ASEN 5044, Fall 2024
Statistical Estimation for Dynamical Systems

Lecture 23: Recursive Linear Least Squares (RLLS) for Static States

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

Tuesday 10/29/2024

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

Announcements

Quiz 6 due tomorrow (Wed 10/30)

HW 6 due this **Fri 11/01** via Gradescope

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

- **Cover HWs 5-7, Quizzes 5-7**

Quiz 7 out this Fri 11/01, Due Tues 11/05

HW 7 to be posted Thurs 10/31, due Thurs Nov 7

- **Solutions to be posted**

Last Time…

- Batch Linear Least Squares (LLS) cost and estimator derivation

$$J(T) = \sum_{k=1}^{T} (y_k - Hx)^T R_k^{-1} (y_k - Hx) = (\vec{y} - \mathbf{H}x)^T \mathbf{R}^{-1} (\vec{y} - \mathbf{H}x) \quad (= \vec{v}^T \mathbf{R}^{-1} \vec{v} \,)$$

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

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

*(*derived assuming for now NO PROCESS NOISE & STATIC $x(k)=x(k+1)=x_0$,*
<u>BUT</u> batch LLS can also handle dynamic x:
*i. without proc. noise, via suitable **H***
[need H(k) for each y(k)as a linear fxn of x_0]
ii. with proc. noise, via suitable cost)

- Batch LLS application example

- Batch LLS estimator error, error bias, and error covariance

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

$$E[e_{LS}] = E[x - \hat{x}_{LS}] = E[-(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} (\mathbf{H}^T \mathbf{R}^{-1}) \vec{v}] = 0$$

$$P_{LS} = E[e_{LS}\, e_{LS}^T] = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} = \begin{bmatrix} \sigma_1^2 & \sigma_{12}^2 & \cdots & \sigma_{1n}^2 \\ \vdots & \ddots & & \vdots \\ & & & \sigma_n^2 \end{bmatrix}$$

Today…

- Numerical considerations for batch LLS
    - E.g. when to use vs. when not to use…
- **Recursive LLS (RLLS)** definition, cost function, and estimator derivation
- Initialization of RLLS estimators
- Analysis of Optimal RLLS gain behavior

READ SIMON TEXT Chapters 4.1-4.~~4~~2, 5.1-5.4

Numerical Considerations for Batch LLS

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

- For solution to exist: $(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$ must exist

  $\rightarrow (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})$ ('information matrix') must be full rank for batch LLS to work!

  (for static x: same as x being observable from $\vec{y}$)

- If x is only weakly observable: i.e. if info matrix $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is *almost* singular (low cond. #)

  $\rightarrow$ then info matrix is *poorly conditioned*

  (some rows/cols of $\mathbf{H}$ will be nearly linearly dependent)

$\rightarrow$ in this case: BAD IDEA to invert $(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})$ directly!

*(*better in such cases to use "square root" batch LLS solution methods instead based on matrix decompositions, e.g. QR, SVD, LU, UDU)*

- For data y_k that arrives sequentially:

  need to construct larger and larger $\mathbf{H}, \vec{y}, \mathbf{R}$ arrays *over and over again*

  in order to re-compute batch $\hat{x}_{LS}$ from scratch (reinvert info matrix) $\rightarrow$ *very expensive!!*

Recursive Linear Least Squares (RLLS)

- Preferred way to handle large and/or sequentially arriving y_k to avoid high computing cost
- Idea: incrementally update "running estimate" ("prior estimate") of x as new y_k data arrives

Given initial estimate $\hat{x}_{k-1}$ at time $k-1$: at time k:

$$(y_k \in \mathbb{R}^{p\times 1},\ v_k \in \mathbb{R}^{p\times 1},\ H_k \in \mathbb{R}^{p\times n})$$

*receive: $y_k \overset{\text{meas. model}}{=} H_k x + v_k$

actual physical meas.

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

$\hat{y}_{k|k-1}$ (predicted meas. @ k based on $\hat{x}_{k-1}$)

updated state estimate via RLLS @ time k

prior result of RLLS @ time k-1

n x p gain matrix: "blending factor"

"surprise factor" = "innovation vector" = "meas. residual"

Predictor-Corrector Formula:
blend prior info from time k-1
with new sensor info at time k

[note: all terms are regular v_k, y_k, H_k terms
(i.e. **NOT** block matrices or vectorized arrays), and still assuming static x here]

note: $e_k = x - \hat{x}_k \rightarrow E[e_k] = E[x - \hat{x}_k] = E[x - (\hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}))]$

$$= E[\underbrace{x - \hat{x}_{k-1}}_{e_{k-1}} - K_k(\underbrace{y_k}_{y_k = H_k x + v_k \text{ (model)}} - H_k\hat{x}_{k-1})]$$

$$= E[e_{k-1} - K_k(H_k x + v_k - H_k\hat{x}_{k-1})] = E[e_{k-1} - K_k H_k (\overbrace{x - \hat{x}_{k-1}}^{e_{k-1}}) - K_k v_k]$$

$$= E[(I - K_k H_k)e_{k-1} - K_k v_k] = (I - K_k H_k)E[e_{k-1}] - K_k E[v_k] = 0, \text{ if } E[e_{k-1}] = 0$$

