

Genetic Fuzzy PID Controller Design and Implementation of DC Servo Motor

直流伺服馬達基因模糊PID控制設計與實現

姚賀騰、謝錦聰
國立勤益科技大學 電機工程系
htyau@ncut.edu.tw

蘇芺鏞
遠東科技大學 電機工程系
evang1204@yahoo.com.tw

摘要

本論文針對直流伺服馬達位置迴授控制系統，選取 IAE (誤差絕對值的積分) 作為適應性函數，利用基因演算法(GA)搜尋 PID 控制器之最佳增益常數值。在此 PID 增益常數值下，結合模糊邏輯控制器(FLC)對增益常數值做微調，可以得到比單獨只用 PID 控制更好的控制響應。最後本論文利用 Quanser 公司所提供的直流伺服馬達為實驗平台，將電腦模擬之結果透過實驗驗證比較，證明了有加模糊邏輯之模糊邏輯 PID 控制器可以讓馬達更快準確定位到想要的角度位置。

關鍵詞：直流伺服馬達，模糊邏輯控制器，基因演算法，比例-積分-微分控制器。

1. 前言

現今科技發達，許多機械系統都需要動力才可以完成動作，而馬達為最廣泛應用的動力來源，況且目前的產業發展還是偏重於馬達的動力為主。馬達的種類繁多，以精密機械工業而言，伺服馬達是主要動力來源，主要原因為伺服馬達的高速度與高精準度。

控制方法就工業界而言，大部分仍然使用 PID(Proportional-Integral-Differential)控制器，其特色為構造簡單、容易實現與維修方便，對大多數的種系統都有良好的性能表現。PID 控制器雖然是簡便的控制器，但是 PID 之參數值卻是一個大問題，於系統中調整是最麻煩的一件事，需耗費大量的人力與時間測試最佳參數值。設計最佳的參數，在學術界方面而言，就有許多學者提出方法與理論，演算法就是其中一種，演算法是計算參數最常使用方法，以基因演算法(Genetic Algorithm, GA)、進化演算法(Evolutionary Algorithm, EA)[4]與進化規劃法(Evolutionary Programming, EP)為最常被提出的演算法。若不考慮使用 PID 控制器，以其他控制器代替而言，FLC(Fuzzy Logic Controller)與 SMC(Sliding Mode Controller)是最常被提出討論的控制方法。文獻[1]以直流伺服馬達為受控體，而控制器使用 PID 控制器，以進化規劃法計算 PID 之參數值，並控制伺服馬達。

在 1989 年，H. Nakamura 與 N. Shimozono 這兩位學者於「Industrial Electronics Society」中發表在直流伺服馬達上應用預測學習控制的方法，利用此方法可以快速收斂且誤差減少至 ± 1 的脈波內，但是必須經過第四次的動作過後才会有這樣的結果，也就是四個週期後才會出現理想的狀態[5]。於 2002 年，葉博仁提出利用個人電腦為基礎介面(PC Based)，設計一個 PID 控制器，透過並列傳輸的方式控制直流伺服馬達，使馬達的運轉能夠更快達到穩定的效果，但是在穩定的狀態中，其穩態誤差表現並不是非常理想，還是會有顫震的現象產生[2]。於 2005 年的「技術學刊，第二十卷，第二期」中，洪美玲與顏錦柱等幾位學者，發表了「直流馬達定位控制-最佳 PID 控制器之設計」，為了提高運算的效率並且快速的搜尋出最佳的 PID 參數值，採用進化規劃法搜尋，其搜尋的 PID 參數值有令人滿意的結果，但是演算的過程中所花費的時間是非常久的[1]。

本論文是利用個人電腦為基礎介面，於 PID 控制器中，運用了基因演算法搜尋最佳參數值，並利用 FLC 克服暫震現象使直流伺服馬達的定位控制更快達到理想的目標。本論文架構如下：在第二節中針對直流伺服馬達作簡單的描述；第三節則介紹基因演算法與模糊邏輯控制器的原理；第四節為簡單介紹硬體與系統架構；第五節為設計與實作的分析；最後一節為結論。

