MECH468 : Modern Control Engineering
MECH509 : Controls

L28 : Least-squares estimation

Dr. Ryozo Nagamune
Department of Mechanical Engineering
University of British Columbia

Zoom lecture to be recorded and posted on Canvas

Course plan

Topics	CT	DT
Modeling	✓	✓
Stability	✓	✓
Controllability/observability	✓	✓
Realization	✓	✓
State feedback/observer	✓	✓
✓ LQR/Kalman filter		

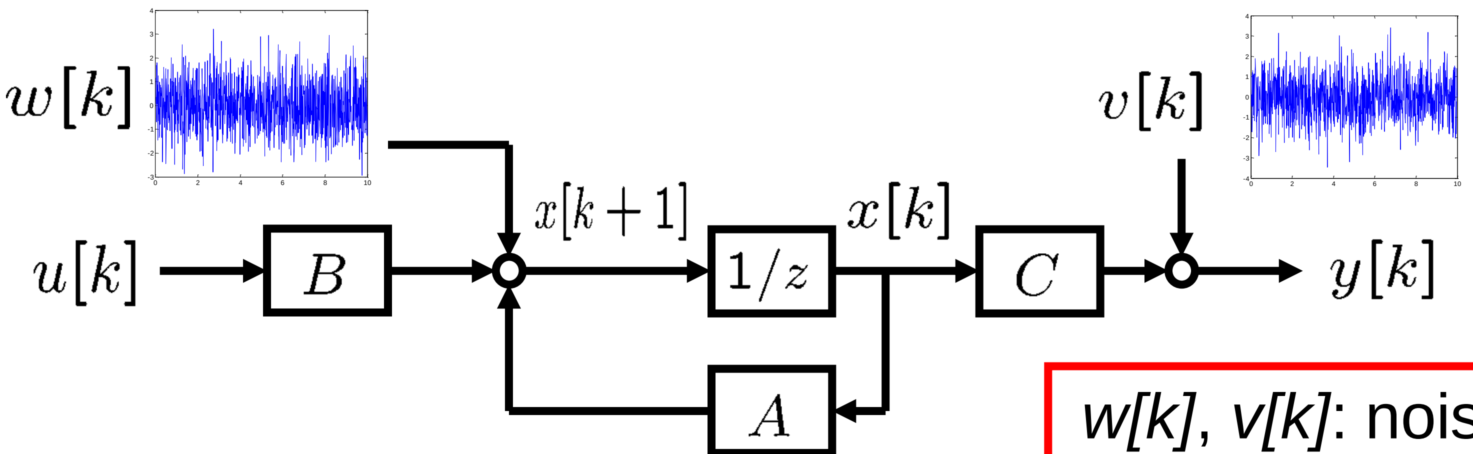
3 lectures

Outline

- Introduction to Kalman filter
- Least-squares (LS) estimation
  - Unweighted LS
  - Weighted LS
  - Recursive LS
- Kalman filter based on least-squares estimation (next lecture)

What is Kalman filter?

- For a discrete-time system: $\begin{cases} x[k+1] &= Ax[k] + Bu[k] + w[k] \\ y[k] &= Cx[k] + v[k] \end{cases}$



w[k], *v[k]*: noise

Kalman filter estimates *x[k]* at each *k* with I/O data up to time *k* in an *optimal & recursive* manner.

Remarks on Kalman filter

- Rudolf E. Kalman, “A New Approach to Linear Filtering and Prediction Problems”, *ASME Journal of Basic Engineering*, 82 (Series D), pp. 35-45, 1960.
- Numerous applications
  - Tracking (missiles, faces etc.)
  - Navigation systems (GPS, IMU)
  - Image processing, computer vision
  - Parameter identification
  - Aerospace industry (Apollo 11)
- Also called linear quadratic estimation (LQE)

