

廣義預測性控制設計變數之自我調整

Self-Tuning of Design Variables for Generalized Predictive Control

林 強

國立清華大學工程與系統科學系

莊哲男

國家高速電腦中心

一、摘要

本文介紹三種技術以決定 廣義預測性控制 設計中需用之模式階次及控制加權。此技術為應用模糊邏輯 (fuzzy logic)，基因演算法 (genetic algorithm)，與模擬退火(simulated annealing)去最佳化一設計之性能指標(performance index)或目標函數(objective function)。由於模糊邏輯在調整變數數值過程中變動較平滑且收斂快，故適用於即時與線上(on-line)使用。基因演算法與模擬退火則較適合於估計起始之模式階次與控制加權，以及這些數值在小範圍內之最終細微調整。本文中以數個多輸入多輸出系統之數值模擬結果，說明這些技術的應用。

二、簡介

在參考文獻一中使用適應預測性回授控制 (adaptive predictive feedback control)，已成功地壓抑平板振動以減少噪音。

適應預測性控制器包括有線上鑑別機構與控制體系。廣義預測性控制 (generalized predictive control, GPC) 的設計變數包括線上鑑別之自我回歸

[autoregressive exogeneous (ARX)] 模式階次與計算控制力 (control force) 之控制加權 (control weighting)。這兩個設計變數必須在線上實現 (implementation) 前決定。這些數值一般皆由實驗或試誤法取得。因此，需由具經驗之工程師進行調整這些數值的工作，但仍有可能無法達到最佳的性能。

此處發展使用模糊規則線上調整的方法，由一組適當的起始設計變數值開始進行調整，GPC 的性能可逐漸改進。此法係收集需要的數據傳送至高層，然後應用模糊規則來調整這些設計變數，重複此步驟直至滿足停止標準，然而此法缺點為無法保證控制器的穩定性。

為保證控制器穩定性，將問題轉化成最佳化問題，首先定義一目標函數，其中包括反應控制器性能的項目與控制器穩定性的處罰 (penalty) 項，使用基因演算法或模擬退火來執行控制加權和 ARX 模式階次的最佳值搜尋。兩種方法皆需評估在解答空間 (solution space) 中的目標函數值，以致在搜尋過程中可能會出現非常差的性能。因此，此法僅適用於控制加權的起始值調整和最終的細微調整。在起始調整時，則作模擬計算，使用鑑別的系模式進行目

標函數評估而不用線上實驗數據。然而在細微調整時，解答空間已限制在某一範圍內，此時控制器性能都可接受，因此在後者的情況，線上調整是可行的。

三、廣義預測性控制

在參考文獻一中定義之預測性矩陣方程式

$$y_s(k) = T u_s(k) + \Theta v_p(k-p) \quad (1)$$

向量 $y_s(k)$ 與 $u_s(k)$ 分別由目前與未來之輸出與輸入數據組成。其中從目前的時序 k 到一未來之時序 $k+s-1$ 。

$$y_s(k) = \begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+s-1) \end{bmatrix} \quad \text{and} \quad u_s(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+s-1) \end{bmatrix} \quad (2)$$

$v_p(k-p)$ 包含過去的輸入與輸出數據，時序從 $k-p$ 到 $k-1$ ，

$$v_p(k-p) = \begin{bmatrix} y(k-p) \\ \vdots \\ y(k-1) \\ u(k-p) \\ \vdots \\ u(k-1) \end{bmatrix} \quad (3)$$

矩陣 T 為一 Toeplitz 矩陣，由脈衝響應(pulse response)序列組成。矩陣 Θ 為一長方形係數矩陣，式中之整數 p 為鑑別模式的階次， s 是預測水平。

預測性控制的目標，是要計算控制量 $u_s(k)$ 使

其最小化下列之成本函數 (cost function)

$$J(k) = \frac{1}{2} \{ [y(k) - \tilde{y}(k)]^T [y(k) - \tilde{y}(k) + u_s(k)^T \Lambda u_s(k)] \} \quad (4)$$

式中的 Λ 是加權 $u_s(k)$ 的正定(positive-definite)矩陣。取式 (4) 對 $u_s(k)$ 的微分，並代入式 (1) 可得

$$u_s(k) = -[T^T T + \Lambda]^{-1} T^T [\Theta v_p(k-p) - \tilde{y}_s(k)] \quad (5)$$

向量 $u_s(k)$ 的前 r 個數值，為前 r 個控制輸入，其餘的數值則不使用，在下個時序中再計算新的控制序列。

為了要進行上述的計算，必須先給定在向量 $\tilde{y}_s(k)$ 中指定的輸出。如果前 r 列之指定輸出為零，則

$$u(k) = \text{前 } r \text{ 列 } \{-[T^T T + \Lambda]^{-1} T^T\} \Theta v_p(k-p) \quad (6)$$

當 Λ 為零矩陣，因為非極小相系統 (non-minimum systems) 其矩陣 T 為秩缺 (rank deficient)，會造成不穩定之閉迴路系統。因此須仔細調整 Λ 的值以保證系統穩定。

模糊邏輯應用

理論上，當 Λ 的數值愈小，則導致較大的控制幅度 (control magnitude)，因此 GPC 的工作性能會愈好。然而控制幅度有其上限，以致性能與控制幅度兩者必須折衷選擇，其中的決定因素即在 Λ 。為了簡化問題，通常是假設加權矩陣 Λ 為一單位矩陣 (identity matrix) 乘以實數 λ 。

本研究發展模糊邏輯規則以調整此 λ 值以得到適當的性能與控制幅度。讀者欲深入瞭解模糊邏輯可參考文獻二。

控制加權調整

在本研究中，模糊規則的前件部(antecedent part)有兩個輸入，即性能指標(performance index, PI)與控制指標(control index, CI)，分別定義如下：

$$PI = \frac{\text{RMS}[y^T(k)y(k)]}{\text{RMS}[y_{op}^T(k)y_{op}(k)]} \quad (7)$$

$$CI = \text{Max}_k |u(k)| / u_{\max} \quad (8)$$

其中 $y(k)$ = 控制中的系統輸出， $y_{op}(k)$ = 無控制下的系統輸出， $u(k)$ = 控制力， u_{\max} = 最大可容許的控制幅度， k = 時序，RMS 意指方均根 (root-mean-square)。

圖 1 為語言變數 (linguist variables) 的歸屬函數，表 1 為規則矩陣。輸出之語言變數，即為本研究中之增殖因數 (multiplication factor)，有十個語言值 (linguist value)，分別為 1 到 10 示於圖 1。圖中之數字 7 表示輸出為零動作。在本研究中增殖因數值為 2 的次方值。以下用一範例來說明如何應用此模糊邏輯。在本文的範例中，為簡化模糊推論中觸發的規則數目，除特別說明外，選擇之數值皆使其語言值之歸屬度為 1，故文中如僅述語言值，即表示其歸屬度為 1。假設由式 (7) 與 (8) 計算得 $PI = 0.5$ ， $CI = 0.8$ ，由圖 1 知 PI 的語言值為 A，CI 為 A。故從表 1 中觸發 $PI = A$ 及 $CI = A$ 的規則，其強度為 1，其餘規則強度皆為 0，此規則之輸出語言值為 6。從圖 1 之輸

出變數歸屬函數知，輸出值為 -0.5。最後的控制加權 λ 之增殖數為 $2^x = 0.70711$ ，其中之 $x = -0.5$ 。此處使用 2 來調整 λ 值，在其它的使用時，可使用其它的整數值。控制加權的遞歸 (recursive) 更新式為

$$\lambda_{\text{new}} = 2^x \lambda_{\text{old}} \quad (9)$$

現考慮當 $PI = 0.4$ ，則從圖 1 知，語言值為 G 與 A 的歸屬度皆為 0.5，如 $CI = 1$ ，從圖 1 知語言值為 U。因此從表 1，觸發兩條規則，即 (1) PI 為 G，CI 為 U 及 (2) PI 為 A，CI 為 U，其對應之輸出語言值分別為 9 與 8，強度為 0.5。從圖 1 可知解模糊化後之輸出值為 0.75，控制加權的增殖數值為 $2^{0.75} = 1.6818$ 。在此設計中， x 的值為負，表示增值數殖小於 1，即降低 λ 值， x 的值為正，表示增殖數殖大於 1，即增加 λ 值。模糊規則簡述如下：

- (1) 假如控制器性能為佳 (G)，且控制幅度接近於最大量值，則增加 λ 值，即使用語言值 8~10。
- (2) 假如控制器性能並不令人滿意 (U) 且控制幅還有增加的空間，則降低 λ 值，即使用語言值 1~6。
- (3) 假如 PI 與 CI 皆屬滿意，或控制幅度設有增加空間，則維持 λ 值不變，即使用語言值 7。

本研究中所定義的模糊變數論域 (universal of discourse) 未必適用於其它系統。但是這問題可藉由一方法來解決，在訊號模糊化前先乘以一數值，以致模糊規則可以適當地應用。此應用說明如下：

按照圖 1 當 PI 值為 0.3，語言值為 G (good)。但在某特定系統，0.3 已屬語言值 E (excellent) 的程度，則可將 PI 訊號乘以 1/3，使 PI 值轉變為 0.1，則經模糊化後其語言值為 E。反之，假如 0.3 僅屬語言值 A (acceptable) 的程度，則此訊號乘上 5/3，使變成 0.5，在模糊化後，其語言值為 A。用以上的修正法，則本研究定義的模糊規則可應用於其它不同的系統。

調整控制加權的計算程序可條列如下：

用一預先設定的輸入序列 $u(k)$ 執行開路實驗以取得開路之輸出響應 $y_{op}(k)$ ，其中 $k = 1, 2, \dots, l$ ，注意此系統的動態響應可能也被未知干擾激發。

1. 設定一起始控制加權值，例如 $\lambda = 1$ 。
2. 執行系統鑑別以計算式(1)中之 T 及 Θ 。
3. 由式 (6) 計算控制量。
4. 系統控制輸入由式 (6) 決定，形成一封閉迴路。
5. 由式 (7) 與 (8) 分別計算性能指標 (PI) 與控制指標 (CI)。
6. 應用此處描述之模糊邏輯更新控制加權。
7. 重複執行步驟 2 至 6 直至加權值 λ 收斂。

ARX 模式階次調整

本研究亦發展模糊規則以調整 ARX 模式之階次。理論上，模式之階次愈高，模式輸出之預測誤差愈小。然而當模式階次超過某個數目，則模式準確度僅會有些微改進但需耗費可觀的計算時間，因此在線上 (on-line) 使用時會限制到取樣速率。因此必須採用一適當之模式階次以平衡計算時間。

在本研究中考慮三個變數以決定採取的措施。此三個變數為正規化之輸出預測誤差 (PE)，PE 之改變量 (CPE) 與正規化之模式階次 (ORD)，分別定義如下：

$$PE = \frac{RMS(\|y(k) - \hat{y}(k)\|)}{RMS(\|y(k)\|)} \quad (10)$$

$$CPE = \text{sign}(N(k) - N(k-1)) \times [PE(k-1) - PE(k)] \quad (11)$$

$$ORD = N(k) / N_{\max} \quad (12)$$

其中 $y(k)$ 與 $\hat{y}(k)$ 分別為輸出與預測輸出， N 和 N_{\max} 為模式階次與最大模式階次， k 為時間步階。式(11)中包含 $\text{sign}(N(k) - N(k-1))$ 項的目的是讓 CPE 能顯示出模式階次改變的方向，如果 CPE 為正，則模式階次改變為正；反之則倒轉改變的方向。

為簡化模糊規則的結構，故採用一層級 (hierarchical) 的結構，第一層的輸入語言變數為 PE 與 CPE，而輸出語言變數為階次乘數 (order multiplier)，在本文中稱為 OUT1。第二層的輸入變數為 ORD 與 OUT1，而輸出語言變數為一乘數，稱為 OUT2。OUT2 用來調整第一層之輸出。

第一層模糊變數的歸屬函數示於圖 2，且規則矩陣亦於表 2。這些規則之設計原則如下：

- (1) 如 CPE 為負，則減少模式階次，即 $OUT1 < 0$
- (2) 如 CPE 為正，則增加模式階次，即 $OUT1 > 0$
- (3) 如 CPE 接近零，則維持同一階次，即 $OUT1 = 0$

第二層模糊變數的歸屬函數與規則矩陣分別示於圖 3 與表 3。這些規則的主要目的是調整第一層輸出值。由於受限於計算機的計算速率，模式階次有一最大的上限，當模式階次接近最大的許可值時，則階次的乘數應予降低；反之則維持原先之乘數。最後之階次乘數為 $OUT2 \times (1 + OUT1)$ 。

下面以一簡單範例說明其應用。如 $PE = 0.35$ ，由圖 2 知其語言值為 B， $CPE = 3 \times 10^{-3}$ ，即 CPE 為 PS。CPE 值為正有兩種可能（1）模式階次增加引致預測誤差減少，或（2）階次減少造成誤差增加。不論屬於那種情況，模式階次都應予以增加。按照表 2，OUT1 的語言值為 4，從圖 2 可得 OUT1 之值為 0.4。在第二層規則時，由圖 3 得 $OUT1 = 0.4$ 對應的語言值為 3，假如此時 $ORD = 0.4$ ，由圖 3 知 ORD 之語言值為 3。由表 3 知 OUT2 的語言值為 4，按照圖 3 解模糊化後其值為 0.8。最後之乘數為 $0.8 \times (1 + 0.4) = 1.12$ ，即在下一步驟中增加模式階段。

現考慮另一情況，當 $PE = 0.1$ ，即 PE 為語言值 G，且 $CPE = -3 \times 10^{-3}$ ，即 PE 為 NS。CPE 值為負，此有兩種可能性：（1）模式階次增加造成預測誤差變劣，或（2）模式階次減少造成誤差減小。因此必須降低模式階次。從表 2，OUT1 為 2，故從圖 2 可知其值為 0。在第二層規則時，從圖 3 中 $OUT1 = 0$ 對應之語言值為 1，如 $ORD = 0.6$ ，即語言值為 4，則從表 3，OUT2 的語言值為 5，經解模糊化後，其值為 0.9。最後之乘數為 $0.9 \times (1 + 0) = 0.9$ ，即在一步驟時降低模式階次。

如 $CPE = 0$ 且 PE 為任意值，則 OUT1 之語言值為 2，在第二層規則時 OUT1 之語言值為 1。假如 ORD 為 1，2，或 3，則 OUT2 為 6，其

數值接近 1。則最終之乘數為 $1 \times (1 + 0) = 1$ ，即保持階次不變。

四、基因演算法與模擬退火應用

在 GPC 中的設計變數如控制加權必須在線上操作前決定。在前述的模糊邏輯中，控制加權可調整至一適當值，此適當值隨系統不同而不同。本節將介紹基因演算法與模擬退火這兩種方法來搜尋在定義的目標函數下之最佳設計變數值。由於這兩種方法皆為不需微分値之最佳化方法，故非常適合本研究所具之特質。

在調整器 (regulator) 的問題中，可定義目標函數為兩項的和，一項為一時段內間輸出偏離輸出設定點的和，另一項為控制器穩定性的處罰項。在最佳化搜尋過程中，兩種方法都需要大量的目標函數評估，這些評估亦可能包含非常差的性能。在實際操作上，無法允許以線上實驗系統來產生此目標函數值。研究結果觀察到，先鑑別一高階模式，再適當的降低模式階次並保留住動態特性，如此產生的模式可代表真實的系統來產生動態響應。

本文將簡單介紹基因演算法與模擬退火，至於詳細內容再參閱參考文獻 3。

基因演算法

基因演算法係將解答空間內的每一點編碼成二進位之字串，此字串稱為染色體。每一個點按定義的目標函數計算得一適當性值 (fitness value)。在每世代 (generation) 中，由一群點組成，稱為族群 (population)。為產生新的世代，基因演算法使用交換 (crossover) 和突變 (mutation) 的方法來產生新的族群。具較高適當性值的個體有較大的機會存活和參

予配對(mating)。經過數代演進後，族群中的個體皆有較好的適當性值。過程陳述如下：

步驟 1：起始

- (a) 在解答空間內以亂數產生起始族群
- (b) 評估每一個體之適當性值

步驟 2：精英主義 (elitism)：保存數個最佳個體至下一代

步驟 3：配對：

- (a) 由族群中按照適當性值選取兩個個體
- (b) 按照交換率(crossover rate)進行交換
- (c) 按照突變率進行突變
- (d) 重複 (a) 至 (c) 直至產生足夠之下一代個體數

步驟 4：重複步驟 2 至 3 直至滿足停止條件。

模擬退火

模擬退火的原則是類似於金屬在控制的溫度速率下冷卻。目前此法廣泛地應用在解決最佳化問題。解答點 (solution points) 在不同的溫度下評估是否為可接受，在搜尋過程中這溫度持續地降低。在高溫時可能接受跳動較遠的解答點。在搜尋過程中，新的點其目標函數值可能比舊的點差，但新的點能否接受與可接受函數有關。可接受函數定義為

$$h(\Delta E, T) = \frac{1}{1 + \exp(\Delta E / (cT))} \quad (13)$$

式中 C 是一與系統相關的常數， T 為溫度， ΔE 是 x_{new} 與 x 的能量差。

$$\Delta E = f(x_{new}) - f(x) \quad (14)$$

式中 f 是系統相關的目標函數。當溫度降至低溫時，此法將只評估在附近變動的一些解答點。搜尋過程如下：

步驟 1：選擇一起始點 x ，且設定一高溫 T 。
評估目標函數 $f(x)$ 值

步驟 2：設定新解答點 $x_{new} = x + \Delta x$ ，其中 Δx 由亂數產生

步驟 3：計算新目標函數值 $f(x_{new})$

步驟 4：(a) 計算可容許函數 $h(\Delta E, T)$

(b) 產生一亂數機率 p

(c) 假如 $h(\Delta E, T) > p$ ，則設定 $x = x_{new}$ 和 $f(x) = f(x_{new})$

重複 (a) 至 (c) 若干次

步驟 5：按照退火程序降低溫度 T

步驟 6：重複步驟 2 至 5 直至停止條件滿足

數值範例

此處用一簡化之機械系統來說明本文所述之方法，對此多輸入多輸出系統，將討論數種情況下之結果。

下面為一九度空間之彈簧-質量- 阻尼(damper) 的系統

$$M\ddot{w} + \Xi\dot{w} + Kw = u$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_8 \\ w_9 \end{bmatrix}; \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_8 \\ u_9 \end{bmatrix}$$

$$M = \begin{bmatrix} m_1 & 0 & \cdots & 0 & 0 \\ 0 & m_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & m_8 & 0 \\ 0 & 0 & \cdots & 0 & m_9 \end{bmatrix}$$

$$\Xi = \begin{bmatrix} \zeta_1 + \zeta_2 & -\zeta_2 & \cdots & 0 & 0 \\ -\zeta_2 & \zeta_2 + \zeta_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \zeta_8 + \zeta_9 & -\zeta_9 \\ 0 & 0 & \cdots & -\zeta_9 & \zeta_9 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & \cdots & 0 & 0 \\ -k_2 & k_2 + k_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & k_8 + k_9 & -k_9 \\ 0 & 0 & \cdots & -k_9 & k_9 \end{bmatrix}$$

式中 m_i, ζ_i 與 k_i 分別為質量，阻尼係數與硬度。這系統的等效狀態空間代表 (state-space representation) 階次為 18。施加在每一個質量 m_i 的控制力量 (control force) 或干擾力量 (disturbance force) 以 u_i 表示。 w_i 為各質量偏離平衡點的位置。在以下的模擬中

$$m_1 = m_2 = \cdots = m_9 = 1 \text{ 公斤}$$

$$k_1 = k_2 = \cdots = k_9 = 1000 \text{ 牛頓/米,}$$

$$\zeta_1 = \zeta_2 = \cdots = \zeta_9 = 1 \text{ 牛頓-秒/米。取樣頻率為}$$

40 赫茲，測量物理量為各質量之加速度，即

$$y_i = d^2 w_i / dt^2 \quad i = 1, 2, \cdots, 9。$$

考慮兩個控制輸入，單干擾輸入與三輸出的狀況，系統的控制輸入為施於質量 3 與 9 (即 u_3, u_9) 的力 (force)，干擾輸入係施加於質量 2 (即 u_2)，

而輸出為第 4，5，與 8 質量的加速度。因此，ARX 模式的最小階次為 18。需提醒的是此為非假設干擾力的函數為 $u_2 = 2\cos(2\pi ft)$ 其中 $f = 6$ 赫茲。施加隨機訊號 u_2 以及獨立隨機訊號 u_3 與 u_9 於系統，以進行開迴路之模擬。兩個隨機訊號皆為正規化分佈。共收集 500 筆數據以供系統鑑別及起始控制器設計用。為模擬真實系統的狀況，將三個輸出都加上測量噪訊，以致訊號噪訊此為 100。加上的噪訊亦為隨機正規化分佈。

模糊邏輯應用

表四為應用開迴路之數據及模糊邏輯進行模式調整的結果，所有的鑑別模式皆使用雙控制輸入與三測量輸出，但不包含干擾。在此處假設干擾為未知。由理論知在週期性輸入干擾下，適當的系統鑑別方法能找出正確的輸入-輸出對應。使用這 500 筆數據計算 PE 與 CPE。因為第一次遞迴運算前並無預測誤差，因此此時並無 CPE 值。仔細檢查表四可發現增加模式階次，確可改進預測準確性。然而當模式階次高於 18 後，預測誤差改變量，即 CPE，減少量小於 1%。因此模式階次為 18 顯然是 GPC 控制器設計的合理選擇。ARX 模式階次為 18 相對於狀態空間代表時的模式大小 (model size) 為 54 (亦即 18 乘上輸出數目)。

表五為應用模糊規則調整控制加權 λ 的結果。任意選擇起始加權為 1。表五顯示約經 12 次運算，控制加權即收斂至常數，而 PI 僅需 9 次運算即收斂，最大控制力設為 15 牛頓。表五顯示控制指標 (CI) 在閉迴路控制模擬中回授控制力量從未超過最大容許力的 80%。表 1 中調整 λ 的規則可以加以修改以調整最大容許 CI 至一設定之

值。每一閉迴路模擬使用方程式 (6) 中之控制力量，其中之係數矩陣由輸入－輸出數據決定和前一步驟決定之控制加權，以及輸入－輸出數據。在每一步驟中皆收集 500 個數據點以計算新的 PI 與 CI 值。注意在此閉迴路模擬的目標函數係減少由未知干擾輸入所造成的響應。

在使用模糊邏輯時，其中的規則係由經驗所建立，而且加權 λ 必須給定一個起始值。即使所得到的 GPC 控制器與系統構成的閉迴路系統為穩定，但 GPC 控制器本身可能為不穩定。理論上，穩定的控制器能造成強健 (robust) 閉迴路系統。由於控制加權與 GPC 控制器穩定性間的直接關係仍屬未知，因此建立規則以保證 GPC 控制器的穩定性是非常困難。

基因演算法與模擬退火應用

對同一範例，設定基因演算法的目標函數如下式

$$J(\lambda, n) = \frac{1}{PI} = \frac{\text{RMS}[y_{opt}^T(k)y_{op}(k)]}{\text{RMS}[y^T(k)y(k)]}$$

最佳化搜尋是最大化 J 且有限制 (constraint) 為 GPC 控制器必須穩定。最佳化變數為控制加權 λ 與 ARX 階次 n 。對不穩定的 GPC 控制器，則設定 J 值為負，以強制此個體，即該組控制加權與階次不進入配對。經 200 次遞迴運算，最佳之變數值為

控制加權： $\lambda = 0.00169$

ARX 階次： $n = 16$

性能指標： $PI = 27.1\%$

對每一組新的變數，都執行閉迴路模擬以收集 500 組數據以計算目標函數。

在本範例中基因演算法所使用的一些參數為
世代的數目 = 10

族群大小 = 20

交換率 = 1.0

突變率 = 0.1

輸入變數的位元數 = 16

加權的數值範圍 = $10^{-3} \leq \lambda \leq 10$

ARX 階次的範圍 = $5 \leq n \leq 30$.

在沒有測量噪訊與週期性干擾的情況下，6 階的 ARX 模式已足以描述一 18 自由度與 3 測量輸出的系統。帶有噪訊與干擾的情況，則需較高的模式階次，以適應系統模態 (mode) 與計劃模態 (Computational modes)，此處執行多個模擬計算，由於每一個模擬中，起始參數為亂數設定，因此最佳結果會有些不同，然而，結果卻相差不多。特別的是 PI 值皆在 27% 與 28% 之間。

模擬退火的目標函數為

$$J(\lambda, n) = PI = \frac{\text{RMS}[y^T(k)y(k)]}{\text{RMS}[y_{opt}^T(k)y_{opt}(k)]}$$

此式與應用模糊邏輯的式 (7) 相同，亦有與基因演算法相同的限制條件，即 GPC 控制器必須穩定。因為模擬退火係搜尋最小值，故需指定一大的正值予 J ，以處理違反限制條件時的情況。經 446 次遞迴運算後，最佳化結果為

控制加權 1： $\lambda_1 = 0.00256$

控制加權 2： $\lambda_2 = 0.00037$

性能指標： $PI = 25.9\%$

其中 λ_1 與 λ_2 分別為控制分量 u_3 與 u_9 所用。 u_9 的加權比 u_3 小一等級 (order)。表示需要較大的控制力量應用於質量 9。此範例之性能比其它範例稍佳。

結語

本研究發展三種方法以決定 GPC 控制器的主要設計變數。此設計變數包括系統／控制器的階次與控制量的加權。應用模糊邏輯，基因演算法與模擬退火來取得起始與細調之設計變數。本研究的主要貢獻包括：導入模糊邏輯所需的適當性能指標與模糊規則，以及基因演算法與模擬退火所需的目標函數。

數值範例的結果顯示模糊邏輯依性能評估則較平滑且快速收斂，收斂過程中新的結果都比前次的結果為佳。至於在基因演算法與模擬退火中所用的變數，則歷經鉅幅振盪與多次遞迴運算才收斂至最終結果。同樣的，其目標函數值也是上下跳動。如果加權值小，則出現很差的性能；例如不穩定的閉迴路響應。由結果很清楚的顯示作即時適應控制時，模糊邏輯比基因演算法和模擬退火適合。但在設計變數值已確定在某小範圍時，後二種方法亦可能以即時／線上使用。模糊邏輯需要專家來建立規則矩陣以及歸屬函數，此外，它並不保證回授控制器的穩定性。例如輸入／輸出間的對應能由輸入與輸出的數據中鑑別出來，而且能代表系統的動態。另外亦能估計出一模式以描述未知的外部干擾，則可用基因演算法與模擬退火以計算出一組最佳之設計變數作為起始值，再由模糊邏輯接手去執行即時／線上適應控制。此處用基因演算法與模擬退火的特徵是保證控制器的穩定性。

參考文獻

1. Juang, Jer-Nan, and Eure, Kenneth W., "Predictive Feedback and Feedforward Control for Systems with Unknown Disturbances," NASA/TM-1998-208744, December 1998.
2. Tsoukalas, Lefteri H., and Uhrig, Robert E., Fuzzy and Neural Approaches in Engineering (Adaptive and Learning Systems for Signal Processing, Communications and Control Series, John Wiley & Sons; ISBN: 0471160032, 1997
3. Jang, Jyh-Shing Roger, Sun, Chuen-Tsai, and Mizutani, Eiji, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall; ISBN: 0132610663, 1996, pp. 175-186

表 1 調整 λ 之規則矩陣

| | | CI | | | | |
|----|---|----|---|---|---|----|
| | | E | G | A | B | U |
| PI | E | 6 | 7 | 7 | 8 | 10 |
| | G | 5 | 6 | 7 | 7 | 9 |
| | A | 5 | 5 | 6 | 7 | 8 |
| | B | 4 | 5 | 5 | 6 | 7 |
| | U | 1 | 1 | 3 | 5 | 7 |

表 2 階次乘數之規則矩陣

| | | CPE | | | | |
|----|---|-----|----|----|----|----|
| | | NM | NS | ZO | PS | PM |
| PE | E | 2 | 2 | 2 | 3 | 3 |
| | G | 1 | 2 | 2 | 3 | 3 |
| | A | 1 | 1 | 2 | 3 | 4 |
| | B | 1 | 1 | 2 | 4 | 5 |
| | U | 1 | 1 | 2 | 4 | 6 |

表 3 調整階次乘數之規則矩陣

| | | ORD | | | | | |
|-----|----|-----|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| OUT | 1 | 6 | 6 | 6 | 5 | 4 | 3 |
| | 2 | 6 | 6 | 5 | 4 | 3 | 2 |
| | 13 | 6 | 6 | 4 | 4 | 3 | 1 |
| | 4 | 6 | 6 | 4 | 3 | 2 | 1 |
| | 5 | 6 | 5 | 3 | 2 | 1 | 1 |

表 4 模式階次調整之模糊邏輯應用

| | Model Order | Prediction Error (PE %) | Change in PE (CPE %) |
|----|-------------|-------------------------|----------------------|
| 1 | 6 | 15.08 | ? |
| 2 | 9 | 9.93 | 5.14 |
| 3 | 12 | 8.22 | 1.71 |
| 4 | 15 | 5.05 | 3.17 |
| 5 | 18 | 3.72 | 1.33 |
| 6 | 21 | 2.86 | 0.86 |
| 7 | 25 | 2.67 | 0.19 |
| 8 | 29 | 2.50 | 0.17 |
| 9 | 33 | 2.34 | 0.17 |
| 10 | 37 | 2.22 | 0.12 |

表 5 控制加權調整之模糊邏輯應用

| | Control Weighting | Performance Index (PI %) | Control Index (CI %) |
|----|-------------------|--------------------------|----------------------|
| 1 | 1 | 62.47 | 27.48 |
| 2 | 0.4061 | 48.02 | 32.89 |
| 3 | 0.2030 | 39.69 | 36.86 |
| 4 | 0.1015 | 33.22 | 40.64 |
| 5 | 0.0508 | 30.07 | 44.63 |
| 6 | 0.0254 | 27.99 | 51.84 |
| 7 | 0.0133 | 26.55 | 61.46 |
| 8 | 0.0078 | 25.26 | 68.84 |
| 9 | 0.0057 | 25.25 | 73.06 |
| 10 | 0.0045 | 24.99 | 77.28 |
| 11 | 0.0041 | 24.88 | 79.51 |
| 12 | 0.0040 | 24.86 | 80.06 |
| 13 | 0.0040 | 24.86 | 80.03 |
| 14 | 0.0040 | 24.86 | 80.00 |
| 15 | 0.0040 | 24.86 | 80.00 |

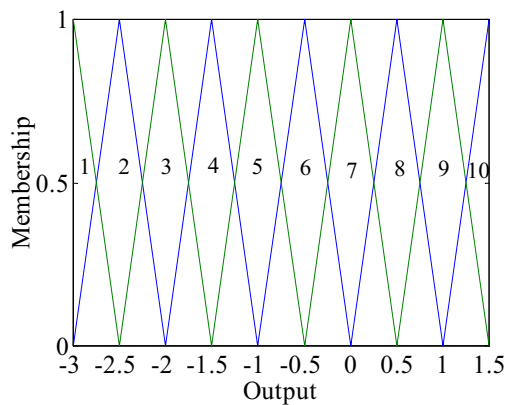
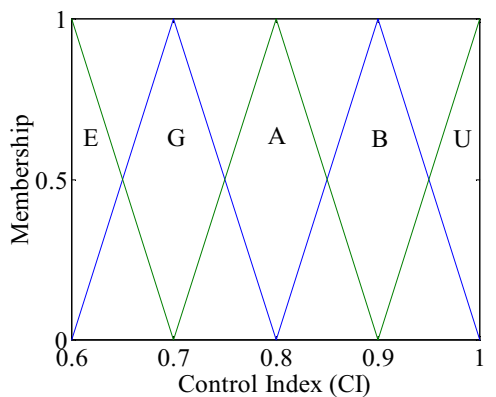
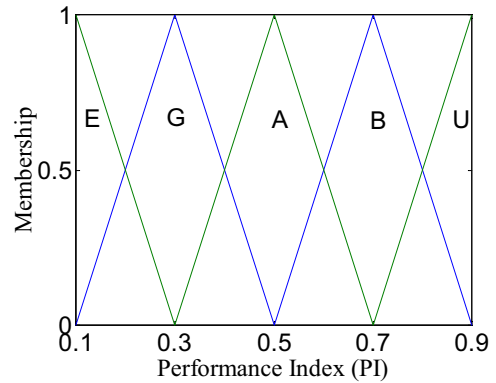


圖 1 調整 λ 之歸屬函數

(E : Excellent G : Good A : Acceptable B : Bad U : Unacceptable)

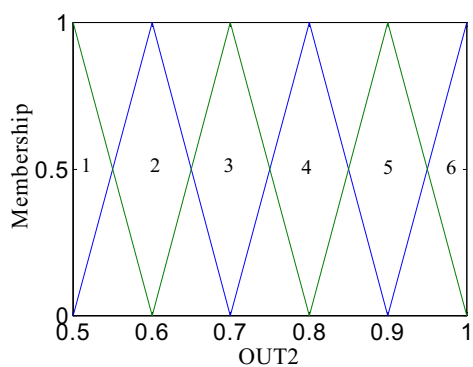
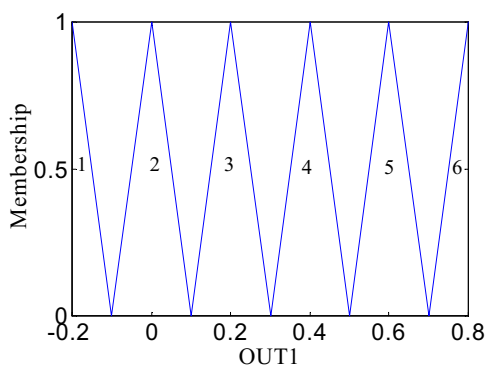
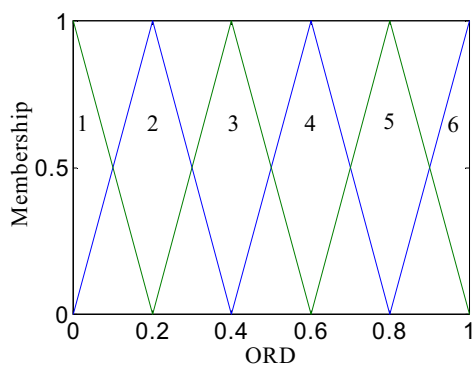
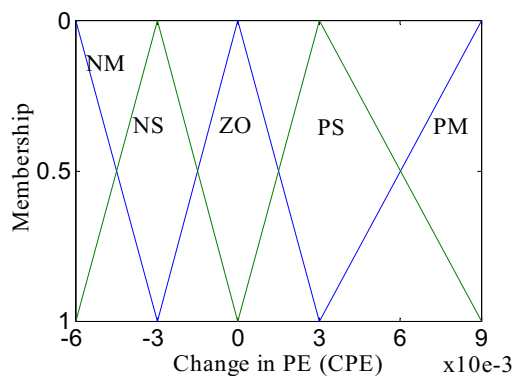
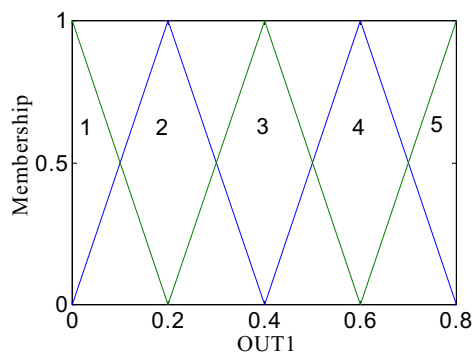
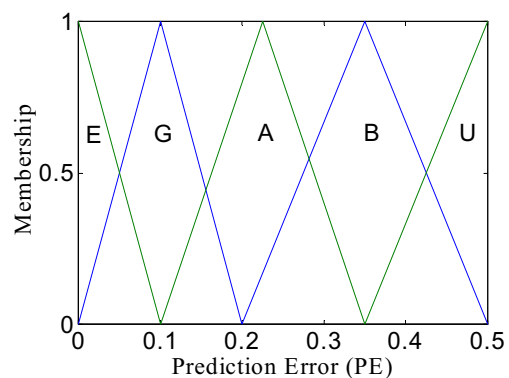


圖 3 調整階次乘數之歸屬函數

圖 2 階次乘數之歸屬函數

(E : Excellent G : Good A : Acceptable
B : Bad U : Unacceptable N : Negative
P : Positive S : Small \M : Medium \ZO :
Zero)