

行政院國家科學委員會專題研究計畫 成果報告

研製包含大量記憶體之快速運算(含並行處理)CMAC 控制
晶片

計畫類別：個別型計畫

計畫編號：NSC91-2213-E-009-058-

執行期間：91年08月01日至92年07月31日

執行單位：國立交通大學電機與控制工程學系

計畫主持人：陳福川

報告類型：精簡報告

處理方式：本計畫可公開查詢

中華民國 92 年 10 月 1 日

行政院國家科學委員會專題研究計畫成果報告

研製包含大量記憶體之快速運算(含並行處理)CMAC 控制晶片

On fast parallel-processing CMAC control chips which contain large amount of in-chip memory

計畫編號：NSC 91-2213-E-009-058

執行期限：91 年 8 月 1 日至 92 年 7 月 31 日

主持人：陳福川 交通大學電機與控制系

一、中文摘要

本計畫的目的主要可以分為系統分析以及硬體設計兩方面。在系統分析方面，本論文研究 CMAC 類神經網路控制系統的控制行為並分析系統的特性。這裡使用 Runge-Kutta 法來模擬受控系統的輸出/輸入特性。CMAC 控制系統不需要關於機械臂的任何資訊，而且可以應付很大的負載變化。當輸入向量緩慢的移動的時候，CMAC 控制器會產生龐大的積分動作，但是當輸入向量快速變化的時候，CMAC 控制器也能有效的產生“遺忘”效應。論文中也探討了 CMAC 控制系統的種種性質與對應的模擬結果。在硬體設計方面，本論文規劃了 CMAC 類神經網路控制晶片的架構以及用硬體描述語言 Verilog 做更進一步的實現。在架構設計上，本論文設計以內嵌 ARM 處理器的結構來實現 CMAC 類神經網路中的演算法。最後，本計畫提供此 CMAC 類神經網路控制晶片平行處理的解決方案，藉以增加晶片內部單元平行處理的套數並做適當的排程來大幅提升不同應用所可能需求的更高處理速度。

Abstract

There are two main objective of this paper, system analysis and hardware design. In the aspect of system analysis, the behavior of the CMAC control system has been studied and stability of the system has been analyzed in this thesis. The Runge-Kutta method is adapted to find out the characteristic between output and input of plant. However, the CMAC requires no information about the robot, and can deal with large variations in load. The CMAC produces enormous integration action when the input

vector moves slowly in the space, but it can also forget efficiently when the input vector moves fast in the space. It is shown that the CMAC control system can converge into the target position. In the aspect of hardware design, on the architecture design and further implementation in Hardware Description Language Verilog have been focused in this thesis. On architecture design, the ARM- based structure is adapted to implement the CMAC neural network. At last but not least, the solution of parallel processing in the CMAC neural network chip is provided, for higher processing speed in accordance to different application.

二、緣由與目的

今日科學最大的挑戰是了解人類腦部的構造，以及感知、學習、動作和記憶的生物性原理，而問題始自於人類腦部極為複雜的構造以及微妙的設計。大部分已建立的知識是關於單一細胞神經元的生化機制等低階運作方式，以及腦部內各式簡單細胞網路間的互動方式。然而，腦部在整個系統層次的運作細節至今亦是一個待研究的課題。從工程的角度，此一研究課題的目的在於嘗試將迄今對於腦部構造的特殊控制機制的了解，應用在機器人等人工智慧控制體上，而且將專注於腦部中跟運動控制相關的部分構造（CMAC 的基礎概念即源自於人類對小腦支配運動控制上的了解）。此研究論文提出一結合 CMAC 控制器與 PD 控制器的控制架構，設計用以做複雜且精準的運動控制，並將此一改善的演算法配合數位訊號處理中的技術及架構，用最有效的方法映射到硬體架構上，最後設計以 CMAC 改進架構為基礎的晶片系統。