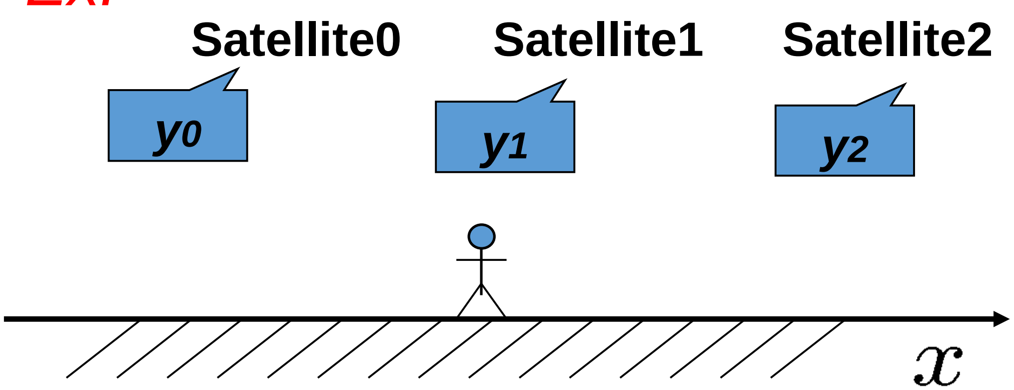
A simpler estimation problem

- First, we consider a simple problem:

$$\begin{cases} \cancel{x[k+1] = Ax[k] + Bu[k] + B_w w[k]} \\ y[\cancel{k}] = Cx[\cancel{k}] + v[\cancel{k}] \end{cases}$$

- Estimate a constant *x* from noisy $y_i, \quad i = 0, 1, \ldots, I$

Ex.



$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_I \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} C \\ C \\ \vdots \\ C \end{bmatrix}}_{H} x + \underbrace{\begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_I \end{bmatrix}}_{V}$$

Least-squares estimation

- If we have no *a priori* information on *v* (y_i, $i=0,1,\ldots$ are "equally trustable"), one natural estimate is the least-squares (LS) estimate

$$\hat{x} := \arg\min_x V^T V = \arg\min_x (Y - Hx)^T (Y - Hx)$$

- LS estimate can be computed by

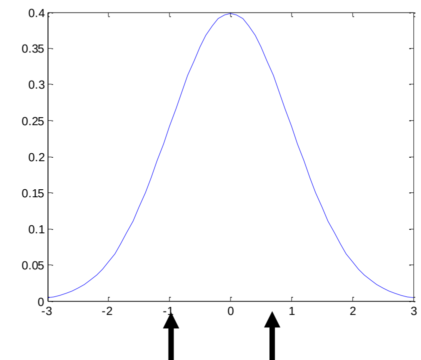
$$\hat{x} := \underbrace{(H^T H)^{-1} H^T}_{\textit{Pseudo-inverse of } H} Y$$

Error analysis of LS estimate

- Imagine that we perform many experiments (You don't need to do in reality! Just imagine!):

experiment	V	$\hat{x}$
1	$V^{(1)}$	$\hat{x}^{(1)}$
2	$V^{(2)}$	$\hat{x}^{(2)}$
$\vdots$	$\vdots$	$\vdots$

Probability Density Function



V

$V^{(2)}$ $V^{(1)}$

- Estimation error

$$\hat{x}^{(n)} - x = (H^T H)^{-1} H^T \underbrace{(Hx + V^{(n)})}_{Y^{(n)}} - x = (H^T H)^{-1} H^T V^{(n)}$$

Error analysis (cont'd)

- Expected value of estimation error

$$E\{\hat{x}-x\} := \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}(\hat{x}^{(n)}-x) = \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}(H^TH)^{-1}H^TV^{(n)} = (H^TH)^{-1}H^TE\{V\}$$

  - Expected value is zero if *E{V}=0* (i.e., unbiased noise).