2. 直流伺服馬達

考慮控制對象直流伺服馬達之轉移函數，架構圖如圖 1 所示，由 Kirchhoff 電壓定律，可得到如下：

$$V_m - R_m I_m - L_m \frac{dI_m}{dt} - E_{emf} = 0 \quad (1)$$

由於 $L_m \ll R_m$ ，因此可以忽略電樞電感可得電樞電流如下式表示：

$$I_m = \frac{V_m - E_{emf}}{R_m} \quad (2)$$

且馬達的反電動勢電壓 E_{emf} 與馬達軸心速度 ω_m 成正比，所以可以表示如下：

$$I_m = \frac{V_m - K_m \omega_m}{R_m} \quad (3)$$

考慮馬達的機械能並在馬達軸心處應用牛頓第二運動定律可得：

$$J_m \dot{\omega}_m = T_m - \frac{T_l}{\eta_g K_g} \quad (4)$$

其中 $\frac{T_l}{\eta_g K_g}$ 為負載轉矩上所看見之等效負載，而

η_g 為齒輪箱的效率。再由馬達之負載端觀察並應用牛頓第二定律，可以得到：

$$J_l \dot{\omega}_l = T_l - B_{eq} \omega_l \quad (5)$$

其中 B_{eq} 為輸出阻尼係數。將(4)式代入(5)式，可以得到：

$$J_l \dot{\omega}_l = \eta_g K_g T_m - \eta_g K_g J_m \dot{\omega}_m - B_{eq} \omega_l \quad (6)$$

其中 $\omega_m = K_g \omega_l$ 且 $T_m = \eta_m K_t I_m$ ，因此(6)式可以改寫如下：

$$J_l \dot{\omega}_l + \eta_g K_g^2 J_m \dot{\omega}_l + B_{eq} \omega_l = \eta_g \eta_m K_g K_t I_m \quad (7)$$

最後，將(3)式代入(7)式，並由 Laplace 轉換可得到速度與輸入電壓之間的轉移函數：

$$\frac{\omega_l(s)}{V_m(s)} = \frac{\eta_g \eta_m K_t K_g}{J_{eq} R_m s + B_{eq} R_m + \eta_g \eta_m K_m K_t K_g^2} \quad (8)$$

其中 $J_{eq} = J_l + \eta_g J_m K_g^2$ 。

因為轉速之積分即為位置，故位置與輸入電壓之間的轉移函數如下：

$$\frac{\theta_l(s)}{V_m(s)} = \frac{\eta_g \eta_m K_t K_g}{s(J_{eq} R_m s + B_{eq} R_m + \eta_g \eta_m K_m K_t K_g^2)} \quad (9)$$

其中 θ_l 為直流馬達之轉動角度。

將受控馬達之參數[10]代入(9)式，如下式所示：

$$\frac{\theta_l(s)}{V_m(s)} = \frac{0.33398}{s(0.00542s + 0.18989)} \quad (10)$$

PID 表示式如下：

$$u(t) = K_p + K_d \frac{de(t)}{dt} + K_i \int_0^t e(\tau) d\tau \quad (11)$$

其中 K_p 為比例項， K_d 為微分項， K_i 微積分項。

3. 原理介紹

3.1 基因演算法

基因演算法是解決最佳化的一種方法[6~11]，不會像傳統梯度法(Gradient)或單純法(Simplex)容易陷入局部最佳解，而且它在求解的空間上限制較少，也不需要許多複雜與嚴格的數學公式，而只需要知道適應性函數，然後根據染色體的適應值大小，逐步的找到全域最佳解[3]。基因演算法有六個主要因素，流程如圖 2 所示，分別如下介紹：

3.1.1 初始族群的設定

一開始先定義參數個數，參數的個數被定義為基因，而基因的數量取決於受控體所需要的參數數量，可以是二進位碼(Binary Code)或是其他的編碼方法，而這些參數的總和被定義為染色體(Chromosome)，不同的染色體分別代表不一樣的解[12]。

3.1.2 適應性函數的設計

適應性函數主要是判別染色體的性能指標(Performance Index)，進而決定是否保存此染色體。在基因演算法的演化過程中，每代所產生的族群都會有適應性函數，當適應性函數比上一代族群還高，表示新一代的族群比上一代族群還要優秀，而新一代則群的染色體被保存下來的機會也越高，在經過幾代的演化後，最後保留下來的族群就是適應性函數最佳的，所以適應性函數的設計好壞會直接影響到是否可得到最佳解[12]。我們以輸入與輸出的誤差取絕對值並積分，再由每次的演算中取最大值來當適應值，式子表示如下。

$$IAE = \int_0^{\infty} |e(\tau)| d\tau \quad (12)$$

3.1.3 選擇

選擇是在群體中選擇生命力較強的染色體所產生新的群體的過程。假設染色體的適應值為 $f(i)$ ，群體大小為 a ，則此族群的每個染色體被保存到下一代的機率為：

$$F(i) = \frac{f(i)}{\sum_{i=1}^a f(i)}, \text{ 且 } i = 1 \sim a \quad (13)$$

根據每個染色體的適應值大小選擇，適應值較高的染色體，表示被保存至下一代群體中的機率較大；適應值較低的染色體被保存至下一代群體中的機率較小。此種方法可以使群體中染色體的適應值不斷的接近最優解。

