Lecture 6
The Kalman filter

- Discrete-time Kalman filter
- Continuous-time Kalman filter
- Kalman filter generalizations
- Nonlinear Kalman filtering

Background

- James Follin, A. G. Carlton, James Hanson, and Richard Bucy developed the continuous-time Kalman filter in unpublished work for the Johns Hopkins Applied Physics Lab in the late 1950s
- Rudolph Kalman independently developed the discrete-time Kalman filter in 1960
- In April 1960 Kalman and Bucy became aware of each other's work and collaborated on the publication of the continuous-time Kalman filter
- The filter is sometimes referred to as the Kalman-Bucy filter

About Kalman

Rudolf Emil Kálmán

- was born in Budapest in 1930
- earned his bachelor's degree in 1953 and his master's degree in 1954, both from the Massachusetts Institute of Technology, completed his doctorate in 1957 at Columbia University in New York City
- has been worked at the Research Institute for Advanced Studies in Baltimore, Maryland, Stanford University, University of Florida, Swiss Federal Institute of Technology in Zürich, Switzerland.
- died on the morning of July 2, 2016, at his home in Gainesville, Florida

About Kalman

Importance of Kalman filter

- is a mathematical technique widely used in the digital computers of control systems, navigation systems, avionics, and outerspace vehicles
- extract a signal from a long sequence of noisy or incomplete measurements, usually those done by electronic and gyroscopic systems.
- was initially used in vast skepticism, **the Apollo program**, and furthermore, in the NASA Space Shuttle, in Navy submarines, and in unmanned aerospace vehicles and weapons, such as cruise missiles

Approach to deriving the Kalman filter

The Kalman filter operates by propagating the mean and covariance of the state through time.

- start with a mathematical description of a dynamic system whose states we want to estimate
- implement equations that describe how the mean of the state and the covariance of the state propagate with time
- take the dynamic system that describes the propagation of the state mean and covariance, and implement the equations on a computer
- every time that we get a measurement, we update the mean and covariance of the state

Contents

- Discrete-time Kalman filter
- Continuous-time Kalman filter
- Kalman filter generalizations
- Nonlinear Kalman filtering

Problem formulation

Suppose we have a linear discrete-time system given as follows:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}$$
$$y_k = H_k x_k + v_k$$

The noise processes $\{w_k\}$ and $\{v_k\}$ are zero-mean, uncorrelated, and have known covariance matrices Q_k and R_k, respectively:

$$w_k \sim \mathcal{N}(0, Q_k), \qquad v_k \sim \mathcal{N}(0, R_k)$$
$$E[w_k w_j^T] = Q_k \delta_{k-j}, \quad E[v_k v_j^T] = R_k \delta_{k-j}$$
$$E[v_k w_j^T] = 0$$

Problem formulation

- our goal is to estimate the state x_k based on our knowledge of the system dynamics and the availability of the noisy measurement $\{y_k\}$
- The amount of information available to us for our state estimate varies depending on the particular problem that we are trying to solve

Different kinds of estimate

- if we have all of the measurements up to and including time k available for use in our estimate of x_k, then we can form an a posteriori estimate, $\hat{x}_k$. One way to formulate the *a posteriori* state estimate is

$$\hat{x}_k = E[x_k | y_1, y_2, \cdots, y_k] = \textit{ a posteriori} \text{ estimate}$$

- if we have all of the measurements up to but not including time k available for use in our estimate of x_k, then we can form an a priori estimate, $\check{x}_k$. One way to formulate the a priori state estimate is

$$\check{x}_k = E[x_k | y_1, y_2, \cdots, y_{k-1}] = \textit{ a priori} \text{ estimate}$$

Different kinds of estimate

- if we have measurements after time k available for use in our estimate of x_k, then we can form a smoothed estimate. One way to formulate the smoothed state estimate is

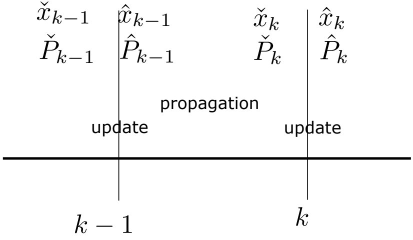
$$\hat{x}_{k|k+N} = E[x_k|y_1, y_2, \cdots, y_k, \cdots, y_{k+N}] = \text{smoothed estimate}$$

- if we want to find the best prediction of x_k more than one time step ahead of the available measurements, then we can form a predicted estimate. One way to form the predicted state estimate is to compute the expected value of x_k is:

$$\hat{x}_{k|k-M} = E[x_k|y_1, y_2, \cdots, y_{k-M}] = \text{predicted estimate}$$

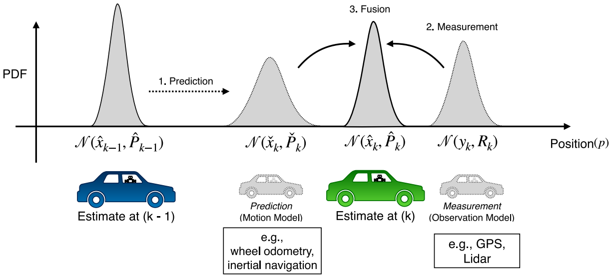
Notations

- $\hat{x}_0$–our initial estimate of x_0 before any measurements are available, in general $\hat{x}_0 = E(x_0)$
- $\check{P}_k$–the covariance of the estimation error of $\check{x}_k$, $\check{P}_k = E[(x_k - \check{x}_k)(x_k - \check{x}_k)^T]$
- $\hat{P}_k$–the covariance of the estimation error of $\hat{x}_k$, $\hat{P}_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$



- after we process the measurement at time $(k-1)$, we have an estimate of x_{k-1} (denoted as $\hat{x}_{k-1}$) and the covariance of that estimate (denoted as $\hat{P}_{k-1}$)
- when time k arrives, before we process the measurement at time k we compute an estimate of x_k (denoted as $\check{x}_k$) and the covariance of that estimate (denoted as $\check{P}_k$)
- we process the measurement at time k to refine our estimate of x_k, the resulting estimate of x_k is denoted as $\hat{x}_k$, and its covariance is denoted as $\hat{P}_k$

The Kalman Filter I Prediction and Correction



From $\hat{x}_{k-1}$ to $\check{x}_k$

- begin with $\hat{x}_0 = E(x_0)$
- we want to set $\check{x}_1 = E(x_1)$, as $\bar{x}_k = F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1}$, we therefore obtain

$$\check{x}_1 = F_0\hat{x}_0 + G_0u_0$$

- time update equation for x:

$$\check{x}_k = F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1}$$

  - we do not have any additional measurements available to help us update our state estimates after time $(k-1)$ and before time k
  - we should just update the state estimate based on our knowledge of the system dynamics

From $\hat{P}_{k-1}$ to $\check{P}_k$

- begin with $\hat{P}_0 = E[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T] = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$
- $\hat{P}_0$ represents the uncertainty in our initial estimate of x_0, so if we know the initial state perfectly, $\hat{P}_0 = 0$, if we have absolutely no idea of the value of x_0, then $\hat{P}_0 = \infty I$
- As $\check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1}$, we have

$$\check{P}_1 = F_0\hat{P}_0F_0^T + Q_0$$

- time-update equation for P (a more general equation):

$$\check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1}$$

Measurement-update equations for x and P

- The quantity $\check{x}_k$ is an estimate of x_k, and the quantity $\hat{x}_k$ is also an estimate of x_k. The only difference between $\check{x}_k$ and $\hat{x}_k$ is that $\hat{x}_k$ takes the measurement y_k into account.
- **Recursive least-square estimate** (the availability of the measurement y_k changes the estimate of a constant x)

$$
\begin{aligned}
K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\
&= P_kH_k^TR_k^{-1} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\
P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \\
&= (P_{k-1}^{-1} + H_k^TR_k^{-1}H_k)^{-1} \\
&= (I - K_kH_k)P_{k-1}
\end{aligned}
$$

Relationships between estimates and covariances in recursive least-square and Kalman filtering

RLS	KF
$\hat{x}_{k-1}$: estimate before y_k is processed	$\check{x}_k$: a priori estimate
P_{k-1}: covariance before y_k is processed	$\check{P}_k$: a priori covariance
$\hat{x}_k$: estimate after y_k is processed	$\hat{x}_k$: a posterior estimate
P_k: covariance after y_k is processed	$\hat{P}_k$: a posterior covariance

Generalization from the RLS formulas

$$\begin{aligned}
K_k &= \check{P}_k H_k^T (H_k \check{P}_k H_k^T + R_k)^{-1} \\
&= \hat{P}_k H_k^T R_k^{-1} \\
\hat{x}_k &= \check{x}_k + K_k (y_k - H_k \check{x}_k) \\
\hat{P}_k &= (I - K_k H_k) \check{P}_k (I - K_k H_k)^T + K_k R_k K_k^T \\
&= ((\check{P}_k)^{-1} + H_k^T R_k^{-1} H_k)^{-1} \\
&= (I - K_k H_k) \check{P}_k
\end{aligned}$$

The discrete-time Kalman filter

- The dynamic system is given by the following equations:

$$
\begin{aligned}
x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\
y_k &= H_k x_k + v_k \\
E(w_k w_j^T) &= Q_k \delta_{k-j} \\
E(v_k v_j^T) &= R_k \delta_{k-j} \\
E(w_k v_j^T) &= 0
\end{aligned}
$$

- The Kalman filter is initialized as follows:

$$\hat{x}_0 = E(x_0)$$

$$\hat{P}_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$$

The discrete-time Kalman filter

- The Kalman filter is given by the following equations, which are computed for each time step $k = 1, 2, \cdots$:

$$
\begin{aligned}
\check{P}_k &= F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1} \\
K_k &= \check{P}_k H_k^T (H_k \check{P}_k H_k^T + R_k)^{-1} \\
&= \hat{P}_k H_k^T R_k^{-1} \\
\check{x}_k &= F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1} \quad \text{a priori state estimate} \\
\hat{x}_k &= \check{x}_k + K_k(y_k - H_k\check{x}_k) \quad \text{a posteriori state estimate} \\
\hat{P}_k &= (I - K_kH_k)\check{P}_k(I - K_kH_k)^T + K_kR_kK_k^T \\
&= ((\check{P}_k)^{-1} + H_k^T R_k^{-1} H_k)^{-1} \\
&= (I - K_kH_k)\check{P}_k
\end{aligned}
$$

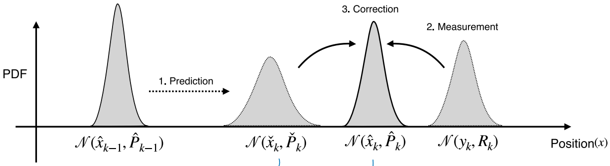
Remark

- the first expression for $\hat{P}_k$ is called the Joseph stabilized version of the covariance measurement update equation, which is more stable and robust than the third expression for $\hat{P}_k$
- the first expression for $\hat{P}_k$ guarantees that $\hat{P}_k$ will always be symmetric semi-positive definite, as long as $\check{P}_k$ is symmetric semi-positive definite
- the third expression for $\hat{P}_k$ is computationally simpler than the first expression, but its form does not guarantee symmetry or semi-positive definiteness for $\hat{P}_k$
- the second form for $\hat{P}_k$ is rarely implemented but will be useful in the derivation of the information filter

Remark

- if the second expression for K_k is used, then the second expression for $\hat{P}_k$ must be used
- if x_k is a constant (random variable unchanged), then $F_k = I, Q_k = 0$ and $u_k = 0$, and **the Kalman filter reduces to the recursive least squares algorithm for the estimation of a constant vector**
- the calculation of $\check{P}_k, K_k$, and $\hat{P}_k$ does not depend on the measurements y_k, but depends only on the system parameters F_k, H_k, Q_k and R_k
- the computational effort of calculating K_k can be saved during real-time operation by precomputing it
- the performance of the filter can be investigated and evaluated before the filter is actually run ($\hat{P}_k$ indicates the accuracy)

The Kalman Filter | Prediction & Correction



1. Prediction

$$\begin{cases} \check{\mathbf{x}}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \\ \check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \end{cases}$$

2,3. Measurement & Correction

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{H}_k^T(\mathbf{H}_k\check{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

$$\begin{cases} \hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\check{\mathbf{x}}_k) \\ \hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\check{\mathbf{P}}_k \end{cases}$$

Bayesian Inference

- Assume we have a joint Gaussian over a pair of variables (x, y)

$$p(x,y) = \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \right)$$

- According to Schur complement, we have

$$\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} = \begin{bmatrix} I & \Sigma_{xy}\Sigma_{yy}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} I & 0 \\ \Sigma_{yy}^{-1}\Sigma_{yx} & I \end{bmatrix}$$

  and the inversion gives,

$$\begin{aligned} &\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} I & 0 \\ -\Sigma_{yy}^{-1}\Sigma_{yx} & I \end{bmatrix} \begin{bmatrix} (\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx})^{-1} & 0 \\ 0 & \Sigma_{yy}^{-1} \end{bmatrix} \begin{bmatrix} I & -\Sigma_{xy}\Sigma_{yy}^{-1} \\ 0 & I \end{bmatrix} \end{aligned}$$

Bayesian Inference

- Suppose $X = [x^T, y^T]^T$, the joint distribution $p(x,y)$ is

$$p(x,y) = \frac{1}{\sqrt{(2\pi)^N \det \Sigma}} \exp\left(-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)\right)$$

- the quadratic part

$$\left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}\right)^T \Sigma^{-1} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}\right)$$
$$= [(x-\mu_x)^T, (y-\mu_y)^T] \cdot$$
$$\begin{bmatrix} I & 0 \\ -\Sigma_{yy}^{-1}\Sigma_{yx} & I \end{bmatrix} \cdot \begin{bmatrix} (\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx})^{-1} & 0 \\ 0 & \Sigma_{yy}^{-1} \end{bmatrix} \cdot \begin{bmatrix} I & -\Sigma_{xy}\Sigma_{yy}^{-1} \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}$$
$$= \begin{bmatrix} (x-\mu_x)^T - (y-\mu_y)^T \Sigma_{yy}^{-1}\Sigma_{yx} & (y-\mu_y)^T \end{bmatrix} \cdot \begin{bmatrix} (\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx})^{-1} & 0 \\ 0 & \Sigma_{yy}^{-1} \end{bmatrix}$$
$$\cdot \begin{bmatrix} x - \mu_x - \Sigma_{xx}\Sigma_{yy}^{-1}(y-\mu_y) \\ y - \mu_y \end{bmatrix}$$
$$= [(x-\mu_x) - \Sigma_{xy}\Sigma_{yy}^{-1}(y-\mu_y)]^T (\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx})^{-1} [\cdots] + (y-\mu_y)^T \Sigma_{yy}^{-1}(y-\mu_y)$$

- the determinant

$$\det\left(\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right) = \det(\Sigma_{yy}) \cdot \det(\Sigma_{xx} - \Sigma_{yx}\Sigma_{yy}^{-1}\Sigma_{xy})$$

Bayesian Inference

As

$$p(x,y) = p(x|y)p(y)$$

we then have

$$p(x|y) = \mathcal{N}\left(\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y-\mu_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\right)$$

and

$$p(y) = \mathcal{N}(\mu_y, \Sigma_{yy})$$

Both are Gaussian!

