Least squares identification

For white noise

Xing Chao

School of Astronautics, Northwestern Polytechnical University

System model based on input/output data

The difference equation of a SISO system is

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

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

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

Vector form

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

Basic least squares method: identification criteria

Identification criterion: least square sum of residuals.

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

Basic least square method: Derivative

$$
\begin{aligned}
\frac{\partial \mathrm{J}}{\partial \hat{\theta}_{\mathrm{k}}} &= \frac{\partial \sum_{\mathrm{i}}(\mathrm{Y}_{\mathrm{i}} - \sum_{\mathrm{m}} \Phi_{\mathrm{i,m}}\hat{\theta}_{\mathrm{m}})^2}{\partial \hat{\theta}_{\mathrm{k}}} \\
&= 2\sum_{\mathrm{i}}(\mathrm{Y}_{\mathrm{i}} - \sum_{\mathrm{m}} \Phi_{\mathrm{i,m}}\hat{\theta}_{\mathrm{m}})\frac{\partial(\mathrm{Y}_{\mathrm{i}} - \sum_{\mathrm{m}} \Phi_{\mathrm{i,m}}\hat{\theta}_{\mathrm{m}})}{\partial \hat{\theta}_{\mathrm{k}}} \\
&= 2\sum_{\mathrm{i}}(\mathrm{Y}_{\mathrm{i}} - \sum_{\mathrm{m}} \Phi_{\mathrm{i,m}}\hat{\theta}_{\mathrm{m}})\frac{\partial(-\sum_{\mathrm{m}} \Phi_{\mathrm{i,m}}\hat{\theta}_{\mathrm{m}})}{\partial \hat{\theta}_{\mathrm{k}}} \\
&= -2\sum_{\mathrm{i}}(\mathrm{Y}_{\mathrm{i}} - \sum_{\mathrm{m}} \Phi_{\mathrm{i,m}}\hat{\theta}_{\mathrm{m}})\Phi_{\mathrm{i,k}} \\
\frac{\partial \mathrm{J}}{\partial \hat{\theta}} &= (-2(\mathrm{Y} - \Phi\hat{\theta})^{\mathrm{T}}\Phi)^{\mathrm{T}} \\
&= -2\Phi^{\mathrm{T}}(\mathrm{Y} - \Phi\hat{\theta})
\end{aligned}
$$

Basic least square method: solution

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

Basic least square method: two order derivative

$$
\begin{aligned}
\frac{\partial^2 \mathrm{J}}{\partial \hat{\theta}^2} &= \frac{\partial(-2\Phi^{\mathrm{T}}(\mathrm{Y}-\Phi\hat{\theta}))}{\partial \hat{\theta}} \\
\frac{\partial \frac{\partial \mathrm{J}}{\partial \hat{\theta}}}{\partial \hat{\theta}_{\mathrm{s}}} &= \frac{\partial(-2\sum_{\mathrm{i}}(\mathrm{Y}_{\mathrm{i}}-\sum_{\mathrm{m}}\Phi_{\mathrm{i,m}}\hat{\theta}_{\mathrm{m}})\Phi_{\mathrm{i,k}})}{\partial \hat{\theta}_{\mathrm{s}}} \\
&= 2\sum_{\mathrm{i}} \frac{\partial \sum_{\mathrm{m}}\Phi_{\mathrm{i,m}}\hat{\theta}_{\mathrm{m}}}{\partial \hat{\theta}_{\mathrm{s}}}\Phi_{\mathrm{i,k}} \\
&= 2\sum_{\mathrm{i}} \Phi_{\mathrm{i,s}}\Phi_{\mathrm{i,k}} \\
\frac{\partial^2 \mathrm{J}}{\partial \hat{\theta}^2} &= 2\Phi^{\mathrm{T}}\Phi
\end{aligned}
$$

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:

$$Y_{N\times n} = \begin{bmatrix} -y_n & \cdots & -y_1 \\ -y_{n+1} & \cdots & -y_2 \\ \vdots & & \vdots \\ y_{n+N-1} & \cdots & -y_N \end{bmatrix}$$

$$U_{N\times(n+1)} = \begin{bmatrix} u_{n+1} & \cdots & u_1 \\ u_{n+2} & \cdots & u_2 \\ \vdots & & \vdots \\ u_{n+N} & \cdots & u_N \end{bmatrix}$$

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

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

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:

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

The requirement of input signal by least square method:

$$\begin{bmatrix} \mathrm{R_y} & \mathrm{R_{yu}} \\ \mathrm{R_{uy}} & \mathrm{R_u} \end{bmatrix}$$