3.1.4 交配

自然界進化的過程中，兩個同樣的染色體可以藉由交配而重組，變成新的染色體，而上述的選擇方法並沒有創造出任何新的染色體，只是單純的保留適應值較高的染色體。但是基因演算法保留了自然界進化過程中基因交換的機制，由交配的過程中，隨機選擇族群中的兩個染色體，交換兩個染色體中的某部位基因，產生新的染色體。

3.1.5 突變

突變的主要目的為改善選擇與交配中無法提供的訊息。由上述可知，選擇主要是保存適應值最佳的染色體，交配是產生新的染色體，若染色體維持在舊一代的基因一直演化，則會過早收斂與陷入局部最佳解的狀況，而突變可以脫離此問題，防止局部最佳解的出現。交配有如從原本的搜索範圍跳至另一個不同的搜索範圍，而突變可加強局部搜索的能力，使得基因演算法能夠得到最佳解。

3.1.6 結束條件

由上述的演算過程中，無法得知停止演算的時間或任何結束的限制，所以需要一個結束的條件使其停止。結束條件的設置主要是防止基因演算法一直演算，而進入無窮迴圈。

3.2 模糊邏輯控制器

一個基本模糊控制器，主要包含四個部分：模糊化介面(Fuzzification Interface, FI)、規則庫(Knowledge Base, KB)、決策邏輯(Decision Making Logic, DML)與解模糊化介面(Defuzzification Interface, DFI)，分別介紹如下：

3.2.1 模糊化介面

模糊化的功能就是將外部的訊號轉化成適當的語意式模糊訊號，也就是將明確數值模糊化成模糊數值。模糊化的過程是依據我們所制定的歸屬函數來轉換，故歸屬函數在模糊時間數列的分析上，佔有很重要的成分。

3.2.2 知識庫

知識庫是由資料庫與規則庫兩者所構成，資料庫主要為制訂相關的必要變數資料，以方便模糊系統判定輸出與輸入變數所定義的模糊集合以及相對應的歸屬函數。規則庫為模糊系統的重心，是用來描述控制目標的控制法則，也是決策邏輯的推論依據。控制規則一般是由專家知識寫成的 IF-THEN 形式的控制規則，並將所有的控制規則整理成一個規則表。

3.2.3 決策邏輯

決策邏輯是整個模糊系統的核心，藉由模糊的推理來模擬人類的思考判斷及決策的模式，因此決策邏輯的任務就是負責將模糊化後的模糊變數，藉由決策邏輯的運算，並且根據規則庫中的模糊規則來進行已決定下一步驟所要採取的控制策略，來達到控制的目的[11]。

3.2.4 解模糊化介面

將決策邏輯所產生的輸出結果，轉換成明確數值訊號的過程，稱為解模糊化[11]。常用的解模糊化方法為重心法(Center of Gravity, COG)主要是尋求物件重心相同的位置，亦即以模糊集合的「中心值」代表整個集合，假設模糊集合A的歸屬函數 $\mu_{C_i}(y_i)$ ， y_i 為第i個模糊規則的明確輸出， y^* 為重心位置，也就是解模糊化輸出，則 y^* 為：

$$y^* = \frac{\sum_i \max_k \mu_{C_i}(y_i) y_i}{\sum_i \max_k \mu_{C_i}(y_i)} \quad (14)$$

4. 硬體架構

本篇論文軟體採用 MATLAB 及 SIMULINK 撰寫程式執行，而硬體示意圖如圖 3 所示，主要由電腦發出訊號給 D/A 卡，透過 D/A 卡轉換成類比訊號傳送給電源驅動馬達 SRV02 轉動，經由感測