三、結果與討論

CMAC 控制系統在工業應用上具有很大的潛力，其中主要的原因是因為 CMAC 控制系統具有很快的學習能力。論文中描述了 CMAC 類神經網路控制系統的控制行為而且分析系統的穩定性。CMAC 控制系統可以在路徑追蹤的過程中，有效的降低系統的追蹤誤差；而在點對點的移動控制中，CMAC 控制系統可以提供較快的上升速度、較低的超越量及較短的安定時間。這些結果都再度說明 CMAC 控制系統的優勢。根據所使用的計算架構及學習法則，當輸入向量緩慢的移動的時候，CMAC 控制器會產生龐大的積分動作。但是當輸入向量快速變化的時候，CMAC 控制器也能有效的產生“遺忘”效應。論文中也證明 CMAC 控制系統能夠在目標位置收斂，另外，我們也在 CMAC 學習過程中加入了 dead-zone 與 $\hat{\sigma}$ -bound 還消除因為長時間控制屬造成的不穩定。在 SoC (System-on-a-chip) 已經成為既定趨勢的現今，在此計畫中我們選擇嵌入式記憶體來實現 CMAC 類神經網路控制晶片內的記憶單元，同時配合該嵌入式記憶體設計記憶體控制單元，並在 AMBA system 上設計包含此部分以及邏輯電路的設計考量，就 ARM-based IP 設計流程前端的設計經驗而言，將系統映射至晶片的設計這樣的訓練已完善。

四、計畫成果自評

CMAC 控制系統模擬結果

首先是有關不同負載的模擬結果，這裡我們的相關控制參數為步階訊號-30度到 30 度， $K_p = 50$ ， $K_d = 45$ ，不同負載為 $m_L = 10\text{kg}$ and $m_L = 50\text{kg}$ 。

圖 3-1 中我們發現對於較大的負載，其機器手臂控制的曲線仍能類似較小的負載，而這最主要的原因為 CMAC 控制器的輸出 f_c ，對於較大的負載能夠積分出足夠大的值，且能夠保持一段時間使得控制輸出 f 能夠反應迅速，而上述的結果為以下兩個原因所造成。(1).如圖 3-1 所見，機器手臂在對於較大的負載其 CMAC 的輸入向量移動的較為緩慢，故能夠積分出足夠大的值。

(2).如圖 3-3 所見，在剛開始時不同的負載其 PD 控制器的輸出 f_p 都是相同的，但對於較大的負載其 PD 控制器的輸出下降的較慢，故 f_c 能夠積分出足夠大的值。

因為上述兩個原因使得 f_c 能夠反應迅速，且能夠面對不同的負載變化，而積分效應也能夠使系統避免 overshoot，在開始控制後 f_p 迅速下降至零，並產生一個負的 f_p 以有效抑制 f_c 。

