3.5 IIDENTIFICATION METHODS

3.5.1 PREDICTION ERROR METHOD

- Estimate model parameters which minimizes the *optimally determined one-step-ahead output prediction error.*
- The output predictor is constructed using the model.
- The existing identification methods such as LSM, OEM, GLSM, ELSM, PLRM, MLM etc. are special cases of PEM which are derived for different model types.
  Hence, the PEM can be considered a kind of generalized framwork for system identification.

Example 1: Identification of an ARMAX process using an ARX Model

True process :

$$y(k) + \bar{a}y(k-1) = \bar{b}u(k-1) + n(k) + \bar{c}_1 n(k-1)$$

where $n(k)$ is white noise($\sim \ (0, \sigma_n^2)$).
Model :

$$y(k) + ay(k-1) = bu(k-1) + e(k) \ \text{ ARX model}$$

where $e(k)$ is assumed to be a zero-mean white noise.

Procedure

1. Given $\{y(k-1), y(k-2), \cdots\}$ and $\{u(k-1), u(k-2), \cdots\}$, the best one-step ahead output prediction is

$$\hat{y(k)} = -ay(k-1)+bu(k-1) = [\ -y(k-1)\ u(k-1)\ ] \begin{bmatrix} a \\ b \end{bmatrix} = \phi(k)^T\theta$$

2. The prediction errror at k is

$$\varepsilon(k,\theta) = y(k) - \hat{y}(k) = y(k) - \phi(k)^T\theta$$

3. θ which minimizes the sum of squared prediction error can be found as

$$\min_{\theta} \quad \sum_{k=1}^{N} \varepsilon(k,\theta)^T\varepsilon(k,\theta)$$

4. The above precedure can be rewritten in the vector form as

$$\begin{bmatrix} \varepsilon(1,\theta) \\ \vdots \\ \varepsilon(N,\theta) \end{bmatrix} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} - \begin{bmatrix} \phi(1)^T \\ \vdots \\ \phi(N)^T \end{bmatrix} \theta \quad \rightarrow \quad E_N(\theta) = Y_N - \Phi_N\theta$$

$$\min_{\theta} E_N(\theta)^T E_N(\theta)$$

$$\hat{\theta}_{LS} = (\Phi_N^T\Phi_N)^{-1}\Phi_N^T Y_N$$

Discussions

- The PEM tries to seek a parameter which minimizes the prediction error. In case that the model has a different structure from the process, the parameter is determined such that the PE is minimized under its structural constraints. This usually leads to *unbiased estimate* as shown below. The process output can be written as

$$\begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \phi(1)^T \\ \vdots \\ \phi(N)^T \end{bmatrix} \bar{\theta} + \begin{bmatrix} n(1) + \bar{c}n(0) \\ \vdots \\ n(N) + \bar{c}n(N-1) \end{bmatrix} \rightarrow Y_N = \Phi_N \bar{\theta} + V_N$$

Hence,

$$\hat{\theta} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T (\Phi_N \bar{\theta} + V_N) = \bar{\theta} + (\Phi_N^T \Phi_N)^{-1} \Phi_N^T V_N$$

Taking expectation gives

$$E\left\{\hat{\theta}_{LS}\right\} = \bar{\theta} + \underbrace{E\left[(\Phi_N^T \Phi_N)^{-1} \Phi_N^T V_N\right]}_{=\ 0\ ?}$$

Now,

$$\begin{aligned} \Phi_N^T V_N &= \begin{bmatrix} -y(0) & -y(1) & \cdots & -y(N-1) \\ u(0) & u(1) & \cdots & u(N-1) \end{bmatrix} \begin{bmatrix} n(1) + \bar{c}n(0) \\ n(2) + \bar{c}n(1) \vdots \\ n(N) + \bar{c}n(N-1) \end{bmatrix} \\ &= \begin{bmatrix} -y(0)\left(n(1) + \bar{c}n(0)\right) - y(1)\left(n(2) + \bar{c}n(1)\right) - \cdots \\ \cdots\cdots \end{bmatrix} \end{aligned}$$

Since $y(k)$ and $n(k)$ have correlation ($E\{y(k)n(k)\} = \sigma^2$),

$$E\left\{\hat{\theta}\right\} \neq \bar{\theta} \quad \rightarrow \quad \text{BIASED !!}$$

If $\bar{c} = 0$, unbased estimate !!

Example 2: Revisit of Example 1 with an ARMAX Model

This time, we consider an ARMAX model

$$y(k) + ay(k-1) = bu(k-1) + e(k) + ce(k-1)$$

- Note that $e(k)$ is not directly measurable. However, it affects $y(k)$. Hence, by treating $y(k)$ we can obtain an estimate of $e(k)$.

Procedure

1. Let the estimate of $e(k)$ be $\hat{e}(k)$.
   At $k-1$, the best one-step-ahead output prediction is

$$\begin{aligned}\hat{y}(k) &= -ay(k-1) + bu(k-1) + c\hat{e}(k-1) \\ &= \begin{bmatrix} -y(k-1) & u(k-1) & \hat{e}(k-1) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \phi_k^T \theta\end{aligned}$$

2. As was in Ex. 1,

$$\hat{\theta} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N$$

3. In the above, $\hat{e}(k)$ can be obtained by inverting the model equation.

$$\hat{e}(k) = -\hat{c}\hat{e}(k-1) + y(k) + \hat{a}y(k-1) - \hat{b}u(k-1), \quad \hat{e}(0) = 0$$

Discussions

- To find $\hat{e}(k)$, $\hat{\theta}$ should known. On the other hand, $\hat{e}(k)$ is needed to find $\hat{\theta}$. $\rightarrow$ Nonlinear equation. Backsubstitution or other nonlinear solver is required.
- Due to the structural consistency, unbiased estimate is obtained.

General Properties of PEM

1. Need a priori infomation on the model structure (model type and orders of each term)
2. Structural inconsistency may lead to biased parameter estimates. The bias is revealed differently for differentl input excitation.
3. An ARMAX model with sufficiently large orders is OK for most applications.
   To find a parsimonious model, however, trial and error procedure with different orders is usaully necessary.
4. Generally, nonlinear equation should be solved to find an estimate. Optimumj solution is not always guaranteed.
5. Recursive (or On-line) PEM algorithms are avaliable, too
6. The PEM can be extended to MIMO identification, too.
   However, lack of an appropriate canonical form for the MIMO ARMAX model leads to an overparameterized model structure.
   The industrial practice for MIMO identification is to seperate the model into n_y MISO (multi-input single-output) susbsystems, conduct MISO identification independently, and combine the results.

$$\mathbf{A}(q^{-1})\mathbf{y}(k) = \mathbf{B}(q^{-1})\mathbf{u}(k) + \mathbf{C}(q^{-1})\mathbf{n}(k) \quad \rightarrow$$

$$\begin{bmatrix} A_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_n \end{bmatrix} \begin{bmatrix} y_1(k) \\ \vdots \\ y_n(k) \end{bmatrix} = \begin{bmatrix} B_{11} & \cdots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{n1} & \cdots & B_{nm} \end{bmatrix} \begin{bmatrix} u_1(k) \\ \vdots \\ u_m(k) \end{bmatrix} + \begin{bmatrix} C_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_n \end{bmatrix} \begin{bmatrix} n_1(k) \\ \vdots \\ n_n(k) \end{bmatrix}$$

$$\begin{aligned} A_1(q^{-1})y_1(k) &= B_{11}(q^{-1})u_1(k) + \cdots + B_{1m}(q^{-1})u_m(k) + C_n(q^{-1})n_1(k) \\ &\vdots \qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\ A_n(q^{-1})y_n(k) &= B_{n1}(q^{-1})u_1(k) + \cdots + B_{nm}(q^{-1})u_m(k) + C_n(q^{-1})n_n(k) \end{aligned}$$

7. MISO identification is in effect same as SISO identification
8. MIMO identification via MISO identification cannot take the directional characterisitcs of MIMO systems into account.

3.5.2 SUBSPACE IDENTIFICATION

- Subspace identification(SSID) is a very powerful method that has been emergered from early 90s.
- SSID
  1. is an off-line identification method (at least currently),
  2. does not require virtually any a priori information on the model structure,
  3. can be seamlessly extended to MIMO identification,
  4. provides an optimally balanced stochastic state space model, $(A,\ B,\ C)$ and noise covariances, of minimum order using input /output data.
- Moreover, SSID does not solve a nonlinear equation as in the PEM, but only relies on numerically stable linear operations.
- The ID package, IPCOS, from SETPOINT is based on a version of SSID.
- More comments will be given later.

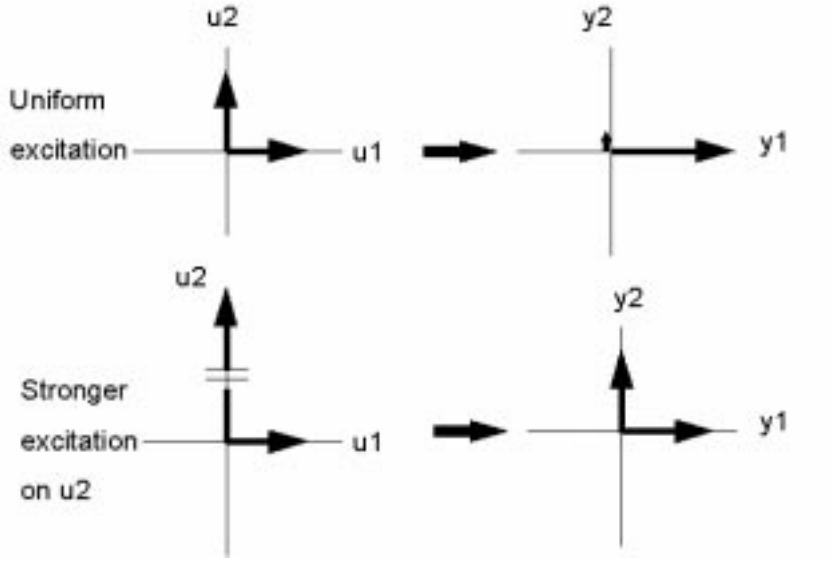
3.6 IDENTIFICATION OF A PROCESS WITH STRONG DIRECTIONALITY

Suppose that we want to control the following 2×2 system.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

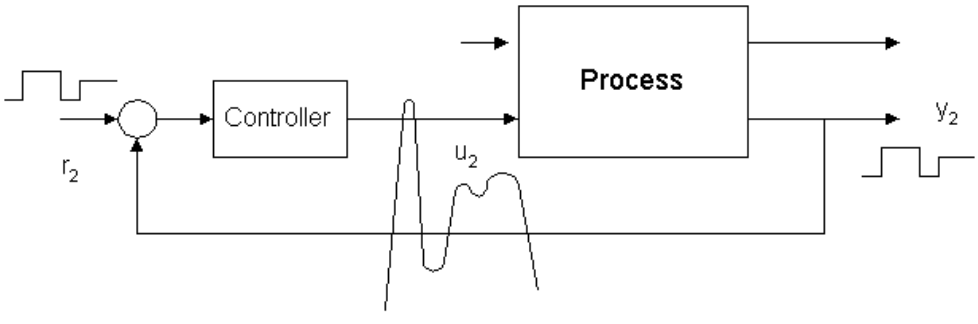
where $a_1 = 100,\ a_2 = 0.1$ and $|e_1|, |e_2| \approx 1$.
For controller design, a_1 and a_2 should be known.

- To identify a_1 and a_2, assume that we apply excitation signals of magnitude 1 to both inputs u_1 and u_2
  S/N(signal to noise ratio) for $y_2 \approx 0.1$ while S/N for $y_1 \approx 100$
  The consequence is that $\hat{a}_2$ is not correctly identified. In the worst case, the sign may be reversed.
  $\Rightarrow$ The y_2 control loop performs poorly or may be unstable.
- In order to get over the trouble,
  - we either apply large exciation to u_2 in open-loop



or

– close the y_2 loop and apply an excitation from outside the loop.



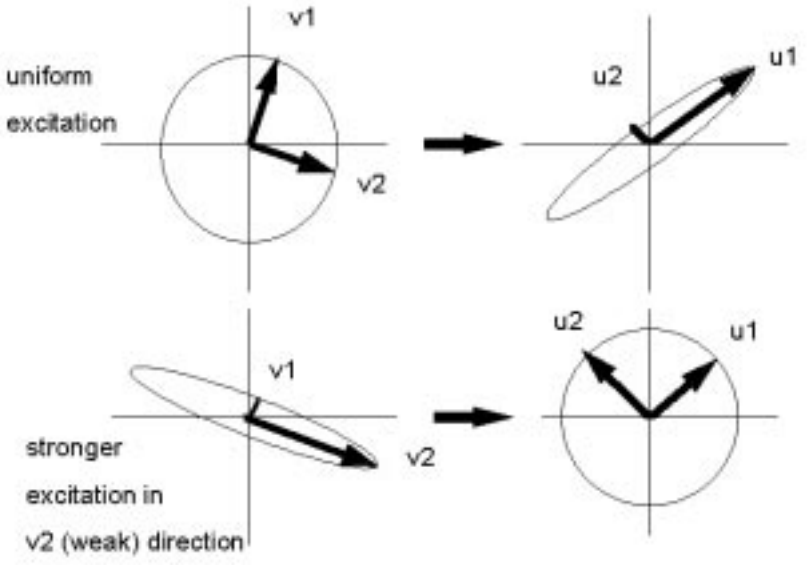
Now consider a more general case.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{G} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

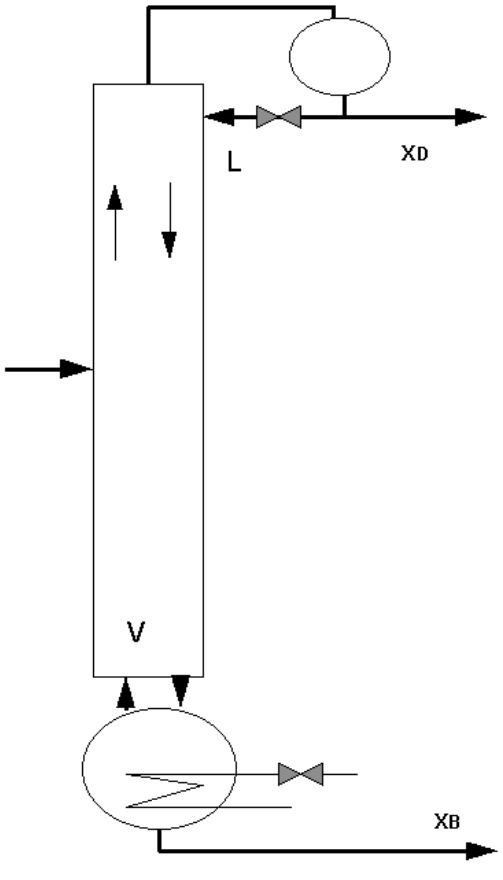
$\mathbf{G}$ is decomposed as

$$\mathbf{G} = [\mathbf{u}_1 \ \mathbf{u}_2] \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix} \quad --\text{SVD}$$

Here, $\mathbf{u}_1 \perp \mathbf{u}_2$, $\mathbf{v}_1 \perp \mathbf{v}_2$, $\|\mathbf{u}_1\| = \|\mathbf{u}_2\| = \|\mathbf{v}_1\| = \|\mathbf{v}_2\| = 1$.
If $\sigma_1 \gg \sigma_2$, the same problem as before arises.
To avoid the problem, it is necessary to apply large input along the weak direction to the process either in an open-loop or a closed-loop manner.



[Example :] High Purity Binary Distillation



In high purity separation,

$$\begin{bmatrix} \Delta L \\ \Delta V \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta x_D \\ \Delta x_B \end{bmatrix} = \begin{bmatrix} +10^{-4} \\ -10^{-4} \end{bmatrix}$$

$$\begin{bmatrix} \Delta L \\ \Delta V \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \Rightarrow \begin{bmatrix} \Delta x_D \\ \Delta x_B \end{bmatrix} = \begin{bmatrix} 10^{-1} \\ 10^{-1} \end{bmatrix}$$

$$\begin{bmatrix} \Delta x_D \\ \Delta x_B \end{bmatrix} = [\mathbf{u}_1 \;\; \mathbf{u}_2] \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \Delta L \\ \Delta V \end{bmatrix}$$

where $\sigma_1 \gg \sigma_2$