Lecture 8.1 Discrete-Time Kalman Filtering

Discrete-time random processes

$$X[k+1] = AX[k] + \overset{\text{ignore}}{\cancel{Bu[k]}} + FV[k]$$

$$Y[k] = CX[k] + W[k]$$

V, W, X, Y discrete-time random processes

$$E\{V[k]\,V^T[l]\} = R_V\,\delta(k-l)$$

discrete time delta fcn ($\delta(0) = 1$, $\delta(m) = 0$ $m \neq 0$)

Assume zero mean, white noise. V, W independent.

Mean: $E\{X[k]\} = E\{X[0]\} + \sum \overset{0}{\cancel{E\{A^{k-i}}} BV[i]\}$

Correlation matrix:

$$R_X(k_1, k_2) = E\{X[k]\,X^T[k+d]\}$$

$$= \sum_{i=0}^{k_1-1}\sum_{j=0}^{k_2-1} A^{k_1-i} F E\{V[i]\,V^T[j]\} F^T (A^{k_j-j})^T$$

$$= \sum_{i=0}^{k_1-1} A^{k_1-i} F R_V F^T (A^{k_2-i})^T$$

$$R_X(k, k+d) = \underbrace{\left(\sum_{i=1}^{k} A^i F R_V F^T (A^i)^T\right)}_{P[k]} A^d$$

$$P[k+1] = AP[k]A^T + FR_VF^T$$

discrete-time Lyapunov eqn

Remarks

1. $R_Y(d) = CR_X(d)C^T$
2. All other definitions and results carry over (stationary, freq domain, spectral factorization, etc)

Discrete-time Kalman filter

Estimation problem:

$$\left.\begin{aligned} x[k+1] &= Ax[k] + Bu[k] + Fv[k] \\ Y[k] &= CX[k] + W[k] \end{aligned}\right\} \text{Process model}$$

$$\hat{x}[k+1] = Ax[k] + Bu[k] + L(y[k] - C\hat{x}[k])$$

Find L such that $E\{(X[k]-\hat{X}[k])(X[k]-\hat{X}[k])^T\}$ is minimized

<u>Thm</u> (Kalman) The observer gain that minimizes the error is given by

$$L[k] = A^T P[k] C^T \overbrace{(R_W + CP[k]C^T)}^{R_\varepsilon}{}^{-1}$$

$$\begin{aligned} P[k+1] &= (A-LC)P[k](A-LC)^T + FR_vF^T + LR_WL^T \\ &= AP[k]A^T + FR_vF^T - AP[k]C^T R_\varepsilon^{-1} CP[k]A^T \end{aligned} \Bigg\} \text{Equiv forms}$$

$$P[0] = E\{X[0]X^T[0]\}$$

<u>Pf</u> Same idea as continuous time: choose L to make $P[k+1]$ as small as possible

<u>Remarks</u>

1. $X[k+1]$ vs $Y[k]$ gives extra factors of A & A^T
2. As in continuous time, get <u>recursive</u> filter ($\Rightarrow$ keep $\hat{X}[k], P[k]$).
3. In steady state, get discrete time algebraic Riccati eqn
4. MATLAB: dlqe(...)

Predictor - Corrector form

Alternative formulation (see references on lecture page)

$$X[k|j] = \text{prediction of } X \text{ at time } k \text{ given measurements up to time } j \le k$$

$$P[k|j] = \text{error covariance at time } k \text{ given} \ldots$$

Rewrite Kalman filter as two step algorithm

$$\left.\begin{aligned} \hat{x}[k|k-1] &= A\hat{x}[k-1|k-1] + Bu[k] \\ P[k|k-1] &= AP[k-1|k-1]A^T + FR_v[k]F^T \end{aligned}\right\}\ \text{Prediction}$$

$$\left.\begin{aligned} (3)\quad \hat{x}[k|k] &= \hat{x}[k|k-1] + L[k]\left(y[k] - C\hat{x}[k|k-1]\right) \\ (1)\quad P[k|k] &= P[k|k-1] - P[k|k-1]C^T R_\varepsilon^{-1}[k]\,C\,P[k|k-1] \\ (2)\quad L[k] &= A^{+} P[k|k]\,C^T R_\varepsilon^{-1}[k] \qquad R_\varepsilon = CP[k|k]C^{+} + R_w \end{aligned}\right\}\ \text{Correction}$$

