Universität Stuttgart

Prof.Dr.
Thomas Hobiger

Dynamic System Estimation

Nonlinear versions of the Kalman filter

9

Nonlinear versions of the Kalman filter

So far we have only dealt with linear systems and derived the Kalman filter algorithm to find an optimal estimate of $\hat{x}$. However, in many cases the state transition model and/or the observation model are nonlinear.

$$\begin{aligned} \boldsymbol{x}_n &= f(\boldsymbol{x}_{n-1}, \boldsymbol{s}_k) + \boldsymbol{u}_n \\ \boldsymbol{z}_n &= h(\boldsymbol{x}_n) + \boldsymbol{v}_n \end{aligned} \tag{9.1}$$

where $\boldsymbol{s}_k$ is a control vector, which is in many cases zero (cf. Eqs. (3.1) and (3.2)). If the functions f and h are differentiable we are able to linearize the system around $\hat{\boldsymbol{x}}_{n-1|n-1}$ by a first order Taylor approximation and denote the prediction step as

$$\boxed{\begin{aligned} \hat{\boldsymbol{x}}_{n|n-1} &= f(\hat{\boldsymbol{x}}_{n-1|n-1}, \boldsymbol{s}_k) \\ \boldsymbol{P}_{n|n-1} &= \boldsymbol{F}_k \boldsymbol{P}_{n-1|n-1} \boldsymbol{F}_n^T + \boldsymbol{Q}_n \quad \text{with } \boldsymbol{F}_n = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{n-1|n-1}, \boldsymbol{s}_k} \end{aligned}} \tag{9.2}$$

While the linearization helps to compute the first part of the covariance prediction, an exact analytic expression of the process noise covariance $\boldsymbol{Q}_n$ very difficult to obtain. Fortunately, in many cases linear state transition models are sufficient to model the dynamic characteristics of the physical system.

Nonlinear versions of the Kalman filter - cont'd

In the update step we can follow a similar strategy and carry out a first-order Taylor expansion of h. However, as we have already predicted to t_n we need to evaluate for $\hat{\boldsymbol{x}}_{n|n-1}$. Thus we get

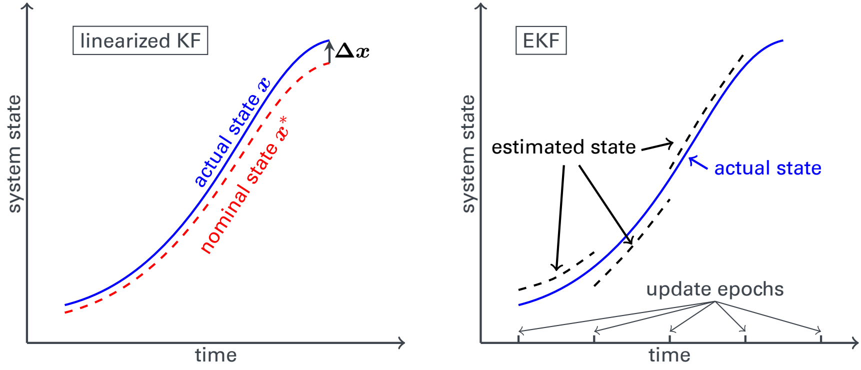
$$
\begin{aligned}
\boldsymbol{H}_n &= \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{n|n-1}} \\
\boldsymbol{K}_n &= \boldsymbol{P}_{n|n-1}\boldsymbol{H}_n^T\left(\boldsymbol{H}_n\boldsymbol{P}_{n|n-1}\boldsymbol{H}_n^T + \boldsymbol{R}_n\right)^{-1} \\
\boldsymbol{P}_{n|n} &= \left(\boldsymbol{I} - \boldsymbol{K}_n\boldsymbol{H}_n\right)\boldsymbol{P}_{n|n-1}
\end{aligned}
\tag{9.3}
$$

Equs. (9.2) and (9.3) together describe the so-called Extended Kalman filter (EKF). Please consider that in case of the EKF the Kalman gain $\boldsymbol{K}_n$ is only near-optimal for minimizing the a posteriori state covariance.

We will later find another way of dealing with non-linearities. One needs to know however, that the EKF is one of the most common used Kalman filter implementation in case of nonlinearities.