Kalman filter via Bayesian Inference

- Assume the Gaussian posteriori estimate at $k-1$ is

$$p(x_{k-1}|\hat{x}_0, u_{0:k-2}, y_{1:k-1}) = \mathcal{N}(\hat{x}_{k-1}, \hat{P}_{k-1})$$

- Prediction step (u_{k-1} is available but y_k is not)

$$p(x_k|\hat{x}_0, u_{0:k-1}, y_{1:k-1}) = \mathcal{N}(\check{x}_k, \check{P}_k)$$

where

$$\check{x}_k = E(x_k) = F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1}$$

$$\check{P}_k = P(x_k) = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1}$$

Kalman filter via Bayesian Inference

- Correction step (y_k is available),

$$p(x_k, y_k|\hat{x}_0, u_{0:k-1}, y_{1:k-1}) = \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right)$$

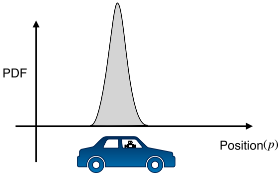
$$= \mathcal{N}\left(\begin{bmatrix} \check{x}_k \\ H_k\check{x}_k \end{bmatrix}, \begin{bmatrix} \check{P}_k & \check{P}_k H_k^T \\ H_k\check{P}_k & H_k\check{P}_k H_k^T + R_k \end{bmatrix}\right)$$

- The conditional density for x_k (the posterior) is,

$$p(x_k|\hat{x}_0, u_{0:k-1}, y_{1:k}) = \mathcal{N}\left(\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y_k - \mu_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\right)$$

- Define $\hat{x}_k$ as the mean and $\hat{P}_k$ as the covariance, and we have the same expressions as before.

The Kalman Filter | Short Example



$$\mathbf{x} = \begin{bmatrix} p \\ \frac{dp}{dt} = \dot{p} \end{bmatrix} \qquad \mathbf{u} = a = \frac{d^2p}{dt^2}$$

Motion/Process Model

$$\mathbf{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

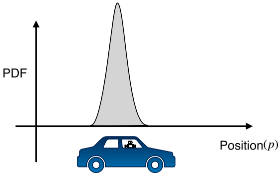
Position Observation

$$y_k = [1 \quad 0]\,\mathbf{x}_k + v_k$$

Noise Densities

$$v_k \sim \mathcal{N}(0,\ 0.05) \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, (0.1)\mathbf{1}_{2\times 2})$$

The Kalman Filter | Short Example



Data

$$\hat{\mathbf{x}}_0 \sim \mathcal{N}(\begin{bmatrix} 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix})$$

$$\Delta t = 0.5\text{s}$$

$$u_0 = -2 \; [m/s^2] \qquad y_1 = 2.2 \; [m]$$

The Kalman Filter | Short Example Solution

Prediction

$$\check{\mathbf{x}}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}$$

$$\begin{bmatrix} \check{p}_1 \\ \check{\dot{p}}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} (-2) = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}$$

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$$

$$\check{\mathbf{P}}_1 = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}^T + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

The Kalman Filter | Short Example Solution

Correction

$$\mathbf{K}_1 = \check{\mathbf{P}}_1\mathbf{H}_1^T(\mathbf{H}_1\check{\mathbf{P}}_1\mathbf{H}_1^T + \mathbf{R}_1)^{-1}$$

$$= \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.05 \right)^{-1}$$

$$= \begin{bmatrix} 0.88 \\ 1.22 \end{bmatrix}$$

$$\hat{\mathbf{x}}_1 = \check{\mathbf{x}}_1 + \mathbf{K}_1(\mathbf{y}_1 - \mathbf{H}_1\check{\mathbf{x}}_1)$$

$$\begin{bmatrix} \hat{p}_1 \\ \hat{\dot{p}}_1 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \begin{bmatrix} 0.88 \\ 1.22 \end{bmatrix} (2.2 - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}) = \begin{bmatrix} 2.24 \\ 3.63 \end{bmatrix}$$

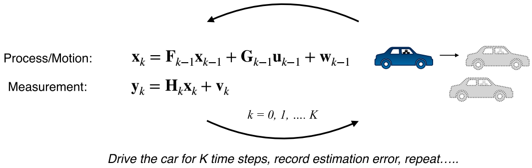
Bonus!

$$\hat{\mathbf{P}}_1 = (\mathbf{1} - \mathbf{K}_1\mathbf{H}_1)\check{\mathbf{P}}_1$$

$$= \begin{bmatrix} 0.04 & 0.06 \\ 0.06 & 0.49 \end{bmatrix}$$

Kalman filter properties: unbiasedness

Bias in State Estimation



- We say an estimator or filter is unbiased if it produces an 'average' error of zero at a particular time step *k*, over many trials

- the error dynamics:

$$\check{e}_k = \check{x}_k - x_k, \hat{e}_k = \hat{x}_k - x_k$$

- Using the Kalman filter equations, we can derive:

$$\check{e}_k = F_{k-1}\hat{e}_{k-1} - w_{k-1}, \hat{e}_k = (I - K_k H_k)\check{e}_k + K_k v_k$$

$$\hat{e}_k = (I - K_k H_k) F_{k-1}\hat{e}_{k-1} - (I - K_k H_k) w_{k-1} + K_k v_k$$

Bias in State Estimation

- For the Kalman filter, for all *k*,

$$\begin{aligned} E[\check{e}_k] &= E[F_{k-1}\hat{e}_{k-1} - w_{k-1}] \\ &= F_{k-1}E[\hat{e}_{k-1}] - E[w_{k-1}] \\ &= 0 \end{aligned}$$

$$\begin{aligned} E[\hat{\mathbf{e}}_k] &= E[(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\check{\mathbf{e}}_k + \mathbf{K}_k\mathbf{v}_k] \\ &= (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)E[\check{\mathbf{e}}_k] + \mathbf{K}_k E[\mathbf{v}_k] \\ &= \mathbf{0} \end{aligned}$$

***Unbiased** predictions!*

So long as $E[\hat{\mathbf{e}}_0] = \mathbf{0}$ $E[\mathbf{v}] = \mathbf{0}$ $E[\mathbf{w}] = \mathbf{0}$

+ white, uncorrelated noise

Note: this does *not* mean that the error on a *given* trial will be zero, but that, with enough trials, our expected error is zero!

Consistency of state estimators

- In the problem of estimating a parameter that is constant, consistency of an estimator (i.e., a static estimator) was defined as convergence of the estimate to the true value.
- This implies that there is a steadily increasing amount of information (in the sense of Fisher) about the parameter that asymptotically reduces to zero the uncertainty about its true value.
- When estimating the state of a dynamic system, in general, no convergence of its estimate occurs.
- What one has, in addition to the “current" estimate of the state, $\hat{x}_k$, is the associated covariance matrix, $\hat{P}_k$.

Consistency of state estimators

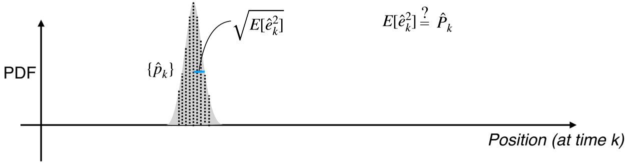
- A state estimator (filter) is called consistent if its state estimation errors satisfy

$$E(x_k - \hat{x}_k) = 0$$

$$E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = P_k$$

- This is a finite-sample consistency property, that is, the estimation errors based on a finite number of samples (measurements) should be consistent with their theoretical statistical properties:
  - Have mean zero (i.e., the estimates are unbiased)
  - Have covariance matrix as calculated by the filter

Consistency in State Estimation



This filter is consistent if **for all *k***,

$$E[\hat{e}_k^2] = E[(\hat{p}_k - p_k)^2] = \hat{P}_k$$

For an estimator,

- If $\hat{P}_k < E[\hat{e}_k \hat{e}_k^T]$, the KF "underestimate" the true uncertainty, the filter is optimistic.
- If $\hat{P}_k > E[\hat{e}_k \hat{e}_k^T]$, the KF "overestimate" the true uncertainty, the filter is pessimistic.

Consistency in State Estimation

- One can also show (with more algebra!) that for all *k*,

$$E[\check{\mathbf{e}}_k \check{\mathbf{e}}_k^T] = \check{\mathbf{P}}_k \qquad E[\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T] = \hat{\mathbf{P}}_k$$

Consistent *predictions!*

- Provided,

$$E[\hat{e}_0 \hat{e}_0^T] = \hat{P}_0, E(v) = 0, E(w) = 0 \quad v, w \text{ are white}$$

- Causes of inconsistency:
  - Modeling errors
  - Numerical errors
  - Programming errors

Practical evaluation of consistency

- Define the normalized (state) estimation error squared (NEES) as,

$$\epsilon(k) = \tilde{x}_k^T \hat{P}_k^{-1} \tilde{x}_k,$$

under the linear Gaussian assumption, one has

$$E[\tilde{x}_k^T \hat{P}_k^{-1} \tilde{x}_k] = n_x$$

that is, the NEES follows a χ^2 distribution and the average equals n_x.

- Hypothesis H_0: the filter is consistent and the linear Gaussian assumption, $\epsilon(k)$ is χ^2 distributed with n_x degrees of freedom, and the test is whether

$$E[\epsilon(k)]$$

can be accepted.

Kalman filter properties: overall optimality

The error $e_k = x_k - \hat{x}_k$ is a random variable determined by the stochastic process $\{w_k\}$ and $\{v_k\}$. Suppose we want to find the estimator that minimizes (at each time step) a weighted two-norm of the expected value of the estimation error e_k:

$$\min \quad E[e_k^T S_k e_k] \tag{1}$$

where S_k is a positive definite user-defined weighting matrix.

- if $\{w_k\}$ and $\{v_k\}$ are Gaussian, zero-mean, uncorrelated, and white, then the Kalman filter is the solution to the problem (1)
- if $\{w_k\}$ and $\{v_k\}$ are zero-mean, uncorrelated, and white, the Kalman filter is the best linear unbiased estimate to the problem (1)

Orthogonality principle in discrete-time Kalman filter

Preliminaries

- Initial estimate $\hat{x}_0 = E(x_0) = 0$ (a fixed value), suppose $u_k = 0$
- Uncorrelated properties:
  - w_i, v_i are uncorrelated with all past or present states, i.e.,

$$E[w_i x_j^T] = 0, E[v_i x_j^T] = 0, \forall j \leq i$$

  - w_i, v_i are orthogonal to past outputs, i.e.,

$$E[w_i y_j^T] = 0, E[v_i y_j^T] = 0, \forall j < i$$

Time update

Since $\check{x}_k$ should be the optimum LMMSE estimate, $x_k - \check{x}_k$ must satisfy the orthogonality condition, i.e.,

$$E[(x_k - \check{x}_k)y_i^T] = 0, \forall i = 1, 2, \ldots, k-1$$

Define $\mathbf{y}_{k-1} = [y_1, \ldots, y_{k-1}]^T$, and we have

$$E[(x_k - \check{x}_k)y_i^T] = E[(F_{k-1}x_{k-1} + w_{k-1} - F_{k-1}\hat{x}_{k-1})\mathbf{y}_{k-1}^T]$$
$$+E[(F_{k-1}\hat{x}_{k-1} - \check{x}_k)\mathbf{y}_{k-1}^T] = 0.$$

As $\hat{x}_{k-1}$ is the LMMSE estimate of x_{k-1} given $\mathbf{y}_{k-1}$, there is

$$E[(x_{k-1} - \hat{x}_{k-1})\mathbf{y}_{k-1}^T] = 0,$$

then we have

$$\check{x}_k = F_{k-1}\hat{x}_{k-1}$$
$$\check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1}$$

Measurement update

As $\hat{x}_{k-1}$ is the LMMSE estimate given $\mathbf{y}_{k-1}$, the following is valid,

$$E[(x_{k-1} - \hat{x}_{k-1})\mathbf{y}_{k-1}^T] = 0,$$

Since $\hat{x}_{k-1}$ is the linear function of $\mathbf{y}_{k-1}$, we can write $\hat{x}_{k-1}$ as,

$$\hat{x}_{k-1} = J_{k-1}\mathbf{y}_{k-1}.$$

Now assume $\hat{x}_k = J_k\mathbf{y}_k$, according to the orthogonality condition, we then have

$$E[(x_k - J_k\mathbf{y}_k)\mathbf{y}_k^T] = 0$$

Measurement update

Suppose $\hat{x}_k = K_k y_k + G_k \mathbf{y}_{k-1}$, substituting the expression of y_k yields,

$$\begin{aligned}\hat{x}_k &= K_k(H_k x_k + v_k) + G_k \mathbf{y}_{k-1} \\ &= K_k(H_k(F_{k-1}x_{k-1} + w_{k-1}) + v_k) + G_k \mathbf{y}_{k-1}\end{aligned}$$

As $\hat{x}_k$ is the LMMSE estimate given $\mathbf{y}_k$, we have

$$E[(x_k - \hat{x}_k)\mathbf{y}_k^T] = 0,$$

i.e.,

$$E\left\{(x_k - \hat{x}_k) \cdot \begin{bmatrix} \mathbf{y}_{k-1}^T \\ y_k^T \end{bmatrix}\right\} = 0$$

From the equality $E[(x_k - \hat{x}_k)\mathbf{y}_{k-1}^T] = 0$, we have

$$E[(Fx_{k-1} + w_{k-1} - K_k H_k F_{k-1} x_{k-1} - K_k H_k w_{k-1} - K_k v_k - G_k \mathbf{y}_{k-1})\mathbf{y}_{k-1}^T] = 0$$

And further the following is true,

$$E[((I - K_k H_k)F(x_{k-1} - \hat{x}_{k-1}) + (I - K_k H_k)F\hat{x}_{k-1} - G_k \mathbf{y}_{k-1})\mathbf{y}_{k-1}^T] = 0$$

as w_{k-1}, v_k are uncorrelated with $\mathbf{y}_{k-1}^T$.

