The Kalman Filter
ImPr Talk

Ged Ridgway
Centre for Medical Image Computing

November, 2006

Outline

What is the Kalman Filter for?

State estimation in Dynamical Systems

What is the Kalman Filter for?

State estimation in Dynamical Systems

In particular

- for discrete-time models
- with continuous, possibly hidden, state

What is the Kalman Filter for?

State estimation in Dynamical Systems

In particular

- for discrete-time models
- with continuous, possibly hidden, state

It is **optimal** for Linear Gaussian systems

Why ‘filter’?

The KF is a state-space filter

Why ‘filter’?

The KF is a state-space filter

- Noisy measurements $\longrightarrow$ cleaned state estimate
- Estimates are updated online with each new measurement

Why ‘filter’?

The KF is a state-space filter

- Noisy measurements $\rightarrow$ cleaned state estimate
- Estimates are updated online with each new measurement

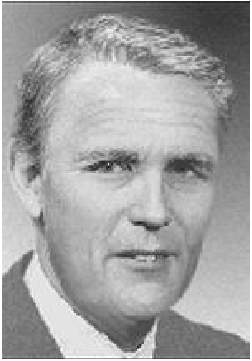
Equivalent to an (optimally) adaptive low-pass (IIR) filter

Why ‘Kalman’? (!)

R. E. Kalman
A new approach to linear filtering and prediction problems.
Journal of Basic Engineering, 82(1):35-45, 1960.

Why 'Kalman'? (!)

R. E. Kalman
A new approach to linear filtering and prediction problems.
Journal of Basic Engineering, 82(1):35-45, 1960.



Rudolf Emil Kalman, born in Budapest, Hungary, May 19, 1930.

Outline

State space examples - dynamical system

Motion of a particle (in 1D)
constant (but noisy) acceleration

State space examples - dynamical system

Motion of a particle (in 1D)
constant (but noisy) acceleration

$$
\begin{aligned}
s_k &= s_{k-1} + \dot{s}_{k-1}\Delta T + \frac{1}{2}\ddot{s}_{k-1}(\Delta T)^2 \\
\dot{s}_k &= \dot{s}_{k-1} + \ddot{s}_{k-1}\Delta T \\
\ddot{s}_k &= \ddot{s}_{k-1}
\end{aligned}
$$

State space examples - dynamical system

Motion of a particle (in 1D)
constant (but noisy) acceleration

$$
\begin{aligned}
s_k &= s_{k-1} + \dot{s}_{k-1}\Delta T + \frac{1}{2}\ddot{s}_{k-1}(\Delta T)^2 + \epsilon_k \\
\dot{s}_k &= \dot{s}_{k-1} + \ddot{s}_{k-1}\Delta T + \dot{\epsilon}_k \\
\ddot{s}_k &= \ddot{s}_{k-1} + \ddot{\epsilon}_k
\end{aligned}
$$

State space examples - dynamical system

Motion of a particle (in 1D)
constant (but noisy) acceleration

$$
\begin{aligned}
s_k &= s_{k-1} + \dot{s}_{k-1}\Delta T + \tfrac{1}{2}\ddot{s}_{k-1}(\Delta T)^2 + \epsilon_k \\
\dot{s}_k &= \dot{s}_{k-1} + \ddot{s}_{k-1}\Delta T + \dot{\epsilon}_k \\
\ddot{s}_k &= \ddot{s}_{k-1} + \ddot{\epsilon}_k
\end{aligned}
$$

$$
\begin{pmatrix} s_k \\ \dot{s}_k \\ \ddot{s}_k \end{pmatrix} = \begin{bmatrix} 1 & \Delta T & (\Delta T)^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} s_{k-1} \\ \dot{s}_{k-1} \\ \ddot{s}_{k-1} \end{pmatrix} + w_k
$$

State space examples - dynamical system

Motion of a particle (in 1D)
constant (but noisy) acceleration

$$
\begin{aligned}
s_k &= s_{k-1} + \dot{s}_{k-1}\Delta T + \tfrac{1}{2}\ddot{s}_{k-1}(\Delta T)^2 + \epsilon_k \\
\dot{s}_k &= \dot{s}_{k-1} + \ddot{s}_{k-1}\Delta T + \dot{\epsilon}_k \\
\ddot{s}_k &= \ddot{s}_{k-1} + \ddot{\epsilon}_k
\end{aligned}
$$

$$
\begin{pmatrix} s_k \\ \dot{s}_k \\ \ddot{s}_k \end{pmatrix} = \begin{bmatrix} 1 & \Delta T & (\Delta T)^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} s_{k-1} \\ \dot{s}_{k-1} \\ \ddot{s}_{k-1} \end{pmatrix} + w_k
$$