Nonlinear versions of the Kalman filter - cont'd

In general we have two ways to deal with nonlinearities: the EKF or the linearized KF. The latter differs from the EKF in the sense that the prediction a nominal state $\boldsymbol{x}^*$ is predicted and the filter only takes care of correction term $\boldsymbol{\Delta x}$ so that at the total state can be computed as $\boldsymbol{x} = \boldsymbol{x}^* + \boldsymbol{\Delta x}$. Another important consideration is the in the case of the linearized KF, the linearization happens around $\boldsymbol{x}^*$ whereas in the case of the EKF, the linearization is made around the actual (predicted/estimated) state.

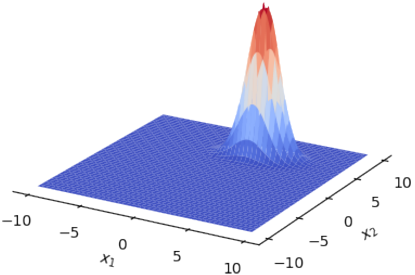


Nonlinear versions of the Kalman filter - cont'd

The EKF might be a suitable solution, but sometimes the functional relation might be highly non-linear. Thus, the first-order Taylor approximation can lead to filter deterioration and wrong estimates of the state vector. The deeper reason behind this is found in a implicit assumption which we have made when we tried to find an "optimal" estimator for our state. In lecture 7 we asked for an optimal state $\hat{\boldsymbol{x}}_{n|n}$ that minimizes $E\left(\left(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n}\right)^2\right)$. However, this requires that the underlying errors are following a multivariate Gaussian probability distribution of the form

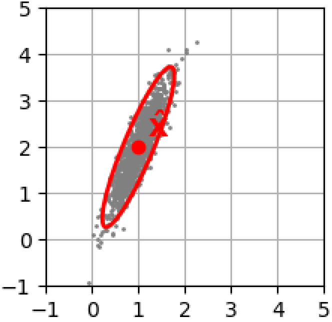
$$p(\boldsymbol{x}_n, \hat{\boldsymbol{x}}_{n|n}, \boldsymbol{P}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{P}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n})^T \boldsymbol{P}^{-1}(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_{n|n})\right) \quad (9.4)$$

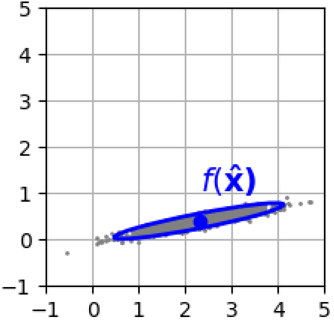
where is the $\boldsymbol{P}$ covariance of the state.



Nonlinear versions of the Kalman filter - cont'd

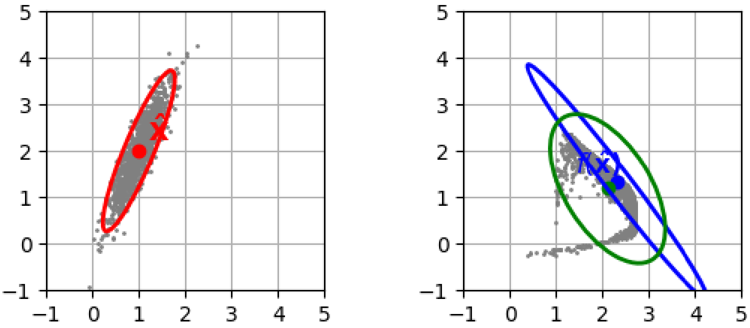
Example: The state $\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\boldsymbol{P} = \begin{bmatrix} 0.1 & 0.2 \\ 0.2 & 0.5 \end{bmatrix}$ undergoes a <u>linear</u> transformation $f(\boldsymbol{x}) = \boldsymbol{x}_f = \hat{\boldsymbol{A}}\boldsymbol{x}$ with $\boldsymbol{A} = \begin{bmatrix} 1.3 & 0.5 \\ -0.2 & 0.3 \end{bmatrix}$. Monte Carlo simulations and error propagation (error ellipses drawn at 95% confidence level) are fully consistent in this case.





Nonlinear versions of the Kalman filter - cont'd

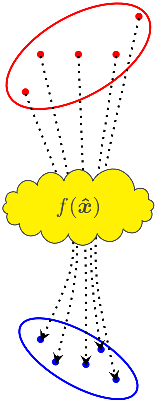
Assume now we apply <u>nonlinear</u> transformation $\boldsymbol{x}_f = \begin{bmatrix} \exp\sin x_1^2 \\ \cos x_1 \exp\sin x_2^2 \end{bmatrix}$ to the same state (and covariance) as before. Then both the transformed state as well as the covariance from error propagation (in blue) differ significantly from the empirically determined values for the mean and its covariance (in green).



Nonlinear versions of the Kalman filter - cont'd

As we have seen, the Monte Carlo approach would allow us to recover the state and its covariance empirically. However, we need a simpler idea which can be summarized as follows

1. Select a set of (a few) representative points (we will call them "sigma points" later)
2. Transform these points through the nonlinear function
3. Assign weights to the transformed points so that these values describe the "importance" of each point
4. Reconstruct the (mean of the) state and its covariance after the transformation, by using the transformed points and their weights



Nonlinear versions of the Kalman filter - cont'd

The "Unscented Transform" (UT) does exactly what we are looking for. Just follow the "cookbook" steps

1. Compute $2n+1$ "sigma points"

$$\boxed{\begin{aligned}\boldsymbol{\chi}^{[0]} &= \boldsymbol{x} \\ \boldsymbol{\chi}^{[i]} &= \boldsymbol{x} + \left(\sqrt{(n+\lambda)\boldsymbol{P}}\right)_i \quad \text{for } i = 1, \ldots, n \\ \boldsymbol{\chi}^{[i]} &= \boldsymbol{x} - \left(\sqrt{(n+\lambda)\boldsymbol{P}}\right)_i \quad \text{for } i = n+1, \ldots, 2n\end{aligned}} \tag{9.5}$$

2. Compute the weights for each of them

$$\boxed{\begin{aligned}& w_m^{[0]} = \tfrac{\lambda}{n+\lambda},\ w_m^{[i]} = w_c^{[i]} = \tfrac{1}{2(n+\lambda)} \\ & w_c^{[0]} = w_m[0] + (1 - \alpha^2 + \beta)\end{aligned}} \tag{9.6}$$

3. Reconstruct the state after and the covariance after the transformation

$$\boxed{\boldsymbol{x}_f \approx \sum_{i=0}^{2n} w_m^{[i]} f(\boldsymbol{\chi}^{[i]}),\ \boldsymbol{P}_f \approx \sum_{i=0}^{2n} w_c^{[i]} \left[f(\boldsymbol{\chi}^{[i]}) - \boldsymbol{x}_f\right] \left[f(\boldsymbol{\chi}^{[i]}) - \boldsymbol{x}_f)\right]^T} \tag{9.7}$$

Nonlinear versions of the Kalman filter - cont'd

Note 1: In Eq. (9.5) the matrix square root $\sqrt{\boldsymbol{P}}$ is required. Computation can be done either by

I. Singular value decomposition (SVD)

$$\boxed{\boldsymbol{P} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^{-1} \ \rightarrow \ \sqrt{\boldsymbol{P}} = \boldsymbol{V}\boldsymbol{D}^{1/2}\boldsymbol{V}^{-1}} \tag{9.8}$$

or by

II. Cholesky decomposition

$$\boxed{\boldsymbol{P} = \boldsymbol{L}\boldsymbol{L}^{T} \ \rightarrow \ \boldsymbol{L} = \sqrt{\boldsymbol{P}}} \tag{9.9}$$

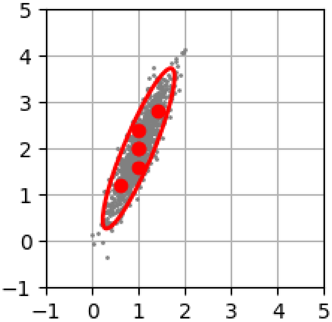
Note 2: While for Gaussians a value $\beta = 2$ has to be used, λ is defined as

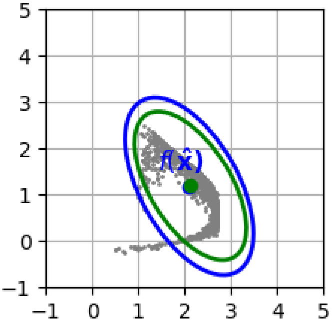
$$\boxed{\lambda = \alpha^2(n+\kappa) - n \ \text{ with } \ \kappa \geq 0 \ \text{ and } \ \alpha \in (0,1]} \tag{9.10}$$

The parameters α and κ determine how far the sigma points are away from $\boldsymbol{x}$ and thus allow for some adaption to certain problems. For most applications values of $\alpha = 0.001$ and $\kappa = 0$ are suitable though.

