Extended Kalman Filter (EKF)

Sensor Fusion

Fredrik Gustafsson
fredrik.gustafsson@liu.se

Gustaf Hendeby
gustaf.hendeby@liu.se

Linköping University

The Kalman Filter

The Kalman filter is the exact solution to the Bayesian filtering recursion for linear Gaussian model

$$\begin{aligned} x_{k+1} &= F_k x_k + G_k v_k, & v_k &\sim \mathcal{N}(0, Q_k) \\ y_k &= H_k x_k + e_k, & e_k &\sim \mathcal{N}(0, R_k). \end{aligned}$$

Kalman Filter Algorithm

Time update:

$$\begin{aligned} \hat{x}_{k+1|k} &= F_k \hat{x}_{k|k} \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + G_k Q_k G_k^T \end{aligned}$$

Meas. update:

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_k) \\ P_{k|k} &= P_{k|k-1} - K_k P_{k|k-1} \\ \hat{y}_k &= H_k \hat{x}_{k|k-1} & \varepsilon_k &= y_k - \hat{y}_k \\ K_k &= P_{k|k-1} H_k^T S^{-1} & S_k &= H_k P_{k|k-1} H_k^T + R_k \end{aligned}$$

Extending to Nonlinear Models

Many phenomena in nature are not linear, especially measurements. Hence, filters to handle more general nonlinear models are needed.

Nonlinear model

Consider the nonlinear model:

$$\begin{aligned} x_{k+1} &= f(x_k, v_k), & \operatorname{Cov}(v_k) &= Q_k \\ y_k &= h(x_k) + e_k, & \operatorname{Cov}(e_k) &= R_k, \end{aligned}$$

assuming $\mathrm{E}(v_k) = 0$, $\mathrm{E}(e_k) = 0$, and mutual independence.

The *extended Kalman filter* (EKF) approximates the model with a linear one using a Taylor series expansion, before applying the regular Kalman filter.

Classic Derivation of EKF (1/2)

For simplicity, assume the simpler nonlinear model:

$$\begin{aligned} x_{k+1} &= f(x_k) + v_k \\ y_k &= h(x_k) + e_k \end{aligned}$$

To derive the **time update**:

$$f(x_{k+1}) \approx f(\hat{x}_{k|k}) + \underbrace{\nabla_x f(\hat{x}_{k|k})}_{F_k}(x_k - \hat{x}_{k|k})$$

The approximate linear model becomes:

$$x_{k+1} = \underbrace{f(\hat{x}_{k|k}) - F_k\hat{x}_{k|k}}_{\text{Known input}} + F_k x_k + v_k.$$

The filter update follows as:

$$\begin{aligned} \hat{x}_{k+1|k} &= F_k\hat{x}_{k|k} + f(\hat{x}_{k|k}) - F_k\hat{x}_{k|k} = f(\hat{x}_{k|k}) \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k. \end{aligned}$$

Classic Derivation of EKF (2/2)

To derive the **measurement update**:

$$h(x_k) \approx h(\hat{x}_{k|k-1}) + \underbrace{\nabla_x h(\hat{x}_{k|k-1})}_{H_k}(x_k - \hat{x}_{k|k-1})$$

The approximate linear model becomes

$$y_k = \underbrace{h(\hat{x}_{k|k-1}) - H_k \hat{x}_{k|k-1}}_{\text{Known input}} + H_k x_k + e_k$$

The measurement update follows as

$$\begin{aligned}
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \big(y_k - (H_k \hat{x}_{k|k-1} + h(\hat{x}_{k|k-1}) - H_k \hat{x}_{k|k-1})\big) \\
&= \hat{x}_{k|k-1} + K_k \big(y_k - h(\hat{x}_{k|k-1})\big) \\
P_{k|k} &= (I - K_k H_k) P_{k|k-1} \\
&= (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T \quad \text{(Joseph's form)}
\end{aligned}$$

EKF1 and EKF2

- The classic extended Kalman filter (EKF) is derived as above using only the first order terms in the Taylor series expansion, *i.e.*, the TT1 NLT previously discussed.
- A TT2 EKF (EKF2) can be obtained similarly, by including the quadratic terms in the Taylor series expansion , *i.e.*, the TT2 NLT previously discussed.
- EKF2 hence compensates for quadratic effects in the model, which results in an additional term in both the mean and covariance expressions.

EKF1 Algorithm

Time update:

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, 0)$$

$$P_{k+1|k} = f'_x(\hat{x}_{k|k})P_{k|k}(f'_x(\hat{x}_{k|k}))^T + f'_v(\hat{x}_{k|k})Q_k(f'_v(\hat{x}_{k|k}))^T$$

Meas. update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - h(\hat{x}_{k|k-1}, 0)\right)$$

$$P_{k|k} = P_{k|k-1} - P_{k|k-1}(h'_x(\hat{x}_{k|k-1}))^T S_k^{-1} h'_x(\hat{x}_{k|k-1})P_{k|k-1}$$

$$S_k = h'_x(\hat{x}_{k|k-1})P_{k|k-1}(h'_x(\hat{x}_{k|k-1}))^T + h'_e(\hat{x}_{k|k-1})R_k(h'_e(\hat{x}_{k|k-1}))^T$$

$$K_k = P_{k|k-1}(h'_x(\hat{x}_{k|k-1}))^T S_k^{-1}$$

EKF1 and EKF2 Algorithm

Time update:

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, 0) + \tfrac{1}{2}\left[\operatorname{tr}(f''_{i,x} P_{k|k})\right]_i$$

$$P_{k+1|k} = f'_x(\hat{x}_{k|k}) P_{k|k} (f'_x(\hat{x}_{k|k}))^T + f'_v(\hat{x}_{k|k}) Q_k (f'_v(\hat{x}_{k|k}))^T + \tfrac{1}{2}\left[\operatorname{tr}(f''_{i,x}(\hat{x}_{k|k}) P_{k|k} f''_{j,x}(\hat{x}_{k|k}) P_{k|k})\right]_{ij}$$