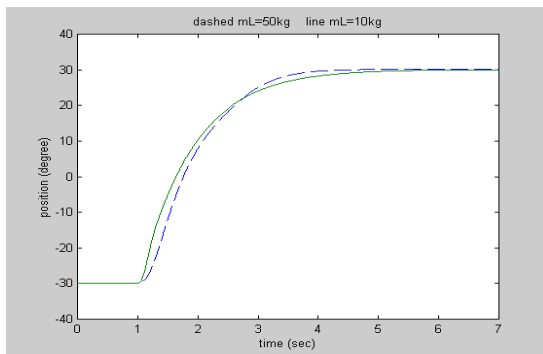


圖 3-1(一). CMAC 控制系統控制結果

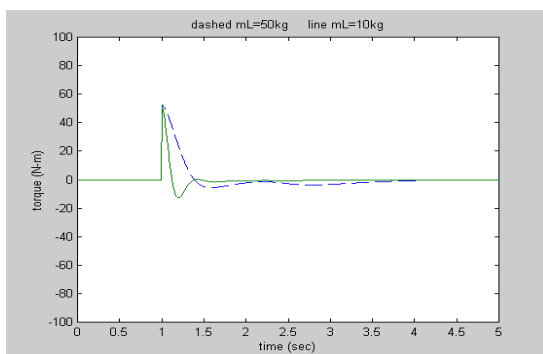


圖 3-3(二). CMAC 控制系統控制結果

接著我們討論在圖 3-4 中在到達控制目標後有關區域穩定(local stability)的問題，在操作點 operating point ($q = \bar{q}, \dot{q} = \bar{\dot{q}}, f = \bar{f}$) 附近，我們的機

器手臂系統 $\left(\frac{m_1}{3} + m_L\right) a_1^2 \ddot{q} + g_0 \left(\frac{m_1}{2} + m_L\right) a_1 \cos(q) + b_1 \dot{q} = \tau$ 可被線性化成

$$\ddot{q} = \frac{g_0 \left(\frac{m_1}{2} + m_L\right) a_1 \sin(\bar{q})}{\left(\frac{m_1}{3} + m_L\right) a_1^2} q - \frac{b_1}{\left(\frac{m_1}{3} + m_L\right) a_1^2} \dot{q} + \frac{1}{\left(\frac{m_1}{3} + m_L\right) a_1^2} \tau \quad \text{在操作點附近,}$$

\ddot{q}, \dot{q}, q and τ 近似於 $\Delta\ddot{q}, \Delta\dot{q}, \Delta q$ and $\Delta\tau$ $\Delta q = q - \bar{q}$ and $\Delta\tau = \tau - \bar{\tau}$ ，在操作點附近
機器手臂停留在 q_d ，因此 $\bar{q} = q_d, \bar{v} = 0, \bar{\tau} = g_0 \left(\frac{m_1}{2} + m_L\right) a_1 \cos(q_d)$

我們可從下式得知當機器手臂停留在 q_d 時，CMAC 輸入向量移動的非常緩慢，
此時控制輸出 τ ($= \tau_c + \tau_p$) 就像是一個 PID 的控制器。

$$\begin{aligned} \tau = \tau_c + \tau_p &= \left(-\frac{K_p}{T} \int q_e dt + \frac{K_d}{T} \int v_e dt \right) + (K_p q_e + K_d v_e) \\ &= \frac{K_p}{T} \int q_e dt + \left(\frac{K_d}{T} + K_p \right) q_e + K_d v_e \end{aligned} \quad \text{因此我們可以總結此非線性}$$

CMAC 控制系統的操作點 ($\bar{q} = q_d, \bar{v} = 0, \bar{\tau} = g_0 \left(\frac{m_1}{2} + m_L\right) a_1 \cos(q_d)$) 是一個穩定的
的狀態點，滿足 PID 穩定的控制法則下。

不過因為上述的穩定假設是近似估計，在經過長時間的控制後，系統中的
非線性參數仍會顯現出來，因此我們針對兩種較為極端的控制函數來討論，並提
出解決不穩定的方法。

我們所使用模擬參數為 $k_p=300, k_d=60, m_l$ (負載)等於 10kg，分 5 層記憶
體，學習參數為 0.3，配置記憶體個數為 512。

dead-zone 效應的模擬結果:

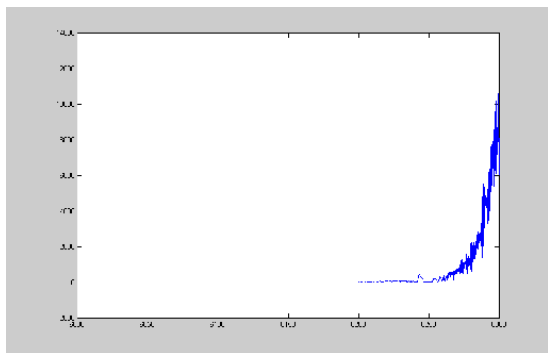


圖 3-7 (三). 45 步階訊號的 CMAC 控制(no dead-zone)