According to the fact $\hat{x}_{k-1}$ is the LMMSE estimate, the orthogonality principle holds, i.e.,

$$E[(I - K_k H_k)F(x_{k-1} - \hat{x}_{k-1})\mathbf{y}_{k-1}^T] = 0,$$

and then we can take

$$G_k \mathbf{y}_{k-1} = (I - K_k H_k)F\hat{x}_{k-1}$$

And we have

$$\hat{x}_k = K_k y_k + G_k \mathbf{y_{k-1}} = K_k y_k + (I - K_k H_k)F_{k-1}\hat{x}_{k-1} = \check{x}_k + K_k[y_k - H_k \check{x}_k]$$

According to the propagation of error

$$\hat{e}_k = x_k - \hat{x}_k = (I - K_k H_k)F_{k-1}\hat{e}_{k-1} + (I - K_k H_k)w_{k-1} - K_k v_k$$

and we write y_k as

$$y_k = H_k x_k + v_k = H_k F_{k-1} x_{k-1} + H_k w_{k-1} + v_k$$

as the equality $E[(x_k - \hat{x}_k)y_k^T] = 0$ holds, we then have

$$\begin{aligned} E[\hat{e}_k y_k^T] &= F_{k-1}\hat{P}_{k-1}F_{k-1}^T H_k^T + Q_{k-1}H_k^T - K_k H_k F_{k-1}\hat{P}_{k-1}F_{k-1}^T H_k^T \\ &\quad - K_k H_k Q_{k-1} H_k^T - K_k R_k = 0 \end{aligned}$$

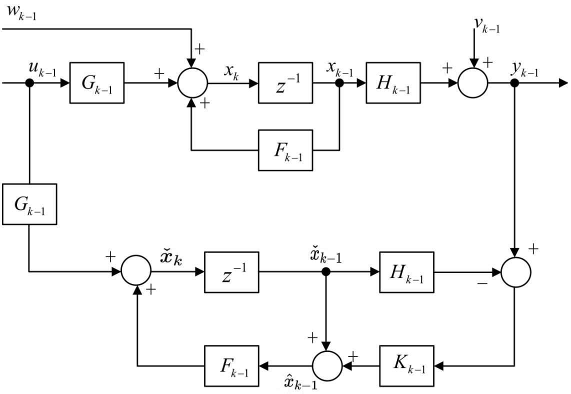
Recalling the expression for $\check{P}_k$, and we have,

$$K_k = \check{P}_k H_k^T [H_k \check{P}_k H_k^T + R_k]^{-1}$$

and the posteriori covariance can be expressed as

$$\hat{P}_k = \check{P}_k - K_k H_k \check{P}_k$$

The block diagram of Kalman filter:



Example

Considering a simple 1-dimensional example,

$$x_{k+1} = 0.5x_k + w_k$$
$$y_k = x_k + v_k$$

w_k and v_k are uncorrelated white noise with zero mean, i.e.,

$$E\{w_k\} = 0,\ E\{v_k\} = 0,\ E\{w_k w_j\} = 1 \cdot \delta_{k-j},\ E\{v_k v_j\} = 2 \cdot \delta_{k-j}$$

Initial value: $\hat{x}_0 = 0$, $P_0 = 1$; Observations $y_1 = 4,\ y_2 = 2$. Find the optimal linear estimate of x_k.

Example

- Initialization: $\hat{x}_0 = 0$, $\hat{P}_0 = P_0 = 1$
- Computation of the gain K_k as well as $\check{P}_k,\ \hat{P}_k$, i.e.,

$$\check{P}_1 = F_0 \hat{P}_0 F_0^T + Q_0 = 1.25$$
$$K_1 = \check{P}_1 H_1^T (H_1 \check{P}_1 H_1^T + R_1)^{-1} = 0.3846$$
$$\hat{P}_1 = [I - K_1 H_1] \check{P}_1 [I - K_1 H_1]^T + K_1 R_1 K_1^T = 0.7692$$
$$\check{P}_2 = F_1 \hat{P}_1 F_1^T + Q_1 = 1.1923$$
$$K_2 = \check{P}_2 H_2^T (H_2 \check{P}_2 H_2^T + R_2)^{-1} = 0.3735$$

- Estimation of the state sequence:

$$\check{x}_1 = F_0 \hat{x}_0 = 0$$

$$\hat{x}_1 = \check{x}_1 + K_1(y_1 - H_1\check{x}_1) = 1.5385$$

$$\check{x}_2 = F_1 \hat{x}_1 = 0.7692$$

$$\hat{x}_2 = \check{x}_2 + K_2(y_2 - H_2\check{x}_2) = 1.2289$$

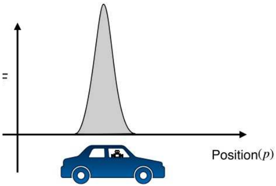
Computation of K_k when time increases:

$$\begin{array}{lll} K_1 = 0.3846 & \check{P}_1 = 1.2500 & \hat{P}_1 = 0.7692 \\ K_2 = 0.3735 & \check{P}_2 = 1.1923 & \hat{P}_2 = 0.7470 \\ K_3 = 0.3724 & \check{P}_3 = 1.1867 & \hat{P}_3 = 0.7448 \\ K_4 = 0.3723 & \check{P}_4 = 1.1862 & \hat{P}_4 = 0.7446 \\ K_5 = 0.3723 & \check{P}_5 = 1.1861 & \hat{P}_5 = 0.7446 \\ K_6 = 0.3723 & \check{P}_6 = 1.1861 & \hat{P}_6 = 0.7446 \\ \vdots & \vdots & \vdots \end{array}$$

The former example:

double

1	2	3	4	5	6	7	8	9	10
.8780	0.8675	0.8435	0.8338	0.8303	0.8291	0.8286	0.8285	0.8285	0.8284
.2195	0.8110	0.6623	0.6120	0.5948	0.5889	0.5869	0.5862	0.5859	0.5858



P_check

2x2x10 double

```
val(:,:,1) =
    0.3600    0.5000
    0.5000    1.1000
val(:,:,2) =
    0.3274    0.3061
    0.3061    0.5902
val(:,:,3) =
    0.2694    0.2116
    0.2116    0.4420
val(:,:,4) =
    0.2508    0.1841
    0.1841    0.4019
val(:,:,5) =
    0.2446    0.1752
    0.1752    0.3893
val(:,:,6) =
    0.2425    0.1723
    0.1723    0.3850
val(:,:,7) =
    0.2418    0.1712
    0.1712    0.3836
val(:,:,8) =
    0.2415    0.1709
```

P_hat

2x2x10 double

```
val(:,:,1) =
    0.0439    0.0610
    0.0610    0.4902
val(:,:,2) =
    0.0434    0.0405
    0.0405    0.3420
val(:,:,3) =
    0.0422    0.0331
    0.0331    0.3019
val(:,:,4) =
    0.0417    0.0306
    0.0306    0.2893
val(:,:,5) =
    0.0415    0.0297
    0.0297    0.2850
val(:,:,6) =
    0.0415    0.0294
    0.0294    0.2836
val(:,:,7) =
    0.0414    0.0293
    0.0293    0.2831
val(:,:,8) =
    0.0414    0.0293
    0.0293    0.2829
```

Steady-state Kalman filter

- when time increases, the Kalman filter gain and error covariance reaches a steady-state value.
- The term "steady-state" Kalman filtering means that the Kalman filter is time-invariant and the Kalman gain is in steady-state

Properties of steady-state Kalman filter

- the dynamic system is time-invariant
- a constant gain K can be pre-computed, $K = PH^T(HPH^T + R)^{-1}$
- P reaches a steady-state

$$P = FPF^T - FPH^T(HPH^T + R)^{-1}HPF^T + Q$$

  and the above equation is called discrete time algebraic Riccati equation (DARE).

- the savings in computations deserve any loss in the estimated state quality

Conditions for the existence of the steady-state Kalman filter

The DARE has a unique positive semidefinite solution P iff both of the following conditions hold.

- (F, H) is detectable
  - A system is detectable if all the unobservable states are stable.
- (F, J) is stabilizable $\left(J \text{ is any matrix such that } JJ^T = Q\right)$
  - A system is said to be stabilizable when all uncontrollable state variables can be made to have stable dynamics.

Conditions for the existence of the steady-state Kalman filter

- If the signal process model is time invariant and asymptotically stable, then

1. For any nonnegative symmetric initial condition $\hat{P}_0$, one has

$$\lim_{k\to\infty} \check{P}_k = P$$

   and P satisfies the discrete time algebraic Riccati equation.

$$P = FPF^T - FPH^T(HPH^T + R)^{-1}HPF^T + Q$$

2. The Kalman filter gain K reaches a constant value and the matrix $(I - KH)F$ is stable.

Steady-state Kalman filter error

Steady-state Kalman filter is a time-invariant observer (also time-invariant system):

$$\check{x}_k = F\hat{x}_{k-1} + Gu_{k-1}$$

$$\begin{aligned}\hat{x}_k &= \check{x}_k + K(y_k - H\check{x}_k)\\ &= (I-KH)F\hat{x}_{k-1} + (I-KH)Gu_{k-1} + KHFx_{k-1} + Kv_k\end{aligned}$$

compared with the state space expression

$$x_k = Fx_{k-1} + Gu_{k-1} + w_{k-1}$$

$$y_k = Hx_k + v_k$$

the state estimation error is

$$\hat{e}_k = x_k - \hat{x}_k = (I-KH)F\hat{e}_{k-1} + KHGu_{k-1} + w_{k-1} - Kv_k$$

Stability of steady-state Kalman filter

- the estimation error propagates according to a linear system, with closed-loop dynamics $(I - KH)F$, driven by the input $KHGu_{k-1}$ and the process $w_{k-1} - Kv_k$, which is IID with zero mean and covariance $KRK^T + Q$
- The stability of $(I - KH)F$ is requisite for the stability of the filter.
- If the DARE has a unique positive semidefinite solution, then the steady-state Kalman filter is stable.

A simple example in 1-dimension

The truth model is given by the following equation,

$$x_{k+1} = \varphi x_k + w_k$$
$$y_k = h x_k + v_k$$

in which w_k and v_k are stationary random process, with $w_k \sim \mathcal{N}(0, q)$ and $v_k \sim \mathcal{N}(0, r)$. Assume the system is stable.

Considering the discrete time algebraic Riccati equation, we have,

$$p = \varphi^2 p - \varphi p h \frac{1}{h^2 p + r} h p \varphi + q$$

By reordering, we have,

$$h^2 p^2 + (r - \varphi^2 r - h^2 q) p - qr = 0 \tag{2}$$

Solving the second-order equation (2) gives the solution of steady-state p.

Next consider two special cases.

- no measurement noise: $r = 0$. then we have $p = q$ and $k = \frac{1}{h}$ and

$$\hat{x}_{k+1} = \frac{\varphi}{h} y_k$$

  at this time, the estimate $\hat{x}_{k+1}$ depends entirely on the measurement, and does not depend on past estimate $\hat{x}_k$. This is because no measurement error exists, and the state can be estimated without the dynamic model.

- the model is accurate: $q = 0$. then we have $p = 0$. at this time,

$$\hat{x}_{k+1} = \varphi \hat{x}_k$$

the estimate depends entirely on the dynamic model, which is due to the model is precise.

Contents

Importance of continuous-time Kalman filter

- Although the vast majority of Kalman filter applications are implemented in digital computers, there are still opportunities to implement Kalman filters in continuous time (i.e., in analog circuits)
- the derivation of the continuous-time filter is instructive from a pedagogical point of view
- steady-state continuous-time estimators can be analyzed using conventional frequency-domain concepts, providing an advantage over discrete-time estimators

Continuous-time dynamic system

Suppose that we have a continuous-time system given as

$$\dot{x} = Ax + Bu + w$$

$$y = Cx + v$$

$$w \sim \mathcal{N}(0, Q_c)$$

$$v \sim \mathcal{N}(0, R_c)$$

Assume $t = t_k$ and $x_k = x(t_k)$, the continuous-time system can be approximated by the following discrete-time system:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}$$
$$y_k = H_k x_k + v_k$$

Next we will derive the expression for F_{k-1}, G_{k-1}, H_k and the stochastic properties of $\{w_k\}$ and $\{v_k\}$.

