

國防科技學術合作協調小組研究計畫成果報告

航空電機致動器數位式適應伺服控制方法之研究

計畫編號：90-2623-7-009-009

執行期間：90年1月1日至90年12月31日

計畫主持人：吳永春 教授

共同主持人：鄒應嶼 教授

執行單位：國立交通大學電機與控制工程學系

中華民國 90 年 12 月 31 日

目 錄

摘要 (中文).....	iii
摘要 (英文).....	iv
第一章 數位式適應伺服控制技術的發展.....	1
1.1 研究背景.....	1
1.2 研究步驟.....	3
第二章 適應伺服控制架構的介紹與比較.....	6
2.1 增益排程.....	6
2.2 參考模型式適應控制系統(MRAC).....	7
2.3 自我校正調整器(STR).....	10
2.4 自調式適應控制系統.....	13
第三章 自調式適應伺服控制之介紹.....	15
3.1 簡介.....	15
3.2 PID 控制器自調法則.....	16
3.2.1 Ziegler-Nichols Method.....	17
3.2.2 Relay-Feedback Method.....	18
3.2.3 Frequency-Domain Method.....	20
3.2.4 Pole Placement Method.....	22
3.2.5 Optimization Method.....	24

第四章 參考模型式數位式映伺服控制.....	25
4.1 永磁式交流馬達數學模型推導.....	25
4.2 參考模型推導.....	27
4.3 適應控制機制推導.....	30
4.4 適應控制系統模擬.....	31
第五章 MATLAB 參數自調控制方法.....	36
5.1 原理說明.....	36
5.1.1 參數邊界值及初始值得決定.....	36
5.1.2 測試訊號的產生.....	37
5.1.3 目標函數的定義.....	38
5.1.4 調整法則.....	39
5.2 MATLAB 軟體實現.....	40
5.2.1 主程式說明.....	40
5.2.2 Full_step 副函式.....	43
5.2.3 Cal_cost 副函式.....	43
5.2.4 Min_cost 副函式.....	44
5.2.5 Record_trend 副函式.....	45
5.2.6 Next_set 副函式.....	46
5.2.7 Converge 副函式.....	49
第六章 實驗結果與分析.....	50
6.1 實驗系統架構.....	50
6.2 實驗結果.....	52
第七章 結論.....	62
參考文獻.....	R-1

摘 要

本計畫擬發展可適用於輕型無人飛行載具之航空電機伺服致動適應控制系統，發展以數位訊號處理器(Digital Signal Processor, DSP) TMS320F240 為核心具有適應控制功能的全數位永磁交流馬達伺服驅動器。全程計畫將以適合航空環境應用之永磁式交流無刷馬達(具備 Resolver)為受控平台(輸入電源直流 28V、輸出功率 1/2 HP)，以此推動滾珠螺桿等機械傳動機構而控制輕型無人飛行載具之翼面，進一步研究先進的適應控制法則，發展以 DSP 軟體控制為核心的適應控制法則。

本計畫第一階段將以發展全數位直流無刷馬達(具備 Resolver)伺服驅動器為目標，而航空電機伺服致動器為未來應用對象，採用以積分比例控制(integral-proportional control)為基礎的多迴路模型配適控制(model matching control, MMC)架構，探討具有實用價值的適應控制機制，如參考模型適應控制(model-reference adaptive control, MRAC)、自動調整適應控制(auto-tuning adaptive control, ATAC)等適應控制法則，發展可應用於高性能航空電機伺服致動器的即時適應控制法則，改善伺服致動器的強韌性與動態響應。研究計畫將以單晶片 DSP 實現所發展的即時適應控制法則，並以永磁式交流無刷馬達(具備 Resolver)為對象，完成系統的實作驗證與性能評估。未來將以此為基礎，進一步發展推動滾珠螺桿等機械傳動機構之伺服致動控制器以控制輕型無人飛行載具之翼面，以期完成一個具有強韌性的航空電機伺服致動器。

關鍵詞：航空電機伺服致動器、永磁式交流無刷伺服馬達、數位訊號處理器、數位馬達控制、積分比例控制、模型配適控制、適應控制

Abstract

This project plans to develop adaptive electrical actuator control technology for advanced electronic control systems in advanced more electronic aircrafts. This project will focus on the development of DSP-based adaptive ac servo control techniques. A permanent-magnet AC (PMAC) servo motor with 28V input, 1/2 HP output power, and an integrated resolver will be chosen as the platform in design of the digital adaptive controller. A DSP-based controller suitable for digital motor control will be constructed, and by using the constructed DSP controller, software control techniques for the torque, velocity, and position control of PMAC servo actuators will be developed. This project will adopt the multiple-loop cascaded integral-proportional control scheme as the basic control structure and try to develop practical adaptive control algorithms for high-performance digital servo actuators. We will investigate practical adaptive control mechanisms, such as model-reference adaptive control (MRAC), auto-tuning adaptive control (ATAC), etc. to improve dynamic responses and robustness of the digital ac servo control system. This project will realize the developed adaptive control algorithm by using a single-chip DSP controller. Experimental verification and performance evaluation will be given by using a brushless dc motor with a resolver. In the future, based on the developed adaptive digital servo control techniques, we can develop high-performance robust servo actuator used in a ball-screw driven mechanical transmission system for the wing actuator of a light manless aircraft.

Keywords: permanent-magnet servo motor, digital signal processor, digital motor control, model matching control, adaptive control

第一章

數位式適應伺服控制技術的發展

1.1 研究背景

根據美國於 1991 年所作的研究顯示，運用電子電力技術取代航空器現有液壓系將可大幅提升可靠度及維護度，並降低系統重量、體積與複雜度。“More Electric Aircraft” (簡稱 MEA) 的研究指導綱要便因應而生，結合政府機構並鼓勵民間業界，將此項技術推廣應用於軍、民用航空器上。美國於 1991 年起便積極投入此項研發計畫，除政府機構外如由民間 DTI 和 TRW Aeronautical Systems (Lucas Aerospace) 所支助的 The LEMAS (Large ElectroMechanical Actuation Systems) 等相關研究計畫越來越普及且投入資金龐大。預計 1999 至 2005 年間平均每年投入政府預算約 2 仟 3 百萬美元進行研究，且未來將持續進行並擴大應用成果展示。跡象顯示，MEA 的技術已是下一代飛機的趨勢，其影響將遍及所有軍用/民用航空器。鑑於國內並未對此趨勢作一廣泛的研究，因應技術發展趨勢並為縮短與國外技術差距，故本研究計畫擬結合學界優異的基礎研究能量，選定航空電機伺服致動控制器為應用標的，由基礎學理評估、數學模型推導、模擬分析來設計適當的適應控制法則並加以驗證。

伺服系統之目的即在於精密的速度與位置控制，隨著高科技的發展與輕薄短小時代的到臨，精密伺服系統日益重要，舉凡積體電路的製造、光電元件與精密陶瓷的加工等，均是其應用領域。伺服馬達主要可分為同步交流伺服馬達與感應交流伺服馬達，根據磁場的分佈方式，又可區分為永磁交流伺

服馬達(正弦分佈)與無刷直流伺服馬達(方波分佈)[1]-[14]。永磁交流伺服馬達廣泛應用於需要精密定位的中小型