bar{x} + \tilde{x})$$

$$= f(\bar{x}) + \frac{\partial f}{\partial x}\Big|_{\bar{x}} (x - \bar{x}) + \frac{1}{2!} \frac{\partial^2 f}{\partial x^2}\Big|_{\bar{x}} (x - \bar{x})^2 + \frac{1}{3!} \frac{\partial^3 f}{\partial x^3}\Big|_{\bar{x}} (x - \bar{x})^3 + \cdots$$

2) κ provides an extra degree of freedom to "fine tune" the higher order moments of the estimation, if x is Gaussian distribution, heuristically, $n + \kappa = 3$

1.2 example

Van der Pol's equation

$$m\ddot{x} + 2c(x^2 - 1)\dot{x} + kx = 0$$

$x_{k+1} = f(x_k, u_k, k) + w_k$

$$\Rightarrow \quad \dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{c}{m}(x_1^2 - 1)x_2 - \frac{k}{m}x_1$$

$$\tilde{y} = x_1 + v \quad \Rightarrow \quad \tilde{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v$$

1.3 Square-root UKF

(textbook Section 4.1 Fractorization method)

$$P = SS^T$$

propagate through S

Summary for UKF

1. it is more accurate than EKF

2. we don't have to compute the Jacobian
   hence it's more computational efficient

3. we have to tune parameters carefully
   (section 3.7 has details)

4. numerically stability is related to

5. There are more methods to generate
   sigma points (Ref: Optimal state estimation)
   Simon

6. the performance depends on the nonlinearity
   and noise distribution

7. For scenarios as multimodal and
   occlusions , UKF won't work.

$\Rightarrow$ Particle filter

Example when the process noise and the measurement noise are correlated

for an aircraft, we have aircraft dynamics $\dot{x}$

sensor: anemometer to measure wind speed (V_k) correlated with W_k

at the same time, wind is one of the input for aircraft dynamics.

2. Colored process noise

Suppose we have $x_{k+1} = \Phi_k x_k + w_k$

$E\{w_k w_k^T\} = Q_k$

w_k is not white (w_k and w_{k+1} is correlated)

Assume that the process noise is the output of a dynamic system:

$$w_{k+1} = \psi_k w_k + \zeta_k$$

ζ_k is a zero-mean white noise, $E\{\zeta_k w_k\} = 0$

The covariance between w_{k+1} and w_k is

$$E\{w_{k+1} w_k^T\} = E\{(\psi_k w_k + \zeta_k) w_k^T\}$$

$$= E\{\psi_k w_k w_k^T\} + E\{\cancel{\zeta_k w_k^T}\}^{\nearrow 0}$$

$$= \psi_k Q_k$$

$$\begin{bmatrix} x_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} \Phi_k & I \\ 0 & \psi_k \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} + \begin{bmatrix} 0 \\ \zeta_k \end{bmatrix}$$

$$x'_{k+1} = \Phi_k' x_k' + w_k'$$

Same strategy can be applied to

$$\tilde{y}_k = H_k x_k + V_k$$

where V_k is colored

$$V_{k+1} = \psi_k V_k + \zeta_k$$

Extra

parameter estimation using KF

example 3.6

$$m\ddot{x} + 2c(x^2-1)\dot{x} + kx = 0$$

$x_{k+1} = f(x_k, u_k, k) + w_k$

$$\Rightarrow \quad \dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{c}{m}(x_1^2-1)x_2 - \frac{k}{m}x_1$$

Suppose $c=1$, $k=1$, but we don't know m

Let $x_3 = m$

$$\dot{x}_3 = 0$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{1}{x_3}(x_1^2-1)x_2 - \frac{1}{x_3}x_1$$

$$\dot{x}_3 = 0$$

measure something (x_1, x_2)

Particle filter

Monte-Carlo method: using a finite number of randomly sampled points to compute a result

In a nutshell, the particle filter is to generate enough points to get a representative sample of the problem, run these points through the system, then compute results on the transformed points based on measurements.

1) Generic PF algorithm

① Randomly generate a bunch of particles

each particle has a weight indicating how likely it matches the actual state

Initialize each particle with same weight.

② predict the next state of particles

propagate all particles based on state dynamics

③ Update

Update the weight of the particles based on measurements. Particles that closely match the measurements are weighted higher than those which don't match the measurement very well.