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

State space examples - dynamical system

Motion of a particle (in 1D)
constant (but noisy) acceleration

$$
\begin{aligned}
s_k &= s_{k-1} + \dot{s}_{k-1}\Delta T + \frac{1}{2}\ddot{s}_{k-1}(\Delta T)^2 + \epsilon_k \\
\dot{s}_k &= \dot{s}_{k-1} + \ddot{s}_{k-1}\Delta T + \dot{\epsilon}_k \\
\ddot{s}_k &= \ddot{s}_{k-1} + \ddot{\epsilon}_k
\end{aligned}
$$

$$
\begin{pmatrix} s_k \\ \dot{s}_k \\ \ddot{s}_k \end{pmatrix} = \begin{bmatrix} 1 & \Delta T & (\Delta T)^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} s_{k-1} \\ \dot{s}_{k-1} \\ \ddot{s}_{k-1} \end{pmatrix} + w_k
$$

$$
\begin{aligned}
\boldsymbol{x}_k &= F\boldsymbol{x}_{k-1} + \boldsymbol{w}_k \\
\boldsymbol{w}_k &\sim \mathcal{N}\left(\boldsymbol{w}_k | 0, Q\right)
\end{aligned}
$$

State space examples - time-series

State space examples - time-series

Autoregressive AR(n) model

$$\begin{pmatrix} s_k \\ s_{k-1} \\ s_{k-2} \\ \vdots \\ s_{k-(n-1)} \end{pmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & & 0 \\ \vdots & & 1 & 0 & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} s_{k-1} \\ s_{k-2} \\ s_{k-3} \\ \vdots \\ s_{k-n} \end{pmatrix} + \begin{bmatrix} \epsilon_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

State space examples - time-series

Autoregressive AR(n) model

$$\begin{pmatrix} s_k \\ s_{k-1} \\ s_{k-2} \\ \vdots \\ s_{k-(n-1)} \end{pmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & & 0 \\ \vdots & & 1 & 0 & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} s_{k-1} \\ s_{k-2} \\ s_{k-3} \\ \vdots \\ s_{k-n} \end{pmatrix} + \begin{bmatrix} \epsilon_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\boldsymbol{x}_k = F\boldsymbol{x}_{k-1} + \boldsymbol{w}_k \qquad Q_k = \begin{bmatrix} \sigma^2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

The state space model

$x_k \in \mathbb{R}^n$	vector of hidden state variables
$z_k \in \mathbb{R}^m$	vector of observations
$u_k \in \mathbb{R}^p$	vector of control inputs

Linear

$$\begin{aligned} x_k &= F_k x_{k-1} + B_k u_k \\ z_k &= H_k x_k \end{aligned} \tag{1}$$

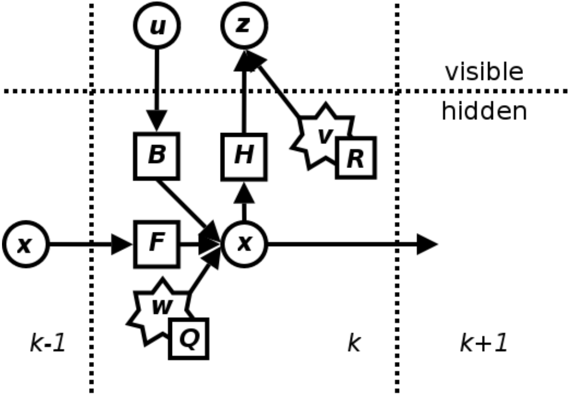
The state space model

$x_k \in \mathbb{R}^n$	vector of hidden state variables
$z_k \in \mathbb{R}^m$	vector of observations
$u_k \in \mathbb{R}^p$	vector of control inputs

Linear Gaussian

$$\begin{aligned} x_k &= F_k x_{k-1} + B_k u_k + w_k \\ z_k &= H_k x_k + v_k \\ w_k &\sim \mathcal{N}(w_k | 0, Q_k) \\ v_k &\sim \mathcal{N}(v_k | 0, R_k) \end{aligned} \tag{1}$$

The state space model



Outline

How does the KF work?

Recursive (online) updating of current state estimate and precision (covariance)

How does the KF work?

Recursive (online) updating of current state estimate and precision (covariance)

1. Using the system model

How does the KF work?

Recursive (online) updating of current state estimate and precision (covariance)

1. Using the system model
   - **Predict** the next state
   - Estimate the new (reduced) precision

How does the KF work?

Recursive (online) updating of current state estimate and precision (covariance)

1. Using the system model
   - **Predict** the next state
   - Estimate the new (reduced) precision
2. Using the new measurement

How does the KF work?

Recursive (online) updating of current state estimate and precision (covariance)

1. Using the system model
   - **Predict** the next state
   - Estimate the new (reduced) precision
2. Using the new measurement
   - **Correct** the state estimate
   - Update the (increased) precision

How is it derived?

Two equivalent alternatives

How is it derived?

Two equivalent alternatives

1. Classical

How is it derived?

Two equivalent alternatives

1. Classical
   - Show form of Kalman gain update is unbiased
   - Derive expressions for posterior covariance (MSE)
   - Analytically find optimal Kalman gain

How is it derived?

Two equivalent alternatives

1. Classical
   - Show form of Kalman gain update is unbiased
   - Derive expressions for posterior covariance (MSE)
   - Analytically find optimal Kalman gain
2. Bayesian

How is it derived?

Two equivalent alternatives

1. Classical
   - Show form of Kalman gain update is unbiased
   - Derive expressions for posterior covariance (MSE)
   - Analytically find optimal Kalman gain
2. Bayesian
   - Treat previous estimates as prior
   - Measurement model gives likelihood
   - Derive posterior distribution

Outline

What is the Kalman Filter?

State Space Models

Kalman Filter Overview

Bayesian Updating of Estimates

Kalman Filter Equations

Extensions

Bayesian updating

A simple example

$$p(x) = \mathcal{N}\left(x|\hat{x}, \sigma_w^2\right) \qquad \text{prior}$$

$$z = x + v \qquad v \sim \mathcal{N}(0, \sigma_v^2)$$

$$p(z|x) = \mathcal{N}\left(z|x, \sigma_v^2\right) \qquad \text{likelihood}$$

Bayesian updating

A simple example

$$p(x) = \mathcal{N}\left(x|\hat{x}, \sigma_w^2\right) \qquad \text{prior}$$

$$z = x + v \qquad v \sim \mathcal{N}(0, \sigma_v^2)$$

$$p(z|x) = \mathcal{N}\left(z|x, \sigma_v^2\right) \qquad \text{likelihood}$$

$$p(x|z) \propto p(z|x)\,p(x) \qquad \text{from Bayes}$$

Bayesian updating

$$p(z|x)\,p(x) = \mathcal{N}\left(z|x, \sigma_v^2\right)\,\mathcal{N}\left(x|\hat{x}, \sigma_w^2\right)$$

Bayesian updating

$$p(z|x)\,p(x) = \mathcal{N}\left(z|x,\sigma_v^2\right)\mathcal{N}\left(x|\hat{x},\sigma_w^2\right)$$

$$\log\left(p(z|x)\,p(x)\right) = \frac{(z-x)^2}{\sigma_v^2} + \frac{(x-\hat{x})^2}{\sigma_w^2} + \cdots$$

Bayesian updating

$$
\begin{aligned}
p(z|x)\,p(x) &= \mathcal{N}\left(z|x,\sigma_v^2\right)\mathcal{N}\left(x|\hat{x},\sigma_w^2\right) \\
\log\left(p(z|x)\,p(x)\right) &= \frac{(z-x)^2}{\sigma_v^2} + \frac{(x-\hat{x})^2}{\sigma_w^2} + \cdots \\
&= x^2\left(\frac{1}{\sigma_v^2}+\frac{1}{\sigma_w^2}\right) - 2x\left(\frac{z}{\sigma_v^2}+\frac{\hat{x}}{\sigma_w^2}\right) + \cdots
\end{aligned}
$$

Bayesian updating

$$
\begin{aligned}
p(z|x)\,p(x) &= \mathcal{N}\left(z|x,\sigma_v^2\right)\,\mathcal{N}\left(x|\hat{x},\sigma_w^2\right) \\
\log\left(p(z|x)\,p(x)\right) &= \frac{(z-x)^2}{\sigma_v^2} + \frac{(x-\hat{x})^2}{\sigma_w^2} + \cdots \\
&= x^2\left(\frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2}\right) - 2x\left(\frac{z}{\sigma_v^2} + \frac{\hat{x}}{\sigma_w^2}\right) + \cdots \\
&= x^2/a - 2bx + \cdots
\end{aligned}
$$

Bayesian updating

$$
\begin{aligned}
p(z|x)\,p(x) &= \mathcal{N}\left(z|x,\sigma_v^2\right)\mathcal{N}\left(x|\hat{x},\sigma_w^2\right) \\
\log\left(p(z|x)\,p(x)\right) &= \frac{(z-x)^2}{\sigma_v^2} + \frac{(x-\hat{x})^2}{\sigma_w^2} + \cdots \\
&= x^2\left(\frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2}\right) - 2x\left(\frac{z}{\sigma_v^2} + \frac{\hat{x}}{\sigma_w^2}\right) + \cdots \\
&= x^2/a - 2bx + \cdots \\
&= \frac{1}{a}(x-ab)^2 + \cdots
\end{aligned}
$$

Bayesian updating (Information form)

$$\log(p(x|z)) \propto \frac{1}{a}(x - ab)^2 + \cdots$$

Bayesian updating (Information form)

$$\log\left(p(x|z)\right) \propto \frac{1}{a}\left(x - ab\right)^2 + \cdots$$

$$\Rightarrow p(x|z) = \mathcal{N}\left(x|ab, a\right)$$

$$\text{where} \quad \frac{1}{a} = \frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2}$$

$$b = \frac{z}{\sigma_v^2} + \frac{\hat{x}}{\sigma_w^2}$$

Bayesian updating (Information form)

$$
\begin{aligned}
\log\left(p(x|z)\right) &\propto \frac{1}{a}(x - ab)^2 + \cdots \\
\Rightarrow p(x|z) &= \mathcal{N}(x|ab, a) \\
\text{where} \quad \frac{1}{a} &= \frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2} \\
b &= \frac{z}{\sigma_v^2} + \frac{\hat{x}}{\sigma_w^2}
\end{aligned}
$$

Note

- prior and measurement *information* add
- the posterior mean is the **information-weighted mean**

Bayesian updating (Information form)

Summary

$$\begin{aligned} \frac{1}{a} &= \frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2} \\ \text{posterior cov} &= a \\ \text{posterior mean} &= a\left(\frac{z}{\sigma_v^2} + \frac{\hat{x}}{\sigma_w^2}\right) \end{aligned} \tag{2}$$

Bayesian updating (Kalman-gain form)

Now rearrange, for reasons that might become clear later...

$$a = \left(\frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2} \right)^{-1}$$

Bayesian updating (Kalman-gain form)

Now rearrange, for reasons that might become clear later...

$$
\begin{aligned}
a &= \left(\frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2} \right)^{-1} \\
&= \frac{\sigma_w^2 \sigma_v^2}{\sigma_w^2 + \sigma_v^2} \\
&= \frac{\sigma_w^2 (\sigma_w^2 + \sigma_v^2) - \sigma_w^4}{\sigma_w^2 + \sigma_v^2} \\
&= \sigma_w^2 - \frac{\sigma_w^4}{\sigma_v^2 + \sigma_w^2}
\end{aligned}
$$

Bayesian updating (Kalman-gain form)

Now rearrange, for reasons that might become clear later...

$$
\begin{aligned}
a &= \left(\frac{1}{\sigma_v^2} + \frac{1}{\sigma_w^2} \right)^{-1} \\
&= \frac{\sigma_w^2 \sigma_v^2}{\sigma_w^2 + \sigma_v^2} \\
&= \frac{\sigma_w^2(\sigma_w^2 + \sigma_v^2) - \sigma_w^4}{\sigma_w^2 + \sigma_v^2} \\
&= \sigma_w^2 - \frac{\sigma_w^4}{\sigma_v^2 + \sigma_w^2} \\
&= (1 - k)\sigma_w^2 \\
k &= \frac{\sigma_w^2}{\sigma_v^2 + \sigma_w^2}
\end{aligned}
$$

Bayesian updating (Kalman-gain form)

Bear with me...

$$a = (1 - k)\sigma_w^2$$

$$k = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_v^2}$$

Bayesian updating (Kalman-gain form)

Bear with me...

$$a = (1 - k)\sigma_w^2$$

$$k = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_v^2}$$

$$(1 - k) = \frac{\sigma_v^2}{\sigma_w^2 + \sigma_v^2} = k\frac{\sigma_v^2}{\sigma_w^2}$$

Bayesian updating (Kalman-gain form)

Bear with me...

$$
\begin{aligned}
a &= (1-k)\sigma_w^2 \\
k &= \frac{\sigma_w^2}{\sigma_w^2 + \sigma_v^2} \\
(1-k) &= \frac{\sigma_v^2}{\sigma_w^2 + \sigma_v^2} = k\frac{\sigma_v^2}{\sigma_w^2} \\
ab &= (1-k)\sigma_w^2 \left(\frac{z}{\sigma_v^2} + \frac{\hat{x}}{\sigma_w^2} \right) \\
&= (1-k)\sigma_w^2 \frac{z}{\sigma_v^2} + (1-k)\hat{x} \\
&= kz + \hat{x} - k\hat{x} = \hat{x} + k(z - \hat{x})
\end{aligned}
$$

Bayesian updating (Kalman-gain form)

Summary

$$\begin{aligned} k &= \frac{\sigma_w^2}{\sigma_w^2 + \sigma_v^2} \\ \text{posterior cov} &= (1-k)\sigma_w^2 \\ \text{posterior mean} &= \hat{x} + k(z - \hat{x}) \end{aligned} \tag{3}$$

Outline

Derivation outline

The **prediction** step is straight-forward.
From the model

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

Derivation outline

The **prediction** step is straight-forward.
From the model

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

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

Derivation outline

The **prediction** step is straight-forward.
From the model

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

Derivation outline

The **prediction** step is straight-forward.
From the model

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

This gives us the time-k prior mean and covariance in terms of the time-$(k-1)$ posteriors.

Derivation outline

The **correction** step is algebraically tricky, but conceptually very similar to the simple example of Bayesian updating.

(5)

Derivation outline

The **correction** step is algebraically tricky, but conceptually very similar to the simple example of Bayesian updating.

$$p(x_k|\hat{x}_{k|k-1}, z_k) \propto p(z_k|x_k, \hat{x}_{k|k-1})\, p(x_k|\hat{x}_{k|k-1}) \quad (5)$$

Derivation outline

The **correction** step is algebraically tricky, but conceptually very similar to the simple example of Bayesian updating.

$$\begin{aligned} p(x_k | \hat{x}_{k|k-1}, z_k) &\propto p(z_k | x_k, \hat{x}_{k|k-1})\, p(x_k | \hat{x}_{k|k-1}) \\ &\propto p(z_k | x_k)\, p(x_k | \hat{x}_{k|k-1}) \end{aligned} \tag{5}$$

Derivation outline

The **correction** step is algebraically tricky, but conceptually very similar to the simple example of Bayesian updating.

$$\begin{aligned} p(x_k | \hat{x}_{k|k-1}, z_k) &\propto p(z_k | x_k, \hat{x}_{k|k-1}) \, p(x_k | \hat{x}_{k|k-1}) \\ &\propto p(z_k | x_k) \, p(x_k | \hat{x}_{k|k-1}) \\ &\propto \mathcal{N}(z_k | H_k x_k, R_k) \, \mathcal{N}\left(x_k | \hat{x}_{k|k-1}, P_{k|k-1}\right) \end{aligned} \quad (4)$$

$$(5)$$

Derivation outline

The **correction** step is algebraically tricky, but conceptually very similar to the simple example of Bayesian updating.

$$
\begin{aligned}
p(x_k|\hat{x}_{k|k-1}, z_k) &\propto p(z_k|x_k, \hat{x}_{k|k-1})\, p(x_k|\hat{x}_{k|k-1}) \\
&\propto p(z_k|x_k)\, p(x_k|\hat{x}_{k|k-1}) \\
&\propto \mathcal{N}\left(z_k|H_k x_k, R_k\right) \mathcal{N}\left(x_k|\hat{x}_{k|k-1}, P_{k|k-1}\right)
\end{aligned} \tag{4}
$$

$$
p(x_k|\hat{x}_{k|k-1}, z_k) = \mathcal{N}\left(x_k|\hat{x}_{k|k}, P_{k|k}\right) \tag{5}
$$

Derivation outline

The mean and covariance of the time-k posterior in (5) can be derived in *information form* by ‘completing the square’ of the quadratic form in x that arises from (4).