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

(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 。

Probabilistic properties of least squares estimation

There are four main aspects of the probability property of least squares estimation,

- unbiasedness of the estimation ;
- Consistency of estimates ;
- Validity of estimation ;
- Asymptotic normality of estimators .

unbiasedness of the estimation

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

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

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

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

Consistent estimation

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

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

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

The validity of the estimation

Cramer-Rao inequality：

$$D\hat{\theta} = M^{-1}$$

其中：

$$M = E\left[\left(\frac{\partial \ln p(y|\theta)}{\partial \theta}\right)^{T}\left(\frac{\partial \ln p(y|\theta)}{\partial \theta}\right)\right]$$

The validity of the estimation

$$\begin{aligned}
y &= \Phi\theta + \xi \\
y &\sim N(\Phi\theta, \sigma^2 I) \\
p(y|\theta) &= (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left[-\frac{1}{2\sigma^2}(y-\Phi\theta)^T(y-\Phi\theta)\right] \\
\frac{\partial \ln p(y|\theta)}{\partial\theta} &= -\frac{1}{\sigma^2}(y-\Phi\theta)^T\Phi \\
M &= E\left[\frac{1}{\sigma^4}\Phi^T(y-\Phi\theta)(y-\Phi\theta)^T\Phi\right] \\
&= \frac{1}{\sigma^4}E[\Phi^T\xi\xi^T\Phi] \\
\lim_{N\to\infty} M^{-1} &= \sigma^4(\sigma^2 E[\Phi^T\Phi])^{-1} \\
&= \frac{\sigma^2}{N}R^{-1}
\end{aligned}$$

Asymptotic normality of estimation

Suppose $\{\xi(\mathrm{k})\}$ is white noise sequence with zero mean and normal distribution. then:

$$
\begin{aligned}
\mathrm{y} &= \Phi\theta + \xi \\
\mathrm{y} &\sim \mathrm{N}(\Phi\theta, \sigma^2 \mathrm{I}) \\
\hat{\theta} &= (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}\mathrm{y}
\end{aligned}
$$

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

System model

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

Identification from n = 0

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

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.

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

from n to n + 1: P_{n+1}

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

from n to n + 1: $\tilde{P}_n$

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

Computation procedure

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

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},

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

Recursive algorithm derivation :$\mathrm{P}_{\mathrm{N+1}}$

$$
\begin{aligned}
\hat{\theta}_{\mathrm{N+1}} &= \left(\begin{bmatrix} \Phi_{\mathrm{N}} \\ \Psi_{\mathrm{N+1}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \Phi_{\mathrm{N}} \\ \Psi_{\mathrm{N+1}}^{\mathrm{T}} \end{bmatrix} \right)^{-1} \begin{bmatrix} \Phi_{\mathrm{N}} \\ \Psi_{\mathrm{N+1}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathrm{Y}_{\mathrm{N}} \\ \mathrm{y}_{\mathrm{N+1}} \end{bmatrix} \\
&= (\Phi_{\mathrm{N}}^{\mathrm{T}} \underbrace{\Phi_{\mathrm{N}}}_{\mathrm{N,2n+1}} + \underbrace{\Psi_{\mathrm{N+1}}}_{\mathrm{2n+1,1}} \Psi_{\mathrm{N+1}}^{\mathrm{T}})^{-1} (\Phi_{\mathrm{N}}^{\mathrm{T}} \underbrace{\mathrm{Y}_{\mathrm{N}}}_{\mathrm{N,1}} + \Psi_{\mathrm{N+1}} \underbrace{\mathrm{y}_{\mathrm{N+1}}}_{\mathrm{1,1}}) \\
\hat{\theta}_{\mathrm{N+1}} &= \mathrm{P}_{\mathrm{N+1}} (\Phi_{\mathrm{N}}^{\mathrm{T}} \mathrm{Y}_{\mathrm{N}} + \Psi_{\mathrm{N+1}} \mathrm{y}_{\mathrm{N+1}})
\end{aligned}
$$

其中：

$$
\begin{aligned}
\mathrm{P}_{\mathrm{N+1}} &= (\mathrm{P}_{\mathrm{N}}^{-1} + \Psi_{\mathrm{N+1}} \Psi_{\mathrm{N+1}}^{\mathrm{T}})^{-1} \\
\mathrm{P}_{\mathrm{N}} &= (\Phi_{\mathrm{N}}^{\mathrm{T}} \Phi_{\mathrm{N}})^{-1}
\end{aligned}
$$