Remarks

1. Same final equation for $P[k] = P[k|k]$
2. By splitting, can handle case where measurement is not available on every update: <u>always</u> apply prediction, correct when new measurement is available

RMM 25 Feb 08

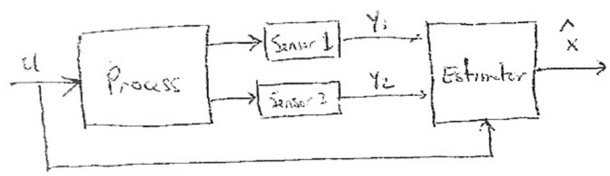
4

~~RMM 23 Jan 07~~

~~1~~

Sensor Fusion

~~Lecture 4-1: Sensor Fusion~~



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

$y = Cx + w$

Disturbances: $R_v(t)$

Noise: $R_w(t)$

Problem statement

- How do we combine multiple sensors to get best estimate?
- If different sensors are available at different times, how do we incorporate data?
- How do uncertainties in individual sensor affect the way their data is included?

Start in continuous time:

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

$$L = PC^T R_w^{-1}$$

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

Multiple sensors:

$$C = \begin{bmatrix} C_1 \\ \vdots \\ C_q \end{bmatrix} \qquad R_w = \begin{bmatrix} R_{11} & & 0 \\ & \ddots & \\ 0 & & R_{qq} \end{bmatrix}$$

← Assumes indep ~~measurements~~ noise in each channel

$$L = PC^T R_w \;\; \cancel{PC^TR_w^{-1}CP} = P\begin{bmatrix} C_1^T & \cdots & C_q^T \end{bmatrix} \begin{bmatrix} R_{11}^{-1} & & \\ & \ddots & \\ & & R_{qq}^{-1} \end{bmatrix}$$

↑ If P small ⇒ current state has good estimate ⇒ lower weight of measurement

↑ Weigh individual sensors according to <u>inverse</u> of covariance ⇒ accurate sensors have ~~lar~~ larger gain

Remarks

1. Same results in discrete time (w/ R_ε complication)

Sensor fusion, ctd

Example Weighted least squares

$$\left.\begin{aligned} x[k+1] &= x[k] + 0 \\ y[k] &= Mx[k] + W \end{aligned}\right\} \quad \text{Trying to solve } Mx = y \text{ given noisy data}$$

Single sensor with independent Gaussian noise $R_W = \sigma_1^2$.

$$\hat{x}[k+1] = \hat{x}[k] + P[k]M^T(\sigma_1^2 + \underbrace{MP[k]M^T}_{\sigma_y[k]})^{-1}(y_1 - M\hat{x})$$

$$= \hat{x}[k] + P[k]M^T \frac{1}{\sigma_1^2 + \sigma_y^2[k]}(y - M\hat{x})$$

$$= \underbrace{\left(I - \frac{\sigma_y^2}{\sigma_1^2 + \sigma_y^2} \frac{PM^TM}{MPM^T}\right)}_{\sim \frac{\sigma_1^2}{\sigma_1^2 + \sigma_y^2} \text{ (scalar)}} x[k] + \frac{\sigma_y^2}{\sigma_1^2 + \sigma_y^2} \frac{PM^TM}{MPM^T} y_1$$

$\Rightarrow$ weight measurement versus current state based on variance.

Multi-sensor case: $M = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$, $R_W = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$.

$$R_\varepsilon = \left(\begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} + \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} P \begin{bmatrix} M_1^T & M_2^T \end{bmatrix} \right) \qquad P = \begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix}$$

$$= \left(\begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} + \begin{bmatrix} M_1PM_1^T & M_1PM_2^T \\ M_2PM_1^T & M_2PM_2^T \end{bmatrix} \right)$$

$$= \left(\underbrace{\begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}}_{\text{Measurement}} + \underbrace{\begin{bmatrix} \sigma_{y_1}^2 & \sigma_{y_1 y_2}^2 \\ \sigma_{y_1 y_2}^2 & \sigma_{y_2}^2 \end{bmatrix}}_{\text{Estimate}} \right) \leftarrow PM^TR_\varepsilon^{-1} \text{ gives optimal weighting of new measurmts}$$