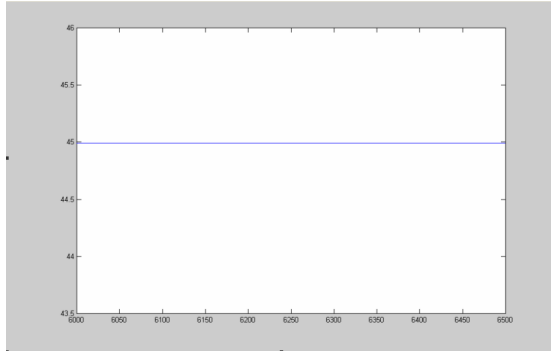


圖 3-8(四). 45 步階訊號的 CMAC 控制(加 dead-zone=0.002degree)

使得系統在 45 度時仍不斷的學習，最後在經過 6250 秒後，系統開始發散了。

圖 3-8 也是一個 45 度步階訊號的 CMAC 控制，因為加了 dead-zone 等於 0.002 degree，使得 τ_p 大於 0.002 degree 時或 τ_p 小於 -0.002 degree 時，系統才做學習，我們發現再經過相同時間 6250 秒後系統仍然維持穩定。

以下是做多次 dead-zone 模擬之關係表：

Dead-zone (degree)	發散時間(second)
no dead-zone	1007.375
0.0005	1014.05
0.0006	1014.975
0.0007	1016.835
0.0008	1017.865
0.0009	1018.795
0.001	10187.95

表(一) Dead-zone 與發散時間關係表

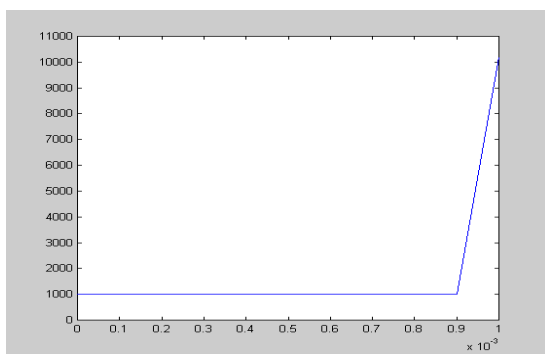


圖 3-11(五) Dead-zone 與發散時間關係圖

再來我們討論有關方波函數控制的不穩定性及用 τ bound 來加以解決。

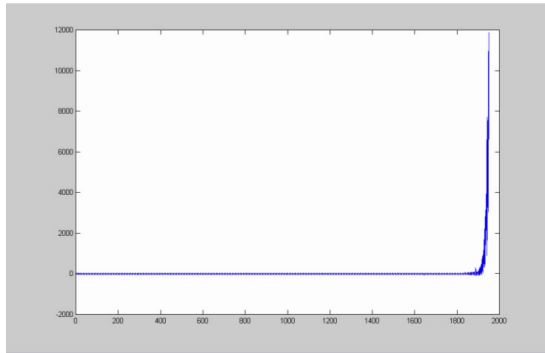


圖 3-12 (六)為等於一個 ± 45 的方波訊號的 CMAC 控制(no τ bound)

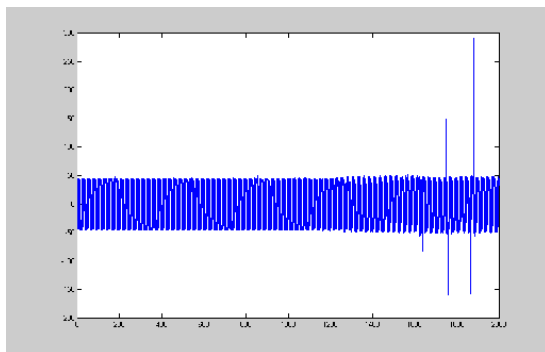


圖 3-13(七)為等於一個 ± 45 的方波訊號的 CMAC 控制(加 τ bound=2000)

