Least squares identification

For white noise

Xing Chao

<LS.1>

1 Least square estimation

System model based on input/output data

The difference equation of a SISO system is

$$y_k + a_1 y_{k-1} + \cdots + a_n y_{k-n} = b_0 u_k + \cdots + b_n u_{k-n} + \xi_k$$

at time $k = n+1, n+2, \cdots, n+N$, there is

$$\begin{gathered}
y_{n+1} + a_1 y_n + \cdots + a_n y_1 = b_0 u_{n+1} + \cdots + b_n u_1 + \xi_{n+1} \\
y_{n+2} + a_1 y_{n+1} + \cdots + a_n y_2 = b_0 u_{n+2} + \cdots + b_n u_2 + \xi_{n+2} \\
\cdots \\
y_{n+N} + a_1 y_{n+N-1} + \cdots + a_n y_N = b_0 u_{n+N} + \cdots + b_n u_N + \xi_{n+N}
\end{gathered}$$

<LS.2>

Vector form

$$\begin{aligned}
Y &= \Phi\theta + \xi \\
Y &= \begin{bmatrix} y_{n+1} & y_{n+2} & \cdots & y_{n+N} \end{bmatrix}^T \\
\Phi &= \begin{bmatrix} -y_n & \cdots & -y_1 & u_{n+1} & \cdots & u_1 \\ -y_{n+1} & \cdots & -y_2 & u_{n+2} & \cdots & u_2 \\ \vdots & & \vdots & \vdots & & \vdots \\ -y_{n+N-1} & \cdots & -y_N & u_{n+N} & \cdots & u_N \end{bmatrix} \\
\theta &= \begin{bmatrix} a_1 & \cdots & a_n & b_0 & \cdots & b_n \end{bmatrix}^T \\
\xi &= \begin{bmatrix} \xi_{n+1} & \xi_{n+2} & \cdots & \xi_{n+N} \end{bmatrix}^T
\end{aligned}$$

<LS.3>

Basic least squares method: identification criteria

Identification criterion: least square sum of residuals.

$$
\begin{aligned}
J &= \sum_{k=n+1}^{n+N} e^2(k) \\
&= (Y - \Phi\hat{\theta})^T (Y - \Phi\hat{\theta}) \\
\hat{\theta}_{LS} &= \arg\min_{\hat{\theta}} J
\end{aligned}
$$

<LS.4>

Basic least square method: Derivative

$$
\begin{aligned}
\frac{\partial J}{\partial \hat{\theta}_k} &= \frac{\partial \sum_i (Y_i - \sum_m \Phi_{i,m}\hat{\theta}_m)^2}{\partial \hat{\theta}_k} \\
&= 2\sum_i (Y_i - \sum_m \Phi_{i,m}\hat{\theta}_m) \frac{\partial (Y_i - \sum_m \Phi_{i,m}\hat{\theta}_m)}{\partial \hat{\theta}_k} \\
&= 2\sum_i (Y_i - \sum_m \Phi_{i,m}\hat{\theta}_m) \frac{\partial (-\sum_m \Phi_{i,m}\hat{\theta}_m)}{\partial \hat{\theta}_k} \\
&= -2\sum_i (Y_i - \sum_m \Phi_{i,m}\hat{\theta}_m)\Phi_{i,k} \\
\frac{\partial J}{\partial \hat{\theta}} &= (-2(Y - \Phi\hat{\theta})^T \Phi)^T \\
&= -2\Phi^T (Y - \Phi\hat{\theta})
\end{aligned}
$$

<LS.5>

Basic least square method: solution

$$
\begin{aligned}
-2\Phi^T (Y - \Phi\hat{\theta}_{LS}) &= 0 \\
\Phi^T Y - \Phi^T \Phi\hat{\theta}_{LS} &= 0 \\
\Phi^T Y &= \Phi^T \Phi\hat{\theta}_{LS} \\
\hat{\theta}_{LS} &= (\Phi^T \Phi)^{-1}\Phi^T Y
\end{aligned}
$$

<LS.6>

Basic least square method: two order derivative

$$
\begin{aligned}
\frac{\partial^2 J}{\partial \hat{\theta}^2} &= \frac{\partial(-2\Phi^T (Y - \Phi\hat{\theta}))}{\partial \hat{\theta}} \\
\frac{\partial \frac{\partial J}{\partial \hat{\theta}}}{\partial \hat{\theta}_s} &= \frac{\partial(-2\sum_i (Y_i - \sum_m \Phi_{i,m}\hat{\theta}_m)\Phi_{i,k})}{\partial \hat{\theta}_s} \\
&= 2\sum_i \frac{\partial \sum_m \Phi_{i,m}\hat{\theta}_m}{\partial \hat{\theta}_s}\Phi_{i,k} \\
&= 2\sum_i \Phi_{i,s}\Phi_{i,k} \\
\frac{\partial^2 J}{\partial \hat{\theta}^2} &= 2\Phi^T \Phi
\end{aligned}
$$

