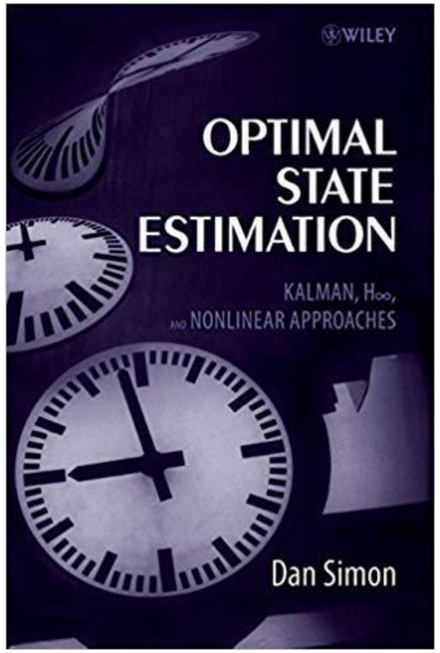
The Discrete-time Kalman Filter

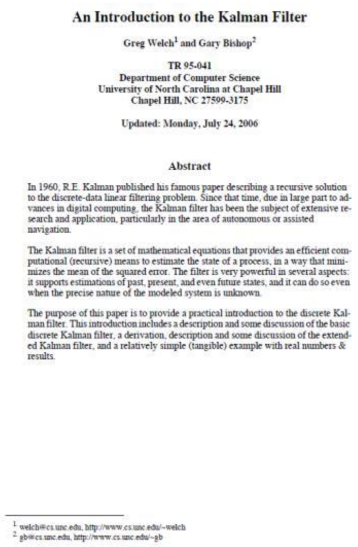
Week 07
2022-04-11

Handong Global University
Smart Sensors and IoT Devices

0. References

1) **Optimal State Estimation: Kalman, H infinity, and nonlinear approaches**
   Simon, Dan. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & sons, 2006.
2) **An Introduction to the Kalman Filter**
   Welch, Greg, and Gary Bishop. 'An introduction to the Kalman filter.' (1995): 41-95.
3) **Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation**
   Faragher, Ramsey. 'Understanding the basis of the Kalman filter via a simple and intuitive derivation.' IEEE Signal processing magazine 29.5 (2012): 128-132.





An Introduction to the Kalman Filter

Greg Welch[1] and Gary Bishop[2]

TR 95-041
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175

Updated: Monday, July 24, 2006

Abstract

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation.

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown.

The purpose of this paper is to provide a practical introduction to the discrete Kalman filter. This introduction includes a description and some discussion of the basic discrete Kalman filter, a derivation, description and some discussion of the extended Kalman filter, and a relatively simple (tangible) example with real numbers & results.

1 welch@cs.unc.edu, http://www.cs.unc.edu/~welch
2 gb@cs.unc.edu, http://www.cs.unc.edu/~gb

[lecture NOTES]

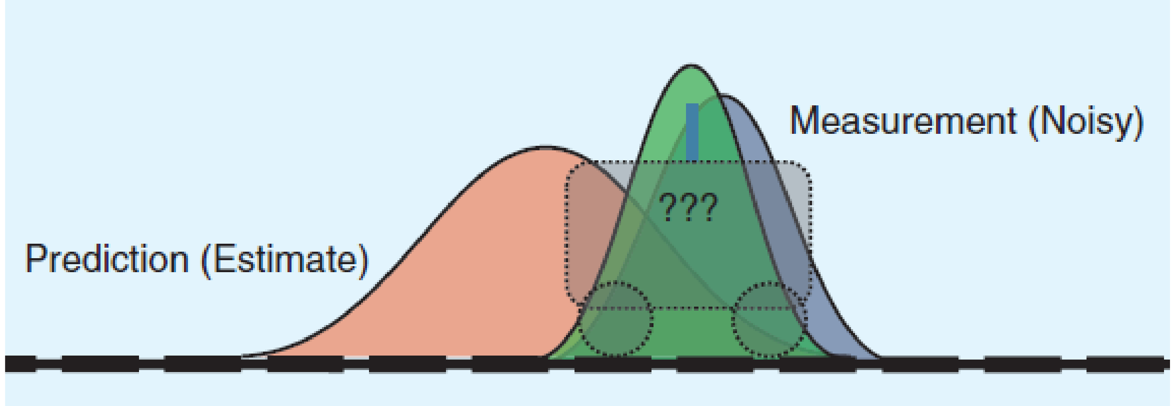
Ramsey Faragher

Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation

This article provides a simple and intuitive derivation of the Kalman filter, with the aim of teaching this useful tool to students from disciplines that do not require a strong ...