Recalling from the sampled-data system, the solution of $x(t)$ when $t = t_k$ can be computed as

$$x(t_k) = e^{A(t_k - t_{k-1})} x(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A(t_k - \tau)} [Bu(\tau) + w(\tau)] d\tau$$

Suppose $u(t) = u(t_{k-1}), \forall t \in [t_{k-1}, t_k]$. This is reasonable if $t_k - t_{k-1} \to 0$. Further, we have

$$x_k = e^{A(t_k - t_{k-1})} x_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k - \tau)} B d\tau u_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k - \tau)} w(\tau) d\tau$$

Suppose the sample time is T, i.e., $T = t_k - t_{k-1}$, Define

$$F = \exp(AT)$$
$$G = \int_{t_{k-1}}^{t_k} e^{A(t_k - \tau)} B d\tau$$

then for small T, we have

$$F \approx I + AT$$

and

$$\begin{aligned} G &= \int_{t_{k-1}}^{t_k} (I + A(t_k - \tau)) B d\tau \\ &\approx BT \end{aligned}$$

Define $w_{k-1} = \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)} w(\tau) d\tau$, then

$$\bar{w}_{k-1} = 0$$

and

$$\begin{aligned} E[w_k w_j^T] &= \int_{t_k}^{t_{k+1}} \int_{t_j}^{t_{j+1}} e^{A(t_{k+1}-\tau)} E[w(\tau) w^T(t)] e^{A^T(t_{j+1}-t)} d\tau dt \\ &= \int_{t_k}^{t_{k+1}} \int_{t_j}^{t_{j+1}} 0 d\tau dt \quad \forall k \neq j \\ &= 0, \forall k \neq j. \end{aligned}$$

The covariance of $\{w_k\}$ is:

$$
\begin{aligned}
E[w_k w_k^T] &= \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} E[w(\tau) w^T(t)] e^{A^T(t_{k+1}-t)} dt d\tau \\
&= \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} Q_c(\tau) \delta(t-\tau) e^{A^T(t_{k+1}-t)} dt d\tau \\
&= \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-t)} Q_c(t) e^{A^T(t_{k+1}-t)} dt \\
&\approx Q_c T \text{ assume } T \text{ is small}
\end{aligned}
$$

The discretization of the measurement equation can be interpreted as:

$$y_k \approx Cx_k + \frac{1}{T}\int_{t_k}^{t_{k+1}} v(t)dt$$

Define $v_k = \frac{1}{T}\int_{t_k}^{t_{k+1}} v(t)dt$, then

$$\bar{v}_k = 0$$

and

$$\begin{aligned}
E[v_k v_j^T] &= \frac{1}{T^2}\int_{t_k}^{t_{k+1}}\int_{t_j}^{t_{j+1}} E[v(\tau)v^T(t)]d\tau dt \\
&= \frac{1}{T^2}\int_{t_k}^{t_{k+1}}\int_{t_j}^{t_{j+1}} 0 d\tau dt \\
&= 0
\end{aligned}$$

Besides,

$$
\begin{aligned}
E[v_k v_k^T] &= \frac{1}{T^2} \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} E[v(t) v^T(\tau)] dt d\tau \\
&= \frac{1}{T^2} \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} R_c \delta(t - \tau) dt d\tau \\
&= \frac{1}{T^2} \int_{t_k}^{t_{k+1}} R_c d\tau \\
&= \frac{R_c}{T}
\end{aligned}
$$

Define

$$H = C$$

we have the discretization of the continuous-time dynamic system with a sample time T:

$$x_k = Fx_{k-1} + Gu_{k-1} + w_{k-1}$$
$$y_k = Hx_k + v_k$$

with the process noise and measurement noise as:

$$w_k \sim \mathcal{N}(0, Q), Q = Q_c T, \quad v_k \sim \mathcal{N}(0, R), R = R_c / T$$

The discrete-time Kalman filter gain for this system was derived as:

$$\begin{aligned} K_k &= \check{P}_k H^T (H \check{P}_k H^T + R)^{-1} \\ &= \check{P}_k C^T (C \check{P}_k C^T + R_c/T)^{-1} \end{aligned}$$

then

$$\frac{K_k}{T} = \check{P}_k C^T (C \check{P}_k C^T T + R_c)^{-1} \tag{3}$$

$$\lim_{T \to 0} \frac{K_k}{T} = \check{P}_k C^T R_c^{-1} \tag{4}$$

The estimation-error covariances were derived as

$$\hat{P}_k = (I - K_k H)\check{P}_k$$

$$\check{P}_{k+1} = F\hat{P}_k F^T + Q$$

For small values of T, this can be written as

$$\begin{aligned}\check{P}_{k+1} &= (I + AT)\hat{P}_k(I + AT)^T + Q_c T \\ &= \hat{P}_k + (A\hat{P}_k + \hat{P}_k A^T + Q_c)T + A\hat{P}_k A^T T^2\end{aligned}$$

Substituting for $\hat{P}_k$ gives

$$\begin{aligned}\check{P}_{k+1} = (I - K_k C)\check{P}_k + A\hat{P}_k A^T T^2 + \\ [A(I - K_k C)\check{P}_k + (I - K_k C)\check{P}_k A^T + Q_c]T\end{aligned}$$

Substracting $\check{P}_k$ from both sides and then dividing by T gives

$$\begin{aligned}\frac{\check{P}_{k+1} - \check{P}_k}{T} =& \frac{-K_k C \check{P}_k}{T} + A\hat{P}_k A^T T + \\ & (A\check{P}_k + AK_k C\check{P}_k + \check{P}_k A^T - K_k C \check{P}_k A^T + Q_c)\end{aligned}$$

Taking the limit as $T \to 0$ and using the expression for K_k gives

$$\begin{aligned}\dot{P} &= \lim_{T\to 0} \frac{\check{P}_{k+1} - \check{P}_k}{T} \\ &= -PC^T R_c^{-1} C P + AP + PA^T + Q_c\end{aligned}$$

Differential Riccati equation

- The equation

$$\dot{P} = -PC^T R_c^{-1} CP + AP + PA^T + Q_c$$

is called a **differential Riccati equation** and can be used to compute the estimation-error covariance for the continuous-time Kalman filter

- this requires n^2 integrations because P is an $n \times n$ matrix
- as P is symmetric, so in practice we only need to integrate $n(n+1)/2$ equations in order to solve for P

The continuous-time Kalman filter equations for $\hat{x}$

The discrete-time version:

$$\check{x}_k = F\hat{x}_{k-1} + Gu_{k-1}$$

$$\hat{x}_k = \check{x}_k + K_k(y_k - H\check{x}_k)$$

Assume that T is small, the measurement update equation can be written as

$$\begin{aligned}
\hat{x}_k &= F\hat{x}_{k-1} + Gu_{k-1} + K_k(y_k - HF\hat{x}_{k-1} - HGu_{k-1}) \\
&\approx (I + AT)\hat{x}_{k-1} + BTu_{k-1} + K_k(y_k - C(I + AT)\hat{x}_{k-1} - CBTu_{k-1})
\end{aligned}$$

Now subtract $\hat{x}_{k-1}$ from both sides, divide by T to obtain

$$\begin{aligned} \frac{\hat{x}_k - \hat{x}_{k-1}}{T} &= A\hat{x}_{k-1} + Bu_{k-1} \\ &\quad + \check{P}_k C^T (C\check{P}_k C^T T + R_c)^{-1}(y_k - C(I + AT)\hat{x}_{k-1} - CBTu_{k-1}) \end{aligned}$$

Taking the limit as $T \to 0$ gives

$$\dot{\hat{x}} = A\hat{x} + Bu + PC^T R_c^{-1}(y - C\hat{x})$$

The framework of the continuous-time Kalman filter

- The continuous-time system dynamics and measurement equations are given as

$$\dot{x} = Ax + Bu + w$$

$$y = Cx + v$$

$$w \sim \mathcal{N}(0, Q_c)$$

$$v \sim \mathcal{N}(0, R_c)$$

in which $w(t)$ and $v(t)$ are continuous-time white noise processes

The framework of the continuous-time Kalman filter

- The continuous-time Kalman filter equations are given as

$$\hat{x}(0) = E[x(0)]$$

$$P(0) = E[(x(0) - \hat{x}(0))(x(0) - \hat{x}(0))^T]$$

$$K = PC^T R_c^{-1} \text{ (This } K \text{ is not the limit of } K_k \text{ as } T \to 0)$$

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x})$$

$$\dot{P} = -PC^T R_c^{-1} CP + AP + PA^T + Q_c$$

Example

Suppose the system dynamic is,

$$\dot{x}(t) = -x(t) + w(t)$$
$$y(t) = x(t) + v(t)$$

in which $w(t)$ and $v(t)$ are white noises with zero mean, and the statistical property is as follows,

$$E\{w(t)\} = E\{v(t)\} = 0$$
$$E\{w(t)w(\tau)\} = 2.5\delta(t-\tau)$$
$$E\{v(t)v(\tau)\} = 2\delta(t-\tau)$$
$$E\{w(t)v(\tau)\} = 0$$

Assume $P(0) = 3$, $E\{x(0)\} = m_0$. Design the continuous-time Kalman filter.

According to the filter design principle,

$$K(t) = 0.5P(t)$$

and the derivation of $P(t)$ is,

$$\dot{P}(t) = -2P(t) - 0.5P^2(t) + 2.5$$

Denote $P = P(t)$, then the above differential equation can be written as,

$$\begin{aligned}
&\tfrac{dP}{dt} = -0.5(P-1)(P+5) \\
&\tfrac{dP}{P-1} - \tfrac{dP}{P+5} = -3dt \\
&\int_3^P \left(\tfrac{dP}{P-1} - \tfrac{dP}{P+5}\right) = \int_0^t -3dt \\
&\ln(P-1) - \ln 2 - \ln(P+5) + \ln 8 = -3t \\
&\tfrac{P-1}{P+5} = e^{-3t-2\ln 2} \\
&\tfrac{P-1}{P+5} = \tfrac{1}{4}e^{-3t}
\end{aligned}$$

Therefore, we have,

$$P(t) = \frac{1 + \frac{5}{4}e^{-3t}}{1 - \frac{1}{4}e^{-3t}}$$

As $t \longrightarrow \infty$, $P(t) \longrightarrow 1$.

The Kalman gain $K(t)$ is,

$$K(t) = \frac{1 + \frac{5}{4}e^{-3t}}{2\left(1 - \frac{1}{4}e^{-3t}\right)}$$

and $K(t) \longrightarrow 0.5$ as $t \longrightarrow \infty$.

When time increases, the Kalman filter gain and error covirance reaches a steady-state value.

Also, if we want to directly know the steady-state covariance, just let $\dot{P}(t) = 0$, which gives,

$$-2P(t) - 0.5P^2(t) + 2.5 = 0$$

that is,

$$P(t) = 1, \quad \text{or} \quad P(t) = -5 (\text{is not reasonable, hence delete})$$

The steady-state covariance and gain is independent of the initial value $P(0)$.

Example

- obtain measurements of the velocity of an object that is moving in one dimension
- the object is subject to random accelerations
- we want to estimate the velocity x from noisy velocity measurements
- the system and measurement equations are given as

$$\dot{x} = w$$

$$y = x + v$$

$$w \sim \mathcal{N}(0, Q)$$

$$v \sim \mathcal{N}(0, R)$$

The covariance matrix is

$$\begin{aligned}\dot{P} &= -PC^TR^{-1}CP + AP + PA^T + Q \\ &= -P^2/R + Q\end{aligned}$$

Integrate both sides from 0 to t yields:

$$\int_{P(0)}^{P(t)} \frac{dP}{Q - P^2/R} = \int_0^t d\tau$$

Then we have (assume $\sqrt{Q} > P/\sqrt{R}$)

$$\frac{\sqrt{R}}{2\sqrt{Q}} \ln\left(\frac{\sqrt{Q} + P/\sqrt{R}}{\sqrt{Q} - P/\sqrt{R}}\right) |_{P(0)}^{P(t)} = t$$

Solving the differential equation for P gives

$$P = \sqrt{QR}\left[\frac{P_0 - \sqrt{QR} + (\sqrt{QR} + P_0)\exp(2t\sqrt{Q/R})}{\sqrt{QR} - P_0 + (\sqrt{QR} + P_0)\exp(2t\sqrt{Q/R})}\right]$$

Take the limit as $t \to \infty$ we have

$$\lim_{t\to\infty} P = \sqrt{QR}$$

The Kalman gain is

$$K = PC^T R^{-1} = P/R$$

Take the limit as $t \to \infty$ we have

$$\lim_{t \to \infty} K = \sqrt{Q/R}$$

The state estimate update expression is

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) = K(y - \hat{x}) \to \sqrt{\frac{Q}{R}}(y - \hat{x})$$

Remarks

- If the process noise increases (i.e., Q increases) then K increases, meaning that we have less confidence in our system model, and relatively more confidence in our measurements. So we change $\hat{x}$ more aggressively to be consistent with our measurements
- if we have large measurement noise (i.e., R is large) then K decreases, meaning that we have less confidence in our measurements. So we change $\hat{x}$ less aggressively to be consistent with our measurements
- if either Q or R increase then P increases. An increase in the noise in either the system model or the measurements will degrade our confidence in our state estimate.

Steady-state continuous-time Kalman filter

- The continuous analytical Riccati equation has a unique positive semidefinite solution P if and only if both of the following conditions hold.

  1. (A, C) is detectable.
  2. (A, G) is stabilizable (G is any matrix such that $GG^T = Q$).

- Furthermore, the corresponding steady-state Kalman filter is stable. That is, the eigenvalues of $(A - KC)$ have negative real parts.

Steady-state continuous-time Kalman filter

- The continuous algebraic Riccati equation has at least one positive semidefinite solution P if (A, C) is detectable.
- Furthermore, at least one such solution results in a marginally stable steady-state Kalman filter.

Example

We consider the following two-state system:

$$\dot{x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + w$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + v$$

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Solve the steady-state continuous-time Kalman filter.

The differential Riccati equation for the Kalman filter is given as

$$\dot{P} = -PC^T R^{-1} CP + AP + PA^T + Q$$

This can be written as the following three coupled differential equations:

$$\dot{p}_{11} = 2p_{11} - p_{11}^2 - p_{12}^2$$

$$\dot{p}_{12} = 2p_{12} - p_{11}p_{12} - p_{12}p_{22}$$

$$\dot{p}_{22} = 2p_{22} - p_{12}^2 - p_{22}^2$$

Let $\dot{P} = 0$ and calculate the steady-state value for P yields,

$$P = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \text{ or } P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ or } P = \begin{bmatrix} c & \pm\sqrt{2c - c^2} \\ \pm\sqrt{2c - c^2} & 2 - c \end{bmatrix}$$

in which $c \in [0, 2]$ is a scalar. Then we have

$$\begin{aligned} K &= PC^T R^{-1} = P \\ &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \text{ or } K = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ or } K = \begin{bmatrix} c & \pm\sqrt{2c - c^2} \\ \pm\sqrt{2c - c^2} & 2 - c \end{bmatrix} \end{aligned}$$

Hence the estimate for x is

$$\dot{\hat{x}} = (A - KC)\hat{x} + Ky$$

$$= (-\hat{x} + Ky) \text{ or } (\hat{x} + Ky) \text{ or } \begin{bmatrix} 1-c & \mp\sqrt{2c-c^2} \\ \mp\sqrt{2c-c^2} & c-1 \end{bmatrix} \hat{x} + Ky$$

in which only the first steady-state continuous-time Kalman filter is stable (the eigenvalues of the first are -1, -1, of the second are 1, 1, of the third are 1, -1). The other two filters are unstable Kalman filters.