<LS.7>

The requirement of input signal by least square method : $\begin{bmatrix} Y_{N\times n} & U_{N\times(n+1)} \end{bmatrix}$

$$
\begin{aligned}
\Phi^T\Phi &= \begin{bmatrix} Y_{N\times n} & U_{N\times(n+1)} \end{bmatrix}^T \begin{bmatrix} Y_{N\times n} & U_{N\times(n+1)} \end{bmatrix} \\
&= \begin{bmatrix} Y_{N\times n}^T Y_{N\times n} & Y_{N\times n}^T U_{N\times(n+1)} \\ U_{N\times(n+1)}^T Y_{N\times n} & U_{N\times(n+1)}^T U_{N\times(n+1)} \end{bmatrix}
\end{aligned}
$$

where:

$$
\begin{aligned}
Y_{N\times n} &= \begin{bmatrix} -y_n & \cdots & -y_1 \\ -y_{n+1} & \cdots & -y_2 \\ \vdots & & \vdots \\ y_{n+N-1} & \cdots & -y_N \end{bmatrix} \\
U_{N\times(n+1)} &= \begin{bmatrix} u_{n+1} & \cdots & u_1 \\ u_{n+2} & \cdots & u_2 \\ \vdots & & \vdots \\ u_{n+N} & \cdots & u_N \end{bmatrix}
\end{aligned}
$$

<LS.8>

The requirement of input signal by least square method $\begin{bmatrix} Y_{N\times n} & U_{N\times(n+1)} \end{bmatrix}$

$$
\begin{aligned}
(Y_{N\times n}^T Y_{N\times n})_{i,j} &= \sum_{k=1}^{N-1+\min\{i,j\}} y_{n-i+k} y_{n-j+k} \\
(Y_{N\times n}^T U_{N\times(n+1)})_{i,j} &= -\sum_{k=1}^{N-1+\min\{i,j-1\}} y_{n-i+k} u_{n+1-j+k} \\
(U_{N\times(n+1)}^T Y_{N\times n}^T)_{i,j} &= -\sum_{k=1}^{N-1+\min\{j,i-1\}} y_{n-j+k} u_{n+1-i+k} \\
(U_{N\times(n+1)}^T U_{N\times(n+1)})_{i,j} &= \sum_{k=1}^{N-2+\min\{i,j\}} u_{n+1-i+k} u_{n+1-j+k}
\end{aligned}
$$

<LS.9>

The requirement of input signal by least square method $\begin{bmatrix} R_y & R_{yu} \\ R_{uy} & R_u \end{bmatrix}$

$$
\begin{aligned}
\lim_{N\to\infty} \frac{\Phi^T\Phi}{N} &= \frac{1}{N} \begin{bmatrix} Y_{N\times n}^T Y_{N\times n} & Y_{N\times n}^T U_{N\times(n+1)} \\ U_{N\times(n+1)}^T Y_{N\times n} & U_{N\times(n+1)}^T U_{N\times(n+1)} \end{bmatrix} \\
&= \begin{bmatrix} R_y & R_{yu} \\ R_{uy} & R_u \end{bmatrix}
\end{aligned}
$$

where:

$$
\begin{aligned}
R_y &= \begin{bmatrix} R_y(0) & R_y(1) & \cdots & R_y(n-1) \\ R_y(1) & R_y(0) & \cdots & R_y(n-2) \\ \vdots & \vdots & & \vdots \\ R_y(n-1) & R_y(n-2) & \cdots & R_y(0) \end{bmatrix} \\
R_{yu} &= \begin{bmatrix} -R_{yu}(1) & -R_{yu}(0) & \cdots & -R_{yu}(1-n) \\ -R_{yu}(2) & -R_{yu}(1) & \cdots & -R_{yu}(2-n) \\ \vdots & \vdots & & \vdots \\ -R_{yu}(n) & -R_{yu}(n-1) & \cdots & -R_{yu}(0) \end{bmatrix}
\end{aligned}
$$

