5 Recursive Technique

5.1 Introduction:

- The model in our textbook:

From the state space observer model above; (page 1-14), we know that

$$y(k) = -\sum_{i=1}^{p} \bar{Y}_i^{(2)} y(k-i) + \sum_{i=1}^{p} \bar{Y}_i^{(1)} u(k-i) + Du(k)$$

$$= \underbrace{\begin{bmatrix} D & -\bar{Y}_1^{(2)} & \bar{Y}_1^{(1)} & \cdots & -\bar{Y}_p^{(2)} & \bar{Y}_p^{(1)} \\ m\times r & m\times m & m\times r & & m\times m & m\times r \end{bmatrix}}_{m\times[p(m+r)+r]} \underbrace{\begin{bmatrix} u(k)_{r\times 1} \\ y(k-1)_{m\times 1} \\ u(k-1)_{r\times 1} \\ \vdots \\ y(k-p)_{m\times 1} \\ u(k-p)_{r\times 1} \end{bmatrix}}_{[p(m+r)+r]\times 1}$$

$$= \begin{bmatrix} D & \bar{Y}_1 & \cdots & \bar{Y}_p \\ m\times r & m\times(m+r) & & m\times(m+r) \end{bmatrix} \begin{bmatrix} u(k)_{r\times 1} \\ v(k-1)_{(m+r)\times 1} \\ \vdots \\ v(k-p)_{(m+r)\times 1} \end{bmatrix}; \text{ where } v(k) = \begin{bmatrix} y(k) \\ u(k) \end{bmatrix} \text{ and}$$

$(\bar{Y}_i)_{m\times(m+r)} = \begin{bmatrix} -\bar{Y}_i^{(2)}{}_{m\times m} & \bar{Y}_i^{(1)}{}_{m\times r} \end{bmatrix}$. Then, we stack them to derive the following; $\begin{bmatrix} y(0) & y(1) & \cdots & y(p) & \cdots & y(k) \end{bmatrix} = \begin{bmatrix} D & \bar{Y}_1 & \bar{Y}_2 & \cdots & \bar{Y}_p \end{bmatrix}$

- $\begin{bmatrix} u(0) & u(1) & u(2) & \cdots & u(p) & \cdots & u(k) \\ 0 & v(0) & v(1) & \cdots & v(p-1) & \cdots & v(k-1) \\ \vdots & \ddots & v(0) & \cdots & v(p-2) & \cdots & v(k-2) \\ \vdots & & \ddots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & v(0) & \cdots & v(k-p) \end{bmatrix}$. $\Rightarrow$ We can simplify the notation by defining that $\mathcal{Y}(k) = \bar{Y}\mathcal{V}_p(k-1)$.

- General FIR model; $Y(k) = U(k)A(k)$:

If we have N experiments on a system which has r input channels, m output channels and order p then

$$Y(k)=\underset{(k+1)N\times m}{\begin{bmatrix} y(k)_{N\times m} \\ y(k-1)_{N\times m} \\ \vdots \\ y(0)_{N\times m} \end{bmatrix}},\quad A(k)=\underset{r(p+1)\times m}{\begin{bmatrix} A_0(k)_{r\times m} \\ A_1(k)_{r\times m} \\ \vdots \\ A_p(k)_{r\times m} \end{bmatrix}},$$

$$U(k)=\underset{(k+1)N\times r(p+1)}{\begin{bmatrix} u(k)_{N\times r} & u(k-1)_{N\times r} & \cdots & u(k-p)_{N\times r} \\ u(k-1)_{N\times r} & u(k-2)_{N\times r} & \cdots & u\binom{k-1}{-p}_{N\times r} \\ \vdots & \vdots & & \vdots \\ u(0)_{N\times r} & u(-1)_{N\times r} & \cdots & u(-p)_{N\times r} \end{bmatrix}}\square\begin{bmatrix} u_p(k)_{N\times r(p+1)} \\ u_p(k-1)_{N\times r(p+1)} \\ \vdots \\ u_p(0)_{N\times r(p+1)} \end{bmatrix},$$

where $A(k)$ is the system parameter matrix.

5.2 Least Square:

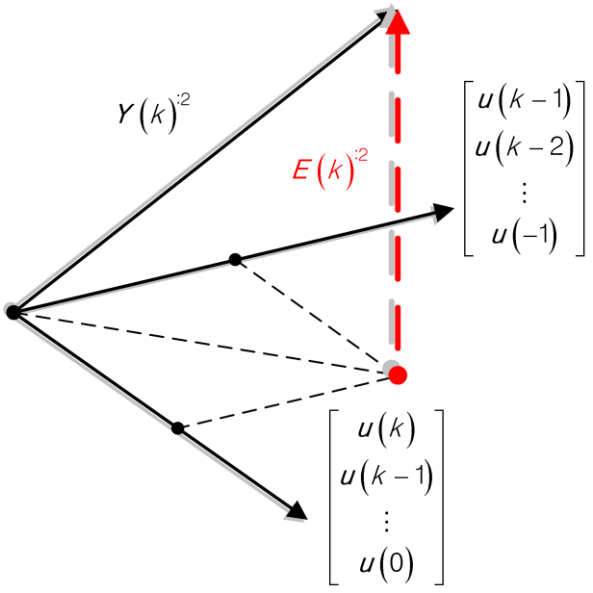
- Minimize the "square" (= 2 norm) distance between $U(k)A(k)$ and $Y(k)$ by choosing a best $A(k)$; Let $E(k)\square Y(k)-U(k)A(k)$ then

$$\min_{A(k)}\left[\operatorname{diag}\left\{E^*(k)E(k)\right\}\right]=\min_{A^{:i};\ i=1\sim m}\left\|E^{:i}\right\|_2^2=\min_{A^{:i};\ i=1\sim m}\left\|Y(k)^{:i}-U(k)A(k)^{:i}\right\|_2^2.$$

➢ We choose $A^{:i}$ according to $Y(k)^{:i}$ independently.

➢ Geometric meanings:

1. $U(k)A(k)$: linear combination of the column vectors of $U(k)$ to minimize the distance of $Y(k)-U(k)A(k)$. $\Rightarrow$ project $Y(k)$ onto the hyperplane spanned by vectors of $U(k)$.



2. $E^{:i}$: the error vector between hy-

perplane of $\text{span}\{U(k)\}$ and $Y(k)^{:i}$. The minimum distance satisfies $E(k)^{:i} \perp \text{span}\{U(k)\}$.

For 2D case, the geometric meaning can be represented by the left-hand side figure.

- L.S. Solution:

$\because E(k) = Y(k) - U(k)A(k) \Rightarrow Y(k) - \bar{Y}(k)$ and $E(k) \perp \text{span}\{U(k)\}$

$\Rightarrow U^*(k)E(k) = U^*(k)Y(k) - U^*(k)U(k)A(k) = 0$

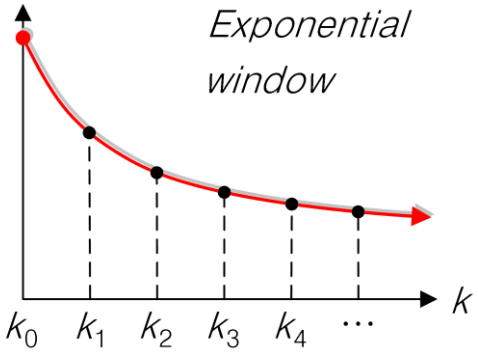
$\Rightarrow A(k) = \left[U^*(k)U(k)\right]^{-1}\left[U^*(k)Y(k)\right] \approx \left[R_{UU}\right]^{-1}\left[R_{UY}\right]$.

➢ L.S. with forgetting factor λ:

$\min_{A(k)}\left[\text{diag}\left\{E^*(k)QE(k)\right\}\right] \Rightarrow \min_{A(k)}\left\|E(k)\right\|_Q$

$= \min_{A^{:i};\ i=1\sim m}\left\|Y(k)^{:i} - U(k)A(k)^{:i}\right\|_Q^2$,

where $\left\|E(k)\right\|_Q \equiv \sum_{\tau=0}^{k} e^*(\tau)e(\tau)\lambda^{k-\tau}$.



So, forgetting factor λ is an exponential weighting window. It means that we redefine inner product of $\langle y(k), u(k)\rangle \equiv \sum_{i=0}^{l}\sum_{\tau=0}^{k} y_i(\tau)u_i(\tau)\lambda^{k-\tau}$.

- R.L.S.; Recursive Least Square:

The computation of $\left[U^*(k)U(k)\right]^{-1}\left[U^*(k)Y(k)\right]$ will increase as k grow.

We can find $\left[U^*(k+1)U(k+1)\right]^{-1}\left[U^*(k+1)Y(k+1)\right]$ by update $\left[U^*(k)U(k)\right]^{-1}\left[U^*(k)Y(k)\right]$ with new $u(k+1)$ and $y(k+1)$ available.

So, we need $A(k)$, $E(k) = Y(k) - U(k)A(k)$, and $\langle E(k), E(k)\rangle$.

➢ Def.: $P(k) \equiv \left[U^*(k)U(k)\right]^{-1}$. $\Rightarrow P^*(k) = P(k)$ We need update $P(k+1)$ when we update $A(k+1)$ from $A(k)$.

1. Matrix inverse lemma: if A and C are square matrices.

$$\left[A+BCD\right]^{-1}=A^{-1}-A^{-1}B\left[C^{-1}+DA^{-1}B\right]^{-1}DA^{-1}$$

[pf]: $\left[A+BCD\right]\left[A^{-1}-A^{-1}B\left[C^{-1}+DA^{-1}B\right]^{-1}DA^{-1}\right]$

$$=I+BCDA^{-1}-\left[A+BCD\right]A^{-1}B\left[C^{-1}+DA^{-1}B\right]^{-1}DA^{-1}$$

$$=I+BCDA^{-1}-BC\left[C^{-1}+DA^{-1}B\right]\left[C^{-1}+DA^{-1}B\right]^{-1}DA^{-1}=I.$$

2. Update $P(k+1)$:

$$\because U(k+1)=\begin{bmatrix}u_p(k+1)\\ U(k)\end{bmatrix}\Rightarrow P^{-1}(k+1)=U^*(k+1)U(k+1)$$

$$=u_p^*(k+1)u_p(k+1)+U^*(k)U(k)=u_p^*(k+1)u_p(k+1)+P^{-1}(k)$$

$$\Rightarrow P(k+1)=\left[P^{-1}(k)+u_p^*(k+1)u_p(k+1)\right]^{-1};\ \text{using above lemma}$$

$$=P(k)-P(k)u_p^*(k+1)\left[I+u_p(k+1)P(k)u_p^*(k+1)\right]^{-1}u_p(k+1)P(k).$$

➢ Define: a-priori estimation; $\hat{y}(k+1)=u_p(k+1)A(k)$,

a-priori estimation error; $\hat{e}(k+1)=y(k+1)-\hat{y}(k+1)$.

a-posteriori estimation; $\bar{y}(k+1)=u_p(k+1)A(k+1)$,

a-posteriori estimation error; $\bar{e}(k+1)=y(k+1)-\bar{y}(k+1)$.

1. Update $A(k+1)$:

First, we let $U=U(k)$, $u_p=u_p(k+1)$, $Y=Y(k)$, $y=y(k+1)$,

$P=P(k)$, $\Delta_{N\times N}=I_{N\times N}+(u_p)_{N\times r(p+1)}P_{r(p+1)\times r(p+1)}(u_p)^*_{r(p+1)\times N}=\Delta^*_{N\times N}$, and

$\hat{Y}=\hat{Y}(k+1)=U(k+1)A(k)$. Then, $A(k+1)=P(k+1)U^*(k+1)Y(k+1)$

$$=\left(P-\frac{Pu_p^*u_pP}{\Delta}\right)\left(U^*Y+u_p^*y\right);\ \left(\because U^*(k+1)=\begin{bmatrix}u_p^* & U^*\end{bmatrix},\ Y(k+1)=\begin{bmatrix}y\\ Y\end{bmatrix}\right)$$

$$=PU^*Y-\frac{Pu_p^*u_pP}{\Delta}U^*Y+Pu_p^*y-\frac{Pu_p^*u_pP}{\Delta}u_p^*y=A(k)-\frac{Pu_p^*}{\Delta}u_pA(k)$$

$$+\frac{1}{\Delta}\left[Pu_p^*\left(I+u_pPu_p^*\right)y-Pu_p^*u_pPu_p^*y\right]=A(k)-\frac{Pu_p^*}{\Delta}\hat{y}+\frac{Pu_p^*}{\Delta}y$$

$$=A(k)+\frac{Pu_p^*}{\Delta}(y-\hat{y})=A(k)+\frac{P(k)u_p^*(k+1)}{I+u_p(k+1)P(k)u_p^*(k+1)}\hat{e}(k+1).\ \text{Let}$$

$K(k+1)\triangleq\dfrac{P(k)u_p^*(k+1)}{I+u_p(k+1)P(k)u_p^*(k+1)}$ then

$A(k+1)=A(k)+K(k+1)\hat{e}(k+1)$. We call this update equation to be in feedback form. So, the update equation of $P(k+1)$ can be written as

$P(k+1)=P(k)-K(k+1)u_p(k+1)P(k)=\left[I-K(k+1)u_p(k+1)\right]P(k)$.

2. The errors of update estimations:

For a-priori, $\hat{e}(k+1)=y(k+1)-\hat{y}(k+1)=y(k+1)-u_p(k+1)A(k)$.

