The Extended Kalman Filter

Dr. Steve Weller

steven.weller@newcastle.edu.au

School of Electrical Engineering and Computer Science

The University of Newcastle, Callaghan, NSW 2308

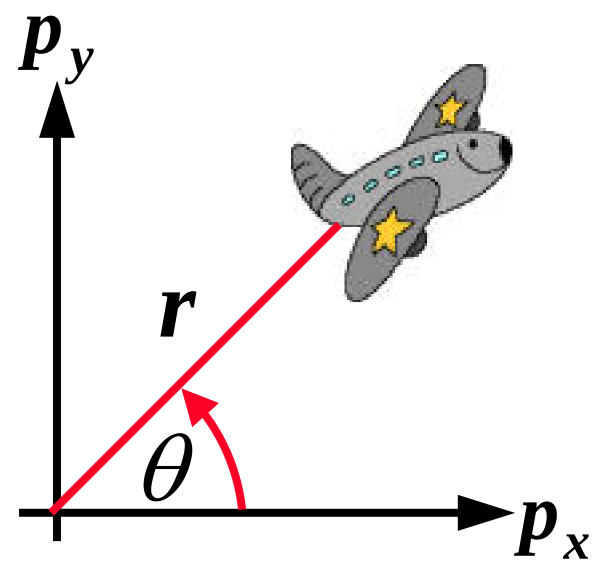
Australia

The extended Kalman filter

- conventional Kalman filter applies to *linear* state estimation problems:
  - next states are *linear* combinations of current states and inputs
  - measurements are *linear* combinations of current states and inputs
- often interested in *nonlinear* problems
  - next states and/or measurements are *nonlinear* combinations of present states and inputs
- very commonly used approach is to *linearise* system model about current best estimate of the state
  - this is the extended Kalman filter (EKF)

Example—radar tracking

Natural choice of states are Cartesian coordinates (p_x, p_y) and derivatives (v_x, v_y)



For constant velocity, states evolve as follows:

$$\begin{bmatrix} p_x \\ v_x \\ p_y \\ v_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ v_x \\ p_y \\ v_y \end{bmatrix}_k$$

sampling period Δ

But most natural measurements are range and heading:

$$y = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{p_x^{\,2} + p_y^{\,2}} \\ \tan^{-1}\left(\dfrac{p_y}{p_x}\right) \end{bmatrix}$$

Linearisation:

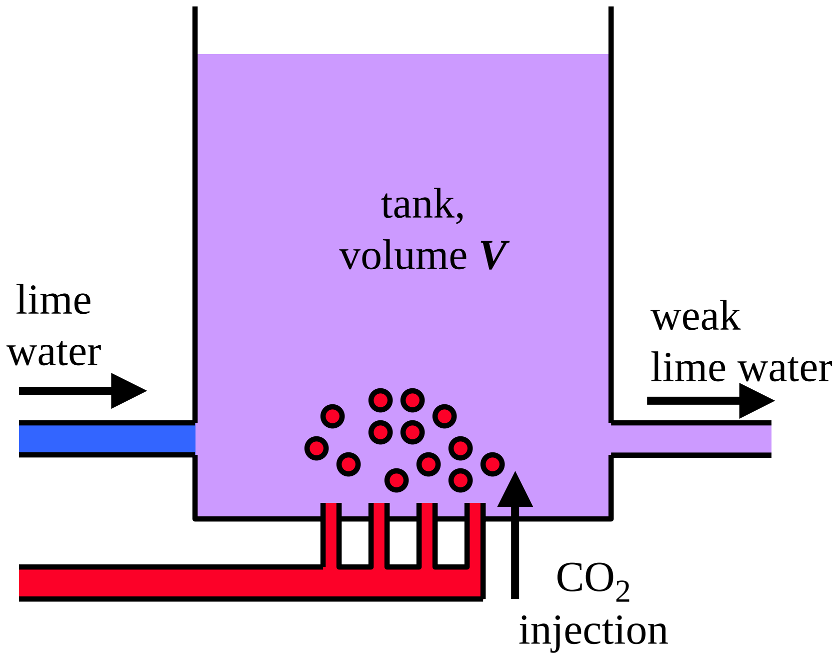
$$\frac{\partial y}{\partial x} = \begin{bmatrix} \dfrac{p_x}{r} & 0 & \dfrac{p_y}{r} & 0 \\ \dfrac{-p_y}{r^2} & 0 & \dfrac{p_x}{r^2} & 0 \end{bmatrix}$$

Example—decalcification plant

$$\dot{x}_1 = -\frac{c}{V}x_1x_2 + \frac{1}{V}u_1$$

$$\dot{x}_2 = -\frac{c}{V}x_1x_2 + \frac{1}{V}u_2$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix}$$