我們在此先為此系統的穩定做一個定義，即到達控制目標後連續十個輸出的平均值超過控制目標的正負 10 度就定義為不穩定，舉例來說，控制訊號如果是正負 45 度的方波訊號，那麼當到達正負 45 度後，若連續十個輸出為 (45,45,45,45,45,45,45,45,45,100)，雖然最後一個值超過 55 度，但此時個數的平均值為 50.5，故此系統仍算穩定，因其保持在 55~35 間或 -55~-35 間，圖 3-12 因為沒有設 τ bound，而方波訊號的控制在其上下擺動時其 tracking-error 很大，而使此時的學習的量很大，經過長時間的累積後，系統會發生發散現象，而圖 3-13 為等於一個正負 45 度的方波訊號的 CMAC 控制，但因我們加入了 τ bound=2000 degree 使得在 τ 大於 2000 degree 或小於 2000 degree 時 τ 仍等於 2000 degree，我們發現雖然其輸出的角度保持在一定範圍，但在大約 1620 秒後其輸出仍到達接近 150 度，其系統達到不穩定。

以下是做多次 τ bound 模擬之關係表：

Bound(degree)	發散時間(second)
no bound	1850
2000	1600
1800	1700
1600	4300
1400	2100
1200	2400
1000	9000

900	15000
-----	-------

表(二) Bound 與發散時間關係表

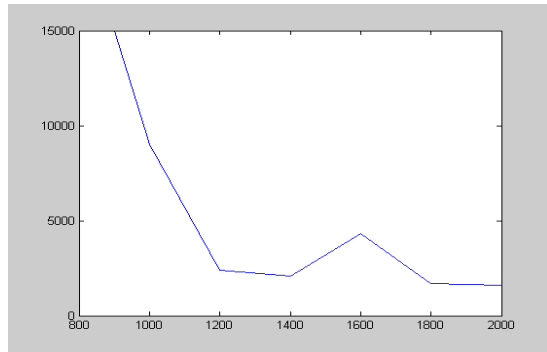


圖 3-15(八) Bound 與發散時間關係圖

CMAC 系統晶片架構

本計畫以系統單晶片的觀點來規劃所需要的 CMAC 控制系統，其晶片內需要有微處理器，專門處理計算及讀取複雜度高的特殊硬體，記憶體，及數位類比轉換器等，我們並且設計四顆特殊用途的硬體來處理 CMAC 演算法中的複雜運算。AHB Bus 及 APB Bus 為 AMBA System 中的兩種規格，WRAPPER 為 AMBA Bus 與 ASIC 間的介面轉換電路。下圖 4-1 為訊號流程說明如下：