For a-posteriori, $\bar{e}(k+1)=y(k+1)-\bar{y}(k+1)=y-u_pA(k+1)$

$$=y-u_p\left[A(k)+K(k+1)\hat{e}(k+1)\right]=y-u_pA(k)-u_pK(k+1)\hat{e}(k+1)$$

$$=\hat{e}(k+1)-u_pK(k+1)\hat{e}(k+1)=\hat{e}(k+1)\left[I-u_pK(k+1)\right]$$

$$=\hat{e}(k+1)\left[I-\frac{u_pPu_p^*}{I+u_pPu_p^*}\right]=\frac{\hat{e}(k+1)}{I+u_p(k+1)P(k)u_p^*(k+1)}.$$

3. The square norm of update estimation error:

Error vector; $E(k+1)\triangleq Y(k+1)-\bar{Y}(k+1)=\begin{bmatrix}y(k+1)\\Y(k)\end{bmatrix}-\begin{bmatrix}u_pA(k+1)\\U(k)A(k+1)\end{bmatrix}$

$$=\begin{bmatrix}\bar{e}(k+1)\\\begin{pmatrix}Y(k)-U(k)A(k)\\-U(k)K(k+1)\hat{e}(k+1)\end{pmatrix}\end{bmatrix}=\begin{bmatrix}\bar{e}(k+1)\\\begin{pmatrix}Y(k)-\bar{Y}(k)\\-U(k)K(k+1)\hat{e}(k+1)\end{pmatrix}\end{bmatrix}$$

$=\begin{bmatrix}\bar{e}(k+1)\\E(k)-U(k)K(k+1)\hat{e}(k+1)\end{bmatrix}$. Then, Square Norm of error vector;

$$E^*(k+1)E(k+1)=\bar{e}^*(k+1)\bar{e}(k+1)+\left[E(k)-U(k)K(k+1)\hat{e}(k+1)\right]^*$$

$$\left[E(k)-U(k)K(k+1)\hat{e}(k+1)\right]=\bar{e}^*(k+1)\bar{e}(k+1)+E^*(k)E(k)$$

$$+\hat{e}^*(k+1)K^*(k+1)U^*(k)U(k)K(k+1)\hat{e}(k+1);\ (\because E(k)\perp \text{span}\{U(k)\}$$

$$\Rightarrow E^*(k)U(k)=U^*(k)E(k)=0)\ =\bar{e}^*(k+1)\bar{e}(k+1)+E^*(k)E(k)$$

$$+\hat{e}^*(k+1)\frac{u_p(k+1)P(k)P^{-1}(k)P(k)u_p^*(k+1)}{\left[I+u_p(k+1)P(k)u_p^*(k+1)\right]^2}\hat{e}(k+1)=E^*(k)E(k)$$

$$+\bar{e}^*(k+1)\bar{e}(k+1)+\bar{e}^*(k+1)\left[u_p(k+1)P(k)u_p^*(k+1)\right]\bar{e}(k+1)$$

$$=E^*(k)E(k)+\bar{e}^*(k+1)\left[I+u_p(k+1)P(k)u_p^*(k+1)\right]\bar{e}(k+1)$$

$$=E^*(k)E(k)+\bar{e}^*(k+1)\hat{e}(k+1)=E^*(k)E(k)+\hat{e}^*(k+1)\bar{e}(k+1)$$

$$=E^*(k)E(k)+\hat{e}^*(k+1)\hat{e}(k+1)\left[I+u_p(k+1)P(k)u_p^*(k+1)\right]^{-1}.$$

➢ Algorithm:

1. Original algorithm:

Step 1: Get $u_p(k+1)$.

Step 2: Compute $\hat{y}(k+1)=u_p(k+1)A(k)$ and

$$K(k+1)=\frac{P(k)u_p^*(k+1)}{I+u_p(k+1)P(k)u_p^*(k+1)}.$$

Step 3: Compute $P(k+1)=P(k)-K(k+1)u_p(k+1)P(k)$. For a given $y(k+1)$, we obtain $\hat{e}(k+1)=y(k+1)-\hat{y}(k+1)$ and

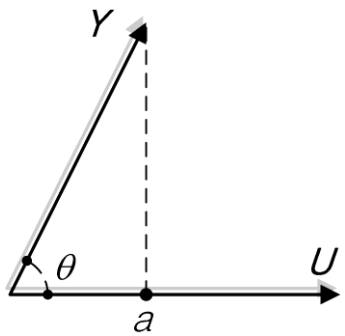
$$A(k+1)=A(k)+K(k+1)\hat{e}(k+1).$$

Step 4: Compute $\bar{e}(k+1)=\dfrac{\hat{e}(k+1)}{I+u_p(k+1)P(k)u_p^*(k+1)}$ and

$$E^*(k+1)E(k+1)=E^*(k)E(k)+\bar{e}^*(k+1)\hat{e}(k+1).$$

2. Two problems:

✓ If $\left[U^*(k)U(k)\right]^{-1}$ is singular when $k<n$ then we take $P(k)$ as $P(k)\square\left[U^*(k)U(k)\right]^{\#}$; pseudo inverse. For a matrix A, $\exists A^{\#}$ э $A^{\#}AA^{\#}=A^{\#}$, and



$AA^{\#}A = A$,. E.g., $A = 0$, $A^{\#} = 0$.

✓ How to start the recursive algorithm: When $k = 0$, we must have $P(0)$ and $A(0)$ in order to be able to start the above algorithm;

✧ We can set $A(0) = 0$ and $P(0) = \sigma I$, where σ is a large enough number. Observe the simple case, Y and U are one dimension. $a = \frac{\|Y\|}{\|U\|}\cos\theta = \frac{\|Y\|}{\|U\|}\frac{\langle Y,U\rangle}{\|Y\|\|U\|} = \frac{\langle Y,U\rangle}{\langle U,U\rangle}$. If we take $A(0) = 0$ and $P(0) = \sigma I$ then $\Rightarrow K(1) = \frac{\sigma u_p^*(1)}{I + \sigma u_p(1)u_p^*(1)}$ and $\hat{y}(1) = u_p(1)A(0) = 0$

$$\Rightarrow A(1) = A(0) + \frac{\sigma u_p^*(1)}{I + \sigma u_p(1)u_p^*(1)}\left[y(1) - \hat{y}(1)\right] = \frac{\sigma u_p^*(1)y(1)}{I + \sigma u_p(1)u_p^*(1)}$$

$$\xrightarrow{\sigma \to \infty} \frac{\langle u_p(1), y(1)\rangle}{\langle u_p(1), u_p(1)\rangle}.$$

✧ If we have initial guess on $A(0)$ as A_0, then we take $A(0) = A_0$ and $P(0) = \sigma I$. It implies that σ is small when we have strong confidence. Otherwise, σ is large when we have low confidence.

3. The algorithm of RLS with forgetting factor λ:

Step 1: Get $u_p(k+1)$.

Step 2: Compute $\hat{y}(k+1) = u_p(k+1)A(k)$ and

$$K(k+1) = \frac{P(k)u_p^*(k+1)}{\lambda I + u_p(k+1)P(k)u_p^*(k+1)}.$$

Step 3: Compute $P(k+1) = \left[I - K(k+1)u_p(k+1)\right]\lambda^{-1}P(k)$. For a given $y(k+1)$, we obtain $\hat{e}(k+1) = y(k+1) - \hat{y}(k+1)$ and

$$A(k+1) = A(k) + K(k+1)\hat{e}(k+1).$$

Step 4: Compute $\bar{e}(k+1)=\dfrac{\hat{e}(k+1)\lambda}{\lambda I+u_p(k+1)P(k)u_p^*(k+1)}$ and

$$E^*(k+1)E(k+1)=\lambda E^*(k)E(k)+\bar{e}^*(k+1)\hat{e}(k+1).$$

- Homework 9:

1. Please verify the above algorithm (RLS with λ).
2. Please write a Matlab program for RLS algorithm which parameters can be set by us freely. As the same above, it must include the detail of the program about parameters and function blocks.
3. Please use your program to identify the system $y(t+1)=ay(t)+by(t-1)$ subjected a sine wave input and compute the eigenvalues of the above difference equation. Please check the result by comparing with its frequency response.

5.3 Problem of General Least Square:

- $\min\limits_{A}\left\|Y(k)-U(k)A(k)\right\|_2^2 \xrightarrow{LSE} \bar{A}(k)=\left[U^*(k)U(k)\right]^{-1}\left[U^*(k)\bar{Y}(k)\right]$

where $\bar{Y}(k)=U(k)\bar{A}(k)$.

Let $U(k)\equiv\left[u_1(k),u_2(k)\right]=\left[u_1(k),u_1(k)+\varepsilon(k)\right]$, $\varepsilon(k)\perp \text{span}\{u_1(k)\}$, and $\left\|\varepsilon(k)\right\|^2 \ll \left\|u_1(k)\right\|^2$, then $\left\|\left[U^*(k)U(k)\right]\right\|=\begin{vmatrix} u_1^*u_1 & u_1^*u_1 \\ u_1^*u_1 & u_1^*u_1+\varepsilon^*\varepsilon \end{vmatrix}$

$=\left|u_1^*u_1\left(u_1^*u_1+\varepsilon^*\varepsilon\right)-u_1^*u_1u_1^*u_1\right|=\left|u_1^*u_1\varepsilon^*\varepsilon\right|$. There is no problem theoretically.

- Finite precision:

  For example, there are five effective digits;

  $10000+0.0100=10000.0100\xrightarrow[digits]{take\ five}10000.$ $\therefore u_1^*u_1+\varepsilon^*\varepsilon=u_1^*u_1$

  when $u_1^*u_1 \gg \varepsilon^*\varepsilon \Rightarrow \left\|\left[U^*(k)U(k)\right]\right\|\approx 0 \Rightarrow \left[U^*(k)U(k)\right]^{-1}$ is numerical unstable when $u_1(k)\to u_2(k)$.

- We can use similarity transformation T to $U(k)$.

  $\Rightarrow \left[U^*(k)U(k)\right] = T^{-1}\left[\bar{U}^*(k)\bar{U}(k)\right]T$, where $\bar{U}(k) = \left[u_1(k), \varepsilon(k)\right]$.

  $= \begin{bmatrix} u_1^*(k)u_1(k) & 0 \\ 0 & \varepsilon_1^*(k)\varepsilon_1(k) \end{bmatrix}$, $\because u_1(k) \perp \varepsilon(k)$

  $\bar{U}^*(k)\bar{U}(k) =$ If $\begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \end{bmatrix}$, where $\sigma_1 \geq \cdots \geq \sigma_n$ and conditional number defined as σ_1 / σ_n. If conditional number are too large then $\left[U^*(k)U(k)\right]^{-1}$ will be numerical unstable. So, we drive $T^{-1}D^{-1}T$ instead of $\left[U^*(k)U(k)\right]^{-1}$, where $D^{-1} = \begin{bmatrix} \sigma_1^{-1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n^{-1} \end{bmatrix}$. It's numerical stable because there is no summation of large and small numbers.

- The process for looking for $U^*(k)U(k) = T^{-1}D(k)T$ is equivalent to looking for $\bar{U} = \left[\bar{u}_1 \quad \cdots \quad \bar{u}_n\right] = UT$ ∋ $\bar{u}_1 \perp \cdots \perp \bar{u}_n$. We call this process as orthogonal decomposition of U to be orthogonal basis $\left[\bar{u}_1 \quad \cdots \quad \bar{u}_n\right]$.

- Orthogonal decomposition:

- G.S.: Given $u_1, \cdots, u_n$ to find the orthogonal basis of a space; $\text{span}\{u_1, \cdots, u_n\}$.

1. $e_1 \square u_1 - U_{2\sim n}\left[U_{2\sim n}^* U_{2\sim n}\right]^{-1}\left[U_{2\sim n}^* u_1\right] \Rightarrow e_1 \perp u_2, \cdots, u_n$.
2. $e_2 \square u_2 - U_{3\sim n}\left[U_{3\sim n}^* U_{3\sim n}\right]^{-1}\left[U_{3\sim n}^* u_2\right] \Rightarrow e_2 \perp u_3, \cdots, u_n$, $\because u_2 \perp e_1$, and $U_{3\sim n}^* \perp e_1$, $\Rightarrow \therefore e_2 \perp e_1$.
3. ….

n−1. $e_{n-1} \square u_{n-1} - U_n \left[U_n^* U_n\right]^{-1} \left[U_n^* u_{n-1}\right] \Rightarrow e_{n-1} \perp u_n,\ e_{n-1} \perp e_1, \cdots, e_{n-2}$.

n. $e_n \square u_n \Rightarrow e_n \perp e_1, \cdots, e_{n-1}$.

Although $\left[e_1, \cdots, e_n\right]$ are orthogonal basis, but the basis of $\left[U_{i\sim n}^* U_{i\sim n}\right]^{-1}$ are not orthogonal. So, it still being possibly numerical unstable.

➢ M.G.S.: We modify the above process to get the following;

1. Take $e_1 = u_1$ and $u_i^{(1)} = u_i - e_1 \left[e_1^* e_1\right]^{-1} \left[e_1^* u_i\right],\ \forall i = 2, \cdots, n$.

2. Take $e_2 = u_2^{(1)}$ and $u_i^{(2)} = u_i^{(1)} - e_2 \left[e_2^* e_2\right]^{-1} \left[e_2^* u_i^{(1)}\right],\ \forall i = 3, \cdots, n$,

   $\because e_1 \perp u_2^{(1)} \Rightarrow e_2 \perp e_1$.

3. ….

n−1. Take $e_{n-1} = u_{n-1}^{(n-2)}$ and $u_i^{(n-1)} = u_i^{(n-2)} - e_{n-1} \left[e_{n-1}^* e_{n-1}\right]^{-1} \left[e_{n-1}^* u_i^{(n-2)}\right]$

   $\forall i = n,\ \Rightarrow e_{n-1} \perp e_1, \cdots, e_{n-2}$.

n. Take $e_n = u_n^{(n-1)},\ \Rightarrow e_n \perp e_1, \cdots, e_{n-1}$.