器測量並傳送位置訊號給 A/D 卡，再由 A/D 卡轉換成數位訊號迴授給電腦計算誤差，如此的反覆使誤差為零，達到控制目的。

5. 設計與實作分析

以 Matlab 之 Simulink 實現，如圖 4 與圖 5 所示，群組大小為 40 組，性能指標採用 IAE 法，IAE 為誤差絕對值的積分，當 IAE 值越小，表示越接近最佳值，如圖 6 所示，可得知最佳值為 0.38436。突變採用均勻突變，而演算 300 代即停止演算。由運算結果所得出的 PID 參數值為 $[K_p \ K_i \ K_d] = [13.982 \ 0.005 \ 0.002]$ 。

於 PID 控制器前加入 FLC，並使用上述基因演算法所搜尋之 PID 控制器的最佳參數值。模糊化採用的方法為非單點模糊化，圖 7 為歸屬函數示意圖。知識庫則定義 R 為輸入之參考值，Y 為系統輸出值，而模糊控制器之輸入設定為兩個，分別為誤差 e_1 與誤差之變化量 e_2 ，其定義如下：

$$\begin{aligned} e_1 &= R(t) - Y(t) \\ e_2 &= \dot{e}_1 \end{aligned} \quad (15)$$

兩個輸入各有 7 個歸屬函數，因此規則庫共有 $7 \times 7 = 49$ 條規則，而決策邏輯採用 Mamdani 法。

綜合兩者 GA 與 FLC+GA 之實驗結果如圖 8 所示，圖中可得到 GA 由 0 度到 45 度之上升時間為 0.163 秒，FLC+GA 為 0.144 秒。由以上的結果可以明顯看出，於 PID 控制器前加入 FLC 控制會得到比單獨 PID 控制器更好的暫態響應。

6. 結論

本文運用在 Quanser 公司所提供的直流伺服馬達為實驗平台，利用基因演算法(GA)，搜尋 PID 控制器之最佳增益常數值。在基因演算法(GA)之 PID 增益常數值下，結合模糊邏輯控制器(FLC)對增益常數值做微調，加強暫態響應的上升時間。經由實驗結果得知，加上模糊邏輯之模糊邏輯 PID 控制器，可以得到比單獨只用 PID 控制更好的控制響應，使馬達更快速準確定位到想要的角度位置。

參考文獻

- [1] 洪美玲，顏錦柱，賴寵文，邱銘彰，劉修綜，直流馬達定位控制-最佳 PID 控制器之設計，技術學刊，第二十卷，第二期，pp.195-200，2005。
- [2] 葉博仁，應用 PC Based 於直流伺服馬達控制，義守大學電機工程研究所，碩士論文，2002。
- [3] 吳培輔，設計具有基因演算法之模糊比例積分控制器於壓電定位平台，國立東華大學電機工程學系，碩士論文，2009。
- [4] P. Power, F. Sweeney and C. F. N. Cowan, "EA

Crossover Schemes for a MLP Channel Equaliser” , *IEEE*, vol.1, pp.407-410, 1999.

- [5] H. Nakamura, N. Shimozono, “ Predictive learning control and application to servo system of DC motor”, *Industrial Electronics Society*, 1989. IECON '89, 15th Annual Conference of IEEE, vol.2, pp.287-290, 1989.
- [6] Chin-Teng Lin and C. S. George Lee, “Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems”, *Prentice-Hall*, 1996.
- [7] Jyh-Shing Roger Jang, Chuen-Tsai Sun and Eiji Mizutani, “Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence”, *Prentice-Hall*, 1997.
- [8] K. Zeger, J. Vaisey. And A. Gersho, “Globally Optimal Vector Quantizer Design by Stochastic Relaxation”, *IEEE Transactions on Signal Processing*, vol.40, no.2, pp.310-322, 1992.
- [9] Melanie Mitchell, “An Introduction to Genetic Algorithms”, *The MIT Press*, 1996.
- [10] P. Scheunders, “A Genetic C-Means Clustering Algorithm Applied to Color Image Quantization”, *Pattern Recognition*, Vol.30, no.6, pp. 859-866, 1997.
- [11] Wen-Jyi Hwang and Yi-Chou Chen, “Robust Transmission Based on Variable-Rate Error Control and Genetic Programming”, *IEEE Communication Letters*, vol.6, no.1, pp.25-27, 2002.
- [12] Semyon D. Savransky, “Engineering of creativity: introduction to TRIZ methodology of inventive problem solving” , *CRC Press LCC*, 2000.