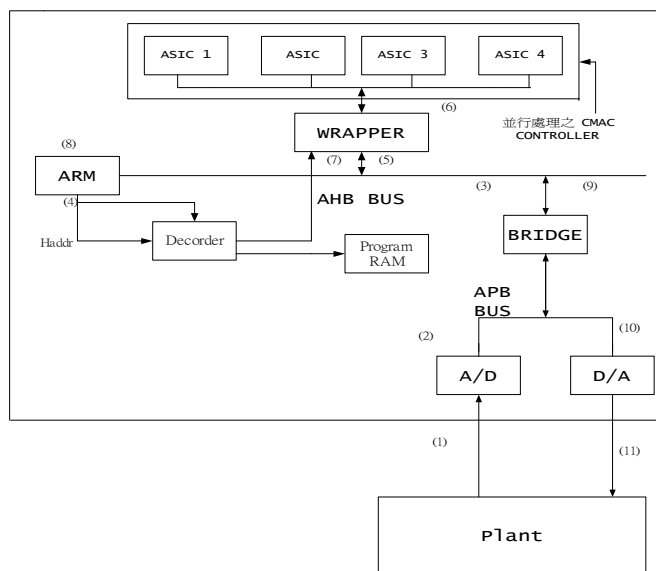


圖 4-1(九) 系統訊號流程圖

(1) Plant 發出輸入類比訊號。(2) 經由類比數位轉換器轉成數位訊號。(3) 訊號在 APB Bus 上經由 Bridge 的轉換到 AHB Bus 上。(4) ARM CPU 經 AHB Bus 抓取訊號，在 ARM 內進行 Quantization 之運算處理。(5) Quantization 後依序送入三個輸入狀態(q_e , q , v_e)及 1 個 t_p 及兩個啟動訊號經由 AHB Bus 線送至

WRAPPER 的 6 個暫存器，以進行訊號之轉換並把 polling 的暫存器由 0 變 1。
 (6)經由 AHB Bus 線傳輸的資料均為 32 位元，進行 Generalization，hash coding，Learning 的運算，做完之後 WRAPPER 從 CAMC Controller 收取控制訊號(\hat{c}_c)。(7)接著 CPU 不斷 polling WRAPPER 中的一個暫存器再經過 19 個 cycle 後，ARM CPU 先取得一個 polling 由 1 變 0 的訊號後得之 CAMC Controller 已完成運算，再取 \hat{c}_c 暫存器的前 32 位元當做輸出訊號 \hat{c}_c 。(8)加上 PD 控制力矩(f_p) (40 位元)成為輸出訊號 \hat{o} 。(9)經 AHB Bus 送至 Bridge 以完成 AHB Bus 到 APB Bus 的訊號轉換。(10)APB Bus 訊號經數位類比轉換器轉成類比訊號 \hat{o} 。(11)輸出類比訊號 \hat{o} ，進行下一時序之控制。

再來，我們撰寫硬體描述語言 verilog 來實現的 CMAC controller 必須符合 AMBA，AMBA 為 ARM Limited 公司[13]所推出的協定，主要目的是為了 on-chip bus 的規範，一個以 AMBA 架構的 SoC，一般來說包含了 high-performance 的 system bus-AHB 與 low-power 的 peripheral bus-APB。System bus 示負責連結例如 ARM 之類的 embedded processor 與 DMA controller，on-chip memory 而 peripheral bus 負責其他週邊元件，其 protocol 相對 AHB 來講較為簡單，與 AHB 之間則透過 Bridge 相連，期待能減少 system bus 的 loading。而 WRAPPER 為 AHB Bus 與所設計的 ASIC 進行訊號轉換的介面電路。

下圖 4-8 為 CMAC 控制器的內部方塊圖，我們針對 CMAC 演算法 generalization=32，記憶體容量為 1024 分成 4 個 block，使其達到並行處理。再依照時序把我們設計的電路分成四層的架構，分別為 generalization，hash coding，ram 的讀寫(代表學習及累加)，最後在把 4 個 block 所累加的 \hat{c}_c 相加。

在此我們解釋圖 4-8 中各個方塊圖所代表的意義，在 generalization 的運算中我們 4 個 block 的分法為將 $p=0\sim7$ ， $p=8\sim15$ ， $p=16\sim23$ ， $p=24\sim31$ 的運算分在同一組，G1-2 代表 $S_1 (q_c)$ 的狀態且 $p=8\sim15$ 的運算方塊圖，依此類推，在 generalization 的運算中共有 12 個運算方塊圖可同時並行處理，而 Hash coding 1 的方塊圖代表 $p=0\sim7$ 的 Hash coding，共有 4 個 Hash coding 的方塊圖表示可同時產生 4 個記憶體的位置。