It implies that the conditional number of every $\left[e_i^* e_i\right]_{1\times 1}^{-1}$ is always one. it promises $\left[e_i^* e_i\right]^{-1}$ to be numerical stable. However, $\left[e_i^* e_i\right]^{-1}$ is replaced by $\left[e_i^* e_i\right]^{\#} = 0$ if $\left\|e_i^*\right\| = 0 \Rightarrow e_{i+1} = u_{i+1}^{(i)} = u_{i+1}^{(i-1)} - e_i \left[e_i^* e_i\right]^{\#} \left[e_i^* u_{i+1}^{(i-1)}\right]$.

Define $\text{span}\left\{e_1 \cdots e_i\right\} = \text{span}\left\{u_1 \cdots u_i\right\} \square H_i$; $\left\{\underset{H_1}{e_1} \ \vdots \ u_2 \quad u_3 \quad \cdots \quad u_n\right\} \rightarrow$

$\left\{\underset{H_1}{e_1} \ \underset{\oplus}{\overset{\perp}{\vdots}} \ \underset{e_2}{u_2^{(1)}} \quad u_3^{(1)} \quad \cdots \quad u_n^{(1)}\right\} \rightarrow \left\{\underset{H_2}{e_1 \quad e_2} \ \underset{\oplus}{\overset{\perp}{\vdots}} \ \underset{e_3}{u_3^{(2)}} \quad \cdots \quad u_n^{(2)}\right\} \Rightarrow H_i = H_{i-1} \underset{\oplus}{\perp} e_i$

is the perpendicularly projected space. Define $P_n u \square H_n \left[H_n^* H_n\right]^{-1} \left[H_n^* u\right]$ then $u_i^{(n)} = u_i - P_n u_i$. M.G.S adds a new vector $e_n = u_n^{(n-1)}$ to form a new H_n every time and this vector is perpendicular to the basis of H_{n-1} only. Because the remainder vectors are also perpendicular to the H_{n-1}, this

will reduce their components on e_n axes. So, we can say $u_i^{(1)}$ is the component of u_i perpendicular to H_1, …, and $u_i^{(n)}$ is the component of u_i perpendicular to H_n. $e_n\left[e_n^* e_n\right]^{-1}\left[e_n^* u_i^{(n-1)}\right]$ is the component of u_i falling on e_n.

- Example: $\left[y \quad 0.9y \quad y+\varepsilon\right]$; ε is a random noise.

$e_1 = y$, $e_2 = y_2^{(1)} = y_2 - e_1\left[e_1^* e_1\right]^{-1}\left[e_1^* y_2\right] = 0.9y - y\left[y^* y\right]^{-1}\left[y^* 0.9y\right]$

$= 0.9y - 0.9y = 0$, $y_3^{(1)} = y + \varepsilon - y\left[y^* y\right]^{-1}\left[y^*\left(y+\varepsilon\right)\right]$ $(\because \left[y^*\varepsilon\right] = 0)$

$= y + \varepsilon - y\left[y^* y\right]^{-1}\left[y^* y\right] = \varepsilon$, $e_3 = y_3^{(2)} = y_3^{(1)} - e_2\left[e_2^* e_2\right]^{-1}\left[e_2^* y_3^{(1)}\right]$

$\because \varepsilon - e_2\left[e_2^* e_2\right]^{\#}\left[e_2^*\varepsilon\right] = \varepsilon - 0 \because 0 = \varepsilon$. $\therefore \Rightarrow \left[e_1 \quad e_2 \quad e_3\right] = \left[y \quad 0 \quad \varepsilon\right]$.

Since the original basis has two vector being linear dependent, the orthogonal decomposition has only two degrees of freedom.

5.4 Lattice Filter:

- L.S. problem: $\min_A \left\|Y(k) - U(k)A\right\|_2^2 = \min_A \left\|E(k)\right\|_2^2$; minimum error.

- Forward prediction model; n[th] order:

$$Y(k) = \begin{bmatrix} z(k) \\ z(k-1) \\ \vdots \end{bmatrix},\quad U(k) = \begin{bmatrix} z(k-1) & z(k-2) & \cdots & z(k-n) \\ z(k-2) & z(k-3) & \cdots & z(k-1-n) \\ \vdots & \vdots & & \vdots \end{bmatrix},$$

$$E_n^f(k) = \begin{bmatrix} e_n^f(k) \\ e_n^f(k-1) \\ \vdots \end{bmatrix} = \begin{bmatrix} z(k) \\ z(k-1) \\ \vdots \end{bmatrix} - \begin{bmatrix} z(k-1) & z(k-2) & \cdots & z(k-n) \\ z(k-2) & z(k-3) & \cdots & z(k-1-n) \\ \vdots & \vdots & & \vdots \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

$= z(k) - H_n(k-1)A_n(k) = z(k) - P_n(k-1)z(k) = P_n^{\perp}(k-1)z(k)$.

- Backward prediction model; n^{th} order:

$$Y(k) \equiv \begin{bmatrix} z(k-n) \\ z(k-1-n) \\ \vdots \end{bmatrix},\ U(k) \equiv \begin{bmatrix} z(k) & z(k-1) & \cdots & z(k-n+1) \\ z(k-1) & z(k-2) & \cdots & z(k-n) \\ \vdots & \vdots & & \vdots \end{bmatrix},$$

$E_n^b(k) =$

$$\begin{bmatrix} e_n^b(k) \\ e_n^b(k-1) \\ \vdots \end{bmatrix} = \begin{bmatrix} z(k-n) \\ z(k-1-n) \\ \vdots \end{bmatrix} - \begin{bmatrix} z(k) & z(k-1) & \cdots & z(k-n+1) \\ z(k-1) & z(k-2) & \cdots & z(k-n) \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

$\equiv z(k-n) - H_n(k)A_n(k) \equiv z(k-n) - P_n(k)z(k-n) \equiv P_n^{\perp}(k)z(k-n)$,

$H_{n+1}(k) = H_n(k-1) \oplus z(k) = H_n(k-1) \underset{\oplus}{\perp} E_n^f(k)$.

$\Rightarrow P_{n+1}(k) = P_n(k-1) + P_n^f(k)$, where $P_n^f(k)$ is an orthogonal projection operator for space $E_n^f(k)$; $P_n^f(k) \equiv E_n^f(k)\left[E_n^f(k)^* E_n^f(k)\right]^{-1} E_n^f(k)^*$. This equation means that the term $P_n^f(k)$ is the difference between

$E_n^f(k) = z(k) - P_n(k-1)z(k)$ & $E_{n+1}^f(k+1) = z(k+1) - P_{n+1}(k)z(k+1)$.

$\Rightarrow E_{n+1}^b(k) = P_{n+1}^{\perp}(k)z(k-n-1) = z(k-n-1) - P_{n+1}(k)z(k-n-1)$

$= z(k-n-1) - \left[P_n(k-1) + P_n^f(k)\right]z(k-n-1)$

$= z(k-n-1) - P_n(k-1)z(k-n-1) - P_n^f(k)z(k-n-1)$

$= E_n^b(k-1) - E_n^f(k)\left[E_n^f(k)^* E_n^f(k)\right]^{-1}\left[E_n^f(k)^* z(k-n-1)\right] \cdots (1)$.

- Lemma: $\forall H, y, u$, the P is the projection onto H.

$\Rightarrow \langle Py, u \rangle = \langle Py, Pu \rangle = \langle P^2 y, Pu \rangle = \langle y, Pu \rangle$

[pf]: $\because P = H\left[H^*H\right]^{-1} H^* \Rightarrow Py = H\left[H^*H\right]^{-1} H^* y$,

$\Rightarrow \langle Py, Pu \rangle = y^* H\left[H^*H\right]^{-1} H^* H\left[H^*H\right]^{-1} H^* u = y^* H\left[H^*H\right]^{-1} H^* u$

$= \langle Py, u \rangle = \langle y, Pu \rangle$, and $P^2 y = H\left[H^*H\right]^{-1} H^* H\left[H^*H\right]^{-1} H^* y$

$= H\left[H^*H\right]^{-1} H^* y = Py$.

➢ $\because E_n^b(k-1) = P_n^{\perp}(k-1)z(k-n-1)$. Let $P = P_n^{\perp}(k-1)$, $Py = E_n^f(k)$ and $u = z(k-n-1)$. From Lemma, $E_n^f(k)^* z(k-n-1) = \langle Py, u \rangle$ $= \langle Py, Pu \rangle = E_n^f(k)^* P_n^{\perp}(k-1)z(k-n-1) = E_n^f(k)^* E_n^b(k-1)$. So, $(1) \Rightarrow$

$$E_{n+1}^b(k) = E_n^b(k-1) - P_n^f(k)z(k-n-1) = E_n^b(k-1) - P_n^f(k)E_n^b(k-1)$$

$$= E_n^b(k-1) - E_n^f(k)\left[E_n^f(k)^* E_n^f(k)\right]^{-1}\left[E_n^f(k)^* E_n^b(k-1)\right]$$

$\Box\ E_n^b(k-1) - E_n^f(k)R_n^f(k)^{-1}K_n(k)$. In similarly,

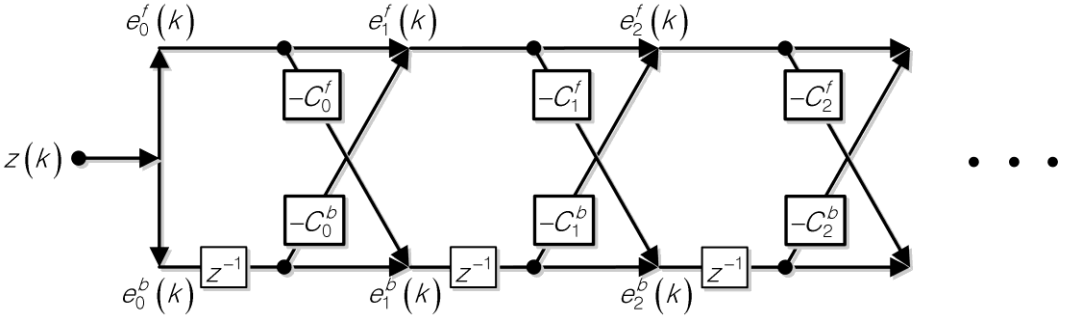
$$E_{n+1}^f(k) = E_n^f(k) - E_n^b(k-1)\left[E_n^b(k-1)^* E_n^b(k-1)\right]^{-1}\left[E_n^b(k-1)^* E_n^f(k)\right]$$

$\Box\ E_n^f(k) - E_n^b(k-1)R_n^b(k-1)^{-1}K_n^*(k)$. The above two equations imply the following equations: $e_{n+1}^b(k) = e_n^b(k-1) - e_n^f(k)R_n^f(k)^{-1}K_n(k)$;

Backward order error update equation and Forward order error update equation $e_{n+1}^f(k) = e_n^f(k) - e_n^b(k-1)R_n^b(k-1)^{-1}K_n^*(k)$.

➢ For stationary signal; $R_n^f(k)^{\#}K_n(k) = cons. \Box C_n^f$:
$R_n^b(k-1)^{\#}K_n^*(k) = cons. \Box C_n^b$

If initial condition is $H_0(k-1) = \varnothing \Rightarrow P_0(k-1)z(k) = 0 \Rightarrow P_0^{\perp}(k-1)z(k)$ $= z(k) - 0 = z(k) \Rightarrow e_0^f(k) = z(k)$, $e_0^b(k) = z(k)$ then



Because the structure looks like a lattice, we call is Lattice filter.

5.5 Adaptive Lattice Filter:

- We need to modify $R_n^f(k)$, $R_n^b(k)$ and $K_n(k)$ with income data.

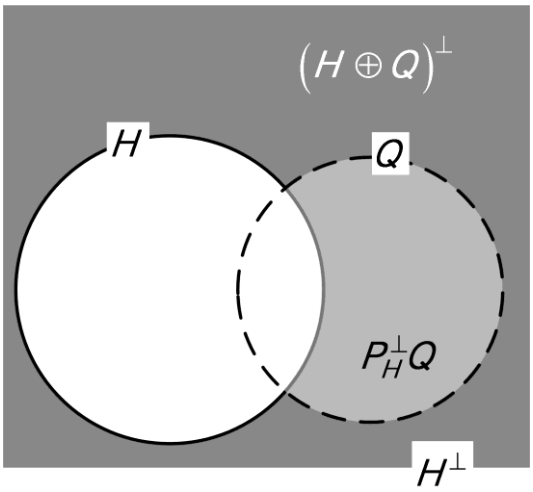
Please look into $H_n(k+1) =$

$$\begin{bmatrix} z(k+1) & z(k) & \cdots & z(k+1-n) \\ \cdots\cdots & & & \\ & H_n(k)_{\infty\times(n+1)} & & \end{bmatrix} \square \begin{bmatrix} h_n(k+1) \\ \cdots\cdots \\ H_n(k) \end{bmatrix}_{(\infty+1)\times(n+1)}.$$

If we define pinning vector as

$$\upsilon \square \begin{bmatrix} 1 \\ \cdots \\ 0 \end{bmatrix}_{(\infty+1)\times 1} \quad \text{then} \quad H \oplus \upsilon = \begin{bmatrix} h \\ \cdots \\ H^- \end{bmatrix} \oplus \begin{bmatrix} 1 \\ \cdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \cdots \\ H^- \end{bmatrix} \overset{\perp}{\oplus} \begin{bmatrix} 1 \\ \cdots \\ 0 \end{bmatrix} \square S.$$



➢ Lemma: $H^{\perp} = (H \oplus Q)^{\perp} \overset{\perp}{\oplus} (P_H^{\perp} Q)$.