<LS.10>

The requirement of input signal by least square method: $\begin{bmatrix} R_y & R_{yu} \\ R_{uy} & R_u \end{bmatrix}$

$$
\begin{aligned}
R_{uy} &= R_{yu}^T \\
R_{uu} &= \begin{bmatrix} R_u(0) & R_u(1) & \cdots & R_u(n) \\ R_u(1) & R_u(0) & \cdots & R_u(n-1) \\ \vdots & \vdots & & \vdots \\ R_u(n) & R_u(n-1) & \cdots & R_u(0) \end{bmatrix}
\end{aligned}
$$

<LS.11>

(n+1) order continuous excitation signal

- defination: $\{u(k)\}$ is called $(n+1)$ order continuous excitation signal if (n+1) order matrix R_u of series $\{u(k)\}$ is positive definate,。
- The requirement of least square method for input signal is: $\{u(k)\}$ is $(n+1)$ order continuous excitation signal
- R_u is positive definate if R_u is an strongly diagonally dominant matrix. The following sginals can satisfy the requirement of positive definate of R_u.
  - White noise sequence ;
  - Pseudo random two bit noise sequence ;
  - Colored noise random signal sequence 。
- ”Pseudo random two bit noise sequence” and ”colored noise random signal sequence” are often used as input signals in Engineering 。

<LS.12>

unbiasedness of the estimation

$\hat{\theta}$ is refered as unbiased estimation of parameter θ if $E\{\hat{\theta}\} = \theta$.

$$
\begin{aligned}
Y &= \Phi\theta + \xi \\
\hat{\theta} &= (\Phi^T\Phi)^{-1}\Phi^T Y \\
E[\hat{\theta}] &= E[(\Phi^T\Phi)^{-1}\Phi^T Y] \\
&= E[(\Phi^T\Phi)^{-1}\Phi^T(\Phi\theta + \xi)] \\
&= E[(\Phi^T\Phi)^{-1}\Phi^T\Phi\theta + (\Phi^T\Phi)^{-1}\Phi^T\xi] \\
&= E[\theta + (\Phi^T\Phi)^{-1}\Phi^T\xi]
\end{aligned}
$$

The necessary and sufficient conditions of least squares estimation for unbiased estimation is:

$$
E[(\Phi^T\Phi)^{-1}\Phi^T\xi] = 0
$$

<LS.13>

Consistent estimation

The estimated value is consistent if the estimated parameter converges to the true value θ in probability 1. defination:

$$
\lim_{N\to\infty} P\{|\hat{\theta} - \theta\} = 1
$$

Suppose $\xi\{(k)\}$ is random sequence with zero mean and independent distribution uncorrelated with $\{u(k)\}$:

$$\begin{aligned}
E(\hat{\theta}-\theta)^2 &= E[(\Phi^T\Phi)^{-1}\Phi^T\xi\xi^T\Phi(\Phi^T\Phi)^{-1}] \\
&= E[\frac{1}{N^2}(\frac{1}{N}\Phi^T\Phi)^{-1}\Phi^T\xi\xi^T\Phi(\frac{1}{N}\Phi^T\Phi)^{-1}] \\
\lim_{N\to\infty} E(\hat{\theta}-\theta)^2 &= \frac{1}{N^2}R^{-1}E[\Phi^T\xi\xi^T\Phi]R^{-1} \\
&= \frac{1}{N^2}R^{-1}\sigma^2E[\Phi^T\Phi]R^{-1} \\
&= \frac{1}{N^2}R^{-1}\sigma^2NRR^{-1} \\
&= \frac{\sigma^2}{N}R^{-1} \\
&= 0
\end{aligned}$$

<LS.14>

2 Model order increasing algorithm

Model order increasing algorithm: algorithm characteristics

- recursive algorithm based on model order n;
- suitable for unknown model order n
- The identification accuracy is the same as that of the basic least square method
- The identification speed is greatly improved than the basic least square method
- It is not necessary to compute the inverse of higher order matrices

<LS.15>

System model

$$\begin{aligned}
Y &= \Phi_n\theta_n+\xi \\
\Phi_n &= \begin{bmatrix} u_{n+1} & -y_n & u_n & \cdots & -y_1 & u_1 \\ u_{n+2} & -y_{n+1} & u_{n+1} & \cdots & -y_2 & u_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ u_{n+N} & -y_{n+N-1} & u_{n+N-1} & \cdots & -y_N & u_N \end{bmatrix} \\
&= \begin{bmatrix} X_1 & \cdots & X_{2n+1} \end{bmatrix} \\
\theta_n &= \begin{bmatrix} b_0 & a_1 & b_1 & \cdots & a_n & b_n \end{bmatrix}^T \\
\xi &= \begin{bmatrix} \xi_{n+1} & \cdots & \xi_{n+N} \end{bmatrix}^T \\
Y &= \begin{bmatrix} y_{n+1} & \cdots & y_{n+N} \end{bmatrix}^T
\end{aligned}$$

