ASEN 5044, Fall 2024
Statistical Estimation for Dynamical Systems

Lecture 22:
Non-deterministic Optimal State Estimation; Batch LLS Estimator and Properties

Prof. Nisar Ahmed (Nisar.Ahmed@Colorado.edu)

Thursday 10/24/2024

Ann and H.J. Smead
Aerospace Engineering Sciences
UNIVERSITY OF COLORADO **BOULDER**

Announcements

HW 6 out, due **NEXT Fri Nov 1** via Gradescope

Quiz 6 to be released this Fri /tomorrow (postponed from last week) – due Tues

- **Midterm 1 grades + solutions will be posted today**

- **Midterm 2: will be released Thurs Nov 7, due Thurs Nov 14**

Overview

Last Time:

- Specifying CT LTI meas noise V PSD + convert to DT LTI meas noise covar R
- **Van Loan's method**: how to actually compute DT covar Q from CT PSD W?
  - Example for 1D robot car
- DT Additive white Gaussian noise (AWGN)
- Monte Carlo simulations of DT stochastic systems with AWGN

Today:

- Intro to "optimal" non-deterministic state estimation
- Set up **Linear least squares (LLS) estimation problem**
  - Basic version: estimating static/constant states from noisy measurements
- Batch solution to LLS (**Batch LLS**) estimator derivation
- LLS estimator properties: estimator error, bias, and error covariance matrix

Introduction to Estimation Problems with Noise

- We now know how to model and simulate stochastic linear systems in DT in their most general state space form:

$$x(k+1) = Fx(k) + Gu(k) + w(k)$$
$$y(k) = Hx(k) + v(k)$$

$$w(k) = \text{process noise (white: } E[w(k)] = 0,\ E[w(k)w^T(j)] = Q \cdot \delta(k,j))$$
$$v(k) = \text{measurement noise (white: } E[v(k)] = 0,\ E[v(k)v^T(j)] = R \cdot \delta(k,j))$$

- Knowing (F,G,H,M,Q,R) lets us mathematically **forward predict** how dynamical systems could behave and what y(k) could be produced in response to:
  - some set of known initial conditions x(0),
  - some set of known control inputs u(k),
  - some set of stochastic/random inputs w(k) and v(k)

State Estimation Problems in the Presence of Noise

$$x(k+1) = Fx(k) + Gu(k) + w(k),$$
$$y(k) = Hx(k) + v(k),$$
$$w(k),\ v(k) = \text{AWGN}$$

- But how to solve the "inverse problem"?
- That is, suppose all you know is (F,G,H,M,Q,R) and **data from the DT system**

  {y(1), y(2), …,y(T)} for T time steps k=1,…,T (noisy measurements)

  {u(0), u(1),…,u(T-1)} (known control inputs)

- Can we use this noisy data to **reconstruct** x(0), x(1), …,x(T)?
- Recall: if there were no process/measurement noise, then can do so exactly if the system (F,H) is observable: a unique sequence of x(0), x(1), …, x(T) exists to perfectly explain the observed data deterministically
- But with noise: we will never know true values of random vectors w(k) and v(k)!
- So even if (F,H) observable, still infinitely many solutions for x(0),…,x(T) which explain the data! → **unique deterministic solution not possible**!!!

The Non-Deterministic State Estimation Problem

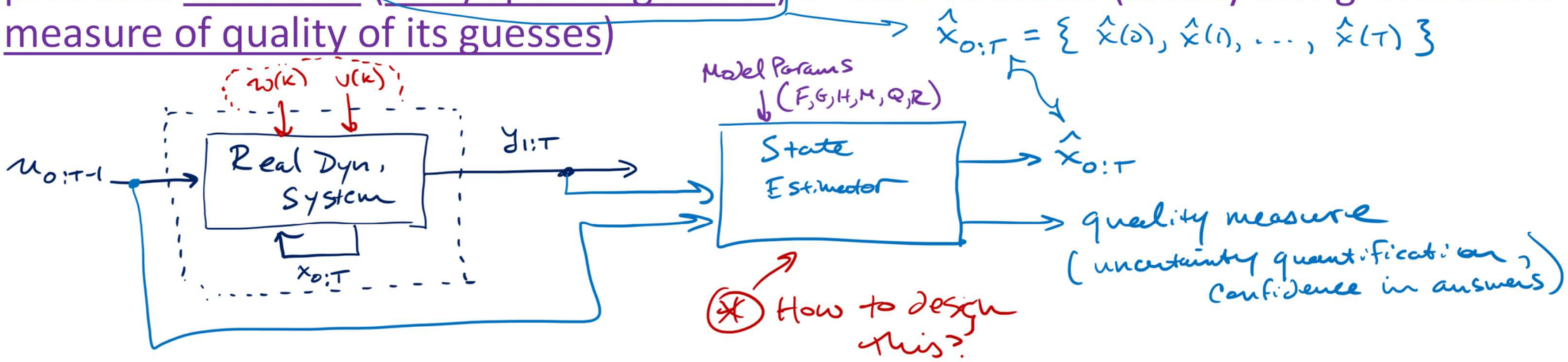
$$x(k+1) = Fx(k) + Gu(k) + w(k),$$
$$y(k) = Hx(k) + v(k),$$
$$w(k),\ v(k) = \text{AWGN}$$

Problem: Given $y_{1:T} = \{y(1), ..., y(T)\}$ and $u_{0:T-1} = \{u(0), ..., u(T-1)\}$,

Find: $x_{0:T} = \{x(0), ..., x(T)\}$

So must devise a "best guess", i.e. "optimal estimate" of $x_{0:T}$ based on what we do know

→we need an estimator: a function of sensor data y, inputs u, and (F,G,H,M,Q,R) that produces estimates (best/optimal guesses) of desired states (ideally along with some measure of quality of its guesses)



DT Static State Estimation (No Dynamics, Sensor Noise Only)

- **Let's start with simplest case (which also turns out to be quite powerful/useful)**
- Suppose state x is constant, i.e. process noise w(k) = 0 always and F = I, so we know that x(0)=x(1)=…x(T) = x for all time steps k=0,1,2,…,T
- Suppose we also know that E[v(k)] = 0 and cov[v(k)] = R(k) (could vary in time)
- **How should we go about estimating x from y(1),…,y(T)?**

Since $y_k = H x_k + v_k$ → $y_1 = Hx + v_1$, $y_2 = Hx + v_2$, …, $y_T = Hx + v_T$

→ Note! $v_k = y_k - Hx$ (actual phys. meas.) (model of meas. for given x)

measurement error or measurement residual

$x \in \mathbb{R}^n$, $H \in \mathbb{R}^{p \times n}$, $y \in \mathbb{R}^?$, $v \in \mathbb{R}^p$

(*) Intuitively: a good est. note $\hat{x}$ will make the est. noise $v_k = y_k - H\hat{x}$ values as small as possible

(*) Define the Linear Least Squares (LLS) Cost Function:

$$J(x) = \sum_{k=1}^{T} [y_k - Hx]^T (R_k)^{-1} [y_k - Hx]$$

(v_k)

(simple scalar for some choice of x w.r.t. given y data) $= \sum_{k=1}^{T} \| y_k - Hx \|^2_{R_k^{-1}} = \sum_{k=1}^{T} \| v_k \|^2_{R_k^{-1}}$

Sum of Mahalanobis distance for each v_k (Sum of weighted 2-norms) cov.

scalar

allows more accurate (less noisy) data y_k to count more

If v_k really were AWGN: $v_k \sim N_{v_k}(0, R_k)$

→ to get $\hat{x}$: want to minimize J(T) w.r.t. x

→ $\hat{x}_{LS} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} J(T)$

Batch Linear Least Squares (LLS) Static State Estimation

- To find analytical solution for estimate of x, re-write LLS cost function in "vectorized" (stacked vector-matrix) form
- Recall:

$$x_{k+1} = x_k = x \text{ (constant, no process noise)}$$

$$y_k = Hx + v_k, \quad v_k \sim \mathcal{N}(0, R_k) \text{ (awgn)}$$

- Define:

$$\underset{[Tp\times 1]}{\vec{y}} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} \quad \underset{[Tp\times n]}{\mathbf{H}} = \begin{bmatrix} H \\ \vdots \\ H \end{bmatrix} \quad \underset{[Tp\times 1]}{\vec{v}} = \begin{bmatrix} v_1 \\ \vdots \\ v_T \end{bmatrix} \quad \underset{[Tp\times Tp]}{\mathbf{R}} = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_T \end{bmatrix}$$

- LLS Cost Function:

$$J(T) = \sum_{k=1}^{T} (y_k - Hx)^T R_k^{-1} (y_k - Hx)$$

$$= (\vec{y} - \mathbf{H}x)^T \mathbf{R}^{-1} (\vec{y} - \mathbf{H}x) \quad \left(= \vec{v}^T \mathbf{R}^{-1} \vec{v} \right)$$

"best"/optimal estimate (according to LLS criterion) minimizes (noise weighted) norm of meas residuals

- "Batch" LLS Optimal Estimate:

$$\hat{x}_{\text{LS}} = \arg\min_{x \in \mathbb{R}^n} J(T)$$

Batch LLS State Estimation

- To minimize $J(T)$ [scalar] w.r.t. $x \in \mathbb{R}^n$ (vector), necessary condition for optimality is:

$$\nabla_x J(T) = \begin{bmatrix} \frac{\partial J}{\partial x_1} \\ \vdots \\ \frac{\partial J}{\partial x_n} \end{bmatrix} = \vec{0}$$

i.e. the gradient vector of $J(T)$ w.r.t. x must vanish
$\rightarrow n$ vector equations with n unknowns!

$\rightarrow$ Note: $J(T) = (\vec{y} - \mathbf{H}x)^T \mathbf{R}^{-1}(\cdots)$

$$= \vec{y}^T \mathbf{R}^{-1} \vec{y} - \vec{y}^T \mathbf{R}^{-1} \mathbf{H} x - (\mathbf{H}x)^T \mathbf{R}^{-1} \vec{y} + (\mathbf{H}x)^T \mathbf{R}^{-1} \mathbf{H} x$$

$$= \vec{y}^T \mathbf{R}^{-1} \vec{y} - \underbrace{\vec{y}^T \mathbf{R}^{-1} \mathbf{H}}_{M_1} x - x^T \underbrace{\mathbf{H}^T \mathbf{R}^{-1} \vec{y}}_{M_2} + x^T \underbrace{\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}}_{M_3} x$$

$$= \vec{y}^T \mathbf{R}^{-1} \vec{y} - \mathbf{M_1} x - x^T \mathbf{M_2} + x^T \mathbf{M_3} x = J(T)$$

FACTS: Can show that:

(i) $\frac{\partial (x^T \mathbf{M} x)}{\partial x} = 2\mathbf{M}x$ (for symmetric $\mathbf{M}$) & (ii) $\frac{\partial (\mathbf{M}^T x)}{\partial x} = \frac{\partial (x^T \mathbf{M})}{\partial x} = \mathbf{M}$ (for any $\mathbf{M}$)

Batch LLS State Estimation

- So: $\frac{\partial J(T)}{\partial x} = \frac{\partial}{\partial x}\left[\vec{y}^T\mathbf{R}^{-1}\vec{y} - \mathbf{M_1}x - x^T\mathbf{M_2} + x^T\mathbf{M_3}x\right] = \vec{0}$ (nec. condition for optimality)

[using facts (i) and (ii)] $= \vec{0} - \mathbf{M_1}^T - \mathbf{M_2} + 2\mathbf{M_3}x$

[using definitions of M_1, M_2, and M_3] $= -2\mathbf{H}^T\mathbf{R}^{-1}\vec{y} + 2\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}x$

$= -2\mathbf{H}^T\mathbf{R}^{-1}(\vec{y} - \mathbf{H}x) = \vec{0}$

[system of linear equations with n unknown elements of $x \in \mathbb{R}^n$]

$\rightarrow$ Simplify and rearrange: $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}x = \mathbf{H}^T\mathbf{R}^{-1}\vec{y}$

$\rightarrow$ If the matrix $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is full rank $\rightarrow (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$ exists! (invertible)

$\rightarrow$ So solving for x gives the LLS optimal estimate:

$$\hat{x}_{LS} = \left[(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\vec{y}\right]$$

Batch LLS Example: Non-stationary Noise Variance

- Suppose GPS noise for 1D robot positioning problem is not constant, but varies with k

$$y_k = x + v_k, \quad E[v_k] = 0, \quad E[v_k^2] = \sigma_k^2, \quad k = 1, ..., T$$

$$\text{stack and vectorize data:} \quad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} x + \begin{bmatrix} v_1 \\ \vdots \\ v_T \end{bmatrix} = \mathbf{H}x + \vec{v}$$

$$\mathbf{R} = \text{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_T^2) = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_T^2 \end{bmatrix} \rightarrow \mathbf{R}^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_T^2} \end{bmatrix}$$

$\rightarrow$ *Batch LLS optimal estimate* is:

$$\hat{x}_{LS} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\vec{y} = \left([1, \cdots, 1] \begin{bmatrix} \frac{1}{\sigma_1^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_T^2} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right)^{-1} \cdot [1, \cdots, 1] \cdot \begin{bmatrix} \frac{1}{\sigma_1^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_T^2} \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix}$$

Noise weighted average of measurements

$$\rightarrow \hat{x}_{LS} = \left(\sum_{k=1}^{T} \frac{1}{\sigma_k^2} \right)^{-1} \cdot \left(\frac{y_1}{\sigma_1^2} + \frac{y_2}{\sigma_2^2} + \cdots + \frac{y_T}{\sigma_T^2} \right)$$

$\rightarrow$ if $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_T^2$, then this reduces to:

$$\hat{x}_{LS} = \frac{1}{T} \sum_{k=1}^{T} y_k$$

Simple average of measurements

Batch LLS State Estimator Behavior

- Examine 1D GPS localization example with R(k) = R = constant w.r.t. time k
- Suppose we get estimates from T = 100 measurements on 10 independent trials
- Will we get exactly the same batch LLS estimates on every single trial?

$$\hat{x}_{LS}^{1} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\vec{y}^{1}$$
$$\vdots$$
$$\hat{x}_{LS}^{10} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\vec{y}^{10}$$

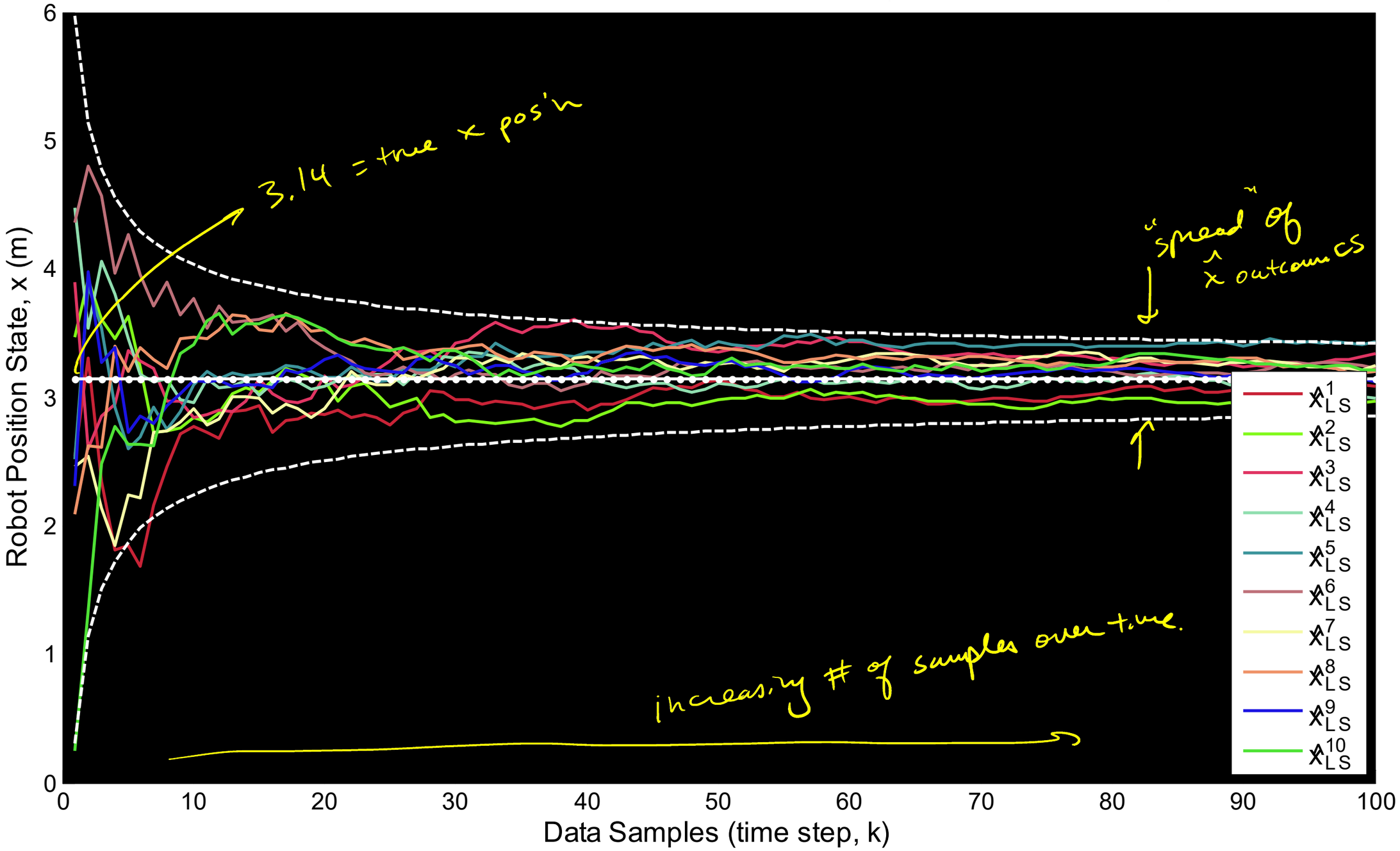
i.e. does $\hat{x}_{LS}^{1} = \hat{x}_{LS}^{2} = \dots \hat{x}_{LS}^{10}$?

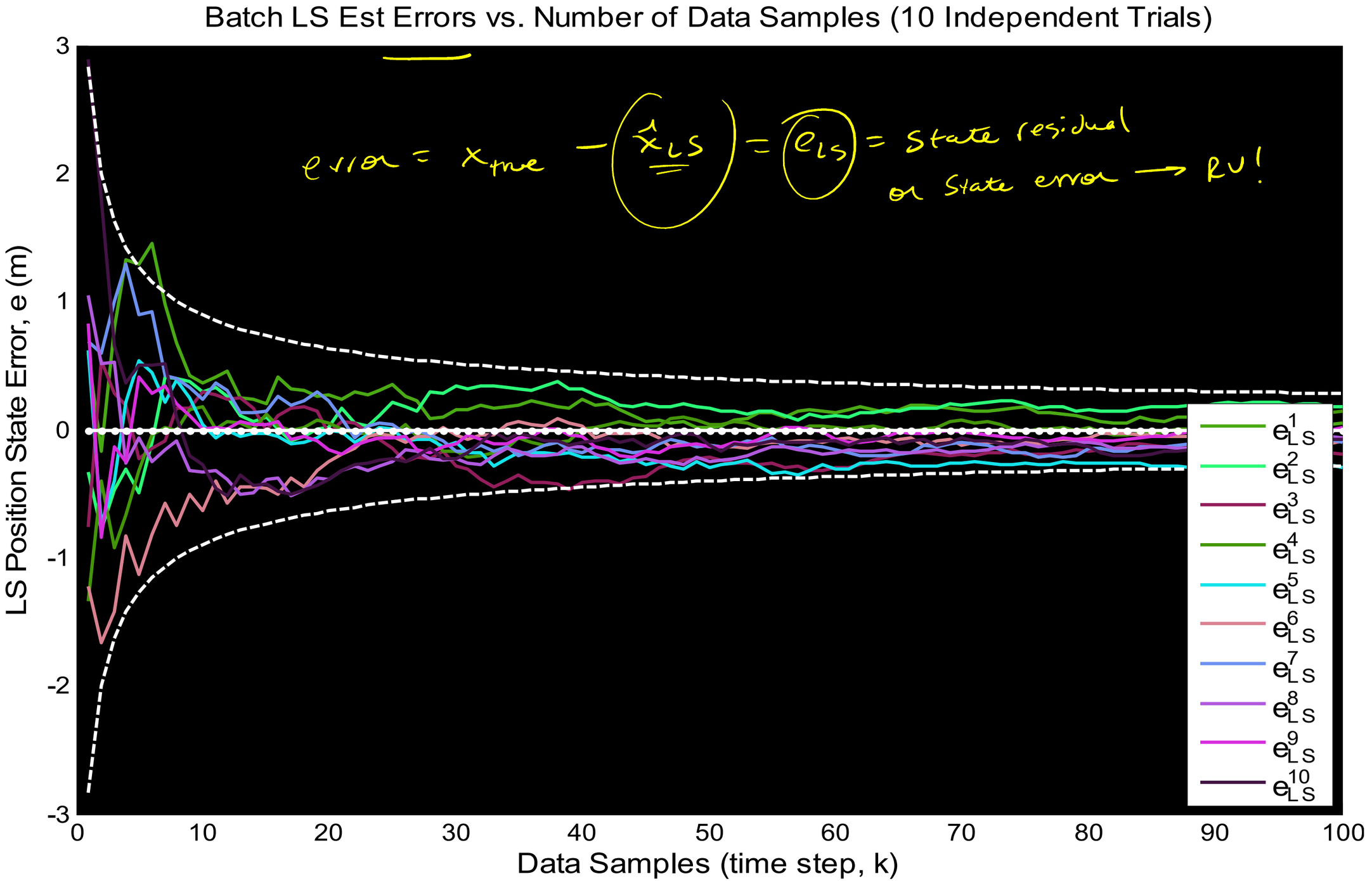
→ NO b/c generally $\vec{y}^1, \vec{y}^2, \dots, \vec{y}^{10}$ are NOT the same (each randomly corrupted by meas. errors $\vec{v}_k$)

→ $\vec{y}$ inputs to $\hat{x}_{LS}$ are Random Vectors

→ ∴ $\hat{x}_{LS}$ must also be a Random Vector!

→ How to characterize randomness in $\hat{x}_{LS}$?


Batch LS Estimates vs. Number of Data Samples (10 Independent Trials)
Robot Position State, x (m)
Data Samples (time step, k)
0
1
2
3
4
5
6
0
10
20
30
40
50
60
70
80
90
100
3.14 = true x pos'n
"spread" of x outcomes
increasing # of samples over time.
$\hat{x}^1_{LS}$
$\hat{x}^2_{LS}$
$\hat{x}^3_{LS}$
$\hat{x}^4_{LS}$
$\hat{x}^5_{LS}$
$\hat{x}^6_{LS}$
$\hat{x}^7_{LS}$
$\hat{x}^8_{LS}$
$\hat{x}^9_{LS}$
$\hat{x}^{10}_{LS}$


Batch LS Est Errors vs. Number of Data Samples (10 Independent Trials)
error = $x_{true} - \hat{x}_{LS} = e_{LS}$ = State residual or State error → RV!
LS Position State Error, e (m)
Data Samples (time step, k)
e^1_{LS}
e^2_{LS}
e^3_{LS}
e^4_{LS}
e^5_{LS}
e^6_{LS}
e^7_{LS}
e^8_{LS}
e^9_{LS}
e^{10}_{LS}

State Estimation Error

- Very important to understand "how centered" & "how uncertain" estimate of x is
- Define the **estimation error vector**: $e_{LS} = x - \hat{x}_{LS}$
- Note that we can plug in the formula for LS estimator to obtain an expression for **theoretical error as a random vector**:

$$e_{LS} = x - \hat{x}_{LS} = x - (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\vec{y}$$

$$= x - (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})(\mathbf{H}x + \vec{v})$$

$$= x - (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})x - (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\vec{v}$$

$$= x - x - (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\vec{v}$$

$$= -(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\vec{v} = e_{LS}$$

State Estimation Error Bias

- Very important to understand "**how centered**" & "how uncertain" estimate of x is
- **Estimation error vector**: $e_{(\cdot)} = x - \hat{x}_{(\cdot)}$ (for estimator optimal w.r.t. $(\cdot)$ criterion)
- "Centeredness" understood as the **estimation error bias (mean estimation error):**

$$E[e_{(\cdot)}] = E[x - \hat{x}_{(\cdot)}] = \text{estimator error bias}$$

$$\rightarrow \text{if } E[e_{(\cdot)}] = 0, \text{ then estimator } \hat{x}_{(\cdot)} \text{ is said to be } \mathbf{unbiased}$$

Is $\hat{x}_{LS}$ unbiased?

$$\begin{aligned} \rightarrow E[e_{LS}] = E[x - \hat{x}_{LS}] &= E[-(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\vec{v}] \\ &= -(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\ E[\vec{v}] \\ &= -(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\ \vec{0} \\ &= 0 \text{ (so, yes: } \hat{x}_{LS} \text{ unbiased!!)} \end{aligned}$$

State Estimation Error Covariance

- Very important to understand "how centered" & "**how uncertain**" estimate of x is
- Uncertainty understood as the **estimation error covariance (i.e. error spread)**

$$P_{(\cdot)} \equiv E[(x - \hat{x}_{(\cdot)})(x - \hat{x}_{(\cdot)})^T] = E[e_{(\cdot)} e_{(\cdot)}^T] = \text{estimator error covar matrix}$$

- Use theoretical LS estimator error to get P_{LS} (error covariance matrix of LS estimator):

$\rightarrow$ From definition of above, it follows for $\hat{x}_{LS}$ that:

$$E[(x - \hat{x}_{LS})^T(\cdots)] = E[e_{LS}\, e_{LS}^T] = E[(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\vec{v}\vec{v}^T(\cdots)^T(\cdots)^{-T}]$$

$$= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})E[\vec{v}\vec{v}^T](\cdots)^T(\cdots)^{-T}$$

$$= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1})\,\mathbf{R}\,(\mathbf{H}^T\mathbf{R}^{-1})^T(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-T}$$

$$= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-T}$$

$$= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-T} = [(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}]^T \boxed{= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1} = E[e_{LS}\, e_{LS}^T] = P_{LS}}$$

$n \times n$

$P_{LS} = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma_n^2 \end{bmatrix}$ $\rightarrow$ variances & covariances for estimation errors for each state

$\rightarrow$ get $2\sigma, 3\sigma$, etc. conf. bounds for each element of $\hat{x}_{LS}$!