Kalman Filter Part II: EKF

Non-linear Dynamical Systems

In Kalman Filter Part I: Kalman Filter, we assume the transition matrix Fk and sensors matrix, Hk are both linear.

For example, the Fk matrix of CV (Constant Velocity) motion model is:

$$F = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

However the real world is not that simple. The objects we want to track may change its acceleration or make turn.

In a solid MOT system, we may assume a more complex motion model such as CTRV (Constant Turn Rate and Velocity) model

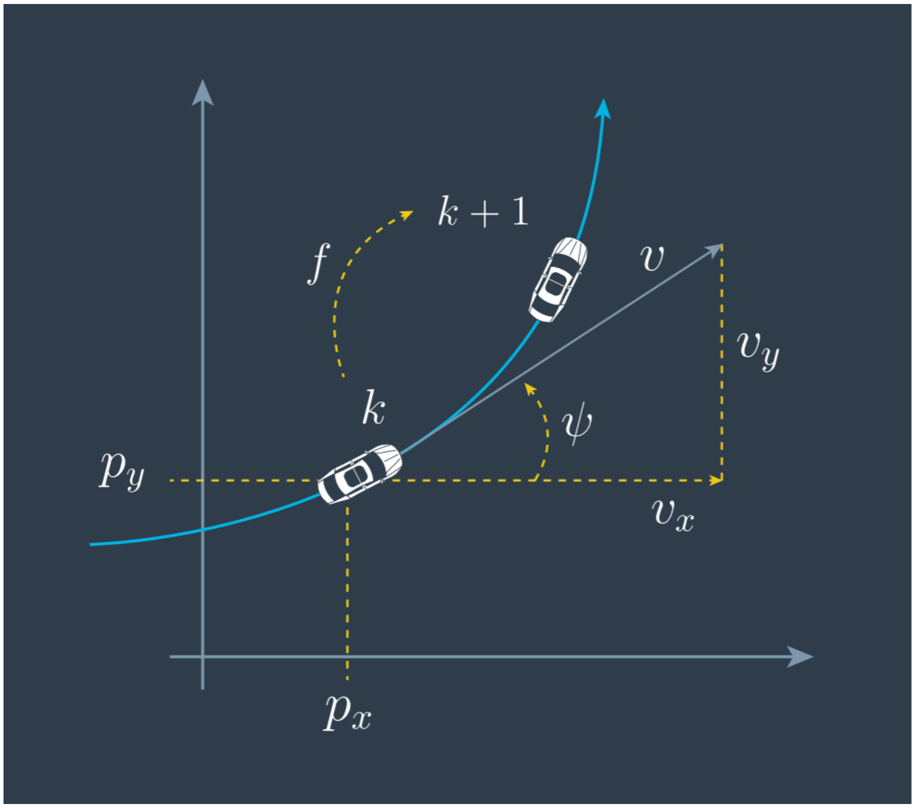
or use IMM (Interactive Multiple Models) for implementation.

Let's take CTRV for an example, the state vector **x** would become:

$$x = \begin{pmatrix} p_x \\ p_y \\ v \\ \psi \\ \dot{\psi} \end{pmatrix}$$

The px,py are the positions, v means the velocity, yaw is the angle between the velocity vector and the x axis and (yaw rate)

is the change rate of



Let's ignore the noise first, the state vector prediction flow would become:

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{p}_x(t) \\ \dot{p}_y(t) \\ \dot{v}(t) \\ \dot{\psi}(t) \\ \ddot{\psi}(t) \end{bmatrix} dt$$

$$x_{k+1} = x_k + \begin{bmatrix} \int_{t_k}^{t_{k+1}} v(t) \cdot cos(\psi(t))dt \\ \int_{t_k}^{t_{k+1}} v(t) \cdot sin(\psi(t))dt \\ 0 \\ \dot{\psi}_k \cdot \Delta t \\ 0 \end{bmatrix}$$

After the integration, it becomes:

$$x_{k+1} = x_k + \begin{bmatrix} \frac{v_k}{\dot{\psi}_k}(sin(\psi_k + \dot{\psi}_k \Delta t) - sin(\psi_k)) \\ \frac{v_k}{\dot{\psi}_k}(-cos(\psi_k + \dot{\psi}_k \Delta t) + cos(\psi_k)) \\ 0 \\ \dot{\psi}_k \cdot \Delta t \\ 0 \end{bmatrix}$$

We need to be careful here when is zero. However, since the cars always move forward, the yaw rate won't change.

Therefore, the equation becomes:

$$x_{k+1} = x_k + \begin{bmatrix} v_k cos(\psi_k) \Delta t \\ v_k sin(\psi_k) \Delta t \\ 0 \\ \dot{\psi}_k \cdot \Delta t \\ 0 \end{bmatrix}$$

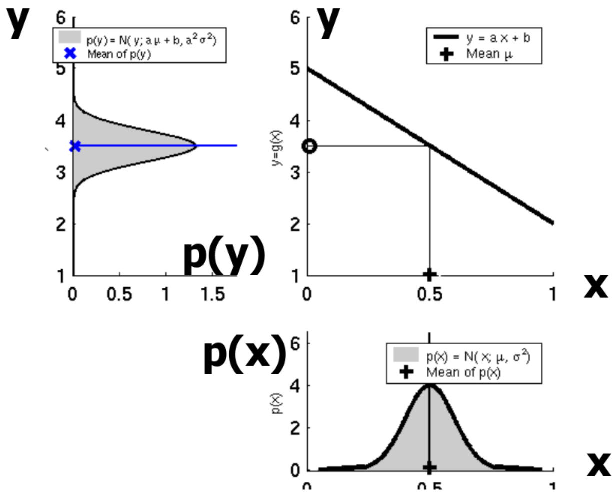
Obviously it's a non-linear translation.

Extended Kalman Filter

In order to make state estimation on nonlinear systems, or parameter estimation, using the Kalman filter, one of the possible approaches is to linearize the system under investigation around its current state and force the filter to use this linearized version of your system as a model. This is the Extended Kalman Filter, or EKF.

With linearity assumption, the prediction and measurement of Kalman Filter look like:

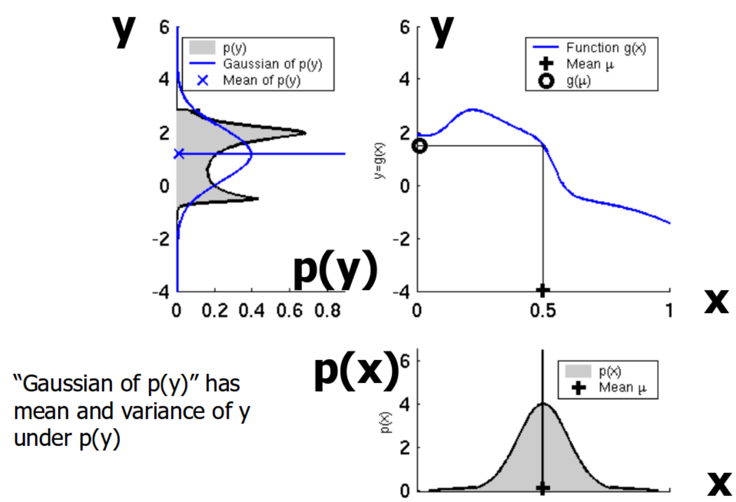
$$\begin{aligned} X_{t+1} &= A_t X_t + B_t u_t + \varepsilon_t \quad \varepsilon_t \sim \mathcal{N}(0, Q_t) \\ Z_t &= C_t X_t + d_t + \delta_t \quad \delta_t \sim \mathcal{N}(0, R_t) \end{aligned}$$



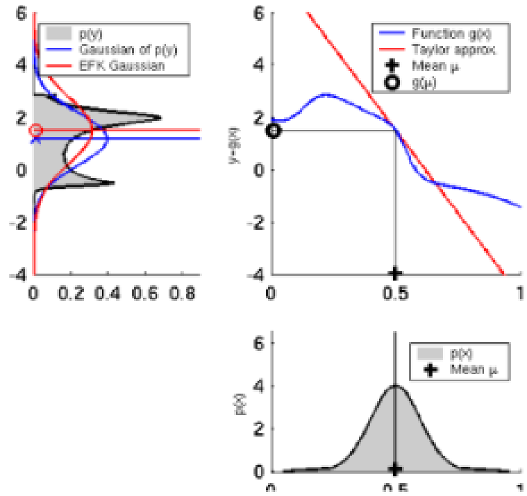
However, most realistic robotic problems involve nonlinear functions:

$$X_{t+1} = f_t(X_t, u_t) + \varepsilon_t \quad \varepsilon_t \sim \mathcal{N}(0, Q_t)$$

$$Z_t = h_t(X_t) + \delta_t \quad \delta_t \sim \mathcal{N}(0, R_t)$$



What EKF does is to take the best estimate (which is the mean of the distribution) and fit a linear function on it using **first order Taylor expansion**. It would look like:



The prediction and measurement would become:

$$
\begin{aligned}
f_t(x_t, u_t) &\approx f_t(\mu_t, u_t) + \frac{\partial f_t(\mu_t, u_t)}{\partial x_t}(x_t - \mu_t) \\
&= f_t(\mu_t, u_t) + F_t(x_t - \mu_t)
\end{aligned}
$$

$$
\begin{aligned}
h_t(x_t) &\approx h_t(\mu_t) + \frac{\partial h_t(\mu_t)}{\partial x_t}(x_t - \mu_t) \\
&= h_t(\mu_t) + H_t(x_t - \mu_t)
\end{aligned}
$$

The second part of the equation is Jacobian matrices, please refer to Appendix A: Jacobian Matrix

EKF Algorithm

- At time 0: $X_0 \sim \mathcal{N}(\mu_{0|0}, \Sigma_{0|0})$
- For t = 1, 2, …
  - Dynamics update: $f_t(x_t, u_t) \approx a_{0,t} + F_t(x_t - \mu_{t|0:t})$

$$
\begin{aligned}
(a_{0,t}, F_t) &= \text{linearize}(f_t, \mu_{t|0:t}, \Sigma_{t|0:t}, u_t) \\
\mu_{t+1|0:t} &= a_{0,t} \\
\Sigma_{t+1|0:t} &= F_t \Sigma_{t|0:t} F_t^\top + Q_t
\end{aligned}
$$

  - Measurement update: $h_{t+1}(x_{t+1}) \approx c_{0,t+1} + H_{t+1}(x_{t+1} - \mu_{t+1|0:t})$

$$
\begin{aligned}
(c_{0,t+1}, H_{t+1}) &= \text{linearize}(h_{t+1}, \mu_{t+1|0:t}, \Sigma_{t+1|0:t}) \\
K_{t+1} &= \Sigma_{t+1|0:t} H_{t+1}^\top (H_{t+1} \Sigma_{t+1|0:t} H_{t+1}^\top + R_{t+1})^{-1} \\
\mu_{t+1|0:t+1} &= \mu_{t+1|0:t} + K_{t+1}(z_{t+1} - c_{0,t+1}) \\
\Sigma_{t+1|0:t+1} &= (I - K_{t+1} H_{t+1}) \Sigma_{t+1|0:t}
\end{aligned}
$$

EKF Summary

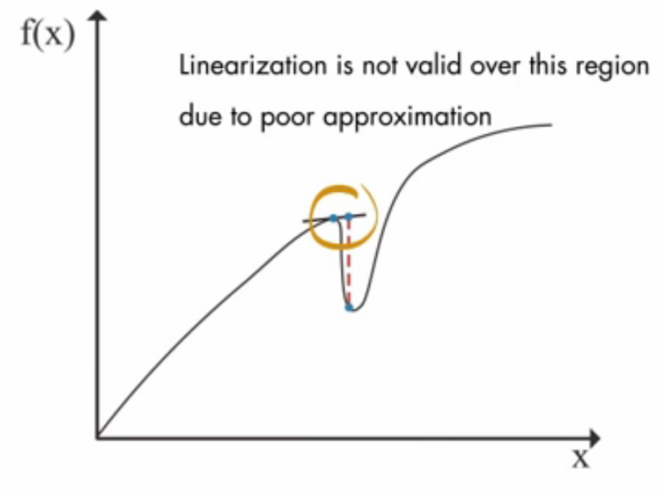
Efficiency:

Polynomial in measurement dimensionality k and state dimensionality n:

$$O(K^{2.376} + n^2)$$

Drawbacks:

1. It is difficult to calculate the Jacobians
2. High computational cost
3. Only works on systems that have differentiable model
4. Not optimal and can diverge if nonlinearities are too large:


f(x)
Linearization is not valid over this region
due to poor approximation
x