<LS.16>

Identification from $n=0$

$$\begin{aligned}
\Phi_0 &= X_1 \\
\hat{\theta}_0 &= (\Phi_0^T\Phi_0)^{-1}\Phi_0^TY \\
&= \frac{\sum_{i=n+1}^{n+N} u_iy_i}{\sum_{i=n+1}^{n+N} u_i^2}
\end{aligned}$$

<LS.17>

from n to $n+1$

Identification result of model order $n+1$ is obtained based on result of model order n。 The solution is divided into two steps. First $\tilde{P}_n$ is solved, and then P_{n+1} is solved.

$$\begin{aligned}
\Phi_{n+1} &= \begin{bmatrix}\Phi_n & X_{2n+2} & X_{2n+3}\end{bmatrix} \\
&= \begin{bmatrix}\tilde{\Phi}_n & X_{2n+3}\end{bmatrix} \\
\tilde{\Phi}_n &\triangleq \begin{bmatrix}\Phi_n & X_{2n+2}\end{bmatrix} \\
P_n &\triangleq (\Phi_n^T\Phi_n)^{-1} \\
\tilde{P}_n &\triangleq (\tilde{\Phi}_n^T\tilde{\Phi}_n)^{-1} \\
P_{n+1} &= (\Phi_{n+1}^T\Phi_{n+1})^{-1}
\end{aligned}$$

<LS.18>

from n to $n+1$：P_{n+1}

$$\begin{aligned}
P_{n+1} &= \begin{bmatrix}\tilde{\Phi}_n^T\tilde{\Phi}_n & \tilde{\Phi}_n^T X_{2n+3} \\ X_{2n+3}^T\tilde{\Phi}_n & X_{2n+3}^T X_{2n+3}\end{bmatrix}^{-1} \\
&= \begin{bmatrix}A_{11} & A_{12} \\ A_{21} & A_{22}\end{bmatrix} \\
A_{22} &= (X_{2n+3}^T X_{2n+3} - X_{2n+3}^T\tilde{\Phi}_n\tilde{P}_n\tilde{\Phi}_n^T X_{2n+3})^{-1} \\
A_{12} &= A_{21}^T \\
&= -\tilde{P}_n\tilde{\Phi}_n^T X_{2n+3}A_{22} \\
A_{11} &= \tilde{P}_n - A_{12}X_{2n+3}^T\tilde{\Phi}_n\tilde{P}_n^T
\end{aligned}$$

<LS.19>

from n to $n+1$：$\tilde{P}_n$

$$\begin{aligned}
\tilde{P}_n &= \begin{bmatrix}\Phi_n^T\Phi_n & \Phi_n^T X_{2n+2} \\ X_{2n+2}^T\Phi_n & X_{2n+2}^T X_{2n+3}\end{bmatrix}^{-1} \\
&= \begin{bmatrix}B_{11} & B_{12} \\ B_{21} & B_{22}\end{bmatrix} \\
B_{22} &= (X_{2n+2}^T X_{2n+2} - X_{2n+2}^T\Phi_n P_n\Phi_n^T X_{2n+2})^{-1} \\
B_{12} &= B_{21}^T \\
&= -P_n\Phi_n^T X_{2n+2}B_{22} \\
B_{11} &= P_n - B_{12}X_{2n+2}^T\Phi_n P_n^T
\end{aligned}$$

<LS.20>

Computation procedure

- Initialize,compute $P_0 = (\Phi_0^T\Phi_0)^{-1}$
- Compute $\hat{\theta}_0 = P_0\Phi_0^T Y$
- iterate
  - Compute $\tilde{P}_n$ based on P_n
  - Compute P_{n+1} based on $\tilde{P}_n$
  - Compute $\hat{\theta}_{n+1} = P_{n+1}\Phi_{n+1}^T Y$

<LS.21>

3 recursive least square

Recursive algorithm derivation: Model

Assuming that the input and output data with length of N have been obtained, the least squares estimation is

$$
\begin{aligned}
Y_N &= \Phi_N \theta + \xi_N \\
\hat{\theta}_N &= (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N \\
\tilde{\theta}_N &= \theta - \tilde{\theta}_N \\
&= -(\Phi_N^T \Phi_N)^{-1} \Phi_N^T \xi_N
\end{aligned}
$$