- state equation is nonlinear
- measurement equation is linear

$$Ca(OH)_2 + CO_2 \rightarrow CaCO_3 + H_2O$$

calcium hydroxide + carbon dioxide → calcium carbonate (precipitate) + water

x_1 = $Ca(OH)_2$ concentration in tank

x_2 = CO_2 concentration in tank

u_1 = $Ca(OH)_2$ inflow rate

u_2 = CO_2 inflow rate

rate of reaction $= c\, x_1 x_2$

Linearisation of nonlinear model

State-space model of discrete-time nonlinear system:

$$
\begin{aligned}
x_{k+1} &= f(x_k) + g(x_k)w_k \\
y_k &= h(x_k) + n_k
\end{aligned}
$$

with noise model:

$$
\begin{aligned}
E\{w_i w_j^T\} &= Q\delta_{ij}, \quad E\{w_i\} = 0 \\
E\{n_i n_j^T\} &= R\delta_{ij}, \quad E\{n_i\} = 0
\end{aligned}
$$

We'll actually use EKF to estimate states of *linearised* state-space system:

$$
\begin{aligned}
x_{k+1} &= A_k x_k + G_k w_k + u_k \\
y_k &= C_k x_k + n_k + \bar{y}_k
\end{aligned}
$$

additional terms due to linearisation

Note: linear system is *time-varying*, since A_k, G_k, C_k will depend on state estimates produced by extended Kalman filter

Linearisation

Key to linearisation is Taylor's Theorem:

$$f(x) = f(a) + (x-a)\,f'(a) + \frac{(x-a)^2}{2!} f''(a) + \cdots$$

- linearising *state equation* around estimate $\hat{x}_{k|k}$ and retaining only first two terms:

$$f(x_k) \approx f(\hat{x}_{k|k}) + A_k\left(x_k - \hat{x}_{k|k}\right)$$

where

$$A_k = \left.\frac{\partial f}{\partial x}\right|_{x = \hat{x}_{k|k}}$$

- also, define:

$$g(x_k) \approx g(\hat{x}_{k|k}) = G_k$$

- similarly for *measurement equation*:

$$h(x_k) \approx h(\hat{x}_{k|k-1}) + C_k\left(x_k - \hat{x}_{k|k-1}\right), \quad \text{where} \quad C_k = \left.\frac{\partial h}{\partial x}\right|_{x = \hat{x}_{k|k-1}}$$

The linearised model

State equation

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{w}_k \\
&\approx \boldsymbol{f}(\hat{\boldsymbol{x}}_{k|k}) + \boldsymbol{A}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}) + \boldsymbol{G}_k\boldsymbol{w}_k \\
&= \boldsymbol{A}_k\boldsymbol{x}_k + \underbrace{\left(\boldsymbol{f}(\hat{\boldsymbol{x}}_{k|k}) - \boldsymbol{A}_k\hat{\boldsymbol{x}}_{k|k}\right)}_{\boldsymbol{u}_k \text{ known at time } \boldsymbol{k}} + \boldsymbol{G}_k\boldsymbol{w}_k
\end{aligned}
$$

Measurement equation

$$
\begin{aligned}
\boldsymbol{y}_k &= \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{n}_k \\
&\approx \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}) + \boldsymbol{C}_k(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}) \\
&= \boldsymbol{C}_k\boldsymbol{x}_k + \underbrace{\left(\boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}) - \boldsymbol{C}_k\hat{\boldsymbol{x}}_{k|k-1}\right)}_{\bar{\boldsymbol{y}}_k \text{ known at time } \boldsymbol{k-1}} + \boldsymbol{n}_k
\end{aligned}
$$

Time and measurement updates

Time update

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} + \text{known term}$$

to predict with KF, simply propagate model assuming noise input is zero

$$= A_k \hat{x}_{k-1|k-1} + \left(f\left(\hat{x}_{k-1|k-1}\right) - A_k \hat{x}_{k-1|k-1} \right)$$

$$= f\left(\hat{x}_{k-1|k-1}\right)$$

Measurement update

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \bar{J}_k \left(y_k - \text{known term} - C_k \hat{x}_{k|k\text{-}1} \right)$$

$$= \hat{x}_{k|k-1} + \bar{J}_k \left(y_k - \left\{ h\left(\hat{x}_{k|k-1}\right) - C_k \hat{x}_{k|k-1} \right\} - C_k \hat{x}_{k|k\text{-}1} \right)$$

$$= \hat{x}_{k|k-1} + \bar{J}_k \left(y_k - h\left(\hat{x}_{k|k-1}\right) \right)$$

