Kalman Filter

Sensor Fusion

Fredrik Gustafsson
fredrik.gustafsson@liu.se

Gustaf Hendeby
gustaf.hendeby@liu.se

Linköping University

Linear Models and Bayesian Filter Recursion

Time-varying linear state-space model

$$x_{k+1} = F_k x_k + G_k v_k, \qquad \mathrm{Cov}(v_k) = Q_k$$
$$y_k = H_k x_k + e_k, \qquad \mathrm{Cov}(e_k) = R_k,$$

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

Bayesian filter recursion

$$p(x_{k+1}|y_{1:k}) = \int_{x_k} p(x_{k+1}|x_k)p(x_k|y_{1:k})\,dx_k \qquad \text{(TU)}$$

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \qquad \text{(MU)}$$

Time Update

Assume $\mathrm{E}(x_k|y_{1:k}) = \hat{x}_{k|k}$ and $\mathrm{Cov}(x_k|y_{1:k}) = P_{k|k}$, and compute the predictive mean and covariance:

$$\begin{aligned}
\hat{x}_{k+1|k} &= \mathrm{E}(F_k x_k + G_k v_k | y_{1:k}) \\
&= F_k \hat{x}_{k|k} + G_k 0 \\
&= F_k \hat{x}_{k|k} \\
P_{k+1|k} &= \mathrm{Cov}(F_k x_k + G_k v_k | y_{1:k}) \\
&= \mathrm{Cov}(F_k x_k | y_{1:k}) + \mathrm{Cov}(G_{k-1} v_{k-1} | y_{1:k}) \\
&= F_k P_{k|k} F_k^T + G_k Q_k G_k^T
\end{aligned}$$

Conditional Gaussian Distribution

Lemma 7.1

If X and Y are two jointly distributed Gaussian stochastic variables according to

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} P_{XX} & P_{XY} \\ P_{YX} & P_{YY} \end{pmatrix}\right),$$

then the conditional distribution of X, given the observed value of $Y = y$, is Gaussian distributed according to

$$(X|Y = y) \sim \mathcal{N}(\mu_X + P_{XY}P_{YY}^{-1}(y - \mu_Y), P_{XX} - P_{XY}P_{YY}^{-1}P_{YX}).$$

Measurement Update (1/2)

Assume $\mathrm{E}(x_k|y_{1:k-1}) = \hat{x}_{k|k-1}$ and $\mathrm{Cov}(x_k|y_{1:k-1}) = P_{k|k-1}$, and compute the mean and covariance conditioned on the new measurement y_k.

First note,

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x_k \\ H_k x_k + e_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_{k|k-1} \\ H\hat{x}_{k|k-1} \end{pmatrix}, \begin{pmatrix} P_{k|k-1} & P_{k|k-1} H_k^T \\ H_k P_{k|k-1} & H_k P_{k|k-1} H_k + R_k \end{pmatrix}\right).$$

Next, apply Lemma 7.1, which yields

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + P_{k|k-1} H^T (H P_{k|k-1} H^T + R_k)^{-1} (y_k - H\hat{x}_{k|k-1})$$

$$P_{k|k} = P_{k|k-1} - P_{k|k-1} H^T (H P_{k|k-1} H^T + R_k)^{-1} H P_{k|k-1}$$

Measurement Update (2/2)

The Kalman filter measurement update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + P_{k|k-1}H^T(HP_{k|k-1}H^T + R_k)^{-1}(y_k - H\hat{x}_{k|k-1})$$
$$P_{k|k} = P_{k|k-1} - P_{k|k-1}H^T(HP_{k|k-1}H^T + R_k)^{-1}HP_{k|k-1}$$

To simplify, introduce variables to highlight the structure

Measurement Update (2/2)

The Kalman filter measurement update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + P_{k|k-1}H^T(HP_{k|k-1}H^T + R_k)^{-1}(y_k - \hat{y}_k)$$
$$P_{k|k} = P_{k|k-1} - P_{k|k-1}H^T(HP_{k|k-1}H^T + R_k)^{-1}HP_{k|k-1}$$

To simplify, introduce variables to highlight the structure

$$\hat{y}_k = H_k\hat{x}_{k|k-1}$$ Predicted measurement.

Measurement Update (2/2)

The Kalman filter measurement update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + P_{k|k-1} H^T (H P_{k|k-1} H^T + R_k)^{-1} \varepsilon_k$$

$$P_{k|k} = P_{k|k-1} - P_{k|k-1} H^T (H P_{k|k-1} H^T + R_k)^{-1} H P_{k|k-1}$$

To simplify, introduce variables to highlight the structure

$\hat{y}_k = H_k \hat{x}_{k|k-1}$ Predicted measurement.

$\varepsilon_k = y_k - \hat{y}_k$ The innovation.

Measurement Update (2/2)

The Kalman filter measurement update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + P_{k|k-1} H^T S_k^{-1} \varepsilon_k$$
$$P_{k|k} = P_{k|k-1} - P_{k|k-1} H^T S_k^{-1} H P_{k|k-1}$$

To simplify, introduce variables to highlight the structure

$\hat{y}_k = H_k \hat{x}_{k|k-1}$	Predicted measurement.
$\varepsilon_k = y_k - \hat{y}_k$	The innovation.
$S_k = H P_{k|k-1} H^T + R_k$	The covariance of the innovation.

Measurement Update (2/2)

The Kalman filter measurement update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \varepsilon_k$$
$$P_{k|k} = P_{k|k-1} - K_k H P_{k|k-1}$$

To simplify, introduce variables to highlight the structure

$\hat{y}_k = H_k \hat{x}_{k|k-1}$	Predicted measurement.
$\varepsilon_k = y_k - \hat{y}_k$	The innovation.
$S_k = H P_{k|k-1} H^T + R_k$	The covariance of the innovation.
$K_k = P_{k|k-1} H_k^T S_k^{-1}$	The Kalman gain.

Measurement Update (2/2)

The Kalman filter measurement update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \varepsilon_k$$
$$P_{k|k} = P_{k|k-1} - K_k H P_{k|k-1}$$

To simplify, introduce variables to highlight the structure

$\hat{y}_k = H_k \hat{x}_{k|k-1}$	Predicted measurement.
$\varepsilon_k = y_k - \hat{y}_k$	The innovation.
$S_k = H P_{k|k-1} H^T + R_k$	The covariance of the innovation.
$K_k = P_{k|k-1} H_k^T S_k^{-1}$	The Kalman gain.

Overview

The Kalman filter:

- the measurements only affect $\hat{x}$ not P, which can be precomputed;
- is a *best linear unbiased estimator* (BLUE);
- is the exact solution to the Bayesian recursion for linear Gaussian models;
- can equivalently be formulated on information form propagating $\iota = P^{-1}\hat{x}$ and $\mathcal{I} = P^{-1}$; and
- has been extended to handle nonlinear problems, e.g., *extened Kalman filter* (EKF), *unscented Kalman filter* (UKF).

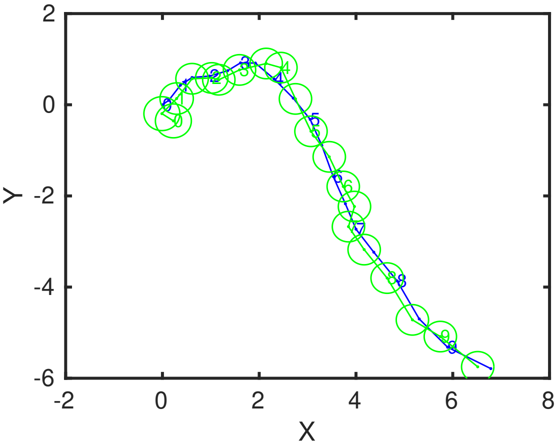
Kalman Filter Tuning

- The **SNR ratio** $\|Q\|/\|R\|$ is the most crucial, it sets the filter speeds. Note the difference between real system and model used in the KF.
- **Recommendation:** fix R according to sensor specification or measured performance, and tune Q.
  (Motion models are anyway subjective approximations of reality).
- **Tune covariances in large steps** (order of magnitudes)!
- **High SNR** in the model, gives a **fast filter** that is quick to adapt to changes/maneuvers, but with larger uncertainty (small bias, large variance).
- **Low SNR** in the model, gives a **slow filter** that is slow to adapt to changes/maneuvers, but with small uncertainty (large bias, small variance).
- P_0 **reflects the belief in the prior** $x_0 \sim \mathcal{N}(\hat{x}_0, P_0)$. Possible to choose P_0 very large (and $\hat{x}_0$ arbitrary), if no prior information exists.

Simulation Example (1/2)

Create a constant velocity model, simulate and Kalman filter.

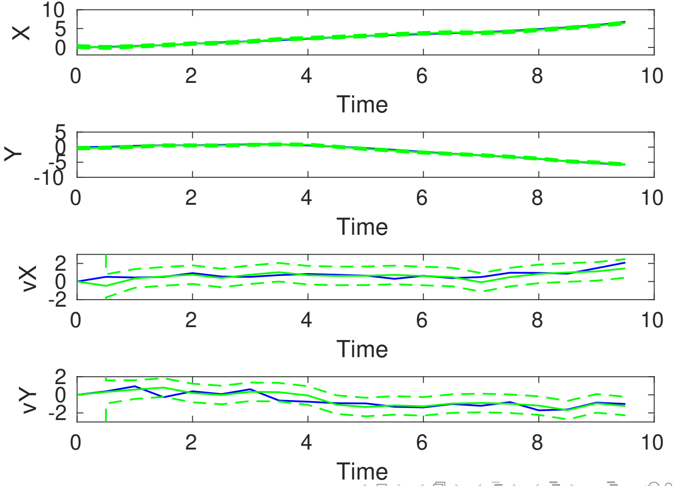
```
T = 0.5;
F = [1 0 T 0; 0 1 0 T; 0 0 1 0; 0 0 0 1];
G = [T^2/2 0; 0 T^2/2; T 0; 0 T];
H = [1 0 0 0; 0 1 0 0];
R = 0.03*eye(2);
m = lss(F, [], H,[], G*G', R, 1/T);
m.xlabel = {'X', 'Y', 'vX', 'vY'};
m.ylabel = {'X', 'Y'};
m.name = 'Constant␣velocity␣motion␣model';
z = simulate(m, 20);
xhat1 = kalman(m, z, 'alg', 2, '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 Kalman filter is the exact solution to the Bayesian filtering recursion for linear Gaussian model

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

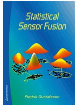
Kalman Filter Algorithm

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:

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



Section 7–7.1. Section 7.1.3 (Lemma 7.1), treated separately.