➢ Lemma: $\forall$ vector, $V = \begin{bmatrix} v \\ \cdots \\ V^- \end{bmatrix}_{(\infty+1)\times 1}$, $H = \begin{bmatrix} h \\ \cdots \\ H^- \end{bmatrix}$, $\upsilon = \begin{bmatrix} 1 \\ \cdots \\ 0 \end{bmatrix}$ and

$$S = H \oplus \upsilon \Rightarrow P_S^{\perp} V = \begin{bmatrix} 0 \\ \cdots\cdots \\ P_{H^-}^{\perp} V^- \end{bmatrix}_{(\infty+1)\times 1}, \quad P_{H^-}^{\perp} \text{ project a vector onto } H^{-\perp}.$$

[Pf]: Let $\bar{H} = \begin{bmatrix} 0 \\ \cdots \\ H^- \end{bmatrix}$ then $P_S^{\perp} V = V - P_S V = V - P_{\bar{H}} V - P_{\upsilon} V$, where

$$P_{\bar{H}} V = \bar{H}\left[\bar{H}^* \bar{H}\right]^{-1} \bar{H}^* V = \begin{bmatrix} 0 \\ \cdots \\ H^- \end{bmatrix} \left\{ \begin{bmatrix} 0 & \vdots & H^{-*} \end{bmatrix} \begin{bmatrix} 0 \\ \cdots \\ H^- \end{bmatrix} \right\}^{-1} \begin{bmatrix} 0 & \vdots & H^{-*} \end{bmatrix} \begin{bmatrix} v \\ \cdots \\ V^- \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \cdots\cdots \\ H^- \left[H^{-*} H^-\right]^{-1} H^{-*} V^- \end{bmatrix} = \begin{bmatrix} 0 \\ \cdots\cdots \\ P_{H^-} V^- \end{bmatrix}_{(\infty+1)\times 1}.$$

$$P_{\upsilon}V=\begin{bmatrix}1\\ \cdots\\ 0\end{bmatrix}\left\{\begin{bmatrix}1 & \vdots & 0\end{bmatrix}\begin{bmatrix}1\\ \cdots\\ 0\end{bmatrix}\right\}^{-1}\begin{bmatrix}1 & \vdots & 0\end{bmatrix}\begin{bmatrix}v\\ \cdots\\ V^{-}\end{bmatrix}=\begin{bmatrix}v\\ \cdots\\ 0\end{bmatrix}_{(\infty+1)\times 1}.$$

$$\Rightarrow P_S^{\perp}V=V-P_SV=\begin{bmatrix}v\\ \cdots\\ V^{-}\end{bmatrix}-\begin{bmatrix}0\\ \cdots\cdots\\ P_{H^-}V^{-}\end{bmatrix}-\begin{bmatrix}v\\ \cdots\\ 0\end{bmatrix}=\begin{bmatrix}0\\ \cdots\cdots\\ P_{H^-}^{\perp}V^{-}\end{bmatrix}_{(\infty+1)\times 1}.$$

➢ Define: $E_n^{\upsilon}(k)\equiv P_n^{\perp}(k)\upsilon$, $P_n^{\upsilon}(k)\equiv E_n^{\upsilon}(k)\left[E_n^{\upsilon}(k)^{*}E_n^{\upsilon}(k)\right]^{-1}E_n^{\upsilon}(k)^{*}$

the latter is the orthogonal projection operator onto $E_n^{\upsilon}(k)$.

1. Time update equation; $E_n^f(k)=P_n^{\perp}(k-1)z(k)=P_n^{\upsilon}(k-1)z(k)$

$$+\begin{bmatrix}0\\ \cdots\cdots\cdots\cdots\cdots\cdots\\ P_n^{\perp}(k-2)z(k-1)\end{bmatrix}=\begin{bmatrix}0\\ \cdots\cdots\cdots\\ E_n^f(k-1)\end{bmatrix}+E_n^{\upsilon}(k-1)\begin{bmatrix}E_n^{\upsilon}(k-1)^{*}\\ E_n^{\upsilon}(k-1)\end{bmatrix}^{-1}\begin{bmatrix}E_n^{\upsilon}(k-1)^{*}\\ z(k)\end{bmatrix}.$$

But $E_n^{\upsilon}(k-1)^{*}z(k)=\left[P_n^{\perp}(k-1)\upsilon\right]^{*}z(k)=\upsilon^{*}\left[P_n^{\perp}(k-1)z(k)\right]$

$$=\upsilon^{*}E_n^f(k)=e_n^f(k)\Rightarrow E_n^f(k)=\begin{bmatrix}0\\ \cdots\cdots\cdots\\ E_n^f(k-1)\end{bmatrix}+E_n^{\upsilon}(k-1)\begin{bmatrix}E_n^{\upsilon}(k-1)^{*}\\ E_n^{\upsilon}(k-1)\end{bmatrix}^{-1}e_n^f(k)$$

$$\equiv\begin{bmatrix}0\\ \cdots\cdots\cdots\\ E_n^f(k-1)\end{bmatrix}+E_n^{\upsilon}(k-1)\sigma_n^{\#}(k-1)e_n^f(k).$$ In the same way,

$$E_n^b(k)\equiv\begin{bmatrix}0\\ \cdots\cdots\cdots\\ E_n^b(k-1)\end{bmatrix}+E_n^{\upsilon}(k)\sigma_n^{\#}(k)e_n^b(k).$$

2. Order update of $\sigma_n(k)$; we know that $\sigma_n(k)=E_n^{\upsilon}(k)^{*}E_n^{\upsilon}(k)$ and

$$E_{n+1}^{\upsilon}(k)\equiv P_{n+1}^{\perp}(k)\upsilon=\upsilon-P_{n+1}(k)\upsilon=\upsilon-\left[P_n(k)+P_n^b(k)\right]\upsilon=P_n^{\perp}(k)\upsilon$$

$$-E_n^b(k)\left[E_n^b(k)^{*}E_n^b(k)\right]^{-1}E_n^b(k)^{*}\upsilon=P_n^{\perp}(k)\upsilon-E_n^b(k)R_n^b(k)^{-1}e_n^b(k)$$

$$=E_n^{\upsilon}(k)-E_n^b(k)R_n^b(k)^{\#}e_n^b(k).\ \therefore\Rightarrow\sigma_{n+1}(k)=E_{n+1}^{\upsilon}(k)^{*}E_{n+1}^{\upsilon}(k)$$

$= E_n^{\upsilon}(k)^* E_n^{\upsilon}(k) - E_n^{\upsilon}(k)^* E_n^b(k) R_n^b(k)^{\#} e_n^b(k)$

$-e_n^b(k)^* R_n^b(k)^{\#} E_n^b(k)^* E_n^{\upsilon}(k) + e_n^b(k)^* R_n^b(k)^{\#} R_n^b(k) R_n^b(k)^{\#} e_n^b(k)$

$= E_n^{\upsilon}(k)^* E_n^{\upsilon}(k) - e_n^b(k)^* R_n^b(k)^{\#} e_n^b(k) = \sigma_n(k) - e_n^b(k)^* R_n^b(k)^{\#} e_n^b(k)$.

Another approach: by $\sigma_{n+1}(k) = \sigma_n(k-1) - e_n^f(k)^* R_n^f(k)^{\#} e_n^f(k)$.

➢ Time update equations for $R_n^f(k)$, $R_n^b(k)$ and $K_n(k)$;

$$R_n^f(k) = E_n^f(k)^* E_n^f(k) = \begin{bmatrix} 0 & \vdots & E_n^f(k-1)^* \end{bmatrix} \begin{bmatrix} 0 \\ \cdots\cdots\cdots \\ E_n^f(k-1) \end{bmatrix}$$

$+e_n^f(k)^* \sigma_n^{\#}(k-1)\sigma_n(k-1)\sigma_n^{\#}(k-1)e_n^f(k) = R_n^f(k-1)$

$+e_n^f(k)^* \sigma_n^{\#}(k-1)e_n^f(k)$ or when LS with Forgetting factor "λ";

$= \lambda R_n^f(k-1) + e_n^f(k)^* \sigma_n^{\#}(k-1)e_n^f(k)$. In the same manner,

$R_n^b(k) = \lambda R_n^b(k-1) + e_n^b(k)^* \sigma_n^{\#}(k)e_n^b(k)$ and

$K_n(k) = \lambda K_n(k-1) + e_n^f(k)^* \sigma_n^{\#}(k-1)e_n^b(k-1)$ with initial condition

$R_n^f(0) = R_n^b(-1) = K_n(0) = 0$.

● Lattice Algorithm (for Estimation Errors):

➢ Program:

Time Initial: $R_n^f(0)$, $R_n^b(-1)$, $K_n(0)$;

For $k = 1:\cdots$

Order initial: $\sigma_0(k-1)$, $e_0^b(k-1)$, $e_0^f(k)$

For $n = 0:\cdots$

$R_n^f(k) = \lambda R_n^f(k-1) + e_n^f(k)^* \sigma_n^{\#}(k-1)e_n^f(k)$,

$R_n^b(k-1) = \lambda R_n^b(k-2) + e_n^b(k-1)^* \sigma_n^{\#}(k-1)e_n^b(k-1)$,

$K_n(k) = \lambda K_n(k-1) + e_n^f(k)^* \sigma_n^{\#}(k-1)e_n^b(k-1)$,

$\sigma_{n+1}(k-1) = \sigma_n(k-1) - e_n^b(k-1)^* R_n^b(k-1)^{\#} e_n^b(k-1)$,

$$e_{n+1}^{b}(k)=e_{n}^{b}(k-1)-e_{n}^{f}(k)R_{n}^{f}(k)^{\#}K_{n}(k),$$

$$e_{n+1}^{f}(k)=e_{n}^{f}(k)-e_{n}^{b}(k-1)R_{n}^{b}(k-1)^{\#}K_{n}^{*}(k),$$

Next n

Next k

1. Time Initial conditions: we assume $z(k)=0\ \forall k\le 0$; (prewindow)

$\Rightarrow z(0)=0,\ z(-1-n)=0,\ H_{n}(-1)=\varnothing$

$\Rightarrow E_{n}^{f}(0)=z(0)-P_{n}(-1)z(0)=0,$

$E_{n}^{b}(-1)=z(-1-n)-P_{n}(-1)z(-1-n)=0,\ R_{n}^{f}(0)=R_{n}^{b}(-1)=K_{n}(0)=0.$

2. Order Initial conditions: $H_{0}(k-1)=\varnothing\Rightarrow$

$E_{0}^{b}(k-1)=z(k-1-n)-P_{0}(k-1)z(k-1-n)=z(k-1-n),$

$E_{0}^{f}(k-1)=z(k)-P_{0}(k-1)z(k)=z(k),\ E_{0}^{\upsilon}(k-1)=\upsilon-P_{0}(k-1)\upsilon=\upsilon.$

$\Rightarrow e_{0}^{b}(k-1)=z(k-1-n),\ e_{0}^{f}(k)=z(k),\ \sigma_{0}(k-1)=\upsilon^{*}\upsilon=1.$

3. What do we get:

$e_{n}^{f}(k),\ e_{n}^{b}(k)$: A-posteriori estimation error for both forward and backward prediction model, order $(0\sim n_{\max})$.

$R_{n}^{f}(k),\ R_{n}^{b}(k-1)$: A-posteriori estimation error correlation (or error energy; $E^{*}E$).

4. What don't we get:

$A_{n}(k),\ B_{n}(k)$: Model parameters.

➢ Update for $A_{n}(k)$ and $B_{n}(k)$:

If we let $A_{n}(k)=\left[a_{n,1}(k)\ \cdots\ a_{n,n}(k)\right]^{T}$, $E_{n}^{f}(k)=z(k)-H_{n}(k-1)A_{n}(k)$

$=-\left[z(k)\ \ z(k-1)\ \cdots\ z(k-n)\right]\left[-1\ \ a_{n,1}(k)\ \cdots\ a_{n,n}(k)\right]^{T}$

$= -\sum_{j=0}^{n} z(k-j)a_{n,j}(k)$, where we let $a_{n,0}(k) = -1$. In the same manner,

$B_n(k) = \left[b_{n,0}(k) \quad \cdots \quad b_{n,n-1}(k) \right]^T$ then $E_n^b(k) = z(k-n) - H_n(k)B_n(k)$

$= -\left[z(k) \quad \cdots \quad z(k+1-n) \quad z(k-n) \right] \left[b_{n,0}(k) \quad \cdots \quad b_{n,n-1}(k) \quad -1 \right]^T$

$= -\sum_{j=0}^{n} z(k-j)b_{n,j}(k)$, where $b_{n,n}(k) = -1$.