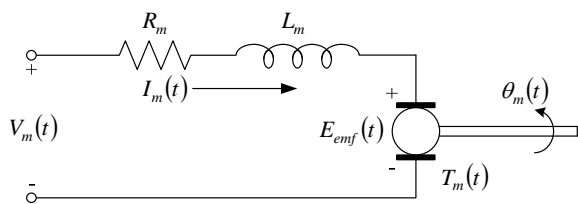


圖 1. 直流馬達架構圖

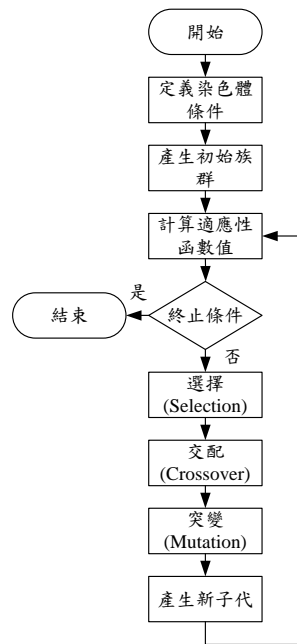


圖 2. 基因演算法流程圖

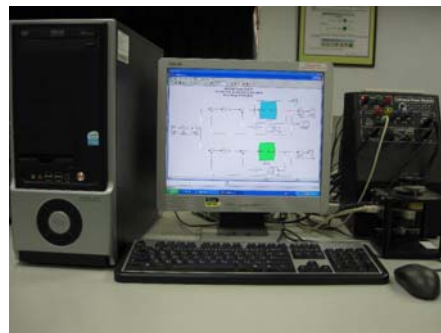


圖 3. 硬體示意圖

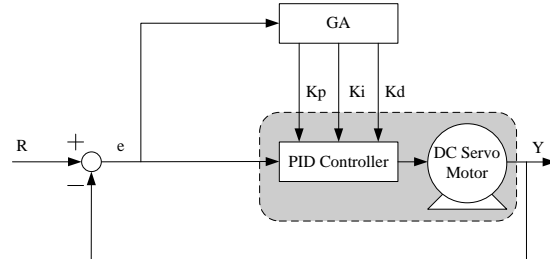


圖 4. GA 基於 PID 控制系統圖

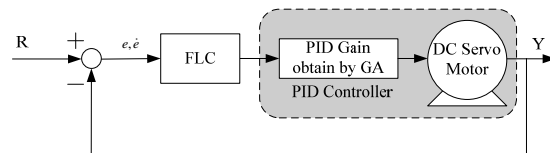


圖 5. FLC+GA 基於 PID 控制系統圖

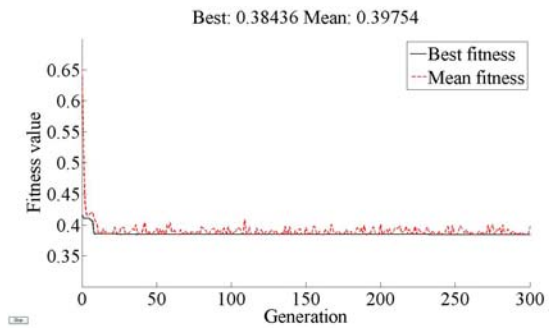


圖 6. IAE 收斂圖

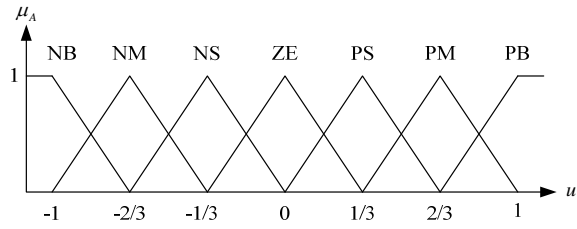


圖 7. 歸屬函數示意圖

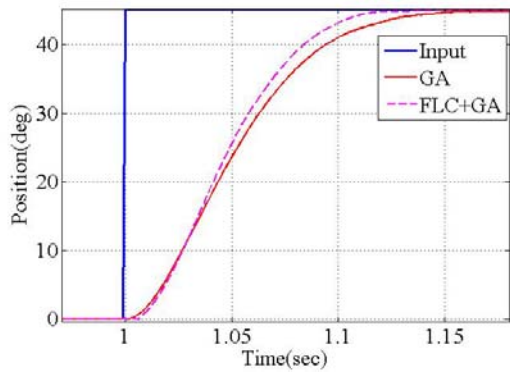


圖 8. GA 與 FLC+GA 比較圖