RAM1 為我們處理 $p=0\sim7$ 的記憶體方塊圖，共有 4 塊記憶體，我們記憶體的傳輸規格為 1 個 cycle 做 1 筆資料的讀或寫，Accumulate1 代表我們處理 $p=0\sim7$ 的累加方塊圖，共有 4 塊累加方塊圖，Learning1 代表我們處理 $p=0\sim7$ 的學習方塊圖，共有 4 塊學習方塊圖可以同時做學習，最後 SUM 代表三個簡單的加法器，分別把 Accumulate1~Accumulate4 最後所產生的 \hat{c}_c 做相加，以得到 CMAC 控制器的輸出。

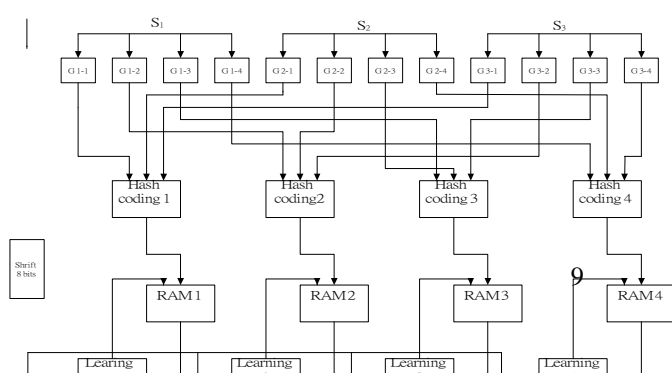


圖 4-8 CMAC Controller 整體架構圖

而完成的 HDL code 在 modelsim 的軟體上進行模擬，來找取邏輯或語法上的錯誤已驗證邏輯功能的正確性，驗證的第一個步驟是對整個電路的規格和功能完整的了解，並能夠依功能上的不同，清楚的劃分數個子區塊 (subblock)，並完成設計與測試，功能驗證的最後一步為整合所有的子區塊，當所有子區塊都發展完成後，接著要寫一個最上層的 code，將所有子區塊連結在一起，以進行完整的功能測試。

做最後測試時，大部分的錯誤都是發生在各子區塊間的介面上，因此開始劃分及設計各子區塊時，就應該對各子區塊之間介面的時序及功能做完整的規劃。

因為在 verilog 的語法中只有一部分是可以合成電路的，其餘的部分為進行 test module 的撰寫，因此我們需在 debussy_nLint 的軟體上對我們的 verilog 檔進行是否可以 synthesis 的偵錯，在改正所有顯示的 error 並確認無誤後，我們才在 synopsys 的軟體下進行我們的電路合成，此項工作是將我們所設計的邏輯功能轉化成 cell 與 cell 的連線關係(netlist)，這包括了 cell mapping(即去元件庫抓取 cell)和邏輯最佳化。

我們受控系統(即機器手臂)輸出角度在 verilog 中的描述為 40bits，包含 12bits 的整數與 28bits 的小數，我們使用 verilog 中的指令 display 將我們所需要的訊號印在 modelsim 的軟體介面上，再使用一個 C Language 的轉碼程式，將我們的數位訊號轉成一般的 10 進位表示法，最後將所得的數據使用 matlab 軟體作圖。經過反覆的驗證後在 $1/2$ 的 28 次方的準確度下，我們實現了內嵌 ARM 處理器的 CMAC 控制器，並與 C Language 的模擬結果相同。圖 5-4 與圖 5-5 為 C Language 及用並行架構的 CMAC Controller 所進行 45 步階訊號的模擬結果。我們發現圖 5-5 在控制初期與圖 5-4 的控制結果非常接近，而在控制後期則有些許不同，而這是因為系統內的 cos 函數我們使用四階的泰勒展開，其誤差值隨角度而慢慢增大，在 45 度時約為 0.0001，另外在系統中的一些 10 進位制

也是以近似表示，例如 g 值(9.8)近似 $9+205/256$ ， $1/6$ 近似 $171/1024$ ，整體來看，圖 5-5 其控制結果大致與圖 5-4 結果相同。