After obtaining new data u_{n+N+1}, y_{n+N+1},

$$
\begin{aligned}
y_{(n+N+1)} &= \Psi^T \theta + \xi_{(n+N+1)} \\
y_{N+1} &= \Psi^T \theta + \xi_{N+1} \\
\Psi_i &= \begin{bmatrix} -y_{(n+i-1)} & \cdots & -y_{(i)} & u_{(n+i)} & \cdots & u_{(i)} \end{bmatrix}^T \\
\begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix} &= \begin{bmatrix} \Phi_N \\ \Psi_{N+1}^T \end{bmatrix} \theta + \begin{bmatrix} \xi_N \\ \xi_{N+1} \end{bmatrix}
\end{aligned}
$$

<LS.22>

Recursive algorithm derivation :P_{N+1}

$$
\begin{aligned}
\hat{\theta}_{N+1} &= \left(\begin{bmatrix} \Phi_N \\ \Psi_{N+1}^T \end{bmatrix}^T \begin{bmatrix} \Phi_N \\ \Psi_{N+1}^T \end{bmatrix} \right)^{-1} \begin{bmatrix} \Phi_N \\ \Psi_{N+1}^T \end{bmatrix}^T \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix} \\
&= (\Phi_N^T \underbrace{\Phi_N}_{N,2n+1} + \underbrace{\Psi_{N+1}}_{2n+1,1} \Psi_{N+1}^T)^{-1} (\Phi_N^T \underbrace{Y_N}_{N,1} + \Psi_{N+1} \underbrace{y_{N+1}}_{1,1}) \\
\hat{\theta}_{N+1} &= P_{N+1} (\Phi_N^T Y_N + \Psi_{N+1} y_{N+1})
\end{aligned}
$$

其中：

$$
\begin{aligned}
P_{N+1} &= (P_N^{-1} + \Psi_{N+1} \Psi_{N+1}^T)^{-1} \\
P_N &= (\Phi_N^T \Phi_N)^{-1}
\end{aligned}
$$

<LS.23>

Recursive algorithm derivation : matrix inversion lemma

If the inverse of the corresponding matrix exists ， then:

$$
(A + BC^T)^{-1} = A^{-1} - A^{-1} B (I + C^T A^{-1} B)^{-1} C^T A^{-1}
$$

therefore：

$$
\begin{aligned}
P_{N+1} &= (P_N^{-1} + \Psi_{N+1} \Psi_{N+1}^T)^{-1} \\
&= P_N - P_N \Psi_{N+1} (1 + \Psi_{N+1}^T P_N \Psi_{N+1})^{-1} \Psi_{N+1}^T P_N \\
\hat{\theta}_{N+1} &= A + B \\
A &= P_{N+1} \Phi_N^T Y_N \\
B &= P_{N+1} \Psi_{N+1} y_{N+1} \\
i &= 1 + \Psi_{N+1}^T P_N \Psi_{N+1}
\end{aligned}
$$

<LS.24>

Recursive algorithm derivation:Simplification

$$\begin{aligned}
A &= (P_N - P_N\Psi_{N+1}i^{-1}\Psi_{N+1}^T P_N)\Phi_N^T Y_N \\
&= P_N\Phi_N^T Y_N - P_N\Psi_{N+1}i^{-1}\Psi_{N+1}^T P_N\Phi_N^T Y_N \\
&= \hat{\theta}_N - P_N\Psi_{N+1}i^{-1}\Psi_{N+1}^T\hat{\theta}_N \\
B &= (P_N - P_N\Psi_{N+1}i^{-1}\Psi_{N+1}^T P_N)\Psi_{N+1}y_{N+1} \\
&= i^{-1}(P_N(1+\Psi_{N+1}^T P_N\Psi_{N+1}) - P_N\Psi_{N+1}\Psi_{N+1}^T P_N)\Psi_{N+1}y_{N+1} \\
&= i^{-1}(P_N + P_N\Psi_{N+1}^T P_N\Psi_{N+1} - P_N\Psi_{N+1}\Psi_{N+1}^T P_N)\Psi_{N+1}y_{N+1} \\
&= i^{-1}(P_N\Psi_{N+1} + P_N\Psi_{N+1}^T P_N\Psi_{N+1}\Psi_{N+1} \\
&\quad -P_N\Psi_{N+1}\Psi_{N+1}^T P_N\Psi_{N+1})y_{N+1} \\
&= i^{-1}(P_N\Psi_{N+1} + P_N\Psi_{N+1}\Psi_{N+1}^T P_N\Psi_{N+1} \\
&\quad -P_N\Psi_{N+1}\Psi_{N+1}^T P_N\Psi_{N+1})y_{N+1} \\
&= i^{-1}P_N\Psi_{N+1}y_{N+1}
\end{aligned}$$