- Error covariance (covariance of estimation error)

$$P := \underbrace{E\left\{(\hat{x}-x)(\hat{x}-x)^T\right\}}_{} = (H^TH)^{-1}H^TE\left\{VV^T\right\}H(H^TH)^{-1}$$

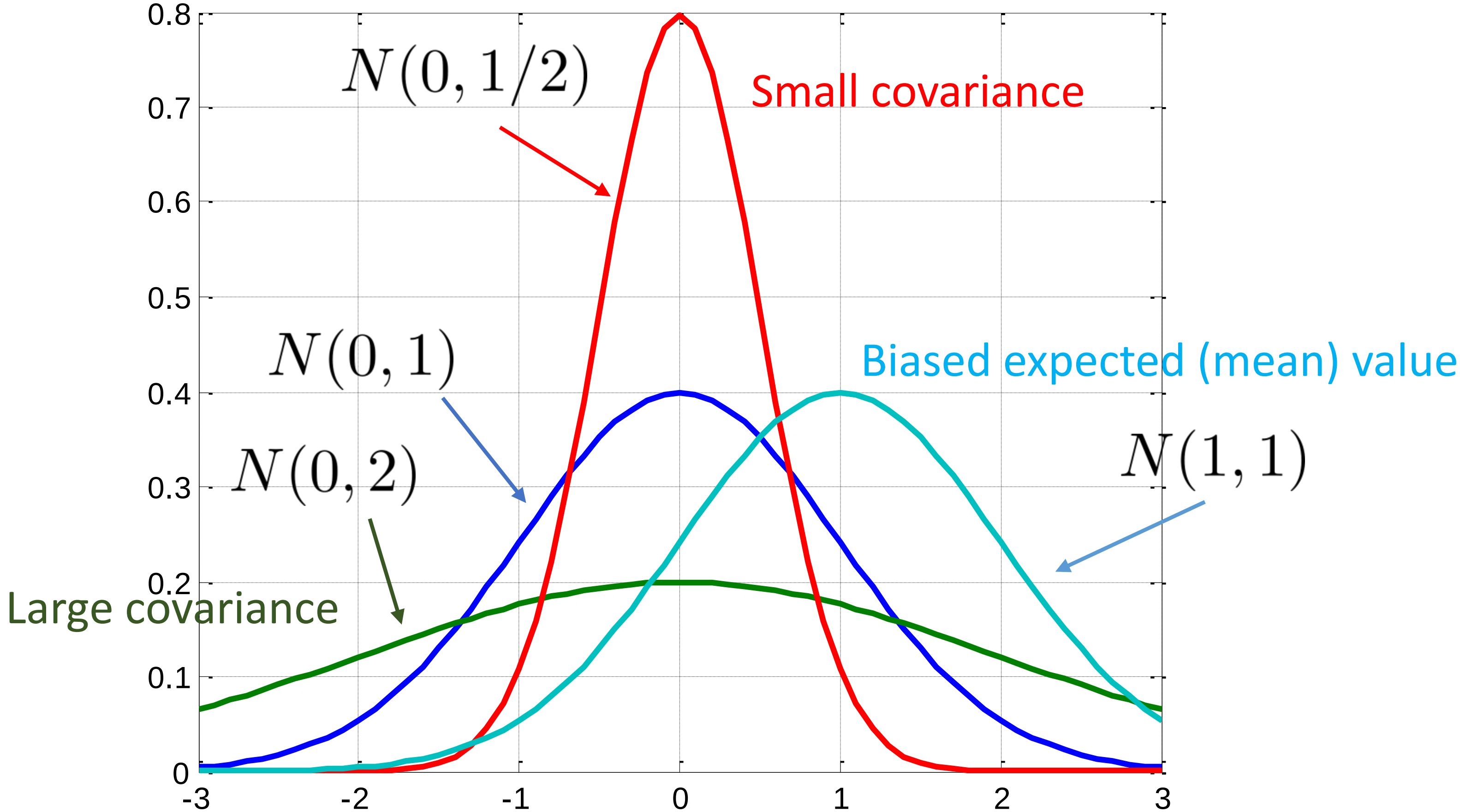
measure of "how much trustable the estimate is"
(small / large covariance → more / less trustable)

