非線性化工程序之模糊模式預測控制

計劃編號 ： NSC89-2214-E-002-009
執行期限 ： 88 年 8 月 1 日至 89 年 7 月 31 日
主持人 ： 與誠亮 國立台灣大學化學工程學系 教授

一、摘要

本文旨在探討非線性動態系統之多重線性模式建構問題，並根據多重線性模式作模式預測控制。多重線性模式做法為將一個非線性動態系統可能的操作範圍劃分成幾個小區域，依照相對應的權重合併各個子區域的線性模式合成為整體非線性模式。對於非線性的系統，採用以多重線性模式為基礎的動態矩陣控制。將操作範圍劃分成幾個小區域，讓每一個線性控制器只負責一個範圍，再依照相對應的權重合併各個子區域的模式或控制器輸出。最後以高度非線性的酸鹼滴定程序和一具有溫度控制與液位控制的攪拌槽來說明本文所提建構與控制方式的可行性。

ABSTRACT

This thesis aims to present the construction of the multi-linear model and multi-linear model based control system design. The multi-linear model divides the possible operating regime into several sub-regimes. The multi-linear model combines several linear models according to the corresponding weighting to a global model. For a nonlinear system, dynamic matrix control based on the multi-linear model is presented in this thesis. One linear controller is designed for each sub-regime, and the current control output is obtained by the weighting sum of all sub-controller outputs, parameters or output changes. Finally, the control strategy is illustrated by two chemical processes. One is the neutralization process and the other is the mixing tank with temperature and level control.

二、計劃緣由與目的

在本文中我們將非線性系統依其操作範圍劃分成幾個子區域，找到可以代表此區域的線性模式，再依相對應權重 (weighting) 合併各區域的線性模式，如此所得的整體 (global) 模式可以代表非線性程序的動態。對於控制策略亦是如此，若只對一個線性模式來作設計時無法得到滿意的結果，那麼可以將操作範圍劃分成幾個小區域，讓每一個控制器只負責一個範圍，同樣地吾人可以預期會得到較佳的結果。在本文中將透過模糊邏輯推論的觀念來建構多重線性模式與控制設計。而面對一個非線性的多變數系統，吾人所採取的方法是使用多重線性的動態矩陣控制 (Dynamic Matrix Control,DMC)，動態矩陣控制乃是模式預測控制中較著名的一種。多變數系統主要的問題來自環路之間交互作用的影響，而動態矩陣控制很適合來解決此一問題。所以對於非線性多變數系統，本文結合多重線性模式與線性的動態矩陣控制，以期避免掉非線性控制的繁複計算，並能有較一般線性控制為佳的結果。

三、研究方法與成果

• 模糊多重線性動態模式

多重線性模式是以模糊模式為基礎，在不同的操作點分別由非線性數學模式的線性化或由傳統系統識別的方法得到一個線性動態模式，故集合所有的模式，即可以得到下列規則庫：

R_1 :

$$\textit{IF}\ \mathbf{x}\text{ is }\mathbf{A}_1\ \textit{and}\ \mathbf{y}\text{ is }\mathbf{B}_1\ \textit{and}\ \mathbf{u}\text{ is }\mathbf{C}_1$$
$$\textit{THEN}\ \begin{cases}\dot{\mathbf{x}} = \mathbf{F}_1\mathbf{x} + \mathbf{G}_1\mathbf{u}\\ \mathbf{y} = \mathbf{H}_1\mathbf{x}\end{cases}$$

R_2 :

$$\textit{IF}\ \mathbf{x}\text{ is }\mathbf{A}_2\ \textit{and}\ \mathbf{y}\text{ is }\mathbf{B}_2\ \textit{and}\ \mathbf{u}\text{ is }\mathbf{C}_2$$
$$\textit{THEN}\ \begin{cases}\dot{\mathbf{x}} = \mathbf{F}_2\mathbf{x} + \mathbf{G}_2\mathbf{u}\\ \mathbf{y} = \mathbf{H}_2\mathbf{x}\end{cases}$$

$$\vdots \quad \vdots$$

R_n :

$$\textit{IF}\ \mathbf{x}\text{ is }\mathbf{A}_n\ \textit{and}\ \mathbf{y}\text{ is }\mathbf{B}_n\ \textit{and}\ \mathbf{u}\text{ is }\mathbf{C}_n$$
$$\textit{THEN}\ \begin{cases}\dot{\mathbf{x}} = \mathbf{F}_n\mathbf{x} + \mathbf{G}_n\mathbf{u}\\ \mathbf{y} = \mathbf{H}_n\mathbf{x}\end{cases}$$

或是以轉移函數 (transfer function) 表示之：

R_1 :

$$\textit{IF}\ \mathbf{x}\text{ is }\mathbf{A}_1\ \textit{and}\ \mathbf{y}\text{ is }\mathbf{B}_1\ \textit{and}\ \mathbf{u}\text{ is }\mathbf{C}_1$$
$$\textit{THEN}\ \mathbf{y}(s) = \mathbf{G}_1(s)\mathbf{u}(s)$$

R_2 :

$$\textit{IF}\ \mathbf{x}\text{ is }\mathbf{A}_2\ \textit{and}\ \mathbf{y}\text{ is }\mathbf{B}_2\ \textit{and}\ \mathbf{u}\text{ is }\mathbf{C}_2$$
$$\textit{THEN}\ \mathbf{y}(s) = \mathbf{G}_2(s)\mathbf{u}(s)$$

$$\vdots \quad \vdots$$

R_n :

$$\textit{IF}\ \mathbf{x}\text{ is }\mathbf{A}_n\ \textit{and}\ \mathbf{y}\text{ is }\mathbf{B}_n\ \textit{and}\ \mathbf{u}\text{ is }\mathbf{C}_n$$
$$\textit{THEN}\ \mathbf{y}(s) = \mathbf{G}_n(s)\mathbf{u}(s)$$

其中

$\mathbf{A}_i$　對應第 i 條規則之狀態 $\mathbf{x}$ 的口語化項目
$\mathbf{B}_i$　對應第 i 條規則之程序輸出 $\mathbf{y}$ 的口語化項目
$\mathbf{C}_i$　對應第 i 條規則之程序輸入 $\mathbf{u}$ 的口語化項目
$\mathbf{u}$　輸入變數向量
$\mathbf{x}$　狀態向量
$\mathbf{y}$　輸出向量

也就是說，多重線性動態模式是將整個程序的操作範圍分割成數個子區域（分割方式可由狀態變數 $\mathbf{x}$，輸出變數 $\mathbf{y}$ 或輸入變數 $\mathbf{u}$ 決定），對於每個子區域以一個線性模式描述其動態行為，再以模糊邏輯的方式合併每個規則的輸出而得到模式的輸出。

• 多重線性模式之推論方法

A. 狀態空間模式

將多重線性模式中的每一個子模式離散化，即可得到線性的代數式，但由於每個子模式的基準點均不相同，故無法直接透過離散化子模式計算模式輸出，也就是說，多重線性模式的推論方式必須做一部份的修正。在文獻中，Banerjee 採取系統參數內插法，利用穩態拓樸空間 (steady state manifold) 上最接近目前觀測值的點作為目前穩態 $\mathbf{y}$、$\mathbf{u}$ 的估測值，並用最適化方法求出當時所有狀態的穩態估計值，再以此得到各參數之權重，求得參數，最後計算模式輸出。此種方法不但需先求出整個穩態拓樸空間，而且在求取狀態的穩態時亦需大量使用最適化技巧，故計算負荷相當繁重。更嚴重的是，由穩態拓樸空間所得到的穩態估測值大多仍有相當大的偏差，所以模式所估測之輸出之準確性易常讓人質疑。因此，若能避免穩態的估計，對模式的準確性會有一些幫助。

對於一離散線性系統，

$$\left\{\begin{array}{l}\mathbf{x}(k+1)=\mathbf{x}_{ss}+\mathbf{\Phi}(\mathbf{x}(k)-\mathbf{x}_{ss})+\mathbf{\Gamma}(\mathbf{u}(k)-\mathbf{u}_{ss})\\ \mathbf{y}(k)=\mathbf{y}_{ss}+\mathbf{H}(\mathbf{x}(k)-\mathbf{x}_{ss})\end{array}\right. \tag{1}$$

若取前一取樣時間相減，即可消去穩態值 $\mathbf{x}_{ss}$、$\mathbf{y}_{ss}$ 及 $\mathbf{u}_{ss}$，則式 (1)可改寫為：

$$\left\{\begin{array}{l}\Delta\mathbf{x}(k+1)=\mathbf{\Phi}\Delta\mathbf{x}(k)+\mathbf{\Gamma}\Delta\mathbf{u}(k)\\ \Delta\mathbf{y}(k)=\mathbf{H}\Delta\mathbf{x}(k)\end{array}\right. \tag{2}$$

其中：

$$\begin{aligned}\Delta\mathbf{x}(k)&=\mathbf{x}(k)-\mathbf{x}(k-1)\\ \Delta\mathbf{y}(k)&=\mathbf{y}(k)-\mathbf{y}(k-1)\\ \Delta\mathbf{u}(k)&=\mathbf{u}(k)-\mathbf{u}(k-1)\\ \Delta\mathbf{x}(k)&=0\qquad\forall k\le 0\end{aligned}$$

故以式 (2)為基礎所建立之多重線性模式成為一模糊動態模式，只不過其輸入變數由 $\mathbf{u}(k)$ 改為 $\Delta\mathbf{u}(k)$，輸出 $\mathbf{y}(k)$ 則變為 $\Delta\mathbf{y}(k)$。

因此實務上多重線性模式應為如下之規則庫：

R_i : ***IF*** $\mathbf{x}$ is $\mathbf{A}_i$ ***and*** $\mathbf{y}$ is $\mathbf{B}_i$ ***and*** $\mathbf{u}$ is $\mathbf{C}_i$ ***THEN***

$$\left\{\begin{array}{rcl}\Delta\mathbf{x}(k+1)&=&\mathbf{\Phi}_i\Delta\mathbf{x}(k)+\mathbf{\Gamma}_i\Delta\mathbf{u}(k)\\ \Delta\mathbf{y}(k)&=&\mathbf{H}_i\Delta\mathbf{x}(k)\end{array}\right. \tag{3}$$

$$\begin{aligned}&i=1,2,\ldots,n\\ &\Delta\mathbf{y}(k)=\mathbf{y}(k)-\mathbf{y}(k-1)\\ &\Delta\mathbf{x}(k)=\mathbf{x}(k)-\mathbf{x}(k-1)\\ &\Delta\mathbf{u}(k)=\mathbf{u}(k)-\mathbf{u}(k-1)\end{aligned}$$

既然式 (3)為一模糊模式，考慮以下兩種模糊推論方式：

1. 模糊化模式參數：以模糊化模式參數計算輸出，則第 $(k+1)$ 個取樣時間之程序輸出可表示成：

$$\mathbf{\Phi}(\boldsymbol{\omega})=\sum_{i=1}^{n}\omega_i\mathbf{\Phi}_i,\quad \mathbf{\Gamma}(\boldsymbol{\omega})=\sum_{i=1}^{n}\omega_i\mathbf{\Gamma}_i,\quad \mathbf{H}(\boldsymbol{\omega})=\sum_{i=1}^{n}\omega_i\mathbf{H}_i$$

$$\begin{aligned}\Delta\mathbf{y}(k)&=\mathbf{H}(\boldsymbol{\omega})\Delta\mathbf{x}(k)\\ &=\mathbf{H}(\boldsymbol{\omega})\mathbf{\Phi}(\boldsymbol{\omega})\Delta\mathbf{x}(k-1)+\mathbf{H}(\boldsymbol{\omega})\mathbf{\Gamma}(\boldsymbol{\omega})\Delta\mathbf{u}(k-1)\end{aligned}$$

$$\begin{aligned}\therefore\ \mathbf{y}(k+1)=\mathbf{y}(k)&+\left(\sum_{i=1}^{n}\omega_i\mathbf{H}_i\right)\left(\sum_{i=1}^{n}\omega_i\mathbf{\Phi}_i\right)\Delta\mathbf{x}(k-1)\\ &+\left(\sum_{i=1}^{n}\omega_i\mathbf{H}_i\right)\left(\sum_{i=1}^{n}\omega_i\mathbf{\Gamma}_i\right)\Delta\mathbf{u}(k-1)\end{aligned} \tag{4}$$

2. TSK 模式計算法

由 TSK 模式計算而得之程序輸出結果為：

$$\begin{aligned}\Delta\mathbf{y}(k)&=\sum_{i=1}^{n}\omega_i\Delta\mathbf{y}_i(k)\\ &=\sum_{i=1}^{n}\omega_i\mathbf{H}_i\Delta\mathbf{x}(k)\\ &=\sum_{i=1}^{n}\left[\omega_i\mathbf{H}_i\sum_{j=1}^{n}\omega_j\Delta\mathbf{x}_j(k)\right]\\ &=\sum_{i=1}^{n}\left\{\omega_i\mathbf{H}_i\sum_{j=1}^{n}\omega_j\left[\mathbf{\Phi}_j\Delta\mathbf{x}(k-1)+\mathbf{\Gamma}_j\Delta\mathbf{u}(k-1)\right]\right\}\\ &=\sum_{i=1}^{n}\omega_i\mathbf{H}_i\left\{\sum_{j=1}^{n}\omega_j\left[\mathbf{\Phi}_j\Delta\mathbf{x}(k-1)+\mathbf{\Gamma}_j\Delta\mathbf{u}(k-1)\right]\right\}\end{aligned}$$

圖 1 多重線性模式的推論方式之一：模糊化程序參數

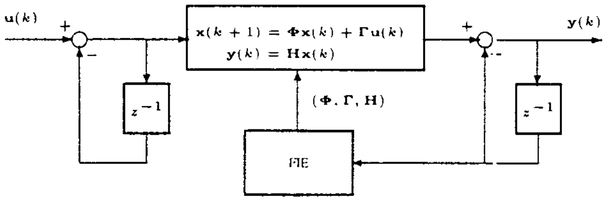
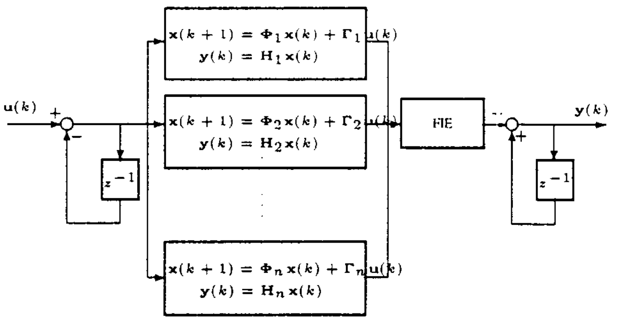


圖 2 多重線性模式的推論方式之二：TSK 模糊模式



$$\therefore \mathbf{y}(k+1) = \mathbf{y}(k) + \left(\sum_{i=1}^{n} \omega_i \mathbf{H}_i\right)\left(\sum_{i=1}^{n} \omega_i \mathbf{\Phi}_i\right) \Delta\mathbf{x}(k-1) + \left(\sum_{i=1}^{n} \omega_i \mathbf{H}_i\right)\left(\sum_{i=1}^{n} \omega_i \mathbf{\Gamma}_i\right) \Delta\mathbf{u}(k-1) \quad (5)$$

其中,

$$\Delta\mathbf{x}(k) = \sum_{i=1}^{n} \omega_i \Delta\mathbf{x}_i(k)$$

由式 (4)和式 (5)可看出，兩式的效果實際上是完全一樣的。

B. 轉移函數模式

轉移函數模式的推論方式在理論上與狀態空間模式完全相同，但其模式輸出表示式會較為簡潔。由於狀態在轉移函數當中可自由定義，因此通常轉移函數模式規則庫中會以輸出和輸入變數 (y、u) 做為分割依據。故由轉移函數離散化後之 SISO 系統規則庫為：

R_i : ***IF*** y is $\mathbf{B}_i$ ***and*** u is C_i ***THEN***

$$\Delta y(k) = a_{i1}\Delta y(k-1) + \ldots + a_{ip}\Delta y(k-p) + b_{i0}\Delta u(k) + \ldots + b_{ip}\Delta u(k-p) \quad (6)$$

至於 MIMO 系統，則應修改為式 (7)(系統為 r 輸入 m 輸出)，

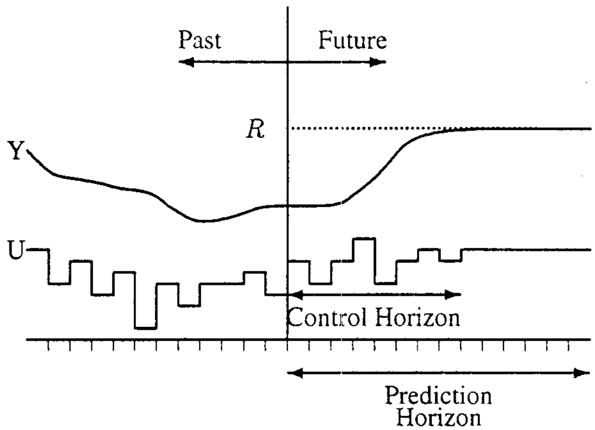
R_i : ***IF*** y is B_i ***and*** u is C_i ***THEN***

$$\begin{aligned}
\Delta y_1(k) &= \sum_{j=1}^{p} a_{1ij}\Delta y_1(k-j) + \sum_{\ell=1}^{r}\sum_{j=0}^{p} b_{1ij\ell}\Delta u_\ell(k-j) \\
\Delta y_2(k) &= \sum_{j=1}^{p} a_{2ij}\Delta y_1(k-j) + \sum_{\ell=1}^{r}\sum_{j=0}^{p} b_{2ij\ell}\Delta u_\ell(k-j) \\
\vdots \quad & \vdots \quad \vdots \\
\Delta y_m(k) &= \sum_{j=1}^{p} a_{mij}\Delta y_1(k-j) + \sum_{\ell=1}^{r}\sum_{j=0}^{p} b_{mij\ell}\Delta u_\ell(k-j)
\end{aligned} \quad (7)$$

- **動態矩陣控制**

在本論文中，主要將使用動態矩陣控制 (Dynamic Matrix Ccontrol, DMC) 的方式控制非線性程序。吾人也利用圖 (3)來說明模式預測控制的精神。

圖 3 模式預測控制示意圖



A. 單變數系統之動態矩陣控制

1. 階梯應答模式：階梯應答模式，也就是褶積模式 (Convolution model)。在時間為零時，給予動態系統一單位階梯變化 ($\Delta u_1 = 1$)，而獲得程序的輸出。在第一個取樣時間 ($t = T_s$)，程序的輸出為 a_1；在第二個取樣時間 ($t = 2T_s$)，程序的輸出為 a_2。依此類推，在第 k 個取樣時間 ($t = kT_s$)，程序的輸出為 a_k。如果 Δu_1 不等於 1，則程序的輸出為階梯應答係數 a_1 乘上 Δu_1。程序的輸出值，若是以階梯應答模式的方式表達，將可以寫成如下：

$$y(k) = \sum_{i=1}^{H_m} a_i \Delta u(k-i) + a_{ss} u(k - H_m - 1) \quad (8)$$

其中

$$
\begin{aligned}
y(k) &: \text{在第 } k \text{ 個瞬間的程序輸出} \\
a_i &: \text{在第 } k \text{ 個瞬間的階梯應答係數} \\
u(k) &: \text{在第 } k \text{ 個瞬間的控制器輸出} \\
\Delta u(k) &= u(k) - u(k-1) \\
H_m &: \text{模式水平 (Model Horizon)}
\end{aligned}
$$

2. 預測模式：程序的輸出值如下：

$$\mathcal{Y}(k+1) = \mathcal{A}\Delta\mathcal{U}(k) + \mathcal{F}(k) \tag{9}$$

其中

$$
\begin{aligned}
\mathcal{Y}(k+1) &= \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+H_p) \end{bmatrix}_{H_p \times 1} \\
\mathcal{A} &= \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{H_c} & a_{H_c-1} & \cdots & a_1 \\ \vdots & \vdots & \ddots & \vdots \\ a_{H_p} & a_{H_p-1} & \vdots & a_{H_p-H_c-1} \end{bmatrix}_{H_p \times H_c} \\
\Delta\mathcal{U} &= \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+H_c-1) \end{bmatrix}_{H_c \times 1} \\
f_j(k) &= \sum_{i=j+1}^{H_m} a_i \Delta u(k+j-i) \\
&\quad + a_{ss} u(k - H_m + j - 1) \\
\mathcal{F}(k) &= \begin{bmatrix} f_1(k) \\ f_2(k) \\ \vdots \\ f_{H_p}(k) \end{bmatrix}_{H_p \times 1}
\end{aligned}
$$

3. 控制律：我們要求系統的目標函數為誤差和的平方和與操作變數變化速率的平方和最小

$$
\begin{aligned}
& \min_{\Delta\mathcal{U}(k)} J \\
=\ & \min_{\Delta\mathcal{U}(k)} \{\|\Gamma E(k)\|^2 + \|\Lambda\Delta\mathcal{U}(k)\|^2\} \qquad (10) \\
=\ & \min_{\Delta\mathcal{U}(k)} \{E^T(k)\Gamma^T\Gamma E(k) + \Delta\mathcal{U}^T(k)\Lambda^T\Lambda\Delta\mathcal{U}(k)\}
\end{aligned}
$$

其中

$$\Gamma = \begin{bmatrix} \Gamma_1 & 0 & 0 & \cdots & 0 \\ 0 & \Gamma_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \Gamma_{H_p} \end{bmatrix}_{H_p \times H_p},$$

對誤差的權重矩陣

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \Lambda_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \Lambda_{H_c} \end{bmatrix}_{H_c \times H_c},$$

對操作變數變化的權重矩陣

$$E(k) = \hat{\mathcal{Y}}(k+1) - \mathcal{R}(k)$$

$$\mathcal{R}(k) = \begin{bmatrix} r(k+1) \\ r(k+2) \\ \vdots \\ r(k+H_p) \end{bmatrix}_{H_p \times 1},$$

在未來 H_p 步的設定點之值

式 (10)即為一個最小平方的問題 (Least Square Problem)，解法為式 (10)對 $\Delta\mathcal{U}(k)$ 偏微分後，解開之後可得模式的控制律：

$$
\begin{aligned}
u(k) &= u(k-1) + \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}_{1 \times H_c} \\
&\quad (\mathcal{A}^T W_1 \mathcal{A} + fI)^{-1} \mathcal{A}^T W_1 \hat{E}(k)
\end{aligned}
$$

B. 多變數系統之動態矩陣控制

多變數系統吾人亦根據階梯應答模式、預測模式推導得到其輸出，控制律亦根據單變數的方式求得，因奈於篇幅有限，故在此不做多述。

● 以多重線性模式為基礎的控制設計

動態矩陣控制是使用階梯應答模式或脈衝應答模式，來做程序的模式預測，並根據程序模式來預測程序的輸出值。此外，藉由最適化的方法，在每一段取樣時間中，計算未來時間裡，未來幾步的操作變數應如何改變，才可使程序輸出的預測值與期望值的誤差最小，但是每次只使用第一個輸出值。而模式的好壞會直接影響控制系統的表現。所以對於非線性程序，將不採用傳統的線性模式來發展控制策略，以避免因程序偏離操作點而產生很大的誤差，使得控制的結果不甚理想。在此將藉由多重線性模式的觀念，分別針對每一個線性模式，並

利用線性的動態矩陣控制的理論，來發展控制策略，然後再依各個模式的適用性，權重合併各模式所建設之控制動作，如此可以獲得較佳的控制效果。

以多重線性模式為基礎之控制設計的步驟，首先是以模糊模式為基礎，在不同的操作點分別由非線性數學模式的線性化或由傳統系統識別的方法得到一個線性動態模式，再集合所有的模式，即可以得到其多重線性動態模式，也就是說，將整個程序的操作範圍分割成數個子區域（分割方式可由狀態變數 x，輸出變數 y 或輸入變數 u 決定），對於每個子區域以一個線性模式描述其動態行為，再以模糊邏輯的方式合併每個規則的輸出而得到模式的輸出。

基於多重線性模式的控制策略，在建構完多重線性模式後，便是分別對每一個線性模式發展控制策略。由於吾人在此選用的是線性的動態矩陣控制，而其基本的控制律規則如下：

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \mathbf{K}_{DMC}\hat{\mathbf{E}}(k) \tag{11}$$

現將動態矩陣控制的技巧帶入多重線性模式中，針對每一個線性模式分別發展其控制策略。在完成每一個線性模式的動態矩陣控制後，便可以建立動態矩陣的控制模式規則庫，再以模糊邏輯的方式合併每個規則的輸出而得到模式的輸出。所以最後可得控制器參數如下：

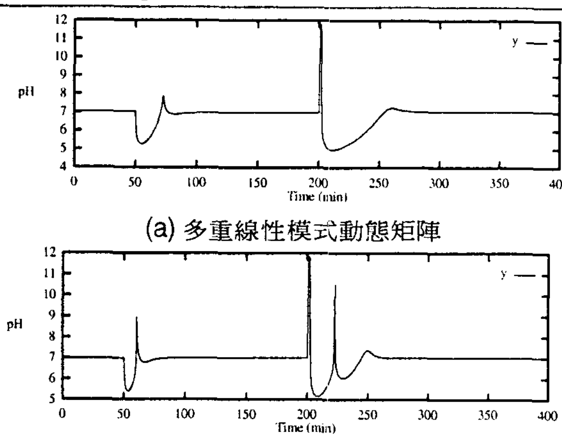
$$\begin{aligned} A &= \frac{\sum_{i=1}^{n}\phi_i A_i}{\sum_{i=1}^{n}\phi_i} && (12)\\ &= \sum_{i=1}^{n}\omega_i A_i \\ \mathbf{K}_{DMC} &= \frac{\sum_{i=1}^{n}\phi_i \mathbf{K}_{DMC_i}}{\sum_{i=1}^{n}\phi_i} && (13)\\ &= \sum_{i=1}^{n}\omega_i \mathbf{K}_{DMC_i} \\ &= \sum \omega_i \left[\mathrm{I}_{n_u\times n_u} \quad 0 \quad \cdots \quad 0\right]_{n_u\times\, n_u H_c} && (14)\\ &\quad (\mathcal{A}_i^T\mathbf{W}_1\mathcal{A}_i + f\mathrm{I}_{(n_u H_c)\times(n_u H_c)})^{-1}\mathcal{A}_i^T\mathbf{W}_1 \end{aligned}$$

• 研究範例 酸鹼滴定程序

酸鹼中和程序的模擬最早由 McAvoy *et al.*於 1972 年提出，是以氫氧化鈉水溶液滴定含醋酸之廢水。其高度非線性成為控制上非常難應付的問題。本節將以對其識別一多重線性模式，並對該模式提供一簡單的多重線性動態矩陣控制系統。並將其控制效果與單一線性模式動態矩陣做比較。我們考慮在時間 $t = 50$ 分鐘時，廢水流量增加至 $15\ell/\min$，$t = 200$ 分鐘時，廢水流量回復至 $10\ell/\min$。結果列於圖 (4)。

由圖 (4)分析，雖然多重線性動態矩陣控制未必控制最好。但可看出相較於單一線性動態矩陣控制的控制結果較佳。因為動態矩陣控制依據所得的模式作模式預測，而多重線性模式提供了較接近真實狀態的模式給動態矩陣，所以可以得到較佳的控制輸出。

圖 4 系統 pH 值的變化：流量負荷



(a) 多重線性模式動態矩陣

(b) 單一線性模式動態矩陣

四、結論與討論

多重線性模式是分段式地使用線性模式來描述整個系統，使得模式本身不再受限於傳統線性模式僅能準確地描述操作點附近行為的缺點，而能在整個操作範圍中對整個程序得到很好的描述。因此對於非線性系統，多重線性模式提供了有別於傳統方法的另一種選擇。多重線性模式是以模糊模式為基礎，在不同的操作點分別由非線性數學模式的線性化或由傳統系統識別的方法得到一個線性動態模式，而一個線性模式只代表一個小區域的動態，再依照相對應的權重合併各個子區域的線性模式合成為整體模式。對於控制策略亦是如此，將操作範圍劃分成幾個小區域，讓每一個控制器只負責一個範圍。再依照相對應的權重合併各個子區域的控制器參數或輸出。在論文中透過模糊邏輯推論的觀念來建構多重線性模式與控制設計。

動態矩陣控制是使用階梯應答模式或脈衝應答模式，來做程序的模式預測，並根據程序模式來預測程序的輸出值。藉由最適化的方法，在每一段取樣時間中，計算未來幾步的操作變數應如何改變。在使用動態矩陣控制的時候，需要找出一個代表真實程序的模式或是直接由工廠的數據，來近似真實的程序。而模式的準確與否，會影響到控制的效果。所以在本文中以多重線性模式取代傳統的單一線性模式，提供較為準確的模式。

五、參考文獻

Banerjee, A., Arkun, Y., Ogunnaike, B. and Pearson, R. Estimation of Nonlinear Systems Using Linear Multiple Models, *AIChE J.*, vol. 43, no. 5, pp. 1204-1226, 1997

Bhat, N., and McAvoy, T. J. Using of Neural Nets for Dynamic Modeling and Control of Chemical Process Systems, *Computers chem. Engng.*, vol. 14, no. 4, pp. 573-587, 1990