1. Summary of the Kalman Filter

How to estimate the position of a car from an imperfect dynamic equation and a noisy measurement?



Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

(3) Update the error covariance

$$P_k = (I - K_k H)P_k^-$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

1. Summary of the Kalman Filter

2-1. Least Squares Method

- Weighted Cost Function with Covariance
- Recursive Expression

2-2. State-Space Equations of a Linear System

- $CT: \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}, \quad DT: \begin{cases} x[k+1] = Ax[k] + Bu[k] \\ y[k] = Cx[k] + Du[k] \end{cases}$

2-3. Predictor & Corrector Algorithm

- Runge-Kutta, K-means, Kalman Filter ...

2-1. Least Squares Method

x: state variables(n elements)
v_k: random measurement noise @ k^{th} time step
y_k: noisy measurement @ k^{th} time step

- Equations of a constant vector, x

$$\begin{cases} y_1 & = & H_{11}x_1 + \cdots + H_{1n}x_n + v_1 \\ \vdots & = & \vdots \\ y_k & = & H_{k1}x_1 + \cdots + H_{kn}x_n + v_k \end{cases} \rightarrow y = H\boldsymbol{x} + v$$

- How to find an optimized estimate of $x, \hat{x}$?
  - Minimize the sum of squares of residuals: cost function, J

$$J = \varepsilon_{y_1}^2 + \varepsilon_{y_2}^2 + \cdots + \varepsilon_{y_k}^2 = \varepsilon_y^T \varepsilon_y = (y - H\hat{x})^T (y - H\hat{x})$$

$$\frac{\partial J}{\partial x} = -y^T H - y^T H + 2\hat{x}^T H^T H = 0$$

$$\therefore \hat{x} = \left(H^T H\right)^{-1} H^T y$$

2-1. Least Squares Method

Example

Voltage measurements with a noisy DMM

$$\begin{cases} y_1 & = & x_1 + v_1 \\ \vdots & = & \vdots \\ y_k & = & x_1 + v_k \end{cases} \rightarrow y = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} x + v \qquad H = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$\hat{x} = \left(H^T H\right)^{-1} H^T y = \left(\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix}$$

$$\therefore \hat{x} = \frac{1}{k}(y_1 + \cdots + y_k)$$

2-2. Weighted Least Squares Method

$$y = Hx + v$$

$$R = E(vv^T) = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k^2 \end{bmatrix}$$, **covariance matrix**

Assume white noise: $E(v_i) = 0, E(v_i^2) = \sigma_i^2, E(v_i v_j) = 0$

- **How to find an optimized estimate of $x, \hat{x}$?**
  - **Minimize the sum of weighted squares of residuals**

$$J = \frac{\varepsilon_{y_1}^2}{\sigma_1^2} + \frac{\varepsilon_{y_2}^2}{\sigma_2^2} + \cdots + \frac{\varepsilon_{y_k}^2}{\sigma_k^2} = \varepsilon_y^T R^{-1} \varepsilon_y = (y - H\hat{x})^T R^{-1} (y - H\hat{x})$$

$$\frac{\partial J}{\partial x} = -y^T R^{-1} H - y^T R^{-1} H + 2\hat{x}^T H^T R^{-1} H = 0$$

$$\therefore \hat{x} = \left(H^T R^{-1} H\right)^{-1} H^T R^{-1} y$$

2-3. Recursive Weighted Least Squares Method

$$y_k = H_k x + v_k$$

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

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

x: state variables
v_k: random measurement noise @ k^{th} time step
y_k: noisy measurement @ k^{th} time step
K_k: correction gain @ k^{th} time step
$\hat{y}_k^-$: priori measurement estimate @ k^{th} time step