④ Resample

Discard highly improbable particle and replace them with copies of the more probable particles

⑤ computed weighted mean & covariance

Mathematical findation for PF: Bayesian state estimation

Suppose we have $\quad x_k = f(x_{k-1}, w_{k-1})$

$$x_{k+1} = f(x_k, w_k) \qquad E\{w_k v_k^T\} = 0$$

$$\tilde{y}_k = h(x_k, v_k)$$

Assume we know the pdf of $\{w_k\}$ and $\{v_k\}$

Goal: approximate pdf of x_k based on $\{\tilde{y}_1, \cdots \tilde{y}_k\}$

denoted as $\underbrace{p(x_k \mid Y_k)}_{\text{pdf}}$ where $Y_k = \{\tilde{y}_1, \cdots \tilde{y}_k\}$

What we have: x_0 and the pdf of x_0, $p(x_0)$

we don't have $\tilde{y}_0$

then $\quad p(x_0 \mid Y_0) = p(x_0) \qquad Y_0 = \{\phi\}$

Bayesian state estimation is to find a recursive way

to compute $p(x_k \mid Y_k)$

First, let's find $p(x_k \mid Y_{k-1})$

Review: Suppose x_1 and x_2 are two random variables

$$p(x_1 \mid x_2) = \frac{p(x_1, x_2)}{p(x_2)}$$

$$p(x_1) = \int_{-\infty}^{\infty} p(x_1, x_2)\, dx_2$$

$$p(x_2) = \int_{-\infty}^{\infty} p(x_1, x_2)\, dx_1$$

$$x_1 \Rightarrow x_k \qquad x_2 \Rightarrow x_{k-1}$$

$$p(x_k \mid Y_{k-1}) = \int p(x_k, x_{k-1} \mid Y_{k-1})\, dx_{k-1}$$

$$= \int p[x_k \mid (x_{k-1}, Y_{k-1})]\, p(x_{k-1} \mid Y_{k-1})\, dx_{k-1}$$

x_k is entirely determined by x_{k-1} and w_{k-1}

$$\Rightarrow p[x_k \mid (x_{k-1}, Y_{k-1})] = p(x_k \mid x_{k-1})$$

Now $p(x_k \mid Y_{k-1}) = \int p(x_k \mid x_{k-1})\, p(x_{k-1} \mid Y_{k-1})\, dx_{k-1}$

$p(x_{k-1} \mid Y_{k-1})$ is not available yet, but we do know $p(x_0 \mid Y_0)$

$p(x_k \mid x_{k-1})$ is available

$$p(x_k \mid Y_k) = \frac{p(Y_k \mid x_k)\, p(x_k)}{p(Y_k)}$$ Bayes's theorem

$$= \frac{p(Y_k \mid x_k)}{p(Y_k)} \underbrace{\frac{p(x_k \mid Y_{k-1})\, p(Y_{k-1})}{p(Y_{k-1} \mid x_k)}}_{p(x_k)}$$

$$= \frac{p(y_k, Y_{k-1} \mid x_k)}{p(y_k, Y_{k-1})} \frac{p(x_k \mid Y_{k-1})\, p(Y_{k-1})}{p(Y_{k-1} \mid x_k)}$$

$$= \frac{p(x_k, y_k, Y_{k-1})}{p(x_k)\, p(y_k, Y_{k-1})} \frac{p(x_k, Y_{k-1})\, \cancel{p(Y_{k-1})}}{\cancel{p(Y_{k-1})}\, p(Y_{k-1} \mid x_k)}$$

$$p(x_k \mid Y_k) = \frac{p(x_k, y_k, Y_{k-1})}{p(x_k)\, p(y_k, Y_{k-1})} \frac{p(x_k, Y_{k-1})}{p(Y_{k-1} \mid x_k)} \frac{p(x_k, y_k)}{p(x_k, y_k)}$$

$$= \frac{p[Y_{k-1} \mid (x_k, y_k)]\, p(y_k \mid x_k)\, p(x_k \mid Y_{k-1})}{p(y_k \mid Y_{k-1})\, p(Y_{k-1} \mid x_k)}$$

$$p[Y_{k-1} \mid (x_k, y_k)] = p[Y_{k-1} \mid x_k]$$

we have $p[Y_{k-1} \mid (x_k, y_k)] = p[Y_{k-1} \mid x_k]$

$$p(x_k \mid Y_k) = \frac{p(y_k \mid x_k)\, p(x_k \mid Y_{k-1})}{p(y_k \mid Y_{k-1})}$$

we can compute $p(y_k \mid x_k)$ and $p(x_{k+1} \mid x_k)$ through

$h(x_k, v_k)$ and $f(x_k, w_k)$

$$p(y_k \mid Y_{k-1}) = \int p(y_k \mid x_k)\, p(x_k \mid Y_{k-1})\, dx_k$$