$\because E_{n+1}^f(k) = E_n^f(k) - E_n^b(k-1)R_n^b(k-1)^{\#} K_n^*(k) \Rightarrow \sum_{j=0}^{n+1} z(k-j)a_{n,j}(k)$

$= \sum_{j=0}^{n} z(k-j)a_{n,j}(k) - \sum_{j=0}^{n} z(k-1-j)b_{n,j}(k-1)R_n^b(k-1)^{\#} K_n^*(k)$, $i = 1+j$

$= \sum_{j=0}^{n} z(k-j)a_{n,j}(k) - \sum_{j=1}^{n+1} z(k-j)b_{n,j-1}(k-1)R_n^b(k-1)^{\#} K_n^*(k) \Rightarrow$ we can

set that $for\ j = 0,\ a_{1+n,0}(k) = -1 = a_{n,0}(k)$, $for\ j = 1 \sim n,$

$a_{1+n,j}(k) = a_{n,j}(k) - b_{n,j-1}(k-1)R_n^b(k-1)^{\#} K_n^*(k)$, $for\ j = 1+n,$

$a_{1+n,n+1}(k) = -b_{n,n}(k-1)R_n^b(k-1)^{\#} K_n^*(k) = R_n^b(k-1)^{\#} K_n^*(k)$. Or we can

write as $A_{n+1}(k) = \begin{bmatrix} A_n(k) \\ 0 \end{bmatrix} - \begin{bmatrix} B_n(k-1) \\ -1 \end{bmatrix} R_n^b(k-1)^{\#} K_n^*(k)$. In similarly,

$\because E_{n+1}^b(k) = E_n^b(k-1) - E_n^f(k)R_n^f(k)^{-1} K_n(k)$. After setting, we get

$for\ j = 0,\ b_{1+n,0}(k) = R_n^f(k-1)^{\#} K_n(k)$, $for\ j = 1 \sim n,$

$b_{1+n,j}(k) = b_{n,j}(k-1) - a_{n,j}(k)R_n^f(k)^{\#} K_n(k)$, $for\ j = 1+n,$

$b_{1+n,n+1}(k) = -b_{n,n}(k-1) = -1$, or

$B_{n+1}(k) = \begin{bmatrix} 0 \\ B_n(k) \end{bmatrix} - \begin{bmatrix} -1 \\ A_n(k-1) \end{bmatrix} R_n^f(k)^{\#} K_n(k)$. Thus far, the Lattice al-

gorithm with parameters must add the updating equations of $A_{n+1}(k)$

and $B_{n+1}(k)$ after $R_n^f(k)$, $R_n^b(k-1)$ and $K_n(k)$ been updated.

- Advantages of adaptive Lattice Filter:

1. Least computation if we needn't parameters.

2. MGS-like algorithm can provide numerical stability.

- Remain problems:

1. Prewindow assumption introduces some estimation error when initial data is non-zero. We must use sliding window / un-windowed Lattice filter now.
2. $R_n^f(k)^{\#}$ and $R_n^b(k-1)^{\#}$ will cause inversing non-scalar matrix when the problem is multichannel.

a. Using matrix inverse lemma to update $R_n^f(k)^{\#}$ and $R_n^b(k-1)^{\#}$ directly.

b. Using multichannel Lattice filter.

Please refer the papers about sliding window, un-windowed and multi-channel.

- Homework 10:

1. Please write a Lattice filter ID program for single channel which order can be adjusted.
2. Please use above program to identify a mixing wave (including five sine waves) and compare the result with it from RLS algorithm.

6 System Realization Theory

6.1 Introduction:

- In basically, there are three kinds methods used to execute system identification:

1. Frequency domain $\xrightarrow{ID}$ Transfer function $\xrightarrow{IFFT}$ *Markov* parameters.

2. Time domain $\xrightarrow{LSE\ \ ID}$ *Markov* parameters.

3. *Markov* parameters $\rightarrow$ *Hankel* matrix $\xrightarrow{ERA}$ State Space mode (Minimum realization).

- Controllable mode can be excited and Observable mode can be sensed.

- Review of matrix operator (vector, matrix, singular value decomposition):

- vector: a list of quantities e.g. forces (input), stress (output), strain (state variable).

- matrix: it can be seen as a collection of vectors or an operator on vector.

1. Collection of vectors; $\begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}$ $\xrightarrow{Decomposition}$ Transform of basis; $\begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix} = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix} T$ or $\begin{bmatrix} e_1 & \cdots & e_n \end{bmatrix} T^{-1} = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}$.

   Physical meaning: change the representation of a physical quantity from ordinary coordinate system to the new coordinate system. In the other word, the physical quantity is the same but its representation is changed.

2. Operator: $y_{m\times 1} = A_{m\times n} x_{n\times 1}$ is the relationship between vectors x and y, such as $\underset{x}{Stress} \rightarrow \underset{y}{Strain}$; 3D strain can be caused by 2D stress, $System\ input \rightarrow System\ output$; Input number can be different with output number.

   Geometric meaning: $y = Ax = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T$ it can be regarded as that y is the linear combination of a_i and x_i is the

corresponding coefficient.

- singular value decomposition; (SVD):

  If matrix $A_{m\times n}$, $m \geq n$ and $rank\{A\} = r$ then $\Rightarrow \exists\ U_{m\times m}$, $V_{n\times n}$ ∋ $A = U\Sigma V^T$ or $U^T AV = \Sigma$, where $\Sigma = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}$, $U^T U = I_m$, $V^T V = I_n$, and $S = diag\left[\sigma_1 \cdots \sigma_r\right]$.

1. Geometric meaning: $y = Ax = \left[a_1 \quad \cdots \quad a_n\right]\left[x_1 \quad \cdots \quad x_n\right]^T$; A is an operator. The dimension of stress domain is n. The dimension of strain domain is m. The dimension of range is the linear independent number of $\{a_1 \quad \cdots \quad a_n\}$; $rank\{A\} = r$. It means that there is an orthogonal normal basis in strain domain and the r of these base vectors can be mapped from the orthogonal normal base vectors in stress domain. V is the orthogonal normal basis of stress domain. U is the orthogonal normal basis of strain domain and the preceding r of them are produced by the column vectors of A through orthogonal decomposition. $\{\sigma_1 \quad \cdots \quad \sigma_r\}$ denote the gains between the orthogonal bases of stress and strain domains through the mapping operator A.

2. Verify: if v_i and u_i are respectively the i^{th} column vectors of V and U then

$$\begin{aligned} x &= v_i \Rightarrow \\ y &= Ax \\ &= Av_i \\ &= U\Sigma V^T v_i \end{aligned} = U\Sigma \begin{bmatrix} v_1^T \\ \vdots \\ v_i^T \\ \vdots \\ v_n^T \end{bmatrix} v_i = U\Sigma \begin{bmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} = U \begin{bmatrix} \sigma_1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & 0 \\ 0 & \cdots & \sigma_r & \\ & 0 & & 0 \end{bmatrix} \begin{bmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

$$= U\left[\cdots \quad 0 \quad \sigma_i \quad 0 \quad \cdots\right]^T = \left[u_1 \quad \cdots \quad u_i \quad \cdots \quad u_m\right]\left[\cdots \quad 0 \quad \sigma_i \quad 0 \quad \cdots\right]^T$$

$= \sigma_i u_i,\ \forall i = 1 \sim r$.

3. Example: $Stress \rightarrow Strain;\ \sigma = K\varepsilon$, SVD is equal to find the principle stress and principle strain. If we can find the principle stress and strain for bases then *Young's* modular factor will be diagonal.
4. Summary: for any mapping operator, we can find one pair of orthonomal bases for stress and strain domain such that any vector along with the base vector is only scale up / down and maintains being linear independent to each other under this operation.

- Review linear system:

Def 1: A state $x(t)$ is controllable $\Rightarrow \exists\ \Delta t < \infty$ ∋ $x(t + \Delta t) = 0$ and $u(t) \approx u(t + \Delta t)$.

Def 2: A state $x(t)$ is reachable $\Rightarrow$ given $x(0) = 0,\ \exists\ \Delta t < \infty,\ u(\tau),\ \tau = 0 \sim \Delta t,$ ∋ $x(\Delta t) = x$.

Def 3: Completely (controllable and reachable) $\Rightarrow$ every state of system are controllable and reachable.

Thm 1: Linear system is controllable and reachable $\Leftrightarrow$ Linear system is complete.

Thm 2: Linear system is controllable $\Leftrightarrow$ Linear system is reachable.

- Response of LTI system: $x(p) = A^p x(0) + \begin{bmatrix} B & A & A^2B & \cdots & A^{P-1}B \end{bmatrix}$ $\begin{bmatrix} u(p-1) & u(p-2) & u(p-3) & \cdots & u(0) \end{bmatrix}^T$. If $x(0) = 0$ then $x(p) = \begin{bmatrix} B & A & A^2B & \cdots & A^{P-1}B \end{bmatrix}$ $\begin{bmatrix} u(p-1) & u(p-2) & u(p-3) & \cdots & u(0) \end{bmatrix}^T$ $\Rightarrow Q_p \bullet U$, where Q_p is a mapping operator from input history to state. From $x(p) = Q_p \bullet U$, $x(p)$ is the linear combination of the column

vectors of matrix Q_p. So, the degree of freedom of $x(p)$ is the number of linear independent column vectors of Q_p.

- Theorem 6.1: $x(k+1) = Ax(k) + Bu(k)$ is controllable $\Leftrightarrow$ $Q_p = \begin{bmatrix} B & A & A^2B & \cdots & A^{P-1}B \end{bmatrix}$ has *rank n*.

[pf]: from SVD, $Q_p = R\begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix} S^T \equiv \begin{bmatrix} R_k & R_0 \end{bmatrix}\begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} S_k^T \\ S_0^T \end{bmatrix}$.

$\forall x(0) = 0$, $x(p) = \begin{bmatrix} B & A & \cdots & A^{P-1}B \end{bmatrix}\begin{bmatrix} u(p-1) & u(p-2) & \cdots & u(0) \end{bmatrix}^T$.

If R is non-singular, $\Rightarrow R^T x(p) = R^T R\begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix} S^T u_p$, where

$u_p \equiv \begin{bmatrix} u(p-1) & u(p-2) & u(p-3) & \cdots & u(0) \end{bmatrix}^T$

$\Rightarrow \begin{bmatrix} R_k^T x(p) \\ R_0^T x(p) \end{bmatrix} = \begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} S_k^T \\ S_0^T \end{bmatrix} u_p = \begin{bmatrix} \Sigma_k S_k^T u_p \\ 0 \end{bmatrix}$. $\forall\ u_p$, we have a constraint on $x(p)$ $\ni$ $R_0^T x(p) = 0$. So, the range of $x(p)$ is not of dimension *n*, unless, $k = n$ $\ni$ $R_k^T x(p) = R_n^T x(p) = \Sigma_n S_n^T u_p$.

Geometric meaning: Q_p maps input history to the state at time $k = p$, if and only if this mapping has *n* non-singular values. A system is controllable. $\equiv$ A mapping has *n* principle axes that span all state space.

- Notes of singular value and eigenvalues:

1. SVD: $\forall$ operation matrix $A_{m\times n}$ $\exists$ orthonomal basis pairs; v_i, u_i, $i = 1 \sim r$ and singular value σ_i $\ni$ $A = U\Sigma V^T \equiv \begin{bmatrix} u_1 & \cdots & u_r & \cdots & u_m \end{bmatrix}$
   - $\begin{bmatrix} \sigma_1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & 0 \\ 0 & \cdots & \sigma_r & \\ & 0 & & 0 \end{bmatrix}\begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \\ \vdots \\ v_n^T \end{bmatrix}$. Under an operation of *A*, v_i is mapped to u_i with scaling factor σ_i.

2. If operation matrix A is square; $A_{n\times n}$ then the spaces of domain and range may be the same. The "same" operator $A_{n\times n}$ has its singular vector pairs that are equal; $u_i = v_i$. Then, we call them eigenvector and respective singular value σ_i is eigenvalue.

   $\Rightarrow A = U\Sigma V^T = U\Sigma U^T$. Since U is orthonormal, $U^T = U^{-1}$

   $\Rightarrow A = U\Sigma V^{-1} \Rightarrow AU = U\Sigma \Rightarrow A\begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}$

   • $\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n \end{bmatrix} \Rightarrow Au_i = u_i\sigma_i$. The result is the definition of eigen value and eigenvector.

➢ Eigenvalues decomposition:

For a square matrix operator, its eigen normal vectors u_i are mapped to its original direction with scaling factor σ_i. $\Rightarrow$ Eigenvector isn't always existing (for real vector space). Note: Square matrix always have *Jordan* form transformation but not always have eigenvalue transformation.

➢ Theorem 6.2:

For $x(k+1) = Ax(k) + bu(k)$. If A has distinct eigenvalues (i.e. has Eigen decomposition) and single input then $b_m = \Psi^{-1}b$ has no zero elements $\Leftrightarrow$ Controllable.

$\because A\Psi = \Psi\Lambda,\ \Lambda = \begin{bmatrix} \lambda_1^p & & 0 \\ & \ddots & \\ 0 & & \lambda_n^p \end{bmatrix}$ are distinct. Ψ spans whole space $\square^n$

or $A = \Psi\Lambda\Psi^{-1} = \Psi\Lambda\Psi^T$. b: a mapping from 1 dimensional space $u(k)$ to state space $x(k+1)$. $b_m = \Psi^{-1}b = \Psi^T b$: a mapping from 1 dimen-

sional space $u(k)$ to state space $\bar{x}(k+1)=\Psi^{-1}x(k+1)$; represented in the new basis. Non-zero elements in b_m: every base vector of state space is excited.

➢ Observability in Discrete-Time domain:

1. Definitions:

Def 1: $x(p)$ is observable if given $u(k)$ and $y(k)$, $\forall 0\le k\le p$ then the $x(p)$ can be state completely determine.

Def 2: If all state in $\Re^n$ are observable $\Rightarrow$ completely observable.

2. Theorems:

Thm 1: If $x(0)$ can be estimated for given $u(k)$ and $y(k)$, $\forall 0\le k\le p$ then $x(p)$ is observable.

Thm 2: For an observable linear system $\Leftrightarrow$ completely observable.

Thm 3: $\begin{aligned} x_{n\times 1}(k+1) &= Ax_{n\times 1}(k)+bu(k) \\ y(k) &= Cx_{n\times 1}(k)+Du(k) \end{aligned}$ is observable $\Leftrightarrow \mathcal{P}_p \equiv \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-1} \end{bmatrix}$ has *rank n*.

[Pf]: If we have given $u(k)=0,\ \forall 0\le k\le p$,

$$\begin{aligned} y(0) &= Cx(0) \\ y(1) &= Cx(1)=CAx(0) \\ &\vdots \\ y(p-1) &= Cx(p-1)=CA^{p-1}x(0) \end{aligned} \Rightarrow \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(p-1) \end{bmatrix} = \mathcal{P}_p x(0)$$; Output history = Mapping from initial state.