For any choice of gain K_k,
$E[e_k] = 0$
(i.e. $\hat{x}_k$ unbiased)
as long as
$E[e_{k-1}] = 0$
(i.e. $\hat{x}_{k-1}$ unbiased)

Recursive Linear Least Squares (RLLS)

- How to pick gain matrix K_k ? i.e. what is the **optimal gain** K_k ?
- Step 1: modify original LLS cost fxn to start from prior at time k-1; Step 2: minimize this to derive optimal estimator gain K_k for each new y_k at time step k

STEP 1: if $x - \hat{x}_k = e_k = \begin{bmatrix} e_{1,k} \\ \vdots \\ e_{n,k} \end{bmatrix}$, then choose cost function $J(k) =$ sum of squared *state estimation error variances at time k*

(error for each state element at time k)

$$\text{i.e. choose } J(k) = J_k = E[e_{1,k}^2 + e_{2,k}^2 + \cdots + e_{n,k}^2] = E[e_k^T e_k] = E[||e_k||_2^2]$$

$$= E[\text{tr}(e_k e_k^T)] = E[\text{tr}(\text{cov}(e_k))] = \text{trace of est. error covar. matrix}$$

$$= \text{tr}(E[e_k e_k^T]) = \text{tr}(P_{k,LS}) \qquad \boxed{\rightarrow J_k = \text{tr}(P_{k,LS})}$$

$\rightarrow$ since $e_{k-1} \perp v_k$, i.e. $E[v_k e_{k-1}^T] = E[v_k]E[e_{k-1}^T] = 0 \cdot E[e_{k-1}^T] = 0$, easy to show (Simon eqs. 3.23-3.25):

est. error cov. for RLLS using *any* gain $K \in \mathbb{R}^{p \times n}$

$$= P_{k,LS} = E[e_k e_k^T] = E[\{(I - K_k H_k)e_{k-1} - K_k v_k\}\{\cdots\}^T]$$

$$= (\text{...smoke clears...}) = \boxed{(I - K_k H_k)P_{LS,k-1}(I - K_k H_k)^T + K_k R_k K_k^T = P_{LS,k}}$$

(recursive formula to update $P_{LS,k}$ in terms of $P_{LS,k-1}$ (for any gain K_k))

RLLS Optimal Gain K_k

- <u>Step 2</u>: Now take derivative of this new J(k) w.r.t. K_k and set equal to zero:

$$J(k) = J_k = \text{RLLS Cost Fxn } = \text{tr}(P_k) = \text{tr}([I - K_k H_k] P_{k-1} [\cdots]^T + K_k R_k K_k^T)$$

(by linearity of trace operator)

$$= \text{tr}([I - K_k H_k] P_{k-1} [\cdots]^T) + \text{tr}(K_k R_k K_k^T)$$

(so we see 2 terms inside of J(k) that are both fxns of [n x p] K_k gain matrix)

$$\text{Note: } \frac{\partial \text{tr}(ABA^T)}{\partial A} = 2AB \text{ if B symmetric}$$

(as in our case: both P_{k-1} and R_k are symmetric pos def covar matrices)

$$\rightarrow \text{so: } \frac{\partial J(k)}{\partial K_k} = 2(I - K_k H_k) P_{k-1} (\underbrace{-H_k^T}_{\text{(via chain rule)}}) + 2 K_k R_k = 0_{n \times p}$$

(necessary condition to minimize J_k w.r.t. K_k)

(re-arrange & solve for optimal K_k):

$$K_k R_k = (I - K_k H_k) P_{k-1} H_k^T \quad \rightarrow K_k \underbrace{(R_k + H_k P_{k-1} H_k^T)}_{} = P_{k-1} H_k^T$$

$$\text{But } R_k \text{ and } P_{k-1} \text{ both posdef } \Rightarrow (R_k + H_k P_{k-1} H_k^T)^{-1} \text{ exists!!}$$

$$\boxed{\Rightarrow \text{Optimal RLLS gain is: } \underset{[n \times p]}{K_k} = \underset{[n \times n]}{P_{k-1}} \underset{[n \times p]}{H_k^T} (\underset{[p \times p]}{R_k} + \underset{[p \times n]}{H_k} \underset{[n \times n]}{P_{k-1}} \underset{[n \times p]}{H_k^T})^{-1}}$$

Fixed (was typo before)