Meas. update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\big(y_k - h(\hat{x}_{k|k-1}, 0) - \tfrac{1}{2}\left[\operatorname{tr}(h''_{i,x} P_{k|k-1})\right]_i\big)$$

$$P_{k|k} = P_{k|k-1} - P_{k|k-1}(h'_x(\hat{x}_{k|k-1}))^T S_k^{-1} h'_x(\hat{x}_{k|k-1}) P_{k|k-1} + \tfrac{1}{2}\left[\operatorname{tr}(h''_{i,x}(\hat{x}_{k|k-1}) P_{k|k-1} h''_{j,x}(\hat{x}_{k|k-1}) P_{k|k-1})\right]_{ij}$$

$$S_k = h'_x(\hat{x}_{k|k-1}) P_{k|k-1} (h'_x(\hat{x}_{k|k-1}))^T + h'_e(\hat{x}_{k|k-1}) R_k (h'_e(\hat{x}_{k|k-1}))^T + \tfrac{1}{2}\left[\operatorname{tr}(h''_{i,x}(\hat{x}_{k|k-1}) P_{k|k-1} h''_{j,x}(\hat{x}_{k|k-1}) P_{k|k-1})\right]_{ij}$$

$$K_k = P_{k|k-1}(h'_x(\hat{x}_{k|k-1}))^T S_k^{-1}$$

NB!

This form of the EKF2 (as given in the book) is disregarding second order terms of the process noise! See, e.g., my thesis for the full expressions.

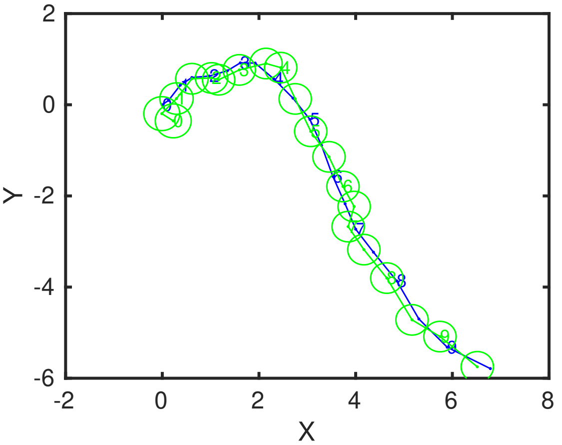
Comments

- The EKF1, using the TT1 transformation, is obtained by letting both Hessians f''_x and h''_x be zero.
- Analytic Jacobian and Hessian needed. If not available, use numerical approximations (done in Signal and Systems Lab by default!)
- The complexity of EKF1 is as in KF n_x^3 due to the FPF^T operation.
- The complexity of EKF2 is n_x^5 due to the $F_i P F_j^T$ operation for $i, j = 1, \dots, n_x$.
- Dithering is good! That is, increase Q and R from the simulated values to account for the approximation errors.

Simulation Example (1/2)

Create a constant velocity model, simulate and Kalman filter.

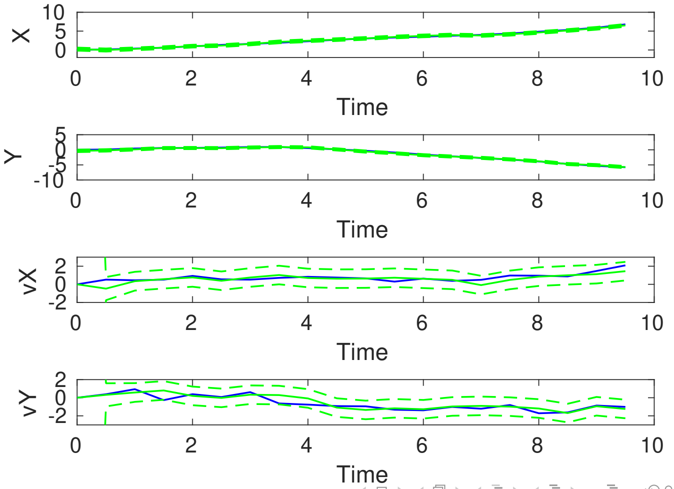
```
T = 0.5;
F = [1 0 T 0; 0 1 0 T; 0 0 1 0; 0 0 0 1];
G = [T^2/2 0; 0 T^2/2; T 0; 0 T];
H = [1 0 0 0; 0 1 0 0];
R = 0.03*eye(2);
m = lss(F, [], H,[], G*G', R, 1/T);
m.xlabel = {'X', 'Y', 'vX', 'vY'};
m.ylabel = {'X', 'Y'};
m.name = 'Constant velocity motion model';
z = simulate(m, 20);
m = nl(m);  % EKF only exist for nl models
xhat1 = ekf(m, z, 'k', 0); % Time-varying
xplot2(z, xhat1, 'conf', 90, [1 2]);
```



Simulation Example (2/2)

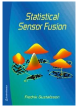
Covariance illustrated as confidence ellipsoids in 2D plots or confidence bands in 1D plots.

```
xplot(z, xhat1, 'conf', 99)
```



Summary

- The extended Kalman filter (EKF) is an extension of the Kalman filter to handle nonlinear models.
- The filter can be derived by first linearizing the model and then applying the normal Kalman filter.
- The EKF can also be derived in the more general NLT framework, similar to the UKF, using TT1 or TT2.
- The EKF loses all optimality properties of the Kalman filter, but does in practice often work very well.



Chapter 8 (EKF related parts)