Recursive Least Squares Parameter Estimation for Linear Steady State and Dynamic Models

Thomas F. Edgar
Department of Chemical Engineering
University of Texas
Austin, TX 78712
Outline

• Static model, sequential estimation
• Multivariate sequential estimation
• Example
• Dynamic discrete-time model
• Closed-loop estimation
Least Squares Parameter Estimation

Linear Time Series Models

ref: PC Young, Control Engr., p. 119, Oct, 1969

scalar example (no dynamics)
model \( y = ax \)
data \( y^* = ax + \varepsilon \quad \varepsilon: \text{error} \)
least squares estimate of a: \( \hat{a} \)

\[
\min_{\hat{a}} \sum_{i=1}^{k} \left( \hat{a} x_i - y_i^* \right)^2 \quad (1)
\]
Simple Example

The analytical solution for the minimum (least squares) estimate is

\[ \hat{a}_k = \left( \sum_{i=1}^{k} x_i^2 \right)^{-1} \left( \sum_{i=1}^{k} x_i y_i^* \right) \]

\[ \begin{aligned}
& p_k \\
& b_k
\end{aligned} \]

(2)

\( p_k, b_k \) are functions of the number of samples

This is the non-sequential form or non-recursive form
Sequential or Recursive Form

To update $\hat{a}_k$ based on a new data pt. $(y_i, x_i)$, in Eq. (2) let

$$p_k^{-1} = \sum_{i=1}^{k} x_i^2 = p_{k-1}^{-1} + x_k^2 \quad (3)$$

and

$$b_k = \sum_{i=1}^{k} x_i y_i^* = b_{k-1} + x_k y_k^* \quad (4)$$
Recursive Form for Parameter Estimation

\[
\hat{a}_k = \hat{a}_{k-1} - K_k \left( x_k \hat{a}_{k-1} - y_k^* \right) \tag{5}
\]

where

\[
K_k = p_{k-1} x_k \left( 1 + p_{k-1} x_k^2 \right)^{-1} \tag{6}
\]

To start the algorithm, need initial estimates \( \hat{a}_0 \) and \( p_0 \). To update \( p \),

\[
p_k = p_{k-1} - p_{k-1} x_k^2 \left( 1 + p_{k-1} x_k^2 \right)^{-1} \tag{7}
\]

(Set \( p_0 = \) large positive number)

Eqn. (7) shows \( p_k \) is decreasing with \( k \)
Estimating Multiple Parameters
(Steady State Model)

\[ y = x^T a = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \quad (8) \]

\[
\min \sum_{i=1}^{k} (x_i^T \hat{a} - y_i^*)^2 
\text{ (non-sequential solution requires } n \times n \text{ inverse)}
\]

To obtain a recursive form for \( \hat{a} \),

\[
P_k^{-1} = P_{k-1}^{-1} + x_k x_k^T \quad (9)
\]

\[
B_k = B_{k-1} + x_k y_k^* \quad (10)
\]
Recursive Solution

\[
\hat{a}_k = \hat{a}_{k-1} - P_{k-1} x_k \left( 1 + x_k^T P_{k-1} x_k \right)^{-1} K_k \left( x_k^T \hat{a}_{k-1} - y_k^* \right)
\]

(11)

\[
P_k = P_{k-1} - P_{k-1} x_k \left[ 1 + x_k^T P_{k-1} x_k \right]^{-1} x_k^T \left( x_k^T P_{k-1} x_k \right)
\]

(12)

need to assume \( \hat{a}_0 \) (vector)

\[P_0\] (diagonal matrix)

\[P_{ii} \] large
Simple Example (Estimate Slope & Intercept)

Linear parametric model

<table>
<thead>
<tr>
<th>input, $u$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>10</th>
<th>12</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>output, $y$</td>
<td>5.71</td>
<td>9</td>
<td>15</td>
<td>19</td>
<td>20</td>
<td>45</td>
<td>55</td>
<td>78</td>
</tr>
</tbody>
</table>

$y = a_1 + a_2 u$
Sequential vs. Non-sequential Estimation of $a_2$ Only ($a_1 = 0$)