- How to find an estimate of x, $\hat{x}$ after each measurement update?
  - ➢ A recursive expression of the weighted least squares estimation

$$J_k = E\left[(x_1 - \hat{x}_1)^2\right] + \cdots + E\left[(x_k - \hat{x}_k)^2\right] = E\left[\varepsilon_{x,k}\varepsilon_{x,k}^T\right] = Tr(P_k)$$

$$P_k = (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T$$

$$\frac{\partial J_k}{\partial K_k} = 2(I - K_k H_k) P_{k-1}(-H_k^T) + 2 K_k R_k = 0$$

$$\therefore K_k = P_{k-1} H_k^T \left(H_k P_{k-1} H_k^T + R_k\right)^{-1}$$

$$Tr(A) = \sum A_{ii}$$

Sum of diagonal elements

$$P_k = E[\varepsilon_{x,k}\varepsilon_{x,k}^T]$$

Covariance matrix of error

2-3. Recursive Weighted Least Squares Method

$$
\begin{aligned}
E(\varepsilon_{x,k}) &= E(x - \hat{x}_k) \\
&= E(x - \hat{x}_{k-1} - K_k(y_k - H_k\hat{x}_{k-1})) \\
&= E(\varepsilon_{x,k-1} - K_k(H_k x + v_k - H_k\hat{x}_{k-1})) \\
&= E(\varepsilon_{x,k-1} - K_k H_k(x - \hat{x}_{k-1}) - K_k v_k) \\
&= E(\varepsilon_{x,k-1} - K_k H_k(\varepsilon_{x,k-1}) - K_k v_k) \\
&= (I - K_k H_k)E(\varepsilon_{x,k-1}) - K_k E(v_k)
\end{aligned}
$$

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

$$
\begin{aligned}
J_k &= E[(x_1 - \hat{x}_1)^2] + \cdots + E[(x_k - \hat{x}_k)^2] \\
&= E\left[\varepsilon^2_{x_1,k} + \cdots + \varepsilon^2_{x_k,k}\right] \\
&= E[\varepsilon_{x,k}\varepsilon^T_{x,k}] \\
&= Tr(P_k)
\end{aligned}
$$

2-3. Recursive Weighted Least Squares Method

$P_k = E[\varepsilon_{x,k}\varepsilon_{x,k}^T]$

Covariance matrix of error

K_k: correction gain

@ k^{th} time step

$$
\begin{aligned}
P_k &= E\left[\varepsilon_{x,k}\varepsilon_{x,k}^T\right] \\
&= E\left[(x_1 - \hat{x}_k)(x_1 - \hat{x}_k)^T\right] \\
&= E\left[\left\{(I - K_k H_k)\varepsilon_{x,k-1} - K_k v_k\right\}\left\{(I - K_k H_k)\varepsilon_{x,k-1} - K_k v_k\right\}^T\right] \\
&= (I - K_k H_k)E\left(\varepsilon_{x,k-1}\varepsilon_{x,k-1}^T\right)(I - K_k H_k)^T + K_k E\left(v_k v_k^T\right)K_k^T \\
&= (I - K_k H_k)P_{k-1}(I - K_k H_k)^T + K_k R_k K_k^T
\end{aligned}
$$

$$\frac{\partial J_k}{\partial K_k} = \frac{\partial Tr(P_k)}{\partial K_k} = 2(I - K_k H_k)P_{k-1}(-H_k^T) + 2K_k R_k = 0$$

$$K_k = P_{k-1}H_k^T\left(H_k P_{k-1} H_k^T + R_k\right)^{-1}$$

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

$$P_k = (I - K_k H_k)P_{k-1}(I - K_k H_k)^T + K_k R_k K_k^T$$

$$K_k = P_{k-1}H_k^T\left(H_k P_{k-1} H_k^T + R_k\right)^{-1}$$

2-3. Recursive Weighted Least Squares Method

Summary

1) $k = 0$

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

$$P_0 = E[(x - \widehat{x_0})(x - \widehat{x_0})^T]$$

2) $k = 1, 2, \dots$

A. Measurement

$$y_k = H_k x + v_k$$

B. Estimation

$$K_k = P_{k-1} H_k^T \left(H_k P_{k-1} H_k^T + R_k \right)^{-1}$$

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

$$P_k = (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T$$

x: state variables
$\hat{x}_k$: estimation of x @ k^{th} time step

v_k: measurement noise @ k^{th} time step
y_k: noisy measurement @ k^{th} time step
white noise: $E(v_i) = 0, E\left(v_i^2\right) = \sigma_i^2, E\left(v_i v_j\right) = 0$

K_k: correction gain @ k^{th} time step
R_k: covariance matrix of measurement noise
P_k: covariance matrix of estimation error