Contents

Correlated process and measurement noise

Suppose that we have a system given by

$$
\begin{aligned}
x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\
y_k &= H_k x_k + v_k \\
w_k &\sim (0, Q_k) \\
v_k &\sim (0, R_k) \\
E[w_k w_j^T] &= Q_k \delta_{k-j} \\
E[v_k v_j^T] &= R_k \delta_{k-j} \\
E[w_k v_j^T] &= M_j \delta_{k-j+1}
\end{aligned}
$$

An example to explain this

- suppose our system is an airplane and winds are buffeting the plane
- we are using an anemometer to measure wind speed as an input to our Kalman filter
- the random gusts of wind affect both the process (i.e., the airplane dynamics) and the measurement (i.e., the sensed wind speed)
- the process noise w_k affects the state x_{k+1}, while v_{k+1} affects the measurement y_{k+1}, and w_k is correlated with v_{k+1}

update equation for the state estimate

$$
\begin{aligned}
\check{x}_k &= F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1} \\
\hat{x}_k &= \check{x}_k + K_k(y_k - H_k\check{x}_k)
\end{aligned}
$$

The gain matrix K_k will not be the same. Define the estimation error as

$$\check{e}_k = x_k - \check{x}_k$$

$$\hat{e}_k = x_k - \hat{x}_k$$

update equation for the estimation error

$$
\begin{aligned}
\check{e}_k &= x_k - \check{x}_k \\
&= (F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}) - (F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1}) \\
&= F_{k-1}\hat{e}_{k-1} + w_{k-1} \\
\hat{e}_k &= x_k - [\check{x}_k + K_k(y_k - H_k\check{x}_k)] \\
&= \check{e}_k - K_k(H_k x_k + v_k - H_k\check{x}_k) \\
&= \check{e}_k - K_k(H_k\check{e}_k + v_k) \\
&= (I - K_k H_k)\check{e}_k - K_k v_k
\end{aligned}
$$

A priori and a posteriori estimation-error covariances

$$
\begin{aligned}
\check{P}_k &= E[\check{e}_k(\check{e}_k)^T] \\
&= F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1} \\
\hat{P}_k &= E[\hat{e}_k(\hat{e}_k)^T] \\
&= E\{[\check{e}_k - K_k(H_k\check{e}_k + v_k)][\cdots]^T\} \\
&= \check{P}_k - K_kH_k\check{P}_k - K_kE[v_k(\check{e}_k)^T] - \check{P}_kH_k^TK_k^T + \\
&\quad K_kH_k\check{P}_kH_k^TK_k^T + K_kE[v_k(\check{e}_k)^T]H_k^TK_k^T - \\
&\quad E(\check{e}_kv_k^T)K_k^T + K_kH_kE(\check{e}_kv_k^T)K_k^T + K_kE(v_kv_k^T)K_k^T
\end{aligned}
$$

Simplify the expression for $\hat{P}_k$

$$
\begin{aligned}
E(\check{e}_k v_k^T) &= E[(x_k - \check{x}_k) v_k^T] \\
&= E(x_k v_k^T - \check{x}_k v_k^T) \\
&= E[(F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + w_{k-1}) v_k^T] - E[\check{x}_k v_k^T] \\
&= 0 + 0 + M_k - 0
\end{aligned}
$$

The last term is 0 because the a priori state estimate at time k is independent of v_k.

Simplify the expression for $\hat{P}_k$

Substituting the expression of $E(\check{e}_k v_k^T)$ into the expression for $\hat{P}_k$ gives

$$\begin{aligned}
\hat{P}_k &= \check{P}_k - K_k H_k \check{P}_k - K_k M_k^T - \check{P}_k H_k^T K_k^T + K_k H_k \check{P}_k H_k^T K_k^T + \\
&\quad + K_k M_k^T H_k^T K_k^T - M_k K_k^T + K_k H_k M_k K_k^T + K_k R_k K_k^T \\
&= (I - K_k H_k)\check{P}_k (I - K_k H_k)^T + K_k R_k K_k^T + \\
&\quad K_k (H_k M_k + M_k^T H_k^T) K_k^T - M_k K_k^T - K_k M_k^T
\end{aligned}$$

Find the optimal K_k

We get the optimal gain matrix K_k by minimizing $\mathrm{Tr}(\hat{P}_k)$. Recall that

$$\frac{\partial \mathrm{Tr}(ABA^T)}{\partial A} = 2AB \text{ if } B \text{ is symmetric.}$$

We can use this fact to derive

$$\begin{aligned}
\tfrac{\partial \mathrm{Tr}(\hat{P}_k)}{\partial K_k} &= -2(I - K_k H_k)\check{P}_k H_k^T + 2K_k R_k + \\
&\quad 2K_k(H_k M_k + M_k^T H_k^T) - M_k - K_k \\
&= 2[K_k(H_k \check{P}_k H_k^T + H_k M_k + M_k^T H_k^T + R_k) - \check{P}_k H_k^T - M_k]
\end{aligned}$$

Setting the partial derivative to be zero gives the optimal gain K_k as

$$K_k = (\check{P}_k H_k^T + M_k)(H_k \check{P}_k H_k^T + H_k M_k + M_k^T H_k^T + R_k)^{-1}$$

The estimation-error covariance propagation

$$
\begin{aligned}
\hat{P}_k &= (I - K_k H_k)\check{P}_k (I - K_k H_k)^T + K_k R_k K_k^T + \\
&\quad K_k (H_k M_k + M_k^T H_k^T) K_k^T - M_k K_k^T - K_k M_k^T \\
&= \check{P}_k - K_k (H_k \check{P}_k + M_k^T)
\end{aligned}
$$

The general discrete-time Kalman filter

- The system and measurement equations are given as

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[v_k v_j^T] &= R_k \delta_{k-j} \\ E[w_k v_j^T] &= M_j \delta_{k-j+1} \end{aligned}$$

- The Kalman filter is initialized as

$$\begin{aligned} \hat{x}_0 &= E(x_0) \\ \hat{P}_0 &= E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \end{aligned}$$

The general discrete-time Kalman filter

- For each time step $k = 1, 2, \ldots,$ the Kalman filter equations are given as

$$
\begin{aligned}
\check{P}_k &= F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_{k-1} \\
K_k &= (\check{P}_k H_k^T + M_k)(H_k\check{P}_k H_k^T + H_k M_k + M_k^T H_k^T + R_k)^{-1} \\
\check{x}_k &= F_{k-1}\hat{x}_{k-1} + G_{k-1}u_{k-1} \\
\hat{x}_k &= \check{x}_k + K_k(y_k - H_k\check{x}_k) \\
\hat{P}_k &= (I - K_k H_k)\check{P}_k(I - K_k H_k)^T + K_k R_k K_k^T + \\
&\quad K_k(H_k M_k + M_k^T H_k^T)K_k^T - M_k K_k^T - K_k M_k^T \\
&= \check{P}_k - K_k(H_k\check{P}_k + M_k^T)
\end{aligned}
$$

Using the orthogonality principle

- initial estimate

$$\hat{x}_0 = E(x_0), \hat{P}_0 = E(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T$$

- the *a priori* estimate at time $k = 1$

$$\check{x}_1 = E(x_1) = F_0\hat{x}_0 + G_0u_0, \check{P}_1 = F_0\hat{P}_0F_0^T + Q_0$$

- Determine the *a posteriori* estimate at time $k = 1$ (the orthogonality principle): find $\hat{x}_1$ such that

$$E(\hat{e}_1 y_1^T) = 0$$

$$\hat{e}_1 = (I - K_1 H_1)\check{e}_1 + K_1 v_1$$

$$E(\hat{e}_1 y_1^T) = E[(I - K_1 H_1)\check{e}_1 + K_1 v_1][H_1 x_1 + v_1]^T = 0$$

$$E[(I - K_1 H_1)\check{e}_1 + K_1 v_1][H_1(\check{x}_1 - \check{e}_1) + v_1]^T = 0$$

$$E(\check{e}_1 v_1^T) = M_1, E(\check{e}_1 \check{x}_1^T) = 0, E(v_1 \check{x}_1^T) = 0$$

- Proof through mathematical induction

Example

Consider the following scalar system:

$$
\begin{aligned}
x_k &= 0.8x_{k-1} + w_{k-1} \\
y_k &= x_k + v_k \\
E[w_k w_j^T] &= 1 \cdot \delta_{k-j} \\
E[v_k v_j^T] &= 0.1 \cdot \delta_{k-j} \\
E[w_k v_j^T] &= M \cdot \delta_{k-j+1}
\end{aligned}
$$

	Standard Filter	Correlated Filter
Correlation M	(M=0 assumed)	(correct M used)
0	0.076	0.076
0.25	0.030	0.019
-0.25	0.117	0.052

Colored process noise

Suppose we have an LTI system given as

$$x_k = Fx_{k-1} + w_{k-1}$$

where the covariance of w_k is equal to Q_k. Further suppose that the process noise is the output of a dynamic system:

$$w_k = \psi w_{k-1} + \zeta_{k-1}$$

where ζ_{k-1} is zero-mean white noise that is uncorrelated with w_{k-1}. Hence, we have

$$E(w_k w_{k-1}^T) = E(\psi w_{k-1} w_{k-1}^T + \zeta_{k-1} w_{k-1}^T) = \psi Q_{k-1}$$

Augmenting the state

Suppose $x'_k = [x_k^T, w_k^T]^T$, we have

$$\begin{bmatrix} x_k \\ w_k \end{bmatrix} = \begin{bmatrix} F & I \\ 0 & \psi \end{bmatrix} \begin{bmatrix} x_{k-1} \\ w_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \zeta_{k-1} \end{bmatrix}$$

i.e.,

$$x'_k = F' x'_{k-1} + w'_{k-1}$$

This is an augmented system with a new state x', a new system matrix F' and a new process noise vector w' whose covariance is given as follows:

$$E(w'_k (w'_k)^T) = \begin{bmatrix} 0 & 0 \\ 0 & E(\zeta_k \zeta_k^T) \end{bmatrix} = Q'_k$$

- computational effort increases because the state dimension has doubled

Colored measurement noise

Now suppose that we have colored measurement noise. The system and measurement equations are given as

$$
\begin{aligned}
x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\
y_k &= H_k x_k + v_k \\
v_k &= \psi_{k-1}v_{k-1} + \eta_{k-1}, \\
w_k &\sim \mathcal{N}(0, Q_k), \quad v_k \sim \mathcal{N}(0, R_k) \\
E[w_k w_j^T] &= Q_k \delta_{k-j} \\
E[\eta_k \eta_j^T] &= Q_{\eta k} \delta_{k-j} \\
E[w_k \eta_j^T] &= 0
\end{aligned}
$$

The measurement noise is itself the output of a linear system with

$$E[v_k v_{k-1}^T] = E[(\psi_{k-1}v_{k-1} + \eta_{k-1})v_{k-1}^T] = \psi_{k-1}E[v_{k-1}v_{k-1}^T]$$

Augmenting the state

We augment the original system model as follows:

$$\begin{bmatrix} x_k \\ v_k \end{bmatrix} = \begin{bmatrix} F_{k-1} & 0 \\ 0 & \psi_{k-1} \end{bmatrix} \begin{bmatrix} x_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} w_{k-1} \\ \eta_{k-1} \end{bmatrix}$$

$$y_k = \begin{bmatrix} H_k & I \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + 0$$

This can be written as

$$x'_k = F'_{k-1} x'_{k-1} + w'_{k-1}$$

$$y_k = H'_k x'_k + v'_k$$

The process noise and measurement noise

$$
\begin{aligned}
E[w_k' {w_k'}^T] &= E\left[\begin{pmatrix} w_k \\ \eta_k \end{pmatrix}\begin{pmatrix} w_k^T & \eta_k^T \end{pmatrix}\right] = \begin{bmatrix} Q_k & 0 \\ 0 & Q_{\eta k} \end{bmatrix} \\
E[v_k' {v_k'}^T] &= 0
\end{aligned}
$$

- There is no measurement noise.
- practically speaking, a singular measurement-noise covariance often results in numerical problems

Measurement differencing

Define an auxiliary signal y'_k as follows:

$$y'_{k-1} = y_k - \psi_{k-1} y_{k-1}$$

Substitute for y_k and y_{k-1} in the above definition,

$$\begin{aligned}
y'_{k-1} &= H_k x_k + v_k - \psi_{k-1}(H_{k-1}x_{k-1} + v_{k-1}) \\
&= H_k(F_{k-1}x_{k-1} + w_{k-1}) + v_k - \psi_{k-1}(H_{k-1}x_{k-1} + v_{k-1}) \\
&= (H_k F_{k-1} - \psi_{k-1}H_{k-1})x_{k-1} + H_k w_{k-1} + v_k - \psi_{k-1}v_{k-1} \\
&= (H_k F_{k-1} - \psi_{k-1}H_{k-1})x_{k-1} + (H_k w_{k-1} + \eta_{k-1}) \\
&= H'_{k-1}x_{k-1} + v'_{k-1}
\end{aligned}$$

The equivalent system

$$
\begin{aligned}
x_k &= F_{k-1} x_{k-1} + w_{k-1} \\
y'_k &= H'_k x_k + v'_k
\end{aligned}
$$

The covariance of the new measurement noise v',

$$
\begin{aligned}
E[v'_k {v'_k}^T] &= E[(H_{k+1} w_k + \eta_k)(H_{k+1} w_k + \eta_k)^T] \\
&= H_{k+1} Q_k H_{k+1}^T + Q_{\eta_k} \\
E[w_k {v'_k}^T] &= E[w_k (H_{k+1} w_k + \eta_k)^T] \\
&= Q_k H_{k+1}^T
\end{aligned}
$$

Find the optimal K_k

- Define the a priori and a posteriori state estimates for the equivalent system as:

$$\check{x}_k = E[x_k | y_1, \ldots, y_k]$$

$$\hat{x}_k = E[x_k | y_1, \ldots, y_k, y_{k+1}] = \check{x}_k + K_k(y'_k - H'_k \check{x}_k)$$

  - this definition is slightly different, as $y'_{k-1} = y_k - \psi_{k-1} y_{k-1}$

- Choose the gain K_k to minimize the trace of the covariance of the estimation error:

$$K_k = \text{argmin Tr } E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$$

Example

Consider the linear system with colored measurement noise

$$
\begin{aligned}
x_k &= \begin{bmatrix} 0.70 & -0.15 \\ 0.03 & 0.79 \end{bmatrix} x_{k-1} + \begin{bmatrix} 0.15 \\ 0.21 \end{bmatrix} w_{k-1} \\
y_k &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k \\
v_k &= \psi v_{k-1} + \zeta_{k-1} \\
E[w_k w_j^T] &= 1 \cdot \delta_{k-j} \\
E[\zeta_k \zeta_j^T] &= \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix} \delta_{k-j} \\
E[w_k \zeta_j^T] &= 0
\end{aligned}
$$

Color ψ	Standard Filter	Augmented Filter	Measurement Differencing
0	0.245	0.245	0.247
0.2	0.260	0.258	0.259
0.5	0.308	0.294	0.295
0.9	0.631	0.407	0.406

Contents

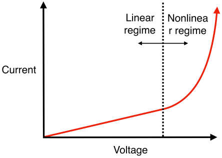
Motivation

- All systems are ultimately nonlinear (Even a device as simple as a resistor is only approximately linear, and even then only in a limited range of operation)
- Many systems are close enough to linear that linear estimation approaches give satisfactory results
- However, there is some system does not behave linearly even over a small range of operation, and our linear approaches for estimation no longer give good results
- Then we need to explore nonlinear estimators
- Some nonlinear estimation methods including nonlinear extensions of the Kalman filter, unscented filtering, and particle filtering have become widespread.