Covariance Matrix vs. Number of Samples Using Eq. (12)
Sequential Estimation of $\hat{a}_1$ and $\hat{a}_2$ Using Eq. (11)
Application to Digital Model and Feedback Control

Linear Discrete Model with Time Delay:

\[ y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_n y(t-n) + b_1 u(t-1-N) + b_2 u(t-2-N) + \cdots + b_r u(t-r-N) + d \quad (13) \]

\( y \): output \quad \( u \): input \quad \( d \): disturbance \quad \( N \): time delay
Recursive Least Squares Solution

\[ y(t) = \Psi^T(t-1)\theta(t-1) + \epsilon(t) \]  \hspace{1cm} (14)

where

\[ \Psi^T(t-1) = [y(t-1), y(t-2), \ldots y(t-n), \\
                u(t-1-N), \ldots u(t-r-N), 1] \]

\[ \theta^T(t-1) = [a_1, a_2, \ldots a_n, b_1, b_2, \ldots b_r, d]. \]

\[
\min_{\hat{\theta}} \sum_{i=1}^{t} \left[ \frac{\Psi^T(i-1)\theta(i) - y(i)}{d} \right]^2
\]

"least squares" (predicted value of \( y \))

\[ \hat{\theta}(t) = \hat{\theta}(t - 1) + P(t)\Psi(t-1)[y(t) - \Psi^T(t-1)\hat{\theta}(t-1)] \] \hspace{1cm} (16)
Recursive Least Squares Solution

\[ P(t) = P(t-1) - P(t-1)\psi(t-1)[\psi^T(t-1)P(t-1)\psi(t-1) + 1]^{-1} \]

\[ \psi^T(t-1)P(t-1) \] \hspace{1cm} (17)

\[ K(t) = \frac{P(t-1)\psi(t-1)}{1 + \psi^T(t-1)P(t-1)\psi(t-1)} \] \hspace{1cm} (18)

\[ P(t) = [I - K(t)\psi^T(t-1)]P(t-1) \] \hspace{1cm} (19)

\[ \hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - \hat{y}(t)] \] \hspace{1cm} (20)

\[ K: \text{ Kalman filter gain} \]
Closed-Loop RLS Estimation

- There are three practical considerations in implementation of parameter estimation algorithms
  - covariance resetting
  - variable forgetting factor
  - use of perturbation signal
Enhance sensitivity of least squares estimation algorithms with forgetting factor $\lambda$

$$J(\theta(t)) = \sum_{i=1}^{t} \lambda^{t-i} \left[ \psi^T(i-1)\theta(i) - y(i) \right]^2$$  \hspace{1cm} (21)

$$P(t) = \frac{1}{\lambda} \left[ P(t-1) - P(t-1)\psi(t-1)[\psi^T(t-1)P(t-1)\psi(t-1) + \lambda]^{-1} \right].$$

$$\psi^T(t-1)P(t-1)]$$  \hspace{1cm} (22)

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\psi(t-1)[y(t) - \psi^T(t-1)\hat{\theta}(t-1)]$$  \hspace{1cm} (23)

$\lambda$ prevents elements of $P$ from becoming too small (improves sensitivity) but noise may lead to incorrect parameter estimates

$$0 \leq \lambda \leq 1.0 \hspace{1cm} \lambda \rightarrow 1.0 \hspace{0.5cm} \text{all data weighted equally}$$

$$\lambda \sim 0.98 \hspace{0.5cm} \text{typical}$$
Closed-Loop Estimation (RLS)

Perturbation signal is added to process input (via set point) to excite process dynamics

large signal: good parameter estimates but large errors in process output

small signal: better control but more sensitivity to noise

1. set forgetting factor $\lambda = 1.0$

2. use covariance resetting (add diagonal matrix $D$ to $P$ when $\text{tr}(P)$ becomes small)
3. use PRBS perturbation signal only when estimation error is large and $P$ is not small. Vary PRBS amplitude with size of elements proportional to $\text{tr}(P)$.

4. $P(0) = 10^4 \ I$

5. filter new parameter estimates

$$\theta_c(k) = \rho \theta_c(k-1) + (1-\rho)\theta(k)$$

$\rho$: tuning parameter
($\theta_c$ used by controller)

6. Use other diagnostic checks such as the sign of the computed process gain