2-3. Recursive Weighted Least Squares Method

Example

Voltage measurement using a DMM with a variance of $(1V)^2$

1) $k = 0$ **(assume the worst-case scenario)**

$$\hat{x}_0 = E(x) = 0$$

$$P_0 = E\left[(x - \widehat{x_0})(x - \widehat{x_0})^T\right] = \infty$$

2) $k = 1$

$$y_1 = x + v_1 \quad (\rightarrow H = 1)$$

$$K_1 = P_0 H_1^T \left(H_1 P_0 H_1^T + R_1\right)^{-1} = P_0 (P_0 + R)^{-1} = \frac{\infty}{\infty + 1} = 1 \quad (\rightarrow R = 1^2)$$

$$\hat{x}_1 = \hat{x}_0 + K_1 (y_1 - \hat{x}_0) = 0 + 1(y_1 - 0) = y_1$$

$$P_1 = (I - K_1 H_1) P_0 (I - K_1 H_1)^T + K_1 R_1 K_1^T = (1 - 1)(\infty)(1 - 1)^T + 1 \cdot 1 \cdot 1^T = 0 + 1 = 1$$

2-3. Recursive Weighted Least Squares Method

Example

Voltage measurement using a DMM with a variance of $(1V)^2$

3) $k = 2$

$$y_2 = x + v_2$$

$$K_2 = P_1 H_2^T \left(H_2 P_1 H_2^T + R\right)^{-1} = P_1 (P_1 + R)^{-1} = \frac{1}{1+1} = \frac{1}{2}$$

$$\hat{x}_2 = \hat{x}_1 + K_2 (y_2 - \hat{x}_1) = y_1 + \frac{1}{2}(y_2 - y_1) = \frac{1}{2}(y_1 + y_2)$$

$$P_2 = (I - K_2 H_2) P_1 (I - K_2 H_2)^T + K_2 R K_2^T = \left(1 - \frac{1}{2}\right)(1)\left(1 - \frac{1}{2}\right)^T + \frac{1}{2} \cdot 1 \cdot \frac{1}{2}^T = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

4) $k = 3$

$$y_3 = x + v_3$$

$$K_3 = P_2 H_3^T \left(H_3 P_2 H_3^T + R\right)^{-1} = P_2 (P_2 + R)^{-1} = \frac{\frac{1}{2}}{\frac{1}{2} + 1} = \frac{1}{3}$$

$$\hat{x}_3 = \hat{x}_2 + K_3 (y_3 - \hat{x}_2) = \frac{1}{2}(y_1 + y_2) + \frac{1}{3}\left(y_3 - \frac{1}{2}(y_1 + y_2)\right) = \frac{1}{3}(y_1 + y_2 + y_3)$$

$$P_3 = (I - K_3 H_3) P_2 (I - K_3 H_3)^T + K_3 R K_3^T = \left(1 - \frac{1}{3}\right)\left(\frac{1}{2}\right)\left(1 - \frac{1}{3}\right)^T + \frac{1}{3} \cdot 1 \cdot \frac{1}{3}^T = \frac{2}{9} + \frac{1}{9} = \frac{1}{3}$$

2-3. Recursive Weighted Least Squares Method

Example

Voltage measurement using a DMM with a variance of $(1V)^2$

$k = k$일때

$$K_k = \frac{1}{k}$$

$$\hat{x}_k = \frac{1}{k}(y_1 + \cdots + y_k)$$

$$P_k = \frac{1}{k}$$

가장 최적화된 전압 추정 값은 계측 값의 평균

오차의 공분산은 1/k이며, 계측을 무한히 반복하면 0에 수렴하여 추정 값도 참값에 수렴

3. State-Space Representation of a linear System

- **State: 상태**

  $t = t_0$의 초기값과 $t \geq t_0$의 입력이 주어졌을 때, 시스템 거동을 완전히 설명하는 변수의 최소 집합

- **State variables: 상태 변수**

  시스템 거동을 완전히 표현하기 위한 최소 개수의 변수들

  초기 값을 가지는 변수들이 대부분 상태변수

- **State space: 상태 공간**

  상태 변수로 이루어진 n차원의 공간

3. State-Space Representation of a linear System

- **State-Space Equations**

$$CT: \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}, \quad DT: \begin{cases} x[k+1] = Ax[k] + Bu[k] \\ y[k] = Cx[k] + Du[k] \end{cases}$$

미분방정식을 행렬형태로 표현

MIMO, Non-linear, Time-variant system을 취급하기에 유용

컴퓨터 시뮬레이션이 쉽다

$\mathbb{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, 상태 벡터

$\mathbb{U} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix}$, 입력벡터 or 제어벡터

$\mathbb{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$, 출력 벡터

$\mathbb{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$

상태 행렬, $n \times n$

(State matrix)

$\mathbb{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1r} \\ b_{21} & b_{22} & \cdots & b_{2r} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nr} \end{bmatrix}$

입력 행렬, $n \times r$ or 제어 행렬

(input matrix)

$\mathbb{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}$

출력 행렬, $m \times n$

(output matrix)

$\mathbb{D} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1r} \\ d_{21} & d_{22} & \cdots & d_{2r} \\ \vdots & \vdots & & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mr} \end{bmatrix}$

직접전송 행렬 or 피드포워드 행렬

(direct transmission matrix)

3. State-Space Representation of a linear System

x_k: state of a system
u_k: known input
w_k: Gaussian zero mean white noise with covariance Q_k

Discrete-time System

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

$$x_1 = F_0x_0 + G_0u_0 + w_0$$

$$x_2 = F_1x_1 + G_1u_1 + w_1 = F_1F_0x_0 + (F_1G_0u_0 + G_1u_1) + (F_1w_0 + w_1)$$

$$x_3 = F_2x_2 + G_2u_2 + w_2 = F_2F_1F_0x_0 + (F_2F_1G_0u_0 + F_2G_1u_1 + G_2u_2) + (F_2F_1w_0 + F_2w_1 + w_2)$$

$$x_k = F_{k,0}x_0 + \sum_{i=0}^{k-1}(F_{k,i+1}G_iu_i + F_{k,i+1}w_i)$$

Where, $F_{k,i} = \begin{cases} F_{k-1}F_{k-2}\dots F_i & ,k > i \\ I & ,k = i \\ 0 & ,k < i \end{cases}$

2-3. Recursive Weighted Least Squares Method

$$\bar{x}_k = E(x_k) = E(F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1})$$
$$= F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1} + 0$$

$$P_k = E\left[(x - \hat{x}_k)(x - \hat{x}_k)^T\right]$$
$$= F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$$

If steady state,

$$P_k = P_{k-1} \rightarrow \therefore P = FPF^T + Q$$

If the system is stable, we get a unique solution

$$\bar{x} = F\bar{x} + Gu \rightarrow \bar{x} = (I - F)^{-1}Gu$$

$$P = \sum_{i=0}^{\infty}(F^i)Q\left(F^T\right)^i$$

3. State-Space Representation of a linear System

Example

- 포식자는 매일 전체 포식자 인구의 20%가 경쟁에서 살아남으며, 전체 먹이 인구의 40%만큼 증가
- 먹이는 매일 전체 포식자 인구의 40%만큼 감소하며, 외부에서 1만큼의 유입이 있음
- 초기 포식자 인구는 10마리, 초기 먹이 인구는 20마리로 설정
- 각 모델은 완벽하지 않아 각각 1,2 만큼의 분산이 있음

상태 공간 방정식을 통하여 steady-state 상태일 때의 인구 변화를 계산하라.

3. State-Space Representation of a linear System

Population of a predator:

$$\boldsymbol{x_1[k+1] = x_1[k] - 0.8x_1[k] + 0.4x_2[k] + w_1[k]}$$

Population of a prey:

$$\boldsymbol{x_2[k+1] = x_2[k] - 0.4x_1[k] + u[k] + w_2[k]}$$

Initial Condition

$$\boldsymbol{x_1[0] = 10, x_2[0] = 10, u[k] = 1, w \sim (0, Q), Q = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}}$$

State-Space Equation

$$\boldsymbol{x[k+1] = \begin{bmatrix} x_1[k+1] \\ x_2[k+1] \end{bmatrix} = \begin{bmatrix} 0.2 & 0.4 \\ -0.4 & 1 \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 + \begin{bmatrix} w_1[k] \\ w_2[k] \end{bmatrix}}$$

3. State-Space Representation of a linear System

Since the system is stable:

$$\bar{x}_\infty = (I-F)^{-1}Gu = \left(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} - \begin{bmatrix}0.2 & 0.4\\-0.4 & 1\end{bmatrix}\right)^{-1}\begin{bmatrix}0\\1\end{bmatrix}1 = \begin{bmatrix}2.5\\5\end{bmatrix}$$

$$P = \sum_{i=0}^{\infty}\left(F^i\right)Q\left(F^T\right)^i = \begin{bmatrix}2.881 & 3.076\\3.076 & 7.959\end{bmatrix}$$