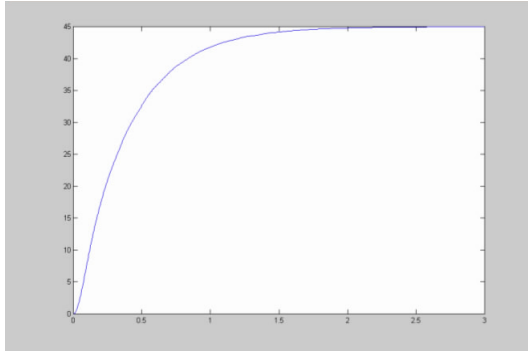


圖 5-4(九) C-language 所得的模擬結果

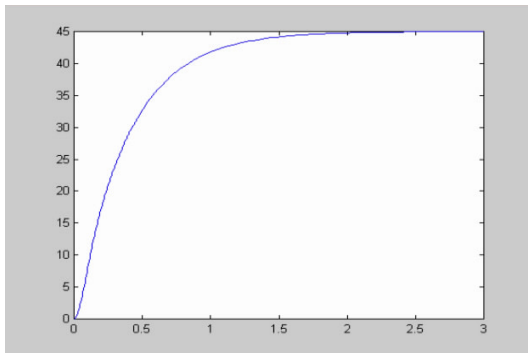


圖 5-5(十) 並行架構的 CMAC Controller 所得的模擬結果

五、參考文獻

- [1] J. S. Albus, "A New Approach in the Cerebellar Model Articulation Controller (CMAC)," *Journal of Dynamic Systems, Measurement and Control*, pp. 220-227, Sep. 1975 .
- [2] J. S. Albus, "Data Storage in The Cerebellar Model Articulation Controller", *Journal of Dynamic Systems, Measurement and Control*, pp. 228-233, Sept. 1975
- [3] W. T. Miller, "A nonlinear learning controller for robotic manipulators," in *Proc. SPIE, Intell. Robots and Computer Vision, vol. 726*, pp. 416-423, Oct. 1986.
- [4] W.T. Miller, F.H. Glanz and L.G. Kraft, "Application of a General Learning Algorithm to the Control of Robotic Manipulators," *International Journal of Robotic Research*, Vol. 6, No. 2, pp. 84-98, 1987
- [5] W.T. Miller, R.H. Hewes, F.H. Glanz and L.G. Kraft, "Real-Time Dynamic Control of an Industrial Manipulator using Neural-Network-Based Learning Controllers," *IEEE Transactions on Robotics and Automation*, pp. 1-9, Feb.

1990

- [6] F.-C. Chen and C.-C. Liu, "Adaptive Control of Nonlinear Continuous-Time Systems using Neural Networks," *IEEE Transactions on Automatic Control*, pp. 1306-1310, June 1994.
- [7] 林桓宇, "以 CMAC 類神經網路控制器實現機器人控制", 國立交通大學電機與控制工程研究所碩士論文, 2000
- [8] F.-C. Chen and C.-H. Chang, "Practical Stability Issues in CMAC Neural Network Control Systems," *IEEE Transactions on Control Systems Technology*, pp. 86-91, January 1996.
- [9] C.-C. Lin and F.-C. Chen, "An Improved CMAC Neural Network Control Scheme," *IEE Electronics Letters*, Vol.35, No.2, pp.157-158, Jan. 1999.
- [10] D. Ellison, "On The Convergence of the Multidimensional Albus Perceptron," *The International Journal of Robotics Research*, Vol. 10, No. 4, pp. 338-357, August 1991.
- [11] 何智南, "以 FPGA 實現 CMAC 類神經網路控制器", 國立交通大學控制工程研究所碩士論文, 1997
- [12] F.-C. Chen and H.K. Khalil, "Adaptive Control of a Class of Nonlinear Discrete-Time Systems using Neural Network," *IEEE Transactions on Automatic Control*, pp. 791-801, May 1995.
- [13] www.arm.com