[p.s.]: $\forall$ mapping operator $\mathcal{P}_p$, it exists a SVD $\ni$ $\mathcal{P}_p=U\Sigma V^T$, where

$\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \boldsymbol{0} \\ 0 & \cdots & \sigma_r & \\ & \boldsymbol{0} & & \boldsymbol{0} \end{bmatrix}$, $rank\{\mathcal{P}_p\} = r$ and U, V are orthonomal basis vectors. From the geometric meaning of SVD, there exist "r" pairs of basis vectors in "state" space and "output history" space. They are one to one mapping. $\Rightarrow$ Only r of state space basis vectors are one to one mapped to output history space. They can be found by examining the output history. If and only if $r = n$ then all state space basis vectors have a mutual independent basis vectors in output history vectors space.

Thm 4: Given $\begin{aligned} x_{n\times 1}(k+1) &= Ax_{n\times 1}(k) + bu(k) \\ y(k) &= Cx_{n\times 1}(k) + Du(k) \end{aligned}$; single output and A has distinct eigenvalues, then it is observable $\Leftrightarrow$ $C_m = C\Psi$ has no zero elements. In addition, for $A\Psi = \Psi\Lambda$, A has distinct eigenvalues $\Rightarrow$ dynamic of system keeps the state space to be whole space. C is the mapping from state space to the single measurement that is a combination of state vector. For $C_m = C\Psi$, C_m is new combination coefficient after changing the state space basis to the eigenvectors. For non-zero in C_m, every state represented in eigenvector basis has influence on measurement $y(k)$.

6.2 Basic concepts of Realization:

- State space model $\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$; $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$.
- *Markov* parameter model

$Y_0 = D,\ Y_1 = CB,\ Y_2 = CAB,\ \cdots,\ Y_k = CA^{k-1}B$.

- Realization: given $Y_0,\ Y_1,\ Y_2,\ \cdots,\ Y_k$ to find $\begin{bmatrix} A & B & C & D \end{bmatrix}$ that satisfies $Y_0 = D,\ Y_1 = CB,\ Y_2 = CAB,\ \cdots,\ Y_k = CA^{k-1}B$. It isn't unique.
- Minimum Realization: The realization with the smallest number of state-space dimensions among all realization. It isn't unique with the same eigenvalues therefore eigenvalues are modal parameters of system.
- $Y_0 = D \Rightarrow$ Independent the state space basis.
- Given state space $\begin{bmatrix} A & B & C \end{bmatrix}$ $\exists$ eigenvalues and eigenvectors are $\Lambda = diag\begin{bmatrix} \lambda_1, \cdots, \lambda_n \end{bmatrix}$ and $\Psi = \begin{bmatrix} \Psi_1 & \cdots & \Psi_n \end{bmatrix}$ $\ni$ $A = \Psi\Lambda\Psi^T;\ A\Psi = \Psi\Lambda$. Use Ψ to transform the state such that state space model become $\begin{bmatrix} \Lambda & \Psi^{-1}B & C\Psi \end{bmatrix}$ and is unique for $|\lambda_1| \geq |\lambda_2| \geq \cdots |\lambda_n|$.
- $\begin{bmatrix} \Lambda & \Psi^{-1}B & C\Psi \end{bmatrix}$ is unique. Λ contains the information of modal damping ratio and undamped natural frequency, $\Psi^{-1}B$ contains initial modal amplitude (excitation effect on eigen states) and $C\Psi$ contains mode shapes (eigen state show up on output measurement). That is $u(k) \xrightarrow{\Psi^{-1}B} x^e(k) \begin{cases} \xrightarrow{\Lambda} x^e(k+1) \\ \xrightarrow{C\Psi} y(k) \end{cases}$, where $x^e(k)$ is the representation of state space on Eigen basis.
- $\Lambda_C = \dfrac{\ln \Lambda}{\Delta t}$ can be used to map discrete time eigen matrix back to continuous time domain. $\because A = e^{\Lambda_C \Delta t} \Rightarrow \Lambda = e^{\Lambda_C \Delta t} \Rightarrow \Lambda_C = \dfrac{\ln \Lambda}{\Delta t}$. You must be careful of the frequency domain aliasing problem when mapping back.

- *Hankel* matrix of order $\alpha \times \beta$ (i.e. $\alpha m \times \beta r$); $H(k-1)_{\alpha\times\beta} = \begin{bmatrix} Y_k & Y_{k+1} & \cdots & Y_{k+\beta-1} \\ Y_{k+1} & Y_{k+2} & & \vdots \\ \vdots & & \ddots & \vdots \\ Y_{k+\alpha-1} & \cdots & \cdots & Y_{k+\alpha+\beta-2} \end{bmatrix} = P_\alpha A^{k-1} Q_\beta$, where $P_\alpha \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\alpha-1} \end{bmatrix}$, and $Q_\beta \triangleq \begin{bmatrix} B & AB & \cdots & A^{\beta-1}B \end{bmatrix}$.

6.3 The Eigensystem Realization Algorithm (ERA):

- *Kalman* and *Ho* use generalized *Hankel* matrix to construct state space model from noise-free data.
- ERA modifies and extends parameters identification to noise data.
- ERA uses matrix modified from generalized *Hankel* matrix; keeps Y_k and eliminate some rows and columns in $H(k-1)$; resuffers the rows and columns of $H(k-1)$.

➢ *Hankel* matrix; $H(k) = P_\alpha A^k Q_\beta \xrightarrow{modify}$ ERA broke matrix;

$\hat{H}(k) = \hat{P}_\alpha A^k \hat{Q}_\beta$, where $\hat{P}_\alpha \triangleq \begin{bmatrix} C^T & \left(C_1 A^{s_1}\right)^T & \cdots & \left(C_\alpha A^{s_\alpha}\right)^T \end{bmatrix}^T$ and $\hat{Q}_\beta \triangleq \begin{bmatrix} B & A^{t_1}B_1 & \cdots & A^{t_\beta}B_\beta \end{bmatrix}$, where $C^T = \begin{bmatrix} c_1^T & c_2^T & \cdots & c_m^T \end{bmatrix}^T$ and C_i is constructed from some of $\{c_1, \cdots, c_m\}$, $B = \begin{bmatrix} b_1 & b_2 & \cdots & b_r \end{bmatrix}$ and B_i is constructed from some of $\{b_1, \cdots, b_r\}$. In addition, $s_1, \cdots, s_\alpha$ and $t_1, \cdots, t_\beta$ are arbitrary integers.

➢ It means that C on the top of P_α remains and the other rows are chosen arbitrarily from $\begin{bmatrix} (CA)^T & \left(CA^2\right)^T & \cdots & \left(CA^{\alpha-1}\right)^T \end{bmatrix}^T$ and are re-arranged according to the magnitudes of signals. In similarly, B on the leftist of Q_β remains and the other columns are chosen arbitrarily

from $\left[AB \quad A^2B \quad \cdots \quad A^{\beta-1}B\right]$ and are rearranged according to the magnitudes of signals.

- Implement: The left-up corner of *Hankel* matrix $H(k)$ keeps Y_{k+1} and the other elements are chosen and rearranged according to the bigger *SN* ratios.
- Meanings: We choose the strongest columns of $\left[AB \quad \cdots \quad A^{\beta-1}B\right]$ and rearrange them. It means that we take the actuators having the strongest affections on system response and prioritize them. In similarly, the strongest rows of $\left[(CA)^T \quad \left(CA^2\right)^T \quad \cdots \quad \left(CA^{\alpha-1}\right)^T\right]^T$ means that the measurements are the strongest affected by system response and prioritize them.

- In the above $\hat{H}(k) = \hat{P}_\alpha A^k \hat{Q}_\beta$, $H(k)$ has been known and we want to know $\left[A \quad B \quad C\right]$. In addition, the top row of $\hat{P}_\alpha$ is C and the leftist column $\hat{Q}_\beta$ is B. So, we will know B and C if $\hat{P}_\alpha$ and $\hat{Q}_\beta$ are known.
- $\because \hat{H}(0) = \hat{P}_\alpha \hat{Q}_\beta$, we can find $\hat{P}_\alpha$ and $\hat{Q}_\beta$ from $\hat{H}(0)$ and then derive B and C. Therefore, $\because \hat{H}(1) = \hat{P}_\alpha A \hat{Q}_\beta$, we can find A from $\hat{P}_\alpha$, $\hat{Q}_\beta$ and $\hat{H}(1)$.
- $\hat{Q}_\beta \equiv \left[B \quad A^{t_1}B_1 \quad \cdots \quad A^{t_\beta}B_\beta\right]$: mapping operator from reorganized input history to state. $\because x(k+1) = \hat{Q}_\beta \begin{bmatrix} u(k) \\ u_1(k-t_1) \\ \vdots \\ u_\beta(k-t_\beta) \end{bmatrix}$, where $u_i(k-t_i)$ is chosen from the part of the elements of $u(k-t_i)$ in order to correspond

to the columns of B_i.

- $\hat{P}_\alpha \triangleq \left[C^T \quad \left(C_1A^{s_1}\right)^T \quad \cdots \quad \left(C_\alpha A^{s_\alpha}\right)^T\right]^T$: mapping operator from state to reorganized output history. $\because \begin{bmatrix} y(k+1) \\ y_1(k+1+s_1) \\ \vdots \\ y_\alpha(k+1+s_\alpha) \end{bmatrix} = \begin{bmatrix} C \\ C_1A^{s_1} \\ \vdots \\ C_\alpha A^{s_\alpha} \end{bmatrix} x(k+1)$

$= \hat{P}_\alpha x(k+1)$, where $y_i(k+1+s_i)$ is chosen from the part of the elements of $y(k+1+s_i)$ in order to correspond to the rows of C_i.

- $\hat{H}(k) = \hat{P}_\alpha A^k \hat{Q}_\beta$: mapping operator from reorganized input history to output history with delay $1+\hat{k}$.

$$\because \begin{bmatrix} y(k+1+\hat{k}) \\ y_1(k+1+s_1+\hat{k}) \\ \vdots \\ y_\alpha(k+1+s_\alpha+\hat{k}) \end{bmatrix} = \begin{bmatrix} C \\ C_1A^{s_1} \\ \vdots \\ C_\alpha A^{s_\alpha} \end{bmatrix} \bullet \; x(k+1+\hat{k}) \qquad \hat{P}_\alpha A^{\hat{k}} \hat{Q}_\beta \bullet$$

$$= \hat{P}_\alpha A^{\hat{k}} x(k+1) = \begin{bmatrix} u(k) \\ u_1(k-t_1) \\ \vdots \\ u_\beta(k-t_\beta) \end{bmatrix}.$$

- If we take SVD for $\hat{H}(0) = \hat{P}_\alpha \hat{Q}_\beta \Rightarrow \hat{H}(0) = R \Sigma S^T$

$= \begin{bmatrix} R_n & R_0 \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} S_n^T \\ S_0^T \end{bmatrix}$, where $\Sigma_n = diag\left[\sigma_1, \cdots, \sigma_n\right]$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$,

and $R_n^T R_n = I = S_n^T S_n$. It will eliminate R_0, S_0^T term

$\Rightarrow \hat{H}(0) = R_n \Sigma_n S_n^T$.

- We can use $\hat{H}(0) = R_n \Sigma_n S_n^T$ to compute pseudo inverse $\hat{H}^{\#}(0) = S_n \Sigma_n^{-1} R_n^T$. Please refer to the definition of $\hat{H}(0)\hat{H}^{\#}(0)\hat{H}(0) = R_n \Sigma_n S_n^T S_n \Sigma_n^{-1} R_n^T R_n \Sigma_n S_n^T = R_n \Sigma_n S_n^T = \hat{H}(0)$.

- We may choose $\hat{P}_\alpha \hat{Q}_\beta$ according to the SVD of $\hat{H}(0)$. Because $\hat{H}(0) = \hat{P}_\alpha \hat{Q}_\beta = R_n \Sigma_n S_n^T$ where the columns of S_n and the rows of R_n

are all unit-vectors, they have not the effect for scaling. So, the scaling effect between input history and output history must be done by Σ_n or achieved by $\hat{P}_\alpha$ and $\hat{Q}_\beta$. If we take SVD for $\hat{P}_\alpha\hat{Q}_\beta = R_n\Sigma_\alpha U_s^T U_s \Sigma_\beta S_n^T$ $= R_n\Sigma_n S_n^T \Rightarrow \Sigma_\alpha\Sigma_\beta = \Sigma_n$ where Σ_α and Σ_β are called as $\hat{P}_\alpha$ and $\hat{Q}_\beta$ scaling, respectively. In addition, there are many divided methods of this.

- Balanced Realization: If we choose $\Sigma_\alpha = \Sigma_\beta = \sqrt{\Sigma_n}$ and use principle axes to state basis then $U_s = I \Rightarrow \hat{P}_\alpha = R_n\sqrt{\Sigma_n}$, $\hat{Q}_\beta = \sqrt{\Sigma_n}S_n^T$, $\hat{P}_\alpha^T\hat{P}_\alpha = \Sigma_n$; observability grammarian and $\hat{Q}_\beta\hat{Q}_\beta^T = \Sigma_n$ controllability grammarian. Using pinning vectors $E_m^T = \begin{bmatrix} I_m & 0_m & \cdots & 0_m \end{bmatrix}$, we can get $\hat{C}$ and $\hat{B}$ from $\hat{C} = E_m^T\hat{P}_\alpha = E_m^T R_n\sqrt{\Sigma_n}$ and $\hat{B} = \hat{Q}_\beta E_m = \sqrt{\Sigma_n}S_n^T E_m$.
- $\because \hat{H}(1) = \hat{P}_\alpha A\hat{Q}_\beta = R_n\sqrt{\Sigma_n}A\sqrt{\Sigma_n}S_n^T$, we can find A from $\hat{P}_\alpha$, $\hat{Q}_\beta$ and $\hat{H}(1)$. Because R_n, S_n^T and $\sqrt{\Sigma_n}$ are all non-singular, $\exists\ R_n^{-1} = R_n^T$, and $S_n^{T^{-1}} = S_n$ such that $\hat{A} = \sqrt{\Sigma_n}^{-1}R_n^T\hat{H}(1)S_n\sqrt{\Sigma_n}^{-1}$.
- Transform $\begin{bmatrix}\hat{A} & \hat{B} & \hat{C}\end{bmatrix}$ to modal coordinate:

Let $\hat{\Psi}$ satisfy $\hat{A}\hat{\Psi} = \hat{\Psi}\hat{\Lambda} \Rightarrow \hat{\Lambda} = \hat{\Psi}^{-1}\hat{A}\hat{\Psi}$, $\hat{B}_m = \hat{\Psi}^{-1}\hat{B}$, $\hat{C}_m = \hat{C}\hat{\Psi}$. So, $\Rightarrow \begin{bmatrix}\hat{\Lambda} & \hat{B}_m & \hat{C}_m\end{bmatrix}$.