If the inverse of the corresponding matrix exists， then:

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

therefore：

$$\begin{aligned}
\mathrm{P}_{\mathrm{N}+1} &= (\mathrm{P}_{\mathrm{N}}^{-1}+\Psi_{\mathrm{N}+1}\Psi_{\mathrm{N}+1}^{\mathrm{T}})^{-1} \\
&= \mathrm{P}_{\mathrm{N}}-\mathrm{P}_{\mathrm{N}}\Psi_{\mathrm{N}+1}(1+\Psi_{\mathrm{N}+1}^{\mathrm{T}}\mathrm{P}_{\mathrm{N}}\Psi_{\mathrm{N}+1})^{-1}\Psi_{\mathrm{N}+1}^{\mathrm{T}}\mathrm{P}_{\mathrm{N}} \\
\hat{\theta}_{\mathrm{N}+1} &= \mathrm{A}+\mathrm{B} \\
\mathrm{A} &= \mathrm{P}_{\mathrm{N}+1}\Phi_{\mathrm{N}}^{\mathrm{T}}\mathrm{Y}_{\mathrm{N}} \\
\mathrm{B} &= \mathrm{P}_{\mathrm{N}+1}\Psi_{\mathrm{N}+1}\mathrm{y}_{\mathrm{N}+1} \\
\mathrm{i} &= 1+\Psi_{\mathrm{N}+1}^{\mathrm{T}}\mathrm{P}_{\mathrm{N}}\Psi_{\mathrm{N}+1}
\end{aligned}$$

Recursive algorithm derivation:Simplification

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

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。

Convergence:P_N

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

Convergence

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

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.

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

Recursive calculation of criterion function

$$
\begin{aligned}
\mathrm{J_i} &= (\mathrm{Y_i} - \Phi_\mathrm{i}\theta_\mathrm{i})^\mathrm{T}(\mathrm{Y_i} - \Phi_\mathrm{i}\theta_\mathrm{i}) \\
\mathrm{J_{i-1}} &= (\mathrm{Y_{i-1}} - \Phi_\mathrm{i-1}\theta_\mathrm{i-1})^\mathrm{T}(\mathrm{Y_{i-1}} - \Phi_\mathrm{i-1}\theta_\mathrm{i-1}) \\
\mathrm{Y_i} - \Phi_\mathrm{i}\theta_\mathrm{i} &= \mathrm{Y_i} - \Phi_\mathrm{i}(\hat{\theta}_\mathrm{i-1} + \mathrm{K_i}\tilde{\mathrm{y}}_\mathrm{i}) \\
&= \begin{bmatrix} \mathrm{Y_{i-1}} \\ \mathrm{y_i} \end{bmatrix} - \begin{bmatrix} \Phi_\mathrm{i-1} \\ \Psi_\mathrm{i}^\mathrm{T} \end{bmatrix} (\hat{\theta}_\mathrm{i-1} + \mathrm{K_i}\tilde{\mathrm{y}}_\mathrm{k}) \\
&= \begin{bmatrix} \mathrm{Y_{i-1}} - \Phi_\mathrm{i-1}\hat{\theta}_\mathrm{i-1} \\ \tilde{\mathrm{y}}_\mathrm{i} \end{bmatrix} - \begin{bmatrix} \Phi_\mathrm{i-1} \\ \Psi_\mathrm{i}^\mathrm{T} \end{bmatrix} \mathrm{K_i}\tilde{\mathrm{y}}_\mathrm{k}
\end{aligned}
$$

Recursive calculation of criterion function

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

he influencTe of the calculation error of gain matrix $\mathrm{K_i}$ on $\mathrm{P_i}$

When there is error $\delta\mathrm{K_i}$ in $\mathrm{K_i}$:

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

Compute new form of $\mathrm{P_i}$:

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

When there is error δK_i in K_i:

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

Stability of recursive algorithms: Difference Equations

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

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 ： $\tilde{\theta}_i \leq \tilde{\theta}_0$。

The relationship between least squares estimation and Kalman filtering

State space model:

$$
\begin{aligned}
\theta_{\mathrm{i+1}} &= \theta_{\mathrm{i}} \\
\mathrm{y_i} &= \Psi_{\mathrm{i}}^{\mathrm{T}}\theta_{\mathrm{i}} + \xi_{\mathrm{i}}
\end{aligned}
$$

Kalman filtering:

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