Nonlinear Kalman Filtering

Linear systems do not exist in reality!

Ohm's Law: $I = V/R$



How can we adapt the Kalman Filter to *nonlinear* discrete-time systems?

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$$

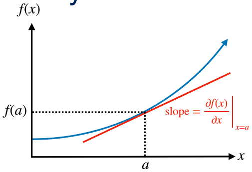
current state · previous state · inputs · process noise

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k)$$

measurement · state · measurement noise

EKF | Linearizing a Nonlinear System

Choose an operating point *a* and approximate the nonlinear function by a tangent line at that point



Mathematically, we compute this linear approximation using a first-order Taylor expansion:

$$f(x) \approx f(a) + \left.\frac{\partial f(x)}{\partial x}\right|_{x=a} (x-a) + \frac{1}{2!}\left.\frac{\partial^2 f(x)}{\partial x^2}\right|_{x=a} (x-a)^2 + \frac{1}{3!}\left.\frac{\partial^2 f(x)}{\partial x^2}\right|_{x=a} (x-a)^3 + \ldots$$

First-order terms

Higher-order terms

The discrete-time extended Kalman filter

Suppose we have the system model

$$
\begin{aligned}
x_k &= f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \\
y_k &= h_k(x_k, v_k) \\
w_k &\sim \mathcal{N}(0, Q_k) \\
v_k &\sim \mathcal{N}(0, R_k)
\end{aligned}
$$

Linearization

We perform a Taylor series expansion of the state equation around $x_{k-1} = \hat{x}_{k-1}$ and $w_{k-1} = 0$ to obtain the following:

$$\begin{aligned} x_k &\approx f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0) + \frac{\partial f_{k-1}}{\partial x}\Big|_{(\hat{x}_{k-1},0)}(x_{k-1} - \hat{x}_{k-1}) \\ &\quad + \frac{\partial f_{k-1}}{\partial w}\Big|_{(\hat{x}_{k-1},0)} w_{k-1} \\ &= f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0) + F_{k-1}(x_{k-1} - \hat{x}_{k-1}) + L_{k-1}w_{k-1} \\ &= F_{k-1}x_{k-1} + [f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0) - F_{k-1}\hat{x}_{k-1}] + L_{k-1}w_{k-1} \\ &= F_{k-1}x_{k-1} + \tilde{u}_{k-1} + \tilde{w}_{k-1} \end{aligned}$$

where $F_{k-1} = \frac{\partial f_{k-1}}{\partial x}\Big|_{(\hat{x}_{k-1},0)}$, $L_{k-1} = \frac{\partial f_{k-1}}{\partial w}\Big|_{(\hat{x}_{k-1},0)}$. The input is $\tilde{u}_k = f_k(\hat{x}_k, u_k, 0) - F_k\hat{x}_k$. The process noise $\tilde{w}_k \sim \mathcal{N}(0, L_kQ_kL_k^T)$.

Linearization

We linearize the measurement equation around $x_k = \check{x}_k$ and $v_k = 0$ to obtain

$$\begin{aligned} y_k &\approx h_k(\check{x}_k, 0) + \tfrac{\partial h_k}{\partial x}|_{(\check{x}_k,0)}(x_k - \check{x}_k) + \tfrac{\partial h_k}{\partial v}|_{(\check{x}_k,0)} v_k \\ &= H_k x_k + [h_k(\check{x}_k, 0) - H_k \check{x}_k] + M_k v_k \\ &= H_k x_k + z_k + \tilde{v}_k \end{aligned}$$

where $H_k = \frac{\partial h_k}{\partial x}|_{(\check{x}_k,0)}$ and $M_k = \frac{\partial h_k}{\partial v}|_{(\check{x}_k,0)}$. The signal z_k and the noise signal $\tilde{v}_k$ are defined as

$$\begin{aligned} z_k &= h_k(\check{x}_k, 0) - H_k \check{x}_k \\ \tilde{v}_k &\sim \mathcal{N}(0, M_k R_k M_k^T) \end{aligned}$$

The discrete-time extended Kalman filter

1. The system and measurement equations are given as follows:

$$
\begin{aligned}
x_k &= f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \\
y_k &= h_k(x_k, v_k) \\
w_k &\sim \mathcal{N}(0, Q_k) \\
v_k &\sim \mathcal{N}(0, R_k)
\end{aligned}
$$

2. Initialize the filter as follows:

$$
\begin{aligned}
\hat{x}_0 &= E(x_0) \\
\hat{P}_0 &= E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]
\end{aligned}
$$

The discrete-time extended Kalman filter

3. For $k = 1, 2, \ldots,$ perform the following.

   - compute the partial derivative matrices:

$$\begin{aligned} F_{k-1} &= \frac{\partial f_{k-1}}{\partial x}\Big|_{(\hat{x}_{k-1},0)} \\ L_{k-1} &= \frac{\partial f_{k-1}}{\partial w}\Big|_{(\hat{x}_{k-1},0)} \end{aligned}$$

   - perform the time update

$$\begin{aligned} \check{P}_k &= F_{k-1}\hat{P}_{k-1}F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T \\ \check{x}_k &= f_{k-1}(\hat{x}_{k-1}, u_{k-1}, 0) \end{aligned}$$

   - compute the partial derivative matrices:

$$\begin{aligned} H_k &= \frac{\partial h_k}{\partial x}\Big|_{(\check{x}_k,0)} \\ M_k &= \frac{\partial h_k}{\partial v}\Big|_{(\check{x}_k,0)} \end{aligned}$$

The discrete-time extended Kalman filter

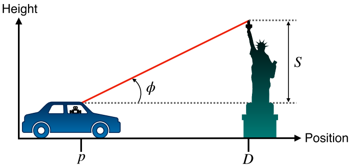
- perform the measurement update

$$\begin{aligned} K_k &= \check{P}_k H_k^T (H_k \check{P}_k H_k^T + M_k R_k M_k^T)^{-1} \\ \hat{x}_k &= \check{x}_k + K_k [y_k - h_k(\check{x}_k, 0)] \\ \hat{P}_k &= (I - K_k H_k) \check{P}_k \end{aligned}$$

Remark

- **The key to the EKF** lies in the **linearization of the original nonlinear dynamic system.**
- if the linearization does not provide a reasonably accurate description of the system dynamics, the state estimates may diverge.
- The computation of partial derivative matrices as well as the error covariance requires the estimates $\hat{x}_k$ and $\check{x}_k$. As a result, the EKF can not be tested off-line; it requires real or simulated data.

EKF | Short Example



$$\mathbf{x} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix} \qquad \mathbf{u} = \ddot{p}$$

S and *D* are known in advance

Motion/Process model

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1} \end{aligned}$$

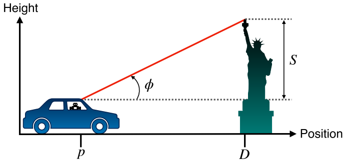
Landmark measurement model

$$\begin{aligned} y_k = \phi_k &= h(p_k, v_k) \\ &= \tan^{-1}\left(\frac{S}{D - p_k}\right) + v_k \end{aligned}$$

Noise densities

$$v_k \sim \mathcal{N}(0,\ 0.01) \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, (0.1)\mathbf{1}_{2\times 2})$$

EKF | Short Example



Motion model Jacobians

$$\mathbf{F}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1},\mathbf{u}_{k-1},\mathbf{0}} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

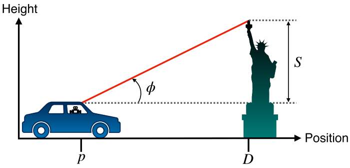
$$\mathbf{L}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{w}_{k-1}}\right|_{\hat{\mathbf{x}}_{k-1},\mathbf{u}_{k-1},\mathbf{0}} = \mathbf{1}_{2\times 2}$$

Measurement model Jacobians

$$\mathbf{H}_{k} = \left.\frac{\partial h}{\partial \mathbf{x}_{k}}\right|_{\check{\mathbf{x}}_{k},\mathbf{0}} = \begin{bmatrix} \frac{S}{(D-\check{p}_k)^2 + S^2} & 0 \end{bmatrix}$$

$$M_{k} = \left.\frac{\partial h}{\partial v_{k}}\right|_{\check{\mathbf{x}}_{k},\mathbf{0}} = 1$$

EKF | Short example



<u>Data</u>

$$\hat{\mathbf{x}}_0 \sim \mathcal{N}\left(\begin{bmatrix}0\\5\end{bmatrix}, \begin{bmatrix}0.01 & 0\\0 & 1\end{bmatrix}\right)$$

$\Delta t = 0.5\text{s}$

$u_0 = -2\ [m/s^2]$ $\quad y_1 = 30\ [\text{deg}]$

$S = 20\ [m]$ $\quad D = 40\ [m]$

Using the Extended Kalman Filter equations, what is our updated position?

$\hat{p}_1$

EKF | Short Example Solution

Prediction

$$\check{\mathbf{x}}_1 = \mathbf{f}_0(\hat{\mathbf{x}}_0, \mathbf{u}_0, \mathbf{0})$$

$$\begin{bmatrix} \check{p}_1 \\ \check{\dot{p}}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}(-2) = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix}$$

This is the same result as in the linear Kalman Filter example because the motion model is already linear!

$$\check{\mathbf{P}}_1 = \mathbf{F}_0\hat{\mathbf{P}}_0\mathbf{F}_0^T + \mathbf{L}_0\mathbf{Q}_0\mathbf{L}_0^T$$

$$\check{\mathbf{P}}_1 = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.5 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

EKF | Short example solution

Correction

$$\mathbf{K}_1 = \check{\mathbf{P}}_1\mathbf{H}_1^T(\mathbf{H}_1\check{\mathbf{P}}_1\mathbf{H}_1^T + \mathbf{M}_1\mathbf{R}_1\mathbf{M}_1^T)^{-1}$$

$$= \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 0.011 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 0.011 & 0 \end{bmatrix} \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} 0.011 \\ 0 \end{bmatrix} + 1(0.01)(1) \right)^{-1}$$

$$= \begin{bmatrix} 0.39 \\ 0.55 \end{bmatrix}$$

$$\hat{\mathbf{x}}_1 = \check{\mathbf{x}}_1 + \mathbf{K}_1(\mathbf{y}_1 - \mathbf{h}_1(\check{\mathbf{x}}_1, \mathbf{0}))$$

Bonus!

$$\hat{\mathbf{P}}_1 = (\mathbf{1} - \mathbf{K}_1\mathbf{H}_1)\check{\mathbf{P}}_1$$

$$= \begin{bmatrix} 0.3585 & 0.4979 \\ 0.4978 & 1.0970 \end{bmatrix}$$

$$\begin{bmatrix} \hat{p}_1 \\ \hat{\dot{p}}_1 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 4 \end{bmatrix} + \begin{bmatrix} 0.39 \\ 0.55 \end{bmatrix} (0.52 - 0.49) = \begin{bmatrix} 2.51 \\ 4.02 \end{bmatrix}$$

Linear parameters identification

Consider a single-input single-output discrete-time system given by the difference equation

$$y_k = f(y_{k-1}, \ldots, y_{k-n_a}, u_{k-1}, \ldots, u_{k-n_b})$$

Approximate it by a linear relationship

$$y_k = \sum_{i=1}^{n_a} a_i y(n-i) + \sum_{i=1}^{n_b} b_i u(n-i) + v_k$$

where v_k is the model error term.

State-space formulation

The linear relationship can be expressed as

$$\begin{aligned} x_{k+1} &= x_k \\ y_k &= c_k x_k + v_k \end{aligned}$$

where

$$x_k = [a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}]^T$$

and

$$c_k = [y_{k-1}, \dots, y_{k-n_a}, u_{k-1}, \dots, u_{k-n_b}]$$

- Then we can use linear discrete-time Kalman filter to estimate x_k, which is actually the system parameters.
- Recall what we have learned in recursive least squares!
- Compare with system identification method you have learned.