Derivation outline

Rearranging into Kalman-gain form then requires some fiddly algebra and the following two matrix inversion identities

Derivation outline

Rearranging into Kalman-gain form then requires some fiddly algebra and the following two matrix inversion identities

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \tag{6}$$

$$(A + BCD)^{-1}BC = A^{-1}B(C^{-1} + DA^{-1}B)^{-1} \tag{7}$$

Results (information form)

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

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

Results (information form)

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

$$P_{k|k}^{-1} = P_{k|k-1}^{-1} + H_k^T R_k^{-1} H_k$$
$$\hat{x}_{k|k} = P_{k|k} \left(P_{k|k-1}^{-1} \hat{x}_{k|k-1} + H_k^T R_k^{-1} z_k \right) \quad (8)$$

(Compare (2) from before)

Results (Kalman-gain form)

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k$$
$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

Results (Kalman-gain form)

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

$$\begin{aligned}K &= P_{k|k-1} H_k^T \left(R_k + H_k P_{k|k-1} H_k^T \right)^{-1} \\ P_{k|k} &= (I - K H_k) P_{k|k-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K \left(z_k - H_k \hat{x}_{k|k-1} \right)\end{aligned} \tag{9}$$

(Compare (3), and see `kalman_update.m`)

Outline

Prediction & Smoothing

The model allows us to predict *ahead* of the measurements, at each iteration we maintain the optimal estimate of the future state given the data we have so far.

Prediction & Smoothing

The model allows us to predict *ahead* of the measurements, at each iteration we maintain the optimal estimate of the future state given the data we have so far.

We can also estimate the state at a point *before* the current measurement — known as ‘fixed lag smoothing’.

Prediction & Smoothing

The model allows us to predict *ahead* of the measurements, at each iteration we maintain the optimal estimate of the future state given the data we have so far.

We can also estimate the state at a point *before* the current measurement — known as ‘fixed lag smoothing’.

If we first collect all the data and want to process it offline (‘fixed interval smoothing’), then a closely related algorithm can be derived.

Prediction & Smoothing

The model allows us to predict *ahead* of the measurements, at each iteration we maintain the optimal estimate of the future state given the data we have so far.

We can also estimate the state at a point *before* the current measurement — known as ‘fixed lag smoothing’.

If we first collect all the data and want to process it offline (‘fixed interval smoothing’), then a closely related algorithm can be derived.

The Rauch Tung Striebel Smoother (or Kalman Smoother) uses a *forward-backward* algorithm to achieve optimal offline smoothing with two passes of a Kalman-like recursion. (see `kalman_smoother.m`).

Nonlinearity

For non-linear state space models

$$x_k = f_k(x_{k-1}, u_k) + w_k$$
$$z_k = h_k(x_k) + v_k$$

Nonlinearity

For non-linear state space models

$$
\begin{aligned}
x_k &= f_k(x_{k-1}, u_k) + w_k \\
z_k &= h_k(x_k) + v_k
\end{aligned}
$$

- Either, repeatedly linearise about the current point (EKF)

Nonlinearity

For non-linear state space models

$$x_k = f_k(x_{k-1}, u_k) + w_k$$
$$z_k = h_k(x_k) + v_k$$

- Either, repeatedly linearise about the current point (EKF)
- Or, non-linearly propagate (deterministic) samples and re-estimate the Gaussian mean and covariance (UKF)

Non-Gaussianity

If the process or measurement noise isn't normally distributed, then for the correction step

$$p(x_k|\hat{x}_{k|k-1}, z_k) \propto p(z_k|x_k)\,p(x_k|\hat{x}_{k|k-1})$$

we can *sample* from the distribution using Markov Chain Monte Carlo methods.

Non-Gaussianity

If the process or measurement noise isn't normally distributed, then for the correction step

$$p(x_k|\hat{x}_{k|k-1}, z_k) \propto p(z_k|x_k)\, p(x_k|\hat{x}_{k|k-1})$$

we can *sample* from the distribution using Markov Chain Monte Carlo methods.

This is known as *Sequential Monte Carlo* (SMC) or *Particle Filtering*, and is a very powerful technique. See, for example, the *Condensation* algorithm in computer vision.