(takes information from $\Delta y_k = y_k - H_k \hat{x}_{k-1} \in \mathbb{R}^p$ and maps to $\Delta \hat{x}_k \in \mathbb{R}^n$ for RLLS update)

General RLLS Algorithm (Simon, p. 88)

- **Problem setup:** given measurement data $y_1,\ldots y_T$ with:

actual physical meas.

$$
\begin{aligned}
y_k &= H_k x + v_k \quad \text{meas. model} \\
x &= \text{constant} \\
E(v_k) &= 0 \\
E(v_k v_i^T) &= R_k \delta_{k-i}
\end{aligned}
\qquad (3.45)
$$

Kronecker: $\delta(k,i)$

- **Initialize estimate at k=0:**

$$
\left.
\begin{aligned}
\hat{x}_0 &= E(x) \\
P_0 &= E[(x-\hat{x}_0)(x-\hat{x}_0)^T]
\end{aligned}
\right\} \text{prior guess}
\qquad (3.46)
$$

- **For k = 1,2,…,T :**

re-used for next step

$$
\begin{aligned}
K_k &= P_{k-1}H_k^T(H_k P_{k-1} H_k^T + R_k)^{-1} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k \hat{x}_{k-1}) \\
P_k &= (I - K_k H_k)P_{k-1}(I - K_k H_k)^T + K_k R_k K_k^T
\end{aligned}
$$

Choosing RLLS Estimator Initial Conditions

- **RLLS Initialization:** at $k = 1:$ $\quad \hat{x}_1 = \hat{x}_0 + K_1(y_1 - H_1\hat{x}_0)$

$$K_1 = P_0 H_1^T (R_1 + H_1 P_0 H_1^T)^{-1}$$

$\rightarrow$ Need an initial state estimate $\hat{x}(0)$ (prior estimate) and covariance P_0 (prior uncertainty)

$\rightarrow$ Where do $\hat{x}_0$ and P_0 come from? $\quad \rightarrow$ Consider 3 possibilities:

#1) *'Engineering know-how':* we know *something* to begin with (some a priori info about static x) in the form of a pdf: $p(x) \sim N_x(m_0, \Sigma_0) \Rightarrow \hat{x}_0 = E[x] = m_0, \; P_0 = \text{cov}(x) = \Sigma_0$

#2) If we know absolutely nothing about x_0 to start, then ideally we ought to set, e.g. $\quad p(x) = \mathcal{N}_x(0, \Sigma_0 = \text{`}\infty\text{'}),$ where $m_0 = 0$ and $\Sigma_0 = \text{`}\infty\text{'} = I_n \cdot (\text{very big } \#)$

(basically saying x has "infinitely large" uniform distribution [not necessarily Gaussian]);

→ Implicitly: this is what batch LLS does (no priors needed!)

#3) *'Hybrid warm start' with batch LLS:* if $(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})$ info invertible after $r \leq n$ measurements, then use batch LLS on first r observations: $(\hat{x}_r, P_r)$ =batch soln $(\hat{x}_r, P_r)$ for $y_1, y_2, ..., y_r$

$\rightarrow$ initialize RLLS at $(\hat{x}_r, P_r)$ $\rightarrow$ then use RLLS for $y_{r+1}, y_{r+2}, \cdots$

RLLS Initialization and Optimal Gain K_k Behavior

- What happens to Optimal RLLS Gain K_1 at k=1 if we have case #2? (i.e $P_0 = `\infty'$)

$$K_1 = P_0 H_1^T \left[H_1 P_0 H_1^T + R_1\right]^{-1}, \text{ where } P_0 `>>' R \quad (\text{e.g. } Z = (H_1 P_0 H_1^T - R_1) \text{ is posdef})$$

$\rightarrow$ Consider scalar x and y_k case, such that K_k, H_k, and R_k are all scalars:

$$K_1 = \frac{P_0 H_1}{H_1^2 P_0 + R_1}, \text{ where } H_1^2 P_0 >> R_1 \text{ because } P_0 = `\infty' \cdot I$$

Then the denominator of K_1 is dominated by P_0 :

$$\lim_{P_0 \to \infty} K_1 = \frac{P_0 H_1}{H_1^2 P_0} = \frac{1}{H_1}$$

So the (scalar) RLLS update becomes: $\hat{x}_1 = \hat{x}_0 + \frac{1}{H_1}(y_1 - H_1 \hat{x}_0)$

$$= \hat{x}_0 + \frac{y_1}{H_1} - \frac{H_1}{H_1}\hat{x}_0 = (\hat{x}_0 - \hat{x}_0) + \frac{y_1}{H_1}$$

Optimal RLLS gain K_1 says to just completely trust new measurement y_1 when $P_0 = `\infty'$…

$$\boxed{\Rightarrow \hat{x}_1 = \frac{y_1}{H_1}}$$

…Exactly what we intuitively expect should happen if no prior info at all about x!