note： $\Psi_{N+1}^T P_N\Psi_{N+1}$ is a scalar

<LS.25>

Recursive algorithm derivation：result

$$\begin{aligned}
\hat{\theta}_{N+1} &= \hat{\theta}_N - P_N\Psi_{N+1}i^{-1}\Psi_{N+1}^T\hat{\theta}_N + i^{-1}P_N\Psi_{N+1}y_{N+1} \\
&= \hat{\theta}_N + i^{-1}P_N\Psi_{N+1}(-\Psi_{N+1}^T\hat{\theta}_N + y_{N+1}) \\
&= \hat{\theta}_N + K_{N+1}(y_{N+1} - \Psi_{N+1}^T\hat{\theta}_N) \\
K_{N+1} &= P_N\Psi_{N+1}(1+\Psi_{N+1}^T P_N\Psi_{N+1})^{-1} \\
P_{N+1} &= P_N - K_{N+1}\Psi_{N+1}^T P_N
\end{aligned}$$

Obtain initial value:

- Basic least squares estimation
- $\hat{\theta}_0 = 0, P_0 = c^2 I$, where c is a sufficient large constant。

<LS.26>

Convergence:P_N

$$\begin{aligned}
P_N &= (P_{N-1}^{-1} + \Psi_N\Psi_N^T)^{-1} \\
P_N^{-1} &= P_{N-1}^{-1} + \Psi_N\Psi_N^T \\
P_{N-1}^{-1} &= P_{N-2}^{-1} + \Psi_{N-1}\Psi_{N-1}^T \\
P_{N-2}^{-1} &= P_{N-3}^{-1} + \Psi_{N-2}\Psi_{N-2}^T \\
P_{N-3}^{-1} &= P_{N-4}^{-1} + \Psi_{N-3}\Psi_{N-3}^T \\
&\vdots \\
P_1^{-1} &= P_0^{-1} + \Psi_1\Psi_1^T \\
P_N^{-1} &= P_0^{-1} + \sum_{i=1}^{N}\Psi_i\Psi_i^T
\end{aligned}$$

<LS.27>

Convergence

Ψ_i is corresponding to the i'th row of Φ_N

$$\begin{aligned}
\Phi_N &= \begin{bmatrix} \Psi_1^T \\ \Psi_2^T \\ \vdots \\ \Psi_N^T \end{bmatrix} \\
P_N^{-1} &= \frac{1}{c^2} I + \begin{bmatrix} \Psi_1 & \Psi_2 & \cdots & \Psi_N \end{bmatrix} \begin{bmatrix} \Psi_1^T \\ \Psi_2^T \\ \vdots \\ \Psi_N^T \end{bmatrix} \\
&= \frac{1}{c^2} I + \Phi^T \Phi \\
\lim_{c\to\infty} P_N^{-1} &= \Phi_N^T \Phi_N \\
\hat{\theta}_N &= P_N \Phi_N^T Y_N \\
&= (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N
\end{aligned}$$

<LS.28>

4 Problem discussion

The relationship between residual and innovation

Innovation $\tilde{y}_i = y_i - \Psi_i^T \hat{\theta}_{i-1}$ is used to describe prediction error at time i. residual $\varepsilon_i = y_i - \Psi_i^T \hat{\theta}_i$ is used to describe the output bias at time i.

$$\begin{aligned}
\varepsilon &= y_i - \Psi_i^T \hat{\theta}_i \\
&= y_i - \Psi_i^T (\hat{\theta}_{i-1} + K_i \tilde{y}_i) \\
&= \tilde{y}_i - \Psi_i^T K_i \tilde{y}_i \\
&= (1 - \Psi_i^T K_i) \tilde{y}_i \\
&= (1 - \Psi_i^T P_{i-1} \Psi_i (\Psi_i^T P_{i-1} \Psi_i + 1)^{-1}) \tilde{y}_i \\
&= \frac{\Psi_i^T P_{i-1} \Psi_i + 1 - \Psi_i^T P_{i-1} \Psi_i}{\Psi_i^T P_{i-1} \Psi_i + 1} \tilde{y}_i \\
&= \frac{\tilde{y}_i}{\Psi_i^T P_{i-1} \Psi_i + 1}
\end{aligned}$$