Steady-state 일 때,

포식자는 2.5마리, 먹이는 5마리에서 수렴

4. Discrete Kalman Filter

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

1) $k = 0$

$$\hat{x}_0^+ = E(x_0)$$
$$P_0^+ = E[(x - \widehat{x_0})(x - \widehat{x_0})^T]$$

2) $k = 1, 2, \ldots$

A. A priori(predictor)

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1}$$
$$P_k^- = F_{k-1}P_{k-1}^+ F_{k-1}^T + Q_{k-1}$$

B. A posteriori(corrector)

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1}$$
$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-)$$
$$P_k = (I - K_k H_k)P_k^-(I - K_k H_k)^T + K_k R_k K_k^T$$

x: state variables
u: system input
w: process noise
white noise: $E(v_i) = 0, E(v_i^2) = \sigma_i^2, E(v_i v_j) = 0$
Q_k: covariance matrix of process noise

y: noisy measurement
v: measurement noise
white noise: $E(v_i) = 0, E(v_i^2) = \sigma_i^2, E(v_i v_j) = 0$
R_k: covariance matrix of measurement noise

$\hat{x}_k^-$: a priori estimate @ k^{th} time step
$\hat{x}_k^+$: a posteriori estimate @ k^{th} time step
K_k: correction gain @ k^{th} time step
P_k: covariance matrix of estimation error

5. Digital gyroscope & accelerometer application

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

$$\Longrightarrow$$

$$x_{k+1} = Fx_k + w_k$$
$$y_k = Hx_k + v_k$$

$$x_{k+1} = Fx_k + w_k$$

$$\Rightarrow \begin{bmatrix}\theta_{k+1}\\ \dot{\theta}_{k+1}\end{bmatrix} = \begin{bmatrix}1 & \Delta t\\ 0 & 1\end{bmatrix}\begin{bmatrix}\theta_k\\ \dot{\theta}_{gyro}\end{bmatrix} + \begin{bmatrix}w_{process}\\ w_{gyro}\end{bmatrix}$$

$$y_k = Hx_k + w_k$$

$$\Rightarrow [\theta_k] = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\theta_{acc}\\ \dot{\theta}_k\end{bmatrix} + [v_{acc}]$$

Assumptions:

1) No input(ignore random, impulse-like inputs by your finger)
2) State equation: ω from gyroscope & Euler method
3) Measurement equation: θ from accelerometer
4) $Q = \begin{bmatrix}\sigma^2_{process} & 0\\ 0 & \sigma^2_{gyro}\end{bmatrix}, R = [\sigma^2_{acc}]$
   - Uncertainties for the two sensors and Euler method

5. Digital gyroscope & accelerometer application

1) $k = 0$

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

$$P_0^+ = E[(x - \widehat{x_0})(x - \widehat{x_0})^T]$$

2) $k = 1, 2, \ldots$

A. A priori(predictor): gyroscope

$$\begin{bmatrix} \theta_k^- \\ \dot{\theta}_k^- \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{k+1}^+ \\ \dot{\theta}_{gyro} \end{bmatrix}$$

$$P_k^- = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} P_{k-1}^+ \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}^T + \begin{bmatrix} \sigma_{process}^2 & 0 \\ 0 & \sigma_{gyro}^2 \end{bmatrix}$$

B. A posteriori(corrector): accelerometer fusion

$$K_k = P_k^- [1 \quad 0]^T \left([1 \quad 0] P_k^- [1 \quad 0]^T + [\sigma_{acc}^2] \right)^{-1}$$

$$\begin{bmatrix} \theta_k^+ \\ \dot{\theta}_k^+ \end{bmatrix} = \begin{bmatrix} \theta_k^- \\ \dot{\theta}_k^- \end{bmatrix} + K_k \left(\theta_{acc} - [1 \quad 0] \begin{bmatrix} \theta_k^- \\ \dot{\theta}_k^- \end{bmatrix} \right)$$

$$P_k^+ = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - K_k [1 \quad 0] \right) P_k^- \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - K_k [1 \quad 0] \right)^T + K_k [\sigma_{acc}^2] K_k^T$$