Various probability density functions

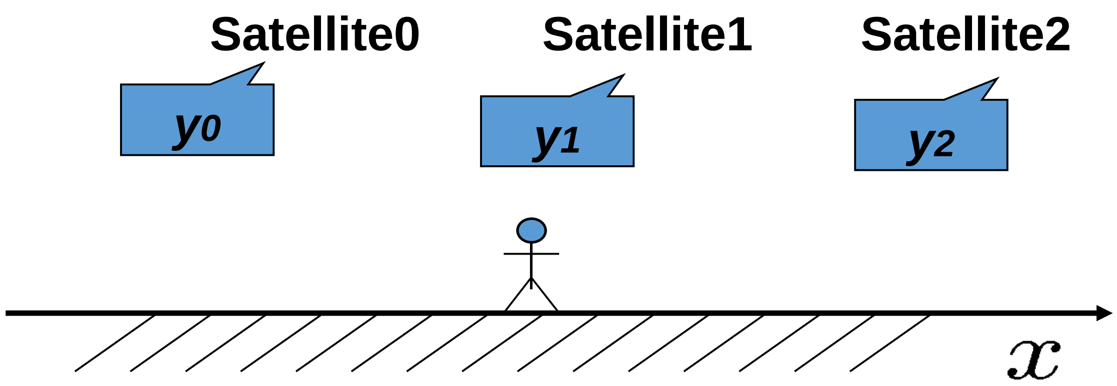
$N(\mu, \sigma)$

μ : mean value

σ : standard deviation $= \sqrt{\text{cov}}$



A simple example
Estimate the human standing position



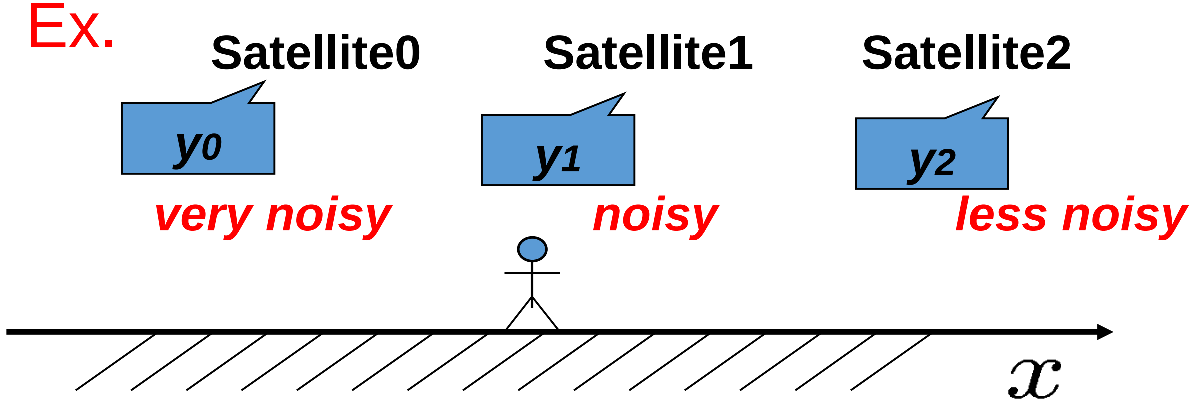
$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{H} x + \underbrace{\begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}}_{V} \Rightarrow \hat{x} = (H^T H)^{-1} H^T Y = \frac{y_0 + y_1 + y_2}{3}$$

$$\Rightarrow E\left\{(\hat{x} - x)(\hat{x} - x)^T\right\} = \frac{1}{9} H^T E\left\{V V^T\right\} H = \begin{cases} \frac{1}{3} & \text{if } E\left\{V V^T\right\} = I_3 \\ \frac{2}{3} & \text{if } E\left\{V V^T\right\} = 2 I_3 \end{cases}$$

Weighted LS estimation

- Suppose that we have *a priori* information on v (accuracy of y_i, i=0,1,...) as $R_V := E\left\{VV^T\right\}$

Ex.



$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{H} x + \underbrace{\begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}}_{V} \qquad R_V = E\left\{VV^T\right\} = \begin{bmatrix} 3 & & \\ & 2 & \\ & & 1 \end{bmatrix}$$

Weighted LS estimation (cont'd)

- It is natural to weight the error V so that more accurate measurement can influence the cost.
- The weighted LS estimate (WLS) is defined by

$$\hat{x} := \arg\min_x V^T R_V^{-1} V = \arg\min_x (Y - Hx)^T R_V^{-1} (Y - Hx)$$

- WLS estimate can be computed by

$$\hat{x} := (H^T R_V^{-1} H)^{-1} H^T R_V^{-1} Y$$

Error analysis of WLS estimate

- Imagine that we perform many experiments (You don't need to do in reality! Just imagine!):

experiment	V	$\hat{x}$
1	$V^{(1)}$	$\hat{x}^{(1)}$
2	$V^{(2)}$	$\hat{x}^{(2)}$
$\vdots$	$\vdots$	$\vdots$

- Estimation error

$$\hat{x}^{(n)} - x = (H^T R_V^{-1} H)^{-1} H^T R_V^{-1} \underbrace{(Hx + V^{(n)})}_{Y^{(n)}} - x = (H^T R_V^{-1} H)^{-1} H^T R_V^{-1} V^{(n)}$$

Error analysis (cont'd)

- Expected value of estimation error

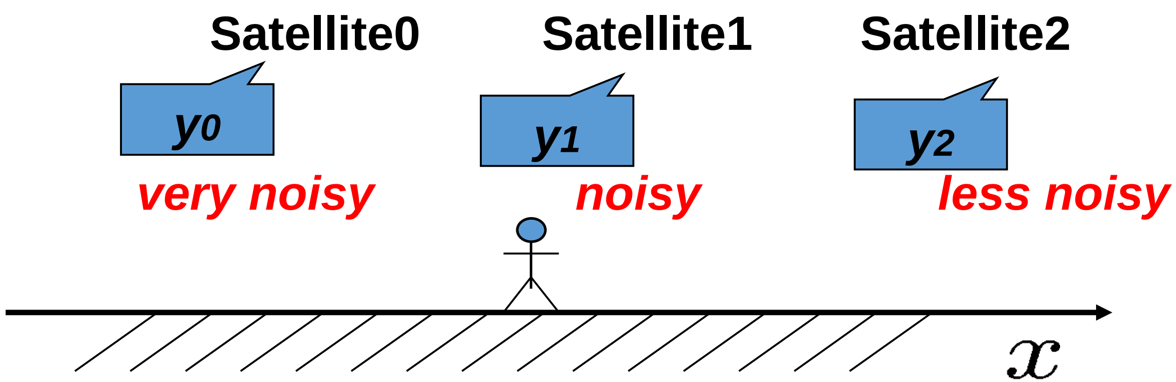
$$E\{\hat{x}-x\} := \lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}(\hat{x}^{(n)}-x) = (H^T R_V^{-1} H)^{-1} H^T R_V^{-1} E\{V\}$$

  - Expected value is zero if *E{V}=0* (i.e. unbiased noise).

- Error covariance (covariance of estimation error)

$$\begin{aligned} P &:= E\left\{(\hat{x}-x)(\hat{x}-x)^T\right\} \\ &= (H^T R_V^{-1} H)^{-1} H^T R_V^{-1} \underbrace{E\left\{V V^T\right\}}_{R_V} R_V^{-1} H (H^T R_V^{-1} H)^{-1} \\ &= (H^T R_V^{-1} H)^{-1} \end{aligned}$$

A simple example
Estimate the human standing position



$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{H} x + \underbrace{\begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}}_{V}$$

$$R_V = E\left\{VV^T\right\} = \begin{bmatrix} 3 & & \\ & 2 & \\ & & 1 \end{bmatrix}$$

$$\Rightarrow$$

$$\begin{aligned} \hat{x} &= (H^T R_V^{-1} H)^{-1} H^T R_V^{-1} Y \\ &= \left(\frac{1}{3} + \frac{1}{2} + 1\right)^{-1} \left(\frac{1}{3}y_0 + \frac{1}{2}y_1 + y_2\right) \\ &= \frac{2y_0 + 3y_1 + 6y_2}{11} \end{aligned}$$

$$P = \frac{6}{11}$$

Recursive LS estimation Motivation

- Suppose that we have weighted LS estimate:

$$\underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_I \end{bmatrix}}_{Y_o} = \underbrace{\begin{bmatrix} C \\ C \\ \vdots \\ C \end{bmatrix}}_{H_o} x + \underbrace{\begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_I \end{bmatrix}}_{V_o} \Rightarrow \begin{cases} \hat{x}_o := (H_o^T R_{V_o}^{-1} H_o)^{-1} H_o^T R_{V_o}^{-1} Y_o \\ P_o = (H_o^T R_{V_o}^{-1} H_o)^{-1} \end{cases}$$

$$R_{V_o} := E\left\{V_o V_o^T\right\}$$

"old"

- Now, we add "new data" to "old data".

$$\begin{bmatrix} Y_o \\ Y_n \end{bmatrix} = \begin{bmatrix} H_o \\ H_n \end{bmatrix} x + \begin{bmatrix} V_o \\ V_n \end{bmatrix} \qquad \begin{aligned} R_{V_o} &:= E\left\{V_o V_o^T\right\} \\ R_{V_n} &:= E\left\{V_n V_n^T\right\} \end{aligned}$$

"new"

- How to compute new estimate & covariance?

Recursive LS estimation Motivation (cont'd)

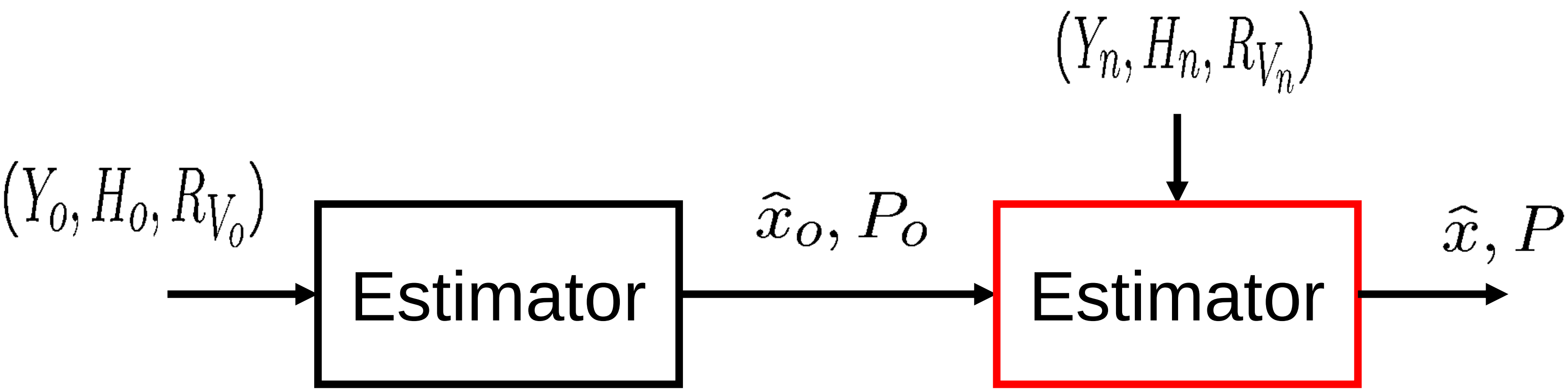
- Batch process (see Slide 13 & Slide 15)

$$\begin{cases} \hat{x} = \left(\begin{bmatrix} H_o \\ H_n \end{bmatrix}^T \begin{bmatrix} R_{V_o}^{-1} & \\ & R_{V_n}^{-1} \end{bmatrix} \begin{bmatrix} H_o \\ H_n \end{bmatrix} \right)^{-1} \begin{bmatrix} H_o \\ H_n \end{bmatrix}^T \begin{bmatrix} R_{V_o}^{-1} & \\ & R_{V_n}^{-1} \end{bmatrix} \begin{bmatrix} Y_o \\ Y_n \end{bmatrix} \\ P = \left(\begin{bmatrix} H_o \\ H_n \end{bmatrix}^T \begin{bmatrix} R_{V_o}^{-1} & \\ & R_{V_n}^{-1} \end{bmatrix} \begin{bmatrix} H_o \\ H_n \end{bmatrix} \right)^{-1} \end{cases}$$

Recursive LS estimation Motivation (cont'd)

- Recursive process (Derivation in Appendix)

$$
\begin{cases}
\hat{x} = \hat{x}_o + P H_n^T R_{V_n}^{-1} (Y_n - H_n \hat{x}_o) \\
P = \left(P_o^{-1} + H_n^T R_{V_n}^{-1} H_n \right)^{-1}
\end{cases}
$$



No need to use "old data"!

Recursive LS algorithm

Initialization: $\hat{x}_o$: Initial estimate P_o: Initial covariance

Step 1: Compute new covariance $P = \left(P_o^{-1} + H_n^T R_{V_n}^{-1} H_n\right)^{-1}$

Step 2: Compute new LS estimate $\hat{x} = \hat{x}_o + P H_n^T R_{V_n}^{-1}(Y_n - H_n \hat{x}_o)$

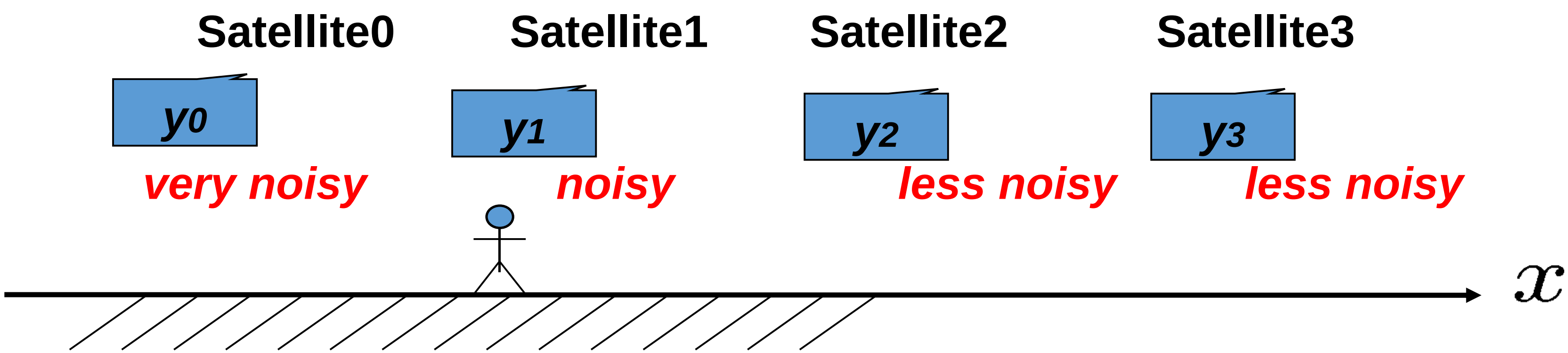
Repeat Steps 1 & 2 for new data.

$$\left.\begin{array}{r}\textbf{Accurate } \widehat{x}_o \\ P_o \approx 0\end{array}\right\} \Longrightarrow P \approx 0 \Longrightarrow \hat{x} \approx \hat{x}_o$$

WLS estimate for new data

$$\left.\begin{array}{r}\textbf{Poor } \widehat{x}_o \\ P_o\text{: large}\end{array}\right\} \Longrightarrow P \approx (H_n^T R_{V_n}^{-1} H_n)^{-1} \Longrightarrow \hat{x} \approx \left(H_n^T R_{V_n}^{-1} H_n\right)^{-1} H_n^T R_{V_n}^{-1} Y_n$$

A simple example
Estimate the human standing position



- Old estimate & cov. (Slide 16) $\hat{x}_o = \dfrac{2y_0 + 3y_1 + 6y_2}{11} \quad P_o = \dfrac{6}{11}$

- New data $\quad Y_n = y_3 \quad H_n = 1 \quad R_{V_n} = 1$
- Updated estimate & cov.

$$P = \left(P_o^{-1} + H_n^T R_{V_n}^{-1} H_n\right)^{-1} = \frac{6}{17} \qquad \begin{aligned} \hat{x} &= \hat{x}_0 + P H_n^T R_{V_n}^{-1}(Y_n - H_n \hat{x}_0) \\ &= (1 - P)\hat{x}_0 + P y_3 \end{aligned}$$

Summary

- Least squares estimation
- Weighted least squares estimation
- Recursive least squares estimation
- Error analysis
  - Expected value
  - Error covariance
- Next,
  - Discrete-time Kalman filter

Recursive LS: derivation

- Suppose $\hat{x}_o$: weighted LS estimate for "old" data

$$H_o^T R_{V_o}^{-1} H_o \hat{x}_o = H_o^T R_{V_o}^{-1} Y_o \quad \text{..............} \quad \text{(A)}$$

- We want to write $\hat{x} = \hat{x}_0 + \delta\hat{x}$ .... (B)
- By substituting (A) & (B) into

$$\begin{bmatrix} H_o \\ H_n \end{bmatrix}^T \begin{bmatrix} R_{V_o}^{-1} & \\ & R_{V_n}^{-1} \end{bmatrix} \begin{bmatrix} H_o \\ H_n \end{bmatrix} \hat{x} = \begin{bmatrix} H_o \\ H_n \end{bmatrix}^T \begin{bmatrix} R_{V_o}^{-1} & \\ & R_{V_n}^{-1} \end{bmatrix} \begin{bmatrix} Y_o \\ Y_n \end{bmatrix}$$

$$\Longleftrightarrow \quad H_o^T R_{V_o}^{-1} H_o \hat{x} + H_n^T R_{V_n}^{-1} H_n \hat{x} = H_o^T R_{V_o}^{-1} Y_o + H_n^T R_{V_n}^{-1} Y_n$$

we have

$$H_o^T R_{V_o}^{-1} H_o (\cancel{\hat{x}_o} + \delta\hat{x}) + H_n^T R_{V_n}^{-1} H_n (\hat{x}_o + \delta\hat{x}) = \cancel{H_o^T R_{V_o}^{-1} Y_o} + H_n^T R_{V_n}^{-1} Y_n$$

Recursive LS: derivation (cont'd)

- Thus, $\delta\hat{x} = \big[\underbrace{H_o^T R_{V_o}^{-1} H_o}_{P_o^{-1}} + H_n^T R_{V_n}^{-1} H_n\big]^{-1} H_n^T R_{V_n}^{-1}(Y_n - H_n\hat{x}_o)$

  $P_o := \left(H_o^T R_{V_o}^{-1} H_o\right)^{-1}$: covariance of the estimate

- Covariance for batch processing is (from Slide 18)

$$P = \left(\begin{bmatrix} H_o \\ H_n \end{bmatrix}^T \begin{bmatrix} R_{V_o}^{-1} & \\ & R_{V_n}^{-1} \end{bmatrix} \begin{bmatrix} H_o \\ H_n \end{bmatrix} \right)^{-1} = \left(P_o^{-1} + H_n^T R_{V_n}^{-1} H_n\right)^{-1}$$

- Therefore,

$$\begin{aligned} \hat{x} &= \hat{x}_o + \delta\hat{x} \\ &= \underbrace{\hat{x}_o}_{\text{Estimate from old data}} + \underbrace{P H_n^T R_{V_n}^{-1}}_{\text{Gain}} \underbrace{(Y_n - H_n\hat{x}_o)}_{\text{Error between measured “y” and estimated “y”}} \end{aligned}$$