<LS.29>

Recursive calculation of criterion function

$$\begin{aligned}
J_i &= (Y_i - \Phi_i \theta_i)^T (Y_i - \Phi_i \theta_i) \\
J_{i-1} &= (Y_{i-1} - \Phi_{i-1} \theta_{i-1})^T (Y_{i-1} - \Phi_{i-1} \theta_{i-1}) \\
Y_i - \Phi_i \theta_i &= Y_i - \Phi_i (\hat{\theta}_{i-1} + K_i \tilde{y}_i) \\
&= \begin{bmatrix} Y_{i-1} \\ y_i \end{bmatrix} - \begin{bmatrix} \Phi_{i-1} \\ \Psi_i^T \end{bmatrix} (\hat{\theta}_{i-1} + K_i \tilde{y}_k) \\
&= \begin{bmatrix} Y_{i-1} - \Phi_{i-1} \hat{\theta}_{i-1} \\ \tilde{y}_i \end{bmatrix} - \begin{bmatrix} \Phi_{i-1} \\ \Psi_i^T \end{bmatrix} K_i \tilde{y}_k
\end{aligned}$$

<LS.30>

Recursive calculation of criterion function

$$
\begin{aligned}
J_i &= J_{i-1} - 2K_i^T \Phi_{i-1}^T (Y_{i-1} - \Phi_{i-1}\hat{\theta}_{i-1})\tilde{y}_i + K_i^T \Phi_{i-1}^T \Phi_{i-1} K_i \tilde{y}_i^2 \\
&\quad + (1 - 2K_i^T \Psi_i + K_i^T \Psi_i \Psi_i^T K_i)\tilde{y}_i^2 \\
&= J_{i-1} - 2K_i^T (\Phi_{i-1}^T Y_{i-1} - \Phi_{i-1}^T \Phi_{i-1}\hat{\theta}_{i-1})\tilde{y}_i \\
&\quad + (1 - 2K_i^T \Psi_i + K_i^T \Phi_i \Phi_i^T K_i)\tilde{y}_i^2 \\
&= J_{i-1} + (1 - 2K_i^T \Psi_i + K_i^T \Phi_i \Phi_i^T K_i)\tilde{y}_i^2 \\
&= J_{i-1} + (1 - 2K_i^T \Psi_i + K_i^T P_i^{-1} K_i)\tilde{y}_i^2 \\
&= J_{i-1} + (1 - 2K_i^T \Psi_i + K_i^T \Psi_i)\tilde{y}_i^2 \\
&= J_{i-1} + (1 - K_i^T \Psi_i)\tilde{y}_i^2 \\
&= J_{i-1} + (1 - \Psi_i^T P_{i-1}\Psi_i(\Psi_i^T P_{i-1}\Psi_i + 1)^{-1})\tilde{y}_i^2 \\
&= J_{i-1} + \frac{\Psi_i^T P_{i-1}\Psi_i + 1 - \Psi_i^T P_{i-1}\Psi_i}{\Psi_i^T P_{i-1}\Psi_i + 1}\tilde{y}_i^2 \\
&= J_{i-1} + \frac{\tilde{y}_i^2}{\Psi_i^T P_{i-1}\Psi_i + 1}
\end{aligned}
$$

<LS.31>

he influencTe of the calculation error of gain matrix K_i on P_i

When there is error δK_i in K_i:

$$
\delta P_i = \delta K_i \Psi_i^T P_{i-1}
$$

Compute new form of P_i:

$$
\begin{aligned}
P_i &= (I - K_i \Psi_i^T) P_{i-1} \\
&= (I - K_i \Psi_i^T) P_{i-1} - P_{i-1}\Psi_i K_i^T + P_{i-1}\Psi_i K_i^T \\
&= (I - K_i \Psi_i^T) P_{i-1} - P_{i-1}\Psi_i K_i^T + K_i(\Psi_i^T P_{i-1}\Psi_i + 1)K_i^T \\
&= (I - K_i \Psi_i^T) P_{i-1} - (I - K_i \Psi_i^T) P_{i-1}\Psi_i K_i^T + K_i K_i^T \\
&= (I - K_i \Psi_i^T)(P_{i-1} - P_{i-1}\Psi_i K_i^T) + K_i K_i^T \\
&= (I - K_i \Psi_i^T) P_{i-1}(I - \Psi_i K_i^T) + K_i K_i^T
\end{aligned}
$$