The EKF algorithm

Given:

$$x_{k+1} = f(x_k) + g(x_k)w_k \qquad E\{w_i w_j^T\} = Q\delta_{ij}, \quad E\{w_i\} = 0$$

$$y_k = h(x_k) + n_k \qquad E\{n_i n_j^T\} = R\delta_{ij}, \quad E\{n_i\} = 0$$

EKF algorithm:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}) \qquad \textit{time update}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \bar{J}_k\left(y_k - h(\hat{x}_{k|k-1})\right) \qquad \textit{measurement update}$$

$$\bar{J}_k = P_{k|k-1}C_k^T\left(C_k P_{k|k-1} C_k^T + R\right)^{-1} \qquad \textit{Kalman gain calculation}$$

$$P_{k+1|k} = A_k P_{k|k} A_k^T + G_k Q G_k^T \qquad \textit{time update}$$

$$P_{k|k} = \left(I - \bar{J}_k C_k\right)P_{k|k-1} \qquad \textit{measurement update}$$

where:

$$A_k = \left.\frac{\partial f}{\partial x}\right|_{x = \hat{x}_{k|k}} \qquad G_k = g(\hat{x}_{k|k}) \qquad C_k = \left.\frac{\partial h}{\partial x}\right|_{x = \hat{x}_{k|k-1}}$$

Iterated Extended Kalman Filter (IEKF)

Basic idea is to *relinearise* and *recompute* $\hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}}$

Replace measurement update

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}} = \hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}-1} + \bar{\boldsymbol{J}}_{\boldsymbol{k}}\left(\boldsymbol{y}_{\boldsymbol{k}} - \boldsymbol{h}(\hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}-1})\right)$$

with

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}}^{(\boldsymbol{i}+1)} = \hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}-1} + \bar{\boldsymbol{J}}_{\boldsymbol{k}}^{(\boldsymbol{i})}\left(\boldsymbol{y}_{\boldsymbol{k}} - \boldsymbol{h}(\hat{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}-1})\right)$$

- ❖ $\bar{\boldsymbol{J}}_{\boldsymbol{k}}^{(\boldsymbol{i})}$ is calculated based on linearisation at $\bar{\boldsymbol{x}}_{\boldsymbol{k}|\boldsymbol{k}}$
- ❖ iterate on $\boldsymbol{i}$ until convergence occurs, or fixed number of iterations completed

Details of IEKF

At each time instant $\boldsymbol{k}$, compute Kalman gain $\bar{\boldsymbol{J}}_{\boldsymbol{k}}$ iteratively as follows:

Initialisation:

$$\boldsymbol{C}_{\boldsymbol{k}}^{(0)} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{x = \hat{x}_{k|k-1}}$$

Steps 0 to L:

$$\bar{\boldsymbol{J}}_{\boldsymbol{k}}^{(j)} = \boldsymbol{P}_{k|k-1}\boldsymbol{C}_{\boldsymbol{k}}^{(j)T}\left[\boldsymbol{C}_{\boldsymbol{k}}^{(j)}\boldsymbol{P}_{k|k-1}\boldsymbol{C}_{\boldsymbol{k}}^{(j)T} + \boldsymbol{R}\right]^{-1}$$

$$\hat{\boldsymbol{x}}_{k|k}^{(j)} = \hat{\boldsymbol{x}}_{k|k} + \bar{\boldsymbol{J}}_{\boldsymbol{k}}^{(j)}\left(\boldsymbol{y} - \boldsymbol{h}\left(\hat{\boldsymbol{x}}_{k|k-1}\right)\right)$$

$$\boldsymbol{C}_{\boldsymbol{k}}^{(j)} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{x = \hat{x}_{k|k}^{(j)}}$$

After L steps:

$$\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{x}_{k|k}^{(L)}, \qquad \hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{f}\left(\hat{\boldsymbol{x}}_{k|k}\right)$$

$$\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \bar{\boldsymbol{J}}_{\boldsymbol{k}}\boldsymbol{C}_{\boldsymbol{k}}^{(L)}\right)\boldsymbol{P}_{k|k-1}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{A}_{\boldsymbol{k}}\boldsymbol{P}_{k|k}\boldsymbol{A}_{\boldsymbol{k}}^{\boldsymbol{T}} + \boldsymbol{G}_{\boldsymbol{k}}\boldsymbol{Q}\boldsymbol{G}_{\boldsymbol{k}}^{\boldsymbol{T}}$$

Performance guarantees for EKF

There are none.

❖ Performance of EKF is clearly dependent on quality of approximations:

$$\begin{aligned} \boldsymbol{x}_{k+1} &\approx \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{u}_k + \boldsymbol{G}_k \boldsymbol{w}_k \\ \boldsymbol{y}_k &\approx \boldsymbol{C}_k \boldsymbol{x}_k + \bar{\boldsymbol{y}}_k + \boldsymbol{n}_k \end{aligned}$$

❖ Linearisation can cause unexpected difficulties:

- bias errors
- divergence

❖ Pragmatic steps to address effects of linearisation errors:

- assume more process noise than there really is: increase $\boldsymbol{Q}$
- scale up covariance matrix $\boldsymbol{P}$
- use numerically robust formulations of KF update equations to keep $\boldsymbol{P}$ positive definite