Nonlinear versions of the Kalman filter - cont'd

If apply the UT for for the example from before we see that the state and its covariance (in blue) are now very close to the empirically determined values (in green) for the mean and its covariance (Note: as this example had very high nonlinearities, values of $\alpha = 0.85$ and $\kappa = 0$ had to be used).





Nonlinear versions of the Kalman filter - cont'd

If we do not linearize, as in the case of the EKF, but use the Unscented Transform, we obtain the so called Unscented Kalman filter, UFK, which has the following steps.

1. If the state transition is nonlinear, we can then use the UT and obtain the predicted state and its covariance. Thus, we need to compute sigma points $\chi^{[i]}$ based on the state $\hat{\boldsymbol{x}}_{n-1|n-1}$ and its covariance $\boldsymbol{P}_{n-1|n-1}$, which is quite straightforward.

$$\boxed{\begin{aligned} \hat{\boldsymbol{x}}_{n|n-1} &= \sum_{i=0}^{2n} w_m^{[i]} f(\boldsymbol{\chi}^{[i]}), \\ \boldsymbol{P}_{n|n-1} &= \sum_{i=0}^{2n} w_c^{[i]} \left[f(\boldsymbol{\chi}^{[i]}) - \hat{\boldsymbol{x}}_{n|n-1}\right] \left[f(\boldsymbol{\chi}^{[i]}) - \hat{\boldsymbol{x}}_{n|n-1}\right]^T + \boldsymbol{Q} \end{aligned}} \tag{9.11}$$

Nonlinear versions of the Kalman filter - cont'd

2. If we have a nonlinear relation between the measurements and our state we can also use the UT here. However, we need to study the expression of the Kalman gain.

$$\boldsymbol{K}_n = \underbrace{\boldsymbol{P}_{n|n-1}\boldsymbol{H}_n^T}_{\boldsymbol{P}_{n|n-1}^{x,z}} \left(\underbrace{\boldsymbol{H}_n \boldsymbol{P}_{n|n-1} \boldsymbol{H}_n^T + \boldsymbol{R}_n}_{\boldsymbol{S}_n} \right)^{-1}$$

$\boldsymbol{P}_{n|n-1}^{x,z}$ is the covariance between the state and the observations. This is usually a square matrix, but can be reconstructed empirically like any other covariance matrix.

$\boldsymbol{S}_n$ is called innovation matrix and in a very similar way as in equ. (9.11) contains the sum of the error propagated covariance plus the contribution of the observation noise.

Nonlinear versions of the Kalman filter - cont'd

If we compute sigma points $\chi^{[i]}$ based on the predicted state $\hat{x}_{n|n-1}$ and its covariance $P_{n|n-1}$ we can update our state as follows

$$
\begin{aligned}
z_n^* &= \sum_{i=0}^{2n} w_m^{[i]} h(\chi'^{[i]}), \\
S_n &= \sum_{i=0}^{2n} w_c^{[i]} \left[h(\chi'^{[i]}) - z_n^*\right] \left[h(\chi'^{[i]}) - z_n^*\right]^T + R \\
P_{n|n-1}^{x,z} &= \sum_{i=0}^{2n} w_c^{[i]} \left[f(\chi^{[i]}) - \hat{x}_{n|n-1}\right] \left[h(\chi'^{[i]}) - z_n^*\right]^T \\
K_n &= P_{n|n-1}^{x,z} S_n^{-1} \\
\hat{x}_{n|n} &= \hat{x}_{n|n-1} + K_n \left(z_n - z_n^*\right) \\
P_{n|n} &= P_{n|n-1} - K_n S_n K_n^T
\end{aligned}
\tag{9.12}
$$

Nonlinear versions of the Kalman filter - cont'd

Can you explain the last term in equ. (9.12) ?

You will find examples for EKF and UKF as Jupyter notebook under
`https://github.com/spacegeodesy/ParameterEstimationDynamicSystems/blob/master/example10.ipynb`