- Due to measurement noise, nonlinearity and computer round-off, $rank\{\hat{H}(k)\}$ >the order of system.
- For ideal data; $\hat{H}(0) = R\Sigma S^T = \begin{bmatrix}R_n & R_0\end{bmatrix}\begin{bmatrix}\Sigma_n & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}S_n^T\\ S_0^T\end{bmatrix}$. But for real data; $\hat{H}(0) = R\Sigma S^T = \begin{bmatrix}R_n & R_e & R_0\end{bmatrix}\begin{bmatrix}\Sigma_n & 0 & 0\\ 0 & \Sigma_e & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}S_n^T\\ S_e^T\\ S_0^T\end{bmatrix}$, where Σ_e is

induced by measurement noise, nonlinearity and computer round-off.

- Singular value $\sigma_1,\cdots,\sigma_n,\sigma_{n+1},\cdots,\sigma_e$ represent the observability and controllability of the correspond mode, $\sigma_{n+1},\cdots,\sigma_e$ are small and negligible compare to $\sigma_1,\cdots,\sigma_n$. So, they are relatively uncontrollable and unobservable therefore better to be eliminate out from model.

6.4 Candidate methods for distinguishing true modes from noise modes:

- Model in modal coordinate $\Rightarrow \left[\hat{\Lambda} \quad \hat{B}_m \quad \hat{C}_m \quad D\right]$.
- $\left[D \quad \hat{C}_m\hat{B}_m \quad \hat{C}_m\hat{\Lambda}\hat{B}_m \quad \cdots \quad \hat{C}_m\hat{\Lambda}^{\ell-2}\hat{B}_m\right] \Rightarrow$ *Markov* parameters in modal coordinate. In modal coordinate, every state is decoupling. For mapping input to state, $\Rightarrow \hat{X}(k) = \left[\hat{B}_m \quad \hat{\Lambda}\hat{B}_m \quad \cdots \quad \hat{\Lambda}^{\ell-2}\hat{B}_m\right]U(k)$; ℓ is the number of *Markov* parameters, $= \begin{bmatrix} \hat{b}_1 & \hat{\lambda}_1\hat{b}_1 & \cdots & \hat{\lambda}_1^{\ell-2}\hat{b}_1 \\ \vdots & \vdots & \cdots & \vdots \\ \hat{b}_n & \hat{\lambda}_n\hat{b}_n & \cdots & \hat{\lambda}_n^{\ell-2}\hat{b}_n \end{bmatrix}U(k) \square \begin{bmatrix} \hat{q}_1 \\ \vdots \\ \hat{q}_n \end{bmatrix}U(k)$

where $\hat{q}_i$ maps input space to the i^{th} state variable. $\because Y(k) = \left[D \quad \hat{C}_m\begin{bmatrix} \hat{q}_1 \\ \vdots \\ \hat{q}_n \end{bmatrix}\right]U(k) = \left[D \quad \begin{bmatrix} \hat{c}_1 & \cdots & \hat{c}_n \end{bmatrix}\begin{bmatrix} \hat{q}_1 \\ \vdots \\ \hat{q}_n \end{bmatrix}\right]U(k) = \left[D \quad \sum_{i=1}^{n}\hat{c}_i\hat{q}_i\right]U(k)$.

So, we can say that $\hat{C}_m = \left[\hat{c}_1 \quad \cdots \quad \hat{c}_n\right]$ is used to map state space to output space where $\hat{c}_i$ maps the i^{th} state variable to output.

$\hat{H}(0) \xrightarrow[\text{broke matrix}]{\text{general ERA}} \left[\hat{A} \quad \hat{B} \quad \hat{C}\right] \xrightarrow[\Psi]{\text{eigen vectors}} \left[\hat{\Lambda} \quad \hat{B}_m \quad \hat{C}_m\right]$

$\rightarrow \left[\hat{B}_m \quad \hat{\Lambda}\hat{B}_m \quad \cdots \quad \hat{\Lambda}^{\ell-2}\hat{B}_m\right] = \begin{bmatrix} \hat{q}_1 \\ \vdots \\ \hat{q}_n \end{bmatrix} \Rightarrow \hat{q}_i$.

- There is another method to find $\hat{q}_i$. We can use singular value de-

composition to decompose *Hankel* matrix as follow; $\hat{H}(0) = R_n \Sigma_n S_n^T$ $= \left[R_n \sqrt{\Sigma_n} \Psi\right]\left[\Psi^{-1} \sqrt{\Sigma_n} S_n^T\right] \equiv \bar{P}_\alpha \bar{Q}_\beta$, where Ψ is an arbitrary nonsingular matrix to be determined by the coordinates chosen for the model. Since $\left[Y_0 \quad Y_1 \quad Y_2 \quad \cdots \quad Y_{\ell-1}\right]$ is directly obtained from pulse response samples and it equal $\left[D \quad \hat{C}_m \hat{B}_m \quad \hat{C}_m \hat{\Lambda} \hat{B}_m \quad \cdots \quad \hat{C}_m \hat{\Lambda}^{\ell-2} \hat{B}_m\right]$ under noise free,

$$\begin{bmatrix} Y_1 & Y_2 & \cdots & Y_{\ell-\alpha} \\ Y_2 & Y_3 & \cdots & Y_{\ell-\alpha+1} \\ \vdots & \vdots & \ddots & \vdots \\ Y_\alpha & Y_{\alpha+1} & \cdots & Y_{\ell-1} \end{bmatrix} = \hat{H}(0) \equiv \bar{P}_\alpha \bar{Q}_\beta = \begin{bmatrix} \hat{C}_m \hat{B}_m & \hat{C}_m \hat{\Lambda} \hat{B}_m & \cdots & \hat{C}_m \hat{\Lambda}^{\ell-\alpha-1} \hat{B}_m \\ \hat{C}_m \hat{\Lambda} \hat{B}_m & \hat{C}_m \hat{\Lambda}^2 \hat{B}_m & \cdots & \hat{C}_m \hat{\Lambda}^{\ell-\alpha} \hat{B}_m \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_m \hat{\Lambda}^{\alpha-1} \hat{B}_m & \hat{C}_m \hat{\Lambda}^{\alpha} \hat{B}_m & \cdots & \hat{C}_m \hat{\Lambda}^{\ell-2} \hat{B}_m \end{bmatrix}$$

$$= \begin{bmatrix} \hat{C}_m \\ \hat{C}_m \hat{\Lambda} \\ \vdots \\ \hat{C}_m \hat{\Lambda}^{\alpha-1} \end{bmatrix} \left[\hat{B}_m \quad \hat{\Lambda} \hat{B}_m \quad \cdots \quad \hat{\Lambda}^{\ell-\alpha-1} \hat{B}_m\right] \Rightarrow \bar{P}_\alpha = \begin{bmatrix} \hat{C}_m \\ \hat{C}_m \hat{\Lambda} \\ \vdots \\ \hat{C}_m \hat{\Lambda}^{\alpha-1} \end{bmatrix} \text{ and}$$

$\bar{Q}_\beta = \left[\hat{B}_m \quad \hat{\Lambda} \hat{B}_m \quad \cdots \quad \hat{\Lambda}^{\ell-\alpha-1} \hat{B}_m\right]$, where α is chosen such that $\alpha m \geq n$.

If noises exist then we define $\bar{Q}_\beta \equiv \left[\bar{B}_m \quad \bar{\Lambda} \bar{B}_m \quad \cdots \quad \bar{\Lambda}^{\ell-\alpha-1} \bar{B}_m\right]$

$$= \begin{bmatrix} \bar{b}_1 & \bar{\lambda}_1 \bar{b}_1 & \cdots & \bar{\lambda}_1^{\ell-\alpha-1} \bar{b}_1 \\ \vdots & \vdots & \cdots & \vdots \\ \bar{b}_n & \bar{\lambda}_n \bar{b}_n & \cdots & \bar{\lambda}_n^{\ell-\alpha-1} \bar{b}_n \end{bmatrix} \equiv \begin{bmatrix} \bar{q}_1 \\ \vdots \\ \bar{q}_n \end{bmatrix}.$$

$$\hat{H}(0) \xrightarrow{SVD} R_n \Sigma_n S_n^T \xrightarrow{\hat{\Psi};\ transform\ to\ modal\ coordinate} \bar{P}_\alpha \bar{Q}_\beta \equiv \left[R_n \sqrt{\Sigma_n} \Psi\right]$$

$$\bullet \left[\Psi^{-1} \sqrt{\Sigma_n} S_n^T\right] \rightarrow \left[\hat{B}_m \quad \hat{\Lambda} \hat{B}_m \quad \cdots \quad \hat{\Lambda}^{\ell-\alpha-1} \hat{B}_m\right] \approx \bar{Q}_\beta \equiv \begin{bmatrix} \bar{q}_1 \\ \vdots \\ \bar{q}_n \end{bmatrix} \Rightarrow \bar{q}_i.$$

- q_i is the mapping from input to modal coordinate state. Compare the modal amplitude from model and from data. $\hat{q}_i$ is constructed from model and $\bar{q}_i$ is constructed from data. If system is free of noise then $\hat{q}_i = \bar{q}_i$.

- MAC; Modal Amplitude Coherence:

It can be thought of as a dot product or a generalized cosine between the vectors of the measured response history $\bar{q}_i$ and the identified model's response history $\hat{q}_i$; $MAC_i \equiv \dfrac{\left|\bar{q}_i \hat{q}_i^*\right|}{\sqrt{\left|\bar{q}_i \bar{q}_i^*\right|\left|\hat{q}_i \hat{q}_i^*\right|}} = \cos\left(\bar{q}_i, \hat{q}_i\right)$ where $i = 1 \sim n$.

In practice, the computation time required to evaluate the *MAC* for each mode is quite small.

- MSV; Mode Singular Value:

It characterizes the contribution of each identified mode to the identified model pulse response history. Because the identification algorithm, such as ERA, attempts to identify a model to match the pulse response history, it's reasonable that a mode that has a large contribution to the system's pulse response data and is then well identified by the algorithm. Since each individual *Markov* parameters can be written as a combination of *n* components contributed from different modal coordinates;

$Y_k = \hat{C}_m \hat{\Lambda}^{k-1} \hat{B}_m = \sum_{i=1}^{n} \left(\hat{c}_i \hat{\lambda}_i^{k-1} \hat{b}_i\right)_{m \times r}$, we can quantified the contribution of each mode by taking its maximum singular value;

$MSV_i \equiv \sqrt{\left|\hat{c}_i\right|\left(1 + \left|\hat{\lambda}_i\right| + \left|\hat{\lambda}_i^2\right| + \cdots + \left|\hat{\lambda}_i^{\ell-2}\right|\right)\left|\hat{b}_i\right|} \approx \sqrt{\dfrac{\left|\hat{c}_i\right|\left|\hat{b}_i\right|}{1 - \left|\hat{\lambda}_i\right|}}$; where the approximation sign is valid only if $\left|\hat{\lambda}_i\right| < 1$ and ℓ is large enough.

- Computational steps of ERA:

1. Construct a block *Hankel* matrix $\hat{H}(0)$ by arranging the *Markov* parameters (pulse response samples) into blocks with given α, β,

s_i; $i = 1 \sim \alpha$, and t_j; $j = 1 \sim \beta$. $\hat{H}(0) = \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_\beta \\ Y_2 & Y_3 & & \vdots \\ \vdots & & \ddots & \vdots \\ Y_\alpha & \cdots & \cdots & Y_{\alpha+\beta-1} \end{bmatrix}$.