Nonlinear system identification

- if the function f is nonlinear, an accurate model can be constructed by using a neural network
- For a simple 2 neurons neural network model

$$\begin{aligned} y_k &= c_1\chi_{1,k} + c_2\chi_{2,k} + v_k \\ \chi_{j,k} &= \mathrm{act}[a_{1j}y_{k-1} + b_{1j}u_k], j = 1, 2 \end{aligned}$$

  where act denotes the activation function.
- The equation for a feedforward NN model with q hidden nodes are

$$y_k = \sum_{j=1}^{q} c_j\chi_{j,k} + v_k$$

$$\chi_{j,k} = \mathrm{act}\left[\sum_{i=1}^{N} a_{ij}y_{k-i} + \sum_{i=0}^{M} b_{ij}u_{k-i}\right], j = 1, \ldots, q$$

State space formulation

Define

$$\theta_j = [a_{1j}, \ldots, a_{Nj}, b_{0j}, \ldots, b_{Mj}]^T \;\; c = [c_1, \ldots, c_q]^T$$

Denote ϕ_k as the $(N + M - 1)$-element vector,

$$\phi_k = [y_{k-1}, \ldots, y_{k-N}, u_k, \ldots, u_{k-M}]^T,$$

we have

$$y_k = \sum_{j=1}^{q} c_j \mathrm{act}[\phi_k^T \theta_j] + v_k$$

Let x_k denote the $q(N + M + 2)$-element state vector defined by

$$x_k = [\theta_1, \ldots, \theta_q, c^T]^T$$

State space formulation

The state space formulation can be obtained,

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

where

$$\gamma(x_k) = \sum_{j=1}^{q} c_j \mathrm{act}[\phi_k^T \theta_j]$$

Time update (Propagation)

$$
\begin{aligned}
\check{x}_{k+1} &= \hat{x}_k \\
\check{P}_{k+1} &= \hat{P}_k
\end{aligned}
$$

Measurement update (Correction)

- Compute the partial derivative matrices:

$$H_k = \frac{\partial \gamma}{\partial x}|_{\check{x}_k} = \begin{bmatrix} \frac{\partial \gamma}{\partial \theta_1} & \dots & \frac{\partial \gamma}{\partial \theta_q} & \frac{\partial \gamma}{\partial c} \end{bmatrix}^T |_{\check{x}_k}$$

  while

$$\frac{\partial \gamma}{\partial \theta_j} = c_j \frac{\partial \mathrm{act}(\eta)}{\partial \eta} \phi_k^T, j = 1, \dots, q, \eta = \phi_k^T \theta_j$$

  and

$$\frac{\partial \gamma}{\partial c} = \begin{bmatrix} \mathrm{act}(\phi_k^T \theta_1) & \dots & \mathrm{act}(\phi_k^T \theta_q) \end{bmatrix}$$

- Update the matrices

$$\begin{aligned} K_k &= \check{P}_k H_k^T (H_k \check{P}_k H_k^T + R_k)^{-1} \\ \hat{x}_k &= \check{x}_k + K_k [y_k - \gamma(\check{x}_k, 0)] \\ \hat{P}_k &= (I - K_k H_k) \check{P}_k \end{aligned}$$

- Comparison with what we have learned in recursive least squares estimation?
- Comparison with the dominant method for neural network training?

Insufficiency of EKF

- The first-order approximation of the dynamic system, however, can introduce large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to sub-optimal performance and sometimes divergence of the filter.
- the UKF approximates the probability distribution instead of the system dynamics

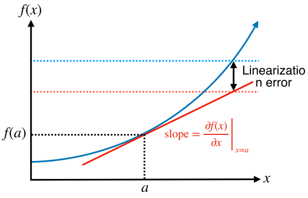
Limitations of the EKF | Linearization error

The EKF works by *linearizing* the nonlinear motion and measurement models to update the mean and covariance of the state

The difference between the linear approximation and the nonlinear function is called *linearization error*

In general, linearization error depends on

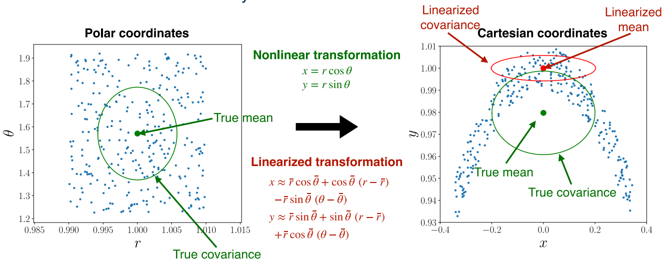
1. How nonlinear the function is
2. How far away from the operating point the linear approximation is being used



$$f(x) \approx f(a) + \left.\frac{\partial f(x)}{\partial x}\right|_{x=a} (x - a)$$

Linearization Error | Example

Let's look at an example of how linearization error affects the mean and covariance of a random variable transformed by a nonlinear function:



Motivation of unscented transform

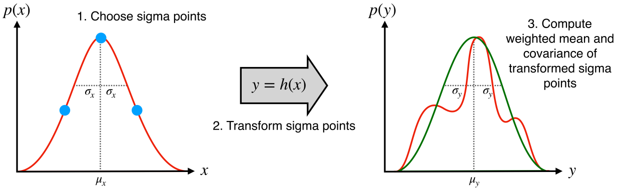
- In 1994 Jeffrey Uhlmann noted that the EKF takes a nonlinear function and partial distribution information of the state of a system but applies an approximation to the known function rather than to the imprecisely-known probability distribution
- He suggested that a better approach would be to use the exact nonlinear function applied to an approximating probability distribution
- Jeffrey Uhlmann explained that "unscented" was an arbitrary name that he adopted to avoid it being referred to as the “Uhlmann filter".

Idea of Unscented Transform

- The unscented transformation is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation
- uses the intuition (which also applies to the particle filter) that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation.

The Unscented Transform

"It is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function" — S. Julier, J. Uhlmann, and H. Durrant-Whyte (2000)



Idea of Unscented Transform: A simple example

Linearization based Gaussian approximation:

- Problem: Determine the mean and covariance of y:

$$\mathbf{x} \sim \mathcal{N}(\mu, \sigma^2)$$
$$\mathbf{y} = \sin(\mathbf{x})$$

- Linearization based approximation:

$$\mathbf{y} = \sin(\mu) + \frac{\partial \sin(\mu)}{\partial \mu}(\mathbf{x} - \mu) + \dots$$

which gives

$$E(\mathbf{y}) \approx E(\sin(\mu) + \cos(\mu)(\mathbf{x} - \mu)) = \sin(\mu)$$
$$Cov(\mathbf{y}) \approx E[(\sin(\mu) + \cos(\mu)(\mathbf{x} - \mu) - \sin(\mu))^2] = \cos^2(\mu)\sigma^2$$

A simple example

- Form 3 sigma points as follows:

$$X_0 = \mu, X_1 = \mu + \sigma, X_2 = \mu - \sigma$$

- We may now select some weights W_0, W_1, W_2 such that the original mean and covariance can be always recovered by

$$\mu = W_0 X_0 + W_1 X_1 + W_2 X_2$$
$$\sigma^2 = \sum_{i=0}^{2} W_i (X_i - \mu)^2$$

- Approximating the distribution of $y = \sin(x)$ as follows:

$$\mu_y = \sum_{i=0}^{2} W_i \sin(X_i)$$
$$\sigma_y^2 = \sum_{i=0}^{2} W_i (\sin(X_i) - \mu_y)^2$$

A simple example

- Set $W_0 = 0, W_1 = W_2 = \frac{1}{2}$
- We get

$$\mu_y \approx \sin\mu\cos\sigma, \sigma_y^2 \approx \cos^2\mu\sin^2\sigma$$

- Compare with the first-order linearization

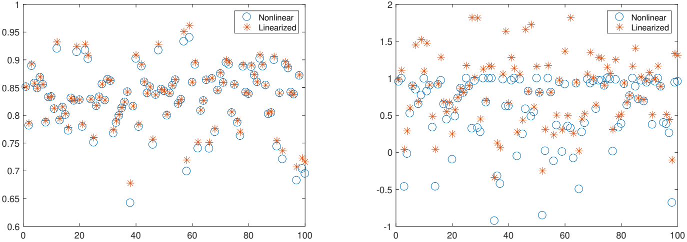


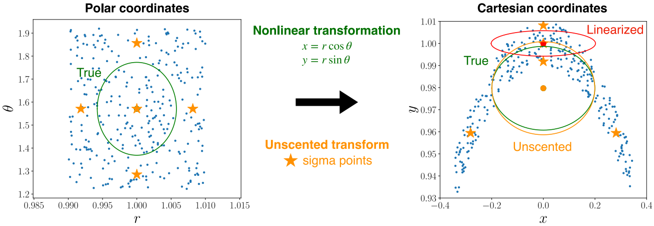
Figure: Comparison of the linearized transformation and unscented transformation.Left: $\mu = 1, \sigma = 0.1$. Right: $\mu = 1, \sigma = 1$.

Table: The mean and variance for the two cases

	$\sigma = 0.1$			$\sigma = 1$		
	nonlinear	linearized	unscented	nonlinear	linearized	unscented
mean	0.8312	0.8355	0.8338	0.5839	0.8499	0.4744
variance	0.0573	0.0549	0.0508	0.4712	0.4720	0.4366

The Unscented Transform vs. Linearization

Let's revisit our nonlinear transformation example from the previous video.



The Unscented Transform gives a much better approximation for similar work!

Principle of Unscented Transform

1. For vectors $x \sim \mathcal{N}(m, P)$, the generalization of standard deviation σ is the Cholesky factor $L = \sqrt{P}$:

$$P = LL^T$$

2. The $(2n+1)$ sigma points can be formed using **columns** of L:

$$X_0 = m$$
$$X_i = m + \sqrt{n+\lambda} L_i$$
$$X_{n+i} = m - \sqrt{n+\lambda} L_i$$

where $[]_i$ denotes the i-th column of the matrix.

3. For transformation $y = g(x)$ the approximation is:

$$E[g(x)] = \sum_{i=0}^{2n} W_i^{(m)} g(X_i)$$
$$Cov[g(x)] = \sum_{i=0}^{2n} W_i^{(c)} (g(X_i) - \mu_y)(g(X_i) - \mu_y)^T.$$

Parameter setting

- λ is a scaling parameter defined as $\lambda = \alpha^2(n+\kappa) - n$
- α and κ determine the spread of the sigma points.
- Weights $W_i^{(m)}$ and $W_i^{(c)}$ are given as follows:

$$W_0^{(m)} = \frac{\lambda}{n+\lambda}, W_0^{(c)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n+\lambda)}, i = 1, \ldots, 2n$$

- β can be used for incorporating priori information on the (non-Gaussian) distribution of x.

Unscented transform approximation of nonlinear augmented function

The unscented transform approximation to the joint distribution of $\mathbf{x}$ and $\mathbf{y} = g(\mathbf{x}) + \mathbf{q}$ where $\mathbf{x} \sim \mathcal{N}(m, P)$ and $\mathbf{q} \sim \mathcal{N}(0, Q)$ is

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m \\ \mu \end{pmatrix}, \begin{pmatrix} P & C \\ C^T & S \end{pmatrix} \right)$$

The sub-matrices are formed as follows:

- Form the set of $(2n+1)$ sigma points as follows:

$$\begin{aligned} X_0 &= m, \\ X_i &= m + \sqrt{n+\lambda}\ L_i, \\ X_{n+i} &= m - \sqrt{n+\lambda}\ L_i, i = 1, \ldots, n \end{aligned}$$

Unscented transform approximation of nonlinear transforms

- Propagate the sigma points through $g(\cdot)$:

$$Y_i = g(X_i), i = 0, \ldots, 2n$$

- The sub-matrices are then given as:

$$\begin{aligned}
\mu &= \sum_{i=0}^{2n} W_i^{(m)} Y_i \\
S &= \sum_{i=0}^{2n} W_i^{(c)} (Y_i - \mu)(Y_i - \mu)^T + Q \\
C &= \sum_{i=0}^{2n} W_i^{(c)} (X_i - m)(Y_i - \mu)^T
\end{aligned}$$

Unscented Kalman Filter (UKF): Derivation

- Assume that the filtering distribution of previous step is Gaussian

$$p(x_{k-1}|y_{1:k-1}) \approx \mathcal{N}(m_{k-1}, P_{k-1})$$

- The joint distribution of x_{k-1} and $x_k = f(x_{k-1}) + q_{k-1}$ can be approximated with UT as Gaussian

$$p(x_{k-1}, x_k|y_{1:k-1}) \approx \mathcal{N}\left(\left[\begin{array}{c} x_{k-1} \\ x_k \end{array}\right] \Big| \left(\begin{array}{c} m'_1 \\ m'_2 \end{array}\right), \left(\begin{array}{cc} P'_{11} & P'_{12} \\ (P'_{12})^T & P'_{22} \end{array}\right)\right)$$

- Form the sigma points X_i of $x_{k-1} \sim \mathcal{N}(m_{k-1}, P_{k-1})$ and compute the transformed sigma points as $\hat{X}_i = f(X_i)$.

Unscented Kalman Filter (UKF): Derivation

- The expected values can now be expressed as

$$m'_1 = m_{k-1}$$
$$m'_2 = \sum_i W_i^{(m)} \hat{X}_i$$

- The blocks of covariance can be expressed as:

$$P'_{11} = P_{k-1}$$
$$P'_{12} = \sum_i W_i^{(c)} (X_i - m_{k-1})(\hat{X}_i - m'_2)^T$$
$$P'_{22} = \sum_i W_i^{(c)} (\hat{X}_i - m'_2)(\hat{X}_i - m'_2)^T + Q_{k-1}$$

Unscented Kalman Filter (UKF): Derivation

- The prediction mean and covariance of x_k are then m'_2 and P'_{22} and thus we get

$$\check{m}_k = \sum_i W_i^{(m)} \hat{X}_i$$
$$\check{P}_k = \sum_i W_i^{(c)} (\hat{X}_i - \check{m}_k)(\hat{X}_i - \check{m}_k)^T + Q_{k-1}$$

- For the joint distribution of x_k and $y_k = h(x_k) + r_k$ we similarly get

$$p(x_k, y_k | y_{1:k-1}) \approx \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix} \middle| \begin{pmatrix} m''_1 \\ m''_2 \end{pmatrix}, \begin{pmatrix} P''_{11} & P''_{12} \\ (P''_{12})^T & P''_{22} \end{pmatrix}\right)$$

Unscented Kalman Filter (UKF): Derivation

- If $\check{X}_i$ are the sigma points of $x_k \sim \mathcal{N}(\check{m}_k, \check{P}_k)$ and $\hat{Y}_i = h(\check{X}_i)$, we get

$$
\begin{aligned}
m_1'' &= \check{m}_k \\
m_2'' &= \sum_i W_i^{(m)} \hat{Y}_i \triangleq \mu_k \\
P_{11}'' &= \check{P}_k \\
P_{12}'' &= \sum_i W_i^{(c)} (\check{X}_i - \check{m}_k)(\hat{Y}_i - m_2'')^T \triangleq C_k \\
P_{22}'' &= \sum_i W_i^{(c)} (\hat{Y}_i - m_2'')(\hat{Y}_i - m_2'')^T + R_k \triangleq S_k
\end{aligned}
$$

Unscented Kalman Filter (UKF): Derivation

- Recall that if

$$\begin{pmatrix} x \\ y \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} A & C \\ C^T & B \end{pmatrix} \right)$$

then

$$x|y \sim \mathcal{N}(a + CB^{-1}(y-b), A - CB^{-1}C^T)$$

- Thus we get the conditional mean and covariance

$$\hat{m}_k = \check{m}_k + P''_{12}(P''_{22})^{-1}(y_k - m''_{22}) = \check{m}_k + C_k S_k^{-1}(y_k - \mu_k)$$
$$\hat{P}_k = \check{P}_k - P''_{12}(P''_{22})^{-1}(P''_{12})^T = \check{P}_k - C_k S_k^{-1} C_k^T$$

Unscented Kalman Filter (UKF)

- Prediction step
  - Form the matrix of sigma points:

$$X_{k-1} = [\ \hat{x}_{k-1} \quad \cdots \quad \hat{x}_{k-1}\ ] + \sqrt{n+\lambda} \begin{bmatrix} 0 & \sqrt{\hat{P}_{k-1}} & -\sqrt{\hat{P}_{k-1}} \end{bmatrix}$$

  - Propagate the sigma points through the dynamic model:

$$\hat{X}_{k,i} = f(X_{k-1,i}), i = 0, 1, \ldots, 2n$$

  - Compute the predicted mean and covariance

$$\check{x}_k = \sum_i W_i^{(m)} \hat{X}_{k,i}$$
$$\check{P}_k = \sum_i W_i^{(c)} (\hat{X}_{k,i} - \check{x}_k)(\hat{X}_{k,i} - \check{x}_k)^T + Q_{k-1}$$

Unscented Kalman Filter (UKF)

- Update step
  - Form the matrix of sigma points:

$$\check{X}_k = [\ \check{x}_k \quad \cdots \quad \check{x}_k\ ] + \sqrt{n+\lambda}[\ 0 \quad \sqrt{\check{P}_k} \quad -\sqrt{\check{P}_k}\ ]$$

  - Propagate the sigma points through the measurement model:

$$\hat{Y}_{k,i} = h(\check{X}_{k,i}), i = 0, 1, \ldots, 2n$$

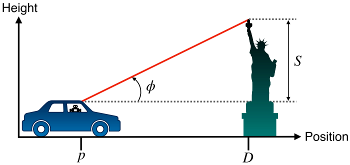
  - Compute the following terms:

$$\mu_k = \sum_i W_i^{(m)} \hat{Y}_{k,i},\ S_k = \sum_i W_i^{(c)} (\hat{Y}_{k,i} - \mu_k)(\hat{Y}_{k,i} - \mu_k)^T + R_k$$
$$C_k = \sum_i W_i^{(c)} (\check{X}_{k,i} - \check{x}_k)(\hat{Y}_{k,i} - \mu_k)$$

  - Compute the filter gain K_k and the filtered state mean m_k and covariance $\hat{P}_k$, conditional to the measurement y_k:

$$K_k = C_k S_k^{-1},\ \hat{x}_k = \check{x}_k + K_k(y_k - \mu_k)$$
$$\hat{P}_k = \check{P}_k - K_k S_k K_k^T$$

UKF | Short example



$$\mathbf{x} = \begin{bmatrix} p \\ \dot{p} \end{bmatrix} \qquad \mathbf{u} = \ddot{p}$$

S and *D* are known in advance

Motion/Process model

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1})$$
$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

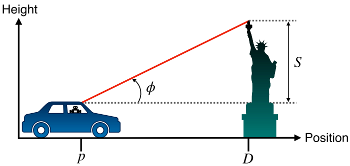
Landmark measurement model

$$y_k = \phi_k = h(p_k, v_k)$$
$$= \tan^{-1}\left(\frac{S}{D - p_k}\right) + v_k$$

Noise densities

$$v_k \sim \mathcal{N}(0,\ 0.01) \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, (0.1)\mathbf{1}_{2\times 2})$$

UKF | Short example



<u>Data</u>

$$\hat{\mathbf{x}}_0 \sim \mathcal{N}\left(\begin{bmatrix}0\\5\end{bmatrix}, \begin{bmatrix}0.01 & 0\\0 & 1\end{bmatrix}\right)$$

$\Delta t = 0.5\text{s}$

$u_0 = -2\ [m/s^2]$ $\quad y_1 = 30\ [\text{deg}]$

$S = 20\ [m]$ $\quad D = 40\ [m]$

Using the Unscented Kalman Filter equations, what is our updated position?
$\hat{p}_1$

UKF: Short example solution

Prediction

- $n = 2$, choose $\lambda = 1$
- $\sqrt{\hat{P}_0} = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}$
- choose 5 sigma points
  - $\hat{x}_0^{(0)} = \hat{x}_0, \hat{x}_0^{(i)} = \hat{x}_0 + \sqrt{3}[\sqrt{\hat{P}_0}]_i, i = 1, 2$
  - $\hat{x}_0^{(i+2)} = \hat{x}_0 - \sqrt{3}[\sqrt{\hat{P}_0}]_i, i = 1, 2$
  - $\hat{x}_0^{(0)} = [0, 5]^T, \hat{x}_0^{(1)} = [0.2, 5]^T, \hat{x}_0^{(2)} = [0, 6.7]^T, \hat{x}_0^{(3)} = [-0.2, 5]^T, \hat{x}_0^{(4)} = [0, 3.3]^T$

UKF: Short example solution

Prediction

- $\check{x}_1^{(i)} = f_0(\hat{x}_0^{(i)}, u_0, 0), i = 0, 1, \ldots, 4$
- $\check{x}_1^{(0)} = [2.5, 4]^T, \check{x}_1^{(1)} = [2.7, 4]^T, \check{x}_1^{(2)} = [3.4, 5.7]^T, \check{x}_1^{(3)} = [2.3, 4]^T, \check{x}_1^{(4)} = [1.6, 2.3]^T$
- $W_0^{(m)} = W_0^{(c)} = 1/3, W_i^{(m)} = W_i^{(c)} = 1/6, i = 1, \ldots, 4$
- $\check{x}_1 = \sum_{i=0}^{4} W_i \check{x}_1^{(i)} = [2.5, 4]^T$
- $\check{P}_k = \sum_{i=0}^{4} W_i (\check{x}_k^i - \check{x}_k)(\check{x}_k^i - \check{x}_k)^T + Q_{k-1} = \begin{bmatrix} 0.36 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$

UKF: Short example solution

Correction

- $\sqrt{\check{P}_1} = \begin{bmatrix} 0.51 & 0 \\ 0.98 & 0.20 \end{bmatrix}$
- choose 5 sigma points
  - $\check{x}_1^{(0)} = \check{x}_1, \check{x}_1^{(i)} = \check{x}_1 + \sqrt{3}[\sqrt{\check{P}_1}]_i, i = 1, 2$
  - $\check{x}_1^{(i+2)} = \check{x}_1 - \sqrt{3}[\sqrt{\check{P}_1}]_i, i = 1, 2$
  - $\check{x}_1^{(0)} = [2.5, 4]^T, \check{x}_1^{(1)} = [3.54, 5.44]^T, \check{x}_1^{(2)} = [2.5, 5.10]^T, \check{x}_1^{(3)} = [1.46, 2.56]^T, \check{x}_1^{(4)} = [2.5, 2.90]^T$
- the output $\hat{y}_1^{(i)} = h_1(\check{x}_1^{(i)}, 0), i = 0, \ldots, 2n$
  - $\hat{y}_1^{(0)} = 28.1, \hat{y}_1^{(1)} = 28.7, \hat{y}_1^{(2)} = 28.1, \hat{y}_1^{(3)} = 27.4, \hat{y}_1^{(4)} = 28.1$

UKF: Short example solution

Correction

- $\mu_1 = \sum_{i=0}^{2n} W_i^{(m)} \hat{y}_1^{(i)} = 28.1$
- $S_1 = \sum_{i=0}^{2n} W_i^{(c)} (\hat{y}_k^{(i)} - \mu_k)(\hat{y}_k^{(i)} - \mu_k)^T + R_k = 0.16$
- $C_1 = \sum_{i=0}^{2n} W_c^{(i)} (\check{x}_k^{(i)} - \check{x}_k)(\hat{y}_k^{(i)} - \mu_k)^T = [0.23, 0.32]^T$
- $K_1 = C_1 S_1^{-1} = [1.47, 2.05]^T$
- $\hat{x}_1 = \check{x}_1 + K_1(y_1 - \mu_1) = [5.33, 7.93]^T$
- $\hat{P}_1 = \check{P}_1 - K_1 S_1 K_1^T = \begin{bmatrix} 0.0143 & 0.0178 \\ 0.0178 & 0.4276 \end{bmatrix}$

Comparison of EKF and UKF

- Local approximation vs larger area approximation
- Require differentiability of F and h vs not require
- Closed form derivatives or expectations vs no such forms are needed
- First order approximation of the nonlinear dynamics vs captures higher order moments of distribution

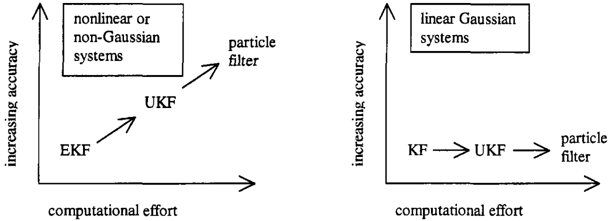
Disadvantages of UKF

- Not a truly global approximation, based on a small set of trial points.
- Does not work well with nearly singular covariances, i.e., with nearly deterministic systems.
- Requires more computations than EKF, e.g., Cholesky factorizations on every step.
- Can **only** be applied to models **driven by Gaussian noises**

Introduction of particle filter

- In a linear system with Gaussian noise, the Kalman filter is optimal.
- In a system that is **nonlinear**, the Kalman filter can be used for state estimation, but the **particle filter** may give **better** results at the price of **additional computational effort.**
- In a system that has **non-Gaussian noise**, the Kalman filter is the **optimal linear filter,** but again the **particle filter may perform better.**
- The UKF provides a **balance** between the **low computational effort** of the Kalman filter and the **high performance** of the particle filter.

An illustration



Introduction of particle filter

- The particle filter has some similarities with the UKF in that it transforms **a set of points** via known nonlinear equations and combines the results to estimate the mean and covariance of the state.
- However, in the particle filter the points are chosen **randomly,** whereas in the UKF the points are chosen on the basis of a specific algorithm (**unscented transform**).
- the number of points used in a particle filter generally needs to be much **greater** than the number of points in a UKF.
- the estimation error in a UKF does not converge to zero in any sense, but the **estimation error in a particle filter does converge to zero as the number of particles (and hence the computational effort) approaches infinity**

Particle filtering methods–Sequential importance sampling algorithm

- The sequential importance sampling (SIS) algorithm is a Monte Carlo (MC) method that forms the basis for most sequential MC filters developed over the past decades.
- Let $\{\mathbf{x}_{0:k}^i, \omega_k^i\}$ denote a random measure that characterizes the posterior pdf $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, where $\{\mathbf{x}_{0:k}^i, i = 0, \ldots, N_s\}$ is a set of support points with associated weights $\{w_k^i, i = 1, \ldots, N_s\}$ and $\mathbf{x}_{0:k} = \{\mathbf{x}_j, j = 0, \ldots, k\}$ is the set of all states up to time k. The posterior density at time k can be approximated as

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i),$$

in which $\sum_{i=1}^{N_s} \omega_k^i = 1$ and δ is the Kronecker delta function.

Sequential importance sampling algorithm

- The weights ω_k^i are chosen using the principle of importance sampling.
  - Suppose $p(x) \propto \pi(x)$ is a probability density from which it is difficult to draw samples but for which $\pi(x)$ can be evaluated.
  - Let $x^i \sim q(x), i = 1, \ldots, N_s$ be samples that are easily generated from a proposal $q(\cdot)$ called an *importance density*.
- A weighted approximation to the density $p(\cdot)$ is given by

$$p(x) \approx \sum_{i=1}^{N_s} \omega^i \delta(x - x^i)$$

where $\omega_i \propto \frac{\pi(x^i)}{q(x^i)}$ is the normalized weight of the i-th particle.

Weights calculation

- If the samples $\mathbf{x}_{0:k}^i$ were drawn from an importance density $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, then the weights becomes

$$\omega_k^i \propto \frac{p(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}.$$

- From ω_{k-1}^i to ω_k^i $(p(\mathbf{x}_{0:k-1}^i|\mathbf{z}_{1:k-1}) \text{—}{>} p(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k}))$
  - Suppose the importance density is chosen as

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$$

Weights update

- Express $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ in terms of $p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}), p(\mathbf{z}_k|\mathbf{x}_k)$, and $p(\mathbf{x}|\mathbf{x}_{k-1})$, i.e.,

$$\begin{aligned} p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) &= \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k},\mathbf{z}_{1:k-1})p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k},\mathbf{z}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}_{0:k-1}\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \cdot p(\mathbf{x}_{0:k-1|\mathbf{z}_{1:k-1}}) \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \cdot p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \\ &\propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \end{aligned}$$

- According to the expression for $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, we have

$$\begin{aligned} \omega_k^i &\propto \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)p(\mathbf{x}_{0:k-1}^i|\mathbf{z}_{1:k-1})}{q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i,\mathbf{z}_{1:k})q(\mathbf{x}_{0:k-1}^i|\mathbf{z}_{1:k-1})} \\ &= \omega_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i,\mathbf{z}_{1:k})} \end{aligned}$$

- Furthermore, if $q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k)$, then we have

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i,\mathbf{z}_k)}$$

SIS particle filter

Algorithm: SIS particle filter $[\{\mathbf{x}_k^i, \omega_k^i\}_{i=1}^{N_s}] = \mathrm{SIS}[\{\mathbf{x}_{k-1}^i, \omega_{k-1}^i\}_{i=1}^{N_s}, \mathbf{z}_k]$

- For $i = 1 : N_s$
  - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  - Assign the particle a weight ω_k^i according to

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)}$$

  - Approximate the posterior as

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i).$$

- End For

Degeneracy problem existed in particle filter

- After a few iterations, all but one particle will have negligible weight.
- the brute force approach: use a very large N_s
- good choice of importance density
- Resampling

Other related particle filters

- sampling importance resampling (SIR) filter
- auxiliary sampling importance resampling (ASIR) filter
- regularized particle filter (RPF)