

This explanation flips the script on normal introductions to the Kalman Filter (KF), working backwards from a full particle filter (PF). This approach makes much more sense because PFs make much more sense - simulate a lot of possibilities, pick the best one, and continue.

Fun application: PFs can be used in multi-modal filters for GPS. Going from a PF to a KF entails adding stronger assumptions about the system:

PF \implies Unscented KF (UKF) : assume gaussian distribution requires unimodal distribution.

UKF \implies EKF : assume mean propagates, differentiable dynamics, find Jacobians.

EKF \implies KF : Linear system

\hat{x} - estimate

$$z = h(x) + v$$

$$x_{k+1} = f(x_k, t) + v$$

$$\hat{x}_{k+1} = f(\hat{x}_k, t)$$

$$P_k = E((x_k - \hat{x}_k)(x_k - \hat{x}_k)^T)$$

Diagonal matrix of variances if we have uncorrelated noise

$$\hat{x}_{k+1} + \Delta x_{k+1} = f(\hat{x}_k) + \frac{\delta f}{dx}|_{\hat{x}_k} \Delta x_k$$

$$\hat{x}_{k+1} = F \Delta x_k + v$$

F is derivative of the dynamics at the state \hat{x}_k Solve for $\Delta x \Delta x^T$

$$E(F \Delta x \Delta x^T F^T + v v^T + F \Delta x_k v^T + v \Delta x_k^T F^T)$$

Expected value of last two terms is zero if noise is white and uncorrelated

$$P_{k+1} = E(F \Delta x_k \Delta x_k^T F^T) + E(v v^T)$$

$$P_{k+1} = F P_k F^T + Q$$

Run extended kalman filter

$$\hat{x}_{k+1}^* = f(\hat{x}_k)$$

$$F = \frac{\delta}{dx} = \text{make} F(\hat{x})$$

$$P_{k+1}^* = F P_k F^T + Q$$

$$\hat{z}_{k+1} = h(\hat{x}_{k+1})$$

$$y = z_{k+1} - \hat{z}_{k+1}$$

Want to know covariance of y (this will be S)

$$S = H P^* H^T + R$$

H is linearized h K - kalman gain

$$K = P * H^T S^{-1}$$

$$\hat{x}_{k+1} = \hat{x}_{k+1} + Ky$$

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

$$P_{k+1} = P *_{k+1} - KHP *_{k+1}$$

Process

1. Figure out f and h
2. Figure out F and H
3. Figure out R and Q
4. Initial state and uncertainty

Where things fall apart: Negative eigenvalues for covariance matrices can happen $P*$ becomes negative How to avoid this: Joseph Form.

$$jf = (1 - KH)P * (1 - KH)^T + KRK^T$$

What if you take P and instead use $P = UDU^T$ makes numerical problems go away UD filters 32 bit or less

What if you have unobservable errors? Considerate covariance - add a bit to covariance $P *_{k+1} = P *_{k+1} + F$ Need to use joeseph form to do updates.

Speed - Big problem is the matrix inverse in $K = P * H^T S^{-1}$ if R is a diagonal matrix, do a bunch of scalar equations

How do you know your filter is running properly Two covariance matrices Record real error over time vs calculated covariance. See if distribution of real error fits inside calculated covariance. Real systems: y should be uncorrelated over time

Real filters IMU type filters

$$x_{k+1} = Fx_k + Gu_k$$

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

Don't model real noise because you have a very good IMU For an IMU type filter, just use ω as the Input $z = [q\theta\gamma\beta]^T$ Process noise becomes ability to sense

Q: best way to deal with kalman filtering 6dof+ systems? Q: how do you come up with the probability distribution ? Q: sensor noise independence?