2. Decompose $\hat{H}(0)$ using singular value decomposition.
   $\hat{H}(0) = R \Sigma S^T$.

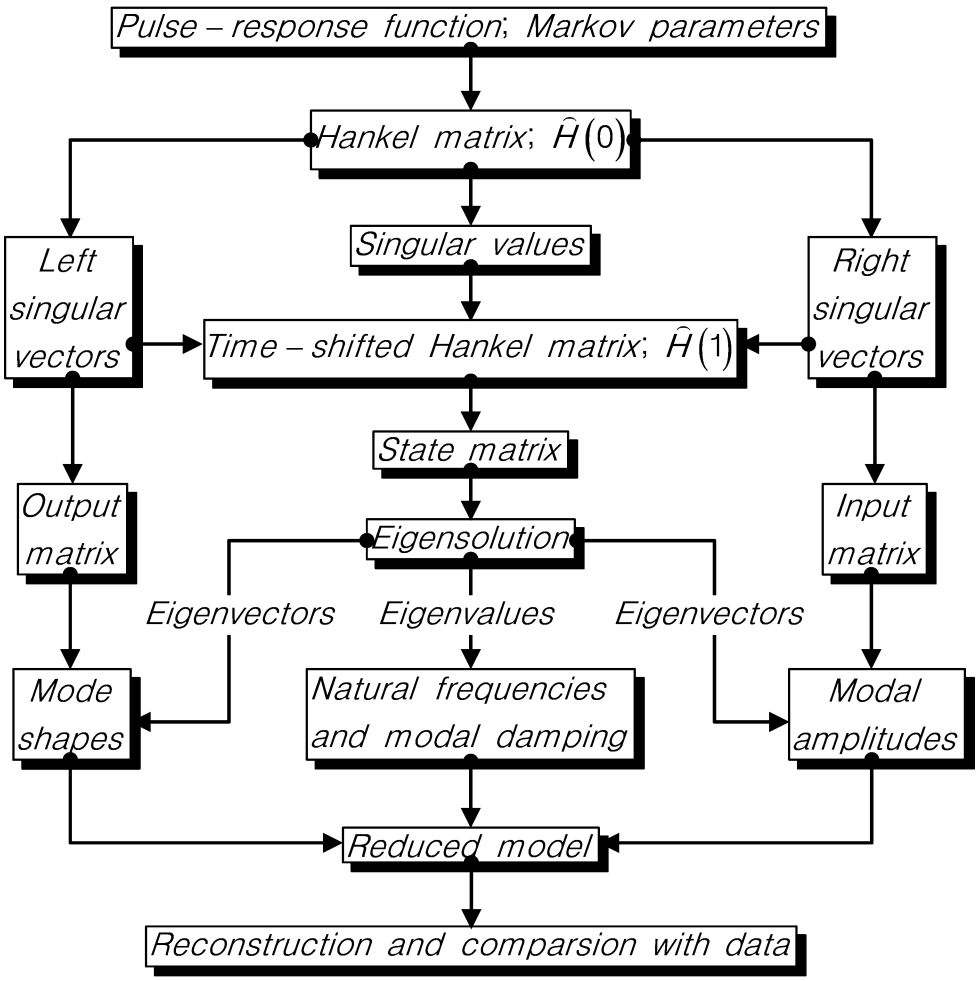
3. Determine the order of the system by examining the singular values of the *Hankel* matrix $\hat{H}(0)$. $\hat{H}(0) = R \Sigma S^T$.

4. Construct a minimum order realization $\begin{bmatrix} \hat{A} & \hat{B} & \hat{C} \end{bmatrix}$ using a shifted block *Hankel* matrix $\hat{H}(1)$. $\hat{A} = \sqrt{\Sigma_n}^{-1} R_n^T \hat{H}(1) S_n \sqrt{\Sigma_n}^{-1}$, $\hat{B} = \sqrt{\Sigma_n} S_n^T E_m$ and $\hat{C} = E_m^T R_n \sqrt{\Sigma_n}$.

5. Find the eigensolution of the realized state matrix and transform the realized model to modal coordinates to $y(k) = \hat{C}_m x_m(k) + \hat{D} u(k)$ $x_m(k+1) = \hat{\Lambda} x_m(k) + \hat{B}_m u(k)$ calculate the system damping and natural frequencies.

6. Calculate the modal amplitude coherence; MAC_i □ $\frac{\left|\bar{q}_i \hat{q}_i^*\right|}{\sqrt{\left|\bar{q}_i \bar{q}_i^*\right| \left|\hat{q}_i \hat{q}_i^*\right|}}$ and mode singular values; MSV_i □ $\sqrt{\left|\hat{c}_i\right| \left(1 + \left|\hat{\lambda}_i\right| + \left|\hat{\lambda}_i^2\right| + \cdots + \left|\hat{\lambda}_i^{\ell-2}\right|\right) \left|\hat{b}_i\right|}$ to quantify the system and noise modes.

7. Determine the reduced system model based on the accuracy indicators computed in step 6, reconstruct *Markov* parameters Y_i and compare with the measured *Markov* parameters.



Note that the optimum determination of α, β, s_i; $i = 1 \sim \alpha$, and t_j; $j = 1 \sim \beta$ in step 1 requires some engineering intuition. This determination is related to the choice of the measurement data to minimize the size of the *Hankel* matrix $\hat{H}(0)$ with the rank unchanged, as will be discussed in a later chapter.

6.5 The ERA with data correlation (ERA/DC):

- When noisy data with uncorrelated noise, the correlation of data isn't as noisy as data. Therefore, ERA/DC is used for data with uncorrelated noise.
- Correlation matrix of *Hankel* matrix: $R_{HH}(k) = H(k)H^T(0)$

$$= \begin{bmatrix} Y_{k+1} & Y_{k+2} & \cdots & Y_{k+\beta} \\ Y_{k+2} & Y_{k+3} & \cdots & Y_{k+\beta+1} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{k+\alpha} & Y_{k+\alpha+1} & \cdots & Y_{k+\alpha+\beta+1} \end{bmatrix} \begin{bmatrix} Y_1 & Y_2 & \cdots & Y_\beta \\ Y_2 & Y_3 & \cdots & Y_{\beta+1} \\ \vdots & \vdots & \ddots & \vdots \\ Y_\alpha & Y_{\alpha+1} & \cdots & Y_{\alpha+\beta+1} \end{bmatrix}^T = P_\alpha A^k Q_\beta Q_\beta^T P_\alpha^T$$

$$= \begin{bmatrix} \sum_{i=1}^{\beta} Y_{k+i} Y_i^T & \sum_{i=1}^{\beta} Y_{k+i} Y_{i+1}^T & \cdots & \sum_{i=1}^{\beta} Y_{k+i} Y_{i+\alpha-1}^T \\ \sum_{i=1}^{\beta} Y_{k+i+1} Y_i^T & \sum_{i=1}^{\beta} Y_{k+i+1} Y_{i+1}^T & \cdots & \sum_{i=1}^{\beta} Y_{k+i+1} Y_{i+\alpha-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{\beta} Y_{k+i+\alpha-1} Y_i^T & \sum_{i=1}^{\beta} Y_{k+i+\alpha-1} Y_{i+1}^T & \cdots & \sum_{i=1}^{\beta} Y_{k+i+\alpha-1} Y_{i+\alpha-1}^T \end{bmatrix}_{\alpha\times\alpha} \square P_\alpha A^k Q_c \text{. For}$$

$R_{HH}(0) = H(0)H^T(0)$, its all elements are cross-correlations except the left top one is auto-correlation.

- If we define block correlation *Hankel* matrix as $\mathcal{H}(k)$,

$$\begin{bmatrix} R_{HH}(k) & R_{HH}(k+\tau) & \cdots & R_{HH}(k+\zeta\tau) \\ R_{HH}(k+\tau) & R_{HH}(k+2\tau) & \cdots & R_{HH}[k+(\zeta+1)\tau] \\ \vdots & \vdots & \ddots & \vdots \\ R_{HH}(k+\xi\tau) & R_{HH}[k+(\xi+1)\tau] & \cdots & R_{HH}[k+(\zeta+\xi)\tau] \end{bmatrix} = \begin{bmatrix} P_\alpha \\ P_\alpha A \\ \vdots \\ P_\alpha A^{\xi\tau} \end{bmatrix} A^k \bullet$$

$$\begin{bmatrix} Q_c & A^\tau Q_c & \cdots & A^{\zeta\tau} Q_c \end{bmatrix} \square \mathcal{P}_\xi A^k \mathcal{Q}_\zeta .$$

- In similarity, $\because Y_{k+1} = CA^kB \Rightarrow H(k) = P_\alpha A^k Q_\beta$, $\therefore R_{HH}(k) = P_\alpha A^k Q_c$ $\Rightarrow \mathcal{H}(k) = \mathcal{P}_\xi A^k \mathcal{Q}_\zeta$. That is $P_\alpha \leftrightarrow C$, $Q_c \leftrightarrow B$, $R_{HH}(k) \leftrightarrow Y_{k+1}$, $\mathcal{P}_\xi \leftrightarrow P_\alpha$, and $\mathcal{Q}_\zeta \leftrightarrow Q_\beta$. We take SVD $H(0)$ for P_α and Q_c to find B and C. In similarity, we take SVD $\mathcal{H}(0)$ for $\mathcal{P}_\xi$ and $\mathcal{Q}_\zeta$ to find $\hat{P}_\alpha$ and $\hat{Q}_c$ then $\hat{B}$ and $\hat{C}$ such that $\mathcal{H}(1) \to \hat{A}$ is finally determined.

- $\mathcal{H}(0) = R\Sigma S^T = R_n\sqrt{\Sigma_n}\sqrt{\Sigma_n}S_n^T \square \hat{\mathcal{P}}_\xi \hat{\mathcal{Q}}_\zeta$.
- Since $\hat{P}_\alpha$ are the first r rows of $\hat{\mathcal{P}}_\xi$; $\hat{P}_\alpha = \begin{bmatrix} I_r & 0 & \cdots \end{bmatrix} \hat{\mathcal{P}}_\xi \square E_r^T \hat{\mathcal{P}}_\xi$ $= E_r^T R_n \sqrt{\Sigma_n}$.

- There are three boundary conditions must be satisfied;

  $P_\alpha \equiv \left[(C)^T \quad (CA)^T \quad \cdots \quad \left(CA^{\alpha-1}\right)^T \right]^T$, $Q_\beta \equiv \left[B \quad AB \quad \cdots \quad A^{\beta-1}B \right]$, and

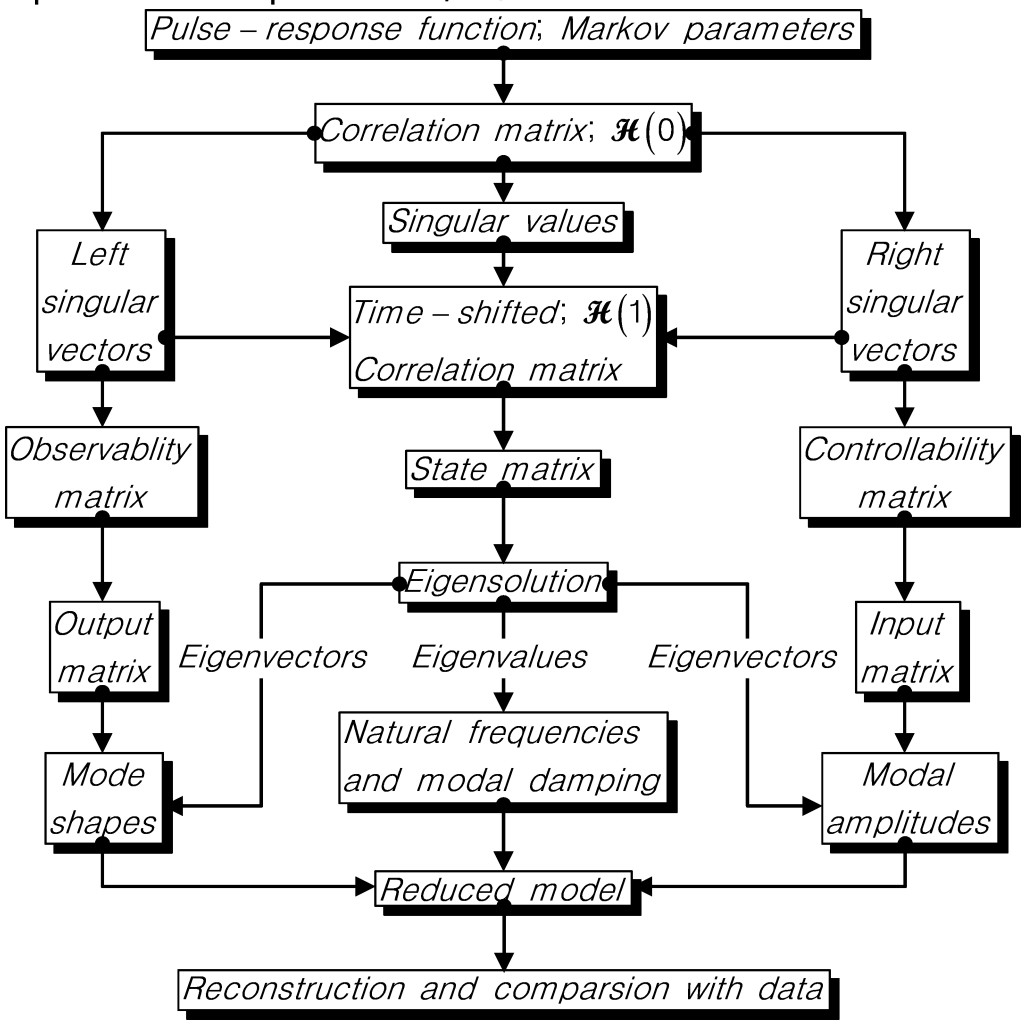
  $H(0) = P_\alpha Q_\beta$. It implies that $\Rightarrow Q_\beta = P_\alpha^{\#} H(0) = \left(E_r^T R_n \sqrt{\Sigma_n} \right)^{\#} H(0)$.

- $\hat{C}$ are the first m rows of $\hat{P}_\alpha$ and $\hat{B}$ are the first r columns of $\hat{Q}_\beta$.

  $\Rightarrow \hat{B} = \left(E_r^T R_n \sqrt{\Sigma_n} \right)^{\#} H(0) E_r$ and $\hat{C} = E_m^T R_n \sqrt{\Sigma_n}$. Then,

  $\because \mathcal{H}(1) = \mathcal{P}_\xi \hat{A} \mathcal{Q}_\zeta = R_n \sqrt{\Sigma_n} \hat{A} \sqrt{\Sigma_n} S_n^T \Rightarrow \hat{A} = \sqrt{\Sigma_n}^{-1} R_n^T \mathcal{H}(1) S_n \sqrt{\Sigma_n}^{-1}$.

- Computational steps of ERA/DC:



[Note]: