Unscented Kalman Filter (UKF)

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

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

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

Kalman Filter Algorithm

Time update:

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k}$$

$$P_{k+1|k} = F_k P_{k|k} F_k^T + G_k Q_k G_k^T$$

Meas. update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_k)$$

$$P_{k|k} = P_{k|k-1} - K_k P_{k|k-1}$$

$$\hat{y}_k = H_k \hat{x}_{k|k-1} \qquad \varepsilon_k = y_k - \hat{y}_k$$

$$K_k = P_{k|k-1} H_k^T S^{-1} \qquad S_k = H_k P_{k|k-1} H_k^T + R_k$$

Nonlinear Model

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:

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

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

Nonlinear Transformation (NLT) (of a stochastic variable)

In many cases it is important to perform nonlinear transformations of stochastic variables, e.g., for estimation of parameters with nonlinear measurement models.

Problem formulation: nonlinear transfomation (NLT)

Given the transform

$$z = g(u)$$

and the mean and covariance of the input,

$$\mathrm{E}(u) = \mu_u, \quad \mathrm{Cov}(u) = P_u \quad \text{(often approximated } u \sim \mathcal{N}(\mu_u, P_u))$$

determine

$$\mathrm{E}(z) = \mu_z \quad \mathrm{Cov}(z) = P_z \quad \text{(often approximated } z \sim \mathcal{N}(\mu_z, P_z)).$$

General Approximate KF: time update

Let

$$\bar{x} = \begin{pmatrix} x_k | y_{1:k} \\ v_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_{k|k} \\ 0 \end{pmatrix}, \begin{pmatrix} P_{k|k} & 0 \\ 0 & Q_k \end{pmatrix} \right)$$

$$z = x_{k+1} = f(x_k | y_{1:k}, v_k) = g(\bar{x}).$$

Any NLT approximation (UT, MCT, TT1, or TT2) gives

$$(x_{k+1} | y_{1:k}) = z \sim \mathcal{N}(\hat{x}_{k+1|k}, P_{k+1|k}).$$

Conditional Gaussian Distribution

Lemma 7.1 (Conditional Gaussian Distribution)

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

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

The Kalman filter can be derived using NLT and Lemma 7.1, which offers a natural extension to nonlinear models, e.g., using the unscented transform (UT) for the NLT approximation.

General Approximate KF: measurement update

Let

$$\bar{x} = \begin{pmatrix} x_k | y_{1:k-1} \\ e_k \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_{k|k-1} \\ 0 \end{pmatrix}, \begin{pmatrix} P_{k|k-1} & 0 \\ 0 & R_k \end{pmatrix} \right)$$

$$z = \begin{pmatrix} x_k | y_{1:k-1} \\ y_k \end{pmatrix} = \begin{pmatrix} x_k | y_{1:k-1} \\ h(x_k | y_{1:k-1}, u_k, e_k) \end{pmatrix} = g(\bar{x}).$$

The transformation approximation (UT, MC, TT1, TT2) gives

$$z \sim \mathcal{N}\left(\begin{pmatrix} \hat{x}_{k|k-1} \\ \hat{y}_{k|k-1} \end{pmatrix}, \begin{pmatrix} P^{xx}_{k|k-1} & P^{xy}_{k|k-1} \\ P^{yx}_{k|k-1} & P^{yy}_{k|k-1} \end{pmatrix} \right).$$

The measurement update is now becomes (direct application of Lemma 7.1):

$$(x_k | y_{1:k}) \sim \mathcal{N}(\hat{x}_{k|k}, P_{k|k})$$

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

$$P_{k|k} = P_{k|k-1} - K_k P^{yx}_{k|k-1},$$

$$K_k = P^{xy}_{k|k-1} \left(P^{yy}_{k|k-1} \right)^{-1}.$$

Unscented Kalman Filter (UKF) Algorithm (1/2)

UKF: time update

Generate sigma points according to:

$$\begin{pmatrix} x_{k|k}^{(0)} \\ w_k^{(0)} \end{pmatrix} = \begin{pmatrix} \hat{x}_{k|k} \\ 0 \end{pmatrix}, \qquad \begin{pmatrix} x_{k|k}^{(\pm i)} \\ w_k^{(\pm i)} \end{pmatrix} = \begin{pmatrix} \hat{x}_{k|k} \\ 0 \end{pmatrix} + \pm\sqrt{n+\lambda}\left[\begin{pmatrix} P_{k|k} & 0 \\ 0 & Q_k \end{pmatrix}\right]_{:,i}$$

$$\omega^{(0)} = \frac{\lambda}{n+\lambda}, \qquad \omega_c^{(\pm i)} = \omega^{(\pm i)}$$

$$\omega_c^{(0)} = \omega^{(0)} + (1 - \alpha^2 + \beta) \qquad \omega^{(\pm i)} = \frac{1}{2(n+\lambda)}.$$

Usually, $\lambda = \alpha^2(n+\kappa) - n$, $\alpha = 10^{-3}$, $\beta = 2$, $\kappa = 0$

Now the **updated mean and covariance** are given by

$$\hat{x}_{k+1|k} = \sum_i \omega_t^{(i)} x_{k+1|k}^{(i)}$$

$$P_{k|k-1} = \sum_{i=0}^{N} \omega_{c,t}^{(i)} \left(x_{k+1|k}^{(i)} - \hat{x}_{k+1|k}\right)\left(x_{k+1|k}^{(i)} - \hat{x}_{k+1|k}\right)^T$$

$$x_{k+1|k}^{(i)} = f(x_{k|k}^{(i)}, w_k^{(i)})$$

Unscented Kalman Filter Algorithm (2/2)

UKF: measurement update

Generate sigma points according to (weights and parameters as before):

$$\begin{pmatrix} x^{(0)}_{k|k-1} \\ e^{(0)}_k \end{pmatrix} = \begin{pmatrix} \hat{x}_{k|k-1} \\ 0 \end{pmatrix}, \qquad \begin{pmatrix} x^{(\pm i)}_{k|k-1} \\ e^{(\pm i)}_k \end{pmatrix} = \begin{pmatrix} \hat{x}_{k|k-1} \\ 0 \end{pmatrix} + \pm\sqrt{n+\lambda}\left[\begin{pmatrix} P_{k|k-1} & 0 \\ 0 & R_{k-1} \end{pmatrix}\right]_{:,i}$$

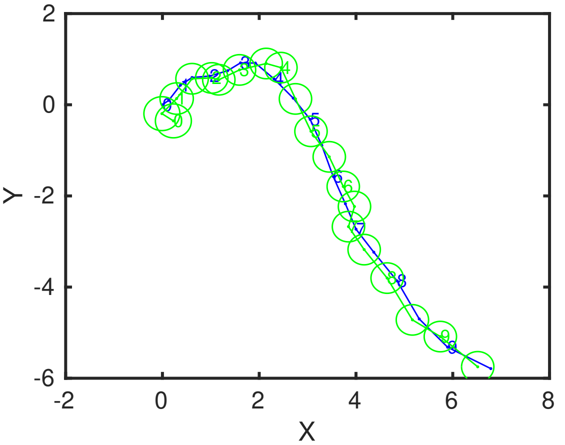
The **updated mean and covariance** are obtained as:

$$\begin{aligned}
\hat{x}_{t|t} &= \hat{x}_{t|t-1} + P^{xy}_{t|t-1} P^{-yy}_{t|t-1} (y_t - \hat{y}_t) \\
P_{t|t} &= P_{t|t-1} - P^{xy}_{t|t-1} P^{-yy}_{t|t-1} P^{xyT}_{t|t-1} \\
y^{(i)}_t &= h(x^{(i)}_{t|t-1}, e^{(i)}_t) \\
\hat{y}_t &= \sum_{i=0}^{N} \omega^{(i)}_t y^{(i)}_t \\
P^{yy}_{t|t-1} &= \sum_{i=0}^{N} \omega^{(i)}_{c,t} \left(y^{(i)}_t - \hat{y}_t\right)\left(y^{(i)}_t - \hat{y}_t\right)^T \\
P^{xy}_{t|t-1} &= \sum_{i=0}^{N} \omega^{(i)}_{c,t} \left(x^{(i)}_{t|t-1} - \hat{x}_{t|t-1}\right)\left(y^{(i)}_t - \hat{y}_t\right)^T.
\end{aligned}$$

Simulation Example (1/2)

Create a constant velocity model, simulate and Kalman filter.

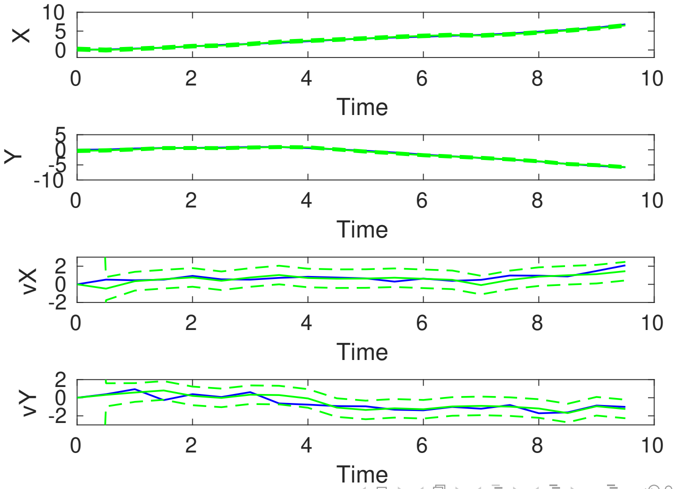
```
T = 0.5;
F = [1 0 T 0; 0 1 0 T; 0 0 1 0; 0 0 0 1];
G = [T^2/2 0; 0 T^2/2; T 0; 0 T];
H = [1 0 0 0; 0 1 0 0];
R = 0.03*eye(2);
m = lss(F, [], H,[], G*G', R, 1/T);
m.xlabel = {'X', 'Y', 'vX', 'vY'};
m.ylabel = {'X', 'Y'};
m.name = 'Constant␣velocity␣motion␣model';
z = simulate(m, 20);
m = nl(m);  %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



Simulation Example (2/2)

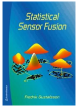
Covariance illustrated as confidence ellipsoids in 2D plots or confidence bands in 1D plots.

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



Summary

- Approximate Kalman filters for nonlinear problems can be derived using nonlinear transforms of stochastic variables.
  - Time update: An NLT is used to transform the current state and the process noise to get the time at the next time step.
  - Measurement update: An NLT is used to transform the current state and measurement noise to a get a joint distribution of the state and the measurement. Then Lemma 7.1 can be applied to get the estimate.
- The *unscented Kalman filter* (UKF) uses the *unscented transform* (UT) as NLT in the above scheme.
- No explicit derivatives (analytic or numerical) are required.
- Captures some, but not all, second order effects.



Chapter 8 (UKF related parts)