<LS.32>

he influencTe of the calculation error of gain matrix K_i on P_i

When there is error δK_i in K_i:

$$
\begin{aligned}
\delta P_i &= (I - (K_i + \delta K_i)\Psi_i^T) P_{i-1}(I - \Psi_i (K_i + \delta K_i)^T) \\
&\quad + (K_i + \delta K_i)(K_i + \delta K_i)^T - P_i \\
&= -\delta K_i \Psi_i^T P_{i-1}(I - \Psi_i K_i^T) + K_i \delta K_i^T \\
&\quad - (I - K_i \Psi_i^T) P_{i-1}\Psi_i \delta K_i^T + \delta K_i K_i^T \\
&\quad + \delta K_i \Psi_i^T P_{i-1}\Psi_i \delta K_i^T + \delta K_i \delta K_i^T \\
&\quad + (I - K_i \Psi_i^T) P_{i-1}(I - \Psi_i K_i^T) + K_i K_i^T - P_i \\
&= -\delta K_i \Psi_i^T P_{i-1}(I - \Psi_i K_i^T) + K_i \delta K_i^T \\
&\quad - (I - K_i \Psi_i^T) P_{i-1}\Psi_i \delta K_i^T + \delta K_i K_i^T + O(\delta K_i) \\
&= -\delta K_i \Psi_i^T P_i^T + \delta K_i K_i^T - P_i \Psi_i \delta K_i^T + K_i \delta K_i^T + O(\delta K_i) \\
&= -\delta K_i K_i^T + \delta K_i K_i^T - K_i \delta K_i^T + K_i \delta K_i^T + O(\delta K_i) \\
&= O(\delta K_i)
\end{aligned}
$$

<LS.33>

Stability of recursive algorithms: Difference Equations

$$
\begin{aligned}
y_i &= \Psi_i^T \theta + \xi_i \\
\tilde{\theta}_i &\overset{def}{=} \theta - \hat{\theta}_i \\
&= \theta - [\hat{\theta}_{i-1} + K_i(y_i - \Psi_i^T \hat{\theta}_{i-1})] \\
&= \tilde{\theta}_{i-1} - K_i(y_i - \Psi_i^T \hat{\theta}_{i-1}) \\
&= \tilde{\theta}_{i-1} - K_i(\Psi_i^T \theta + \xi_i - \Psi_i^T \hat{\theta}_{i-1}) \\
&= \tilde{\theta}_{i-1} - K_i(\Psi_i^T \tilde{\theta}_{i-1} + \xi_i) \\
&= (I - K_i \Psi_i^T)\tilde{\theta}_{i-1} - K_i \xi_i \\
&= P_i P_{i-1}^{-1} \tilde{\theta}_{i-1} - K_i \xi_i \\
&= A_i \tilde{\theta}_{i-1} - K_i \xi_i \\
A_i &= P_i P_{i-1}^{-1}
\end{aligned}
$$

<LS.34>

Stability of recursive algorithms: eigenvalues

$$
\begin{aligned}
A_i x &= \lambda x \\
(P_{i-1}^{-1} + \Psi_i \Psi_i^T)^{-1} P_{i-1}^{-1} x &= \lambda x \\
P_{i-1}^{-1} x &= [P_{i-1}^{-1} + \Psi_i \Psi_i^T]\lambda x \\
(1-\lambda) P_{i-1}^{-1} x &= \lambda \Psi_i \Psi_i^T x \\
(1-\lambda) x^T P_{i-1}^{-1} x &= \lambda x^T \Psi_i \Psi_i^T x
\end{aligned}
$$

where： P_{i-1}^{-1} is positive definite and $\Psi_i \Psi_i^T$ is non-negtive definite ， so $0 < \lambda \leq 1$。 that is ： $\tilde{\theta}_i \leq \tilde{\theta}_0$。

<LS.35>

The relationship between least squares estimation and Kalman filtering

State space model：

$$
\begin{aligned}
\theta_{i+1} &= \theta_i \\
y_i &= \Psi_i^T \theta_i + \xi_i
\end{aligned}
$$

Kalman filtering：

$$
\begin{aligned}
\hat{\theta}_i &= \hat{\theta}_{i-1} + K_i(y_i - \Psi_i^T \hat{\theta}_{i-1}) \\
K_i &= S_i \Psi_i (\Psi_i^T S_i \Psi_i + \sigma^2)^{-1} \\
S_i &= P_{i-1} \\
P_i &= (I - K_i \Psi_i^T) P_{i-1} \\
\hat{\theta}_0 &= 0
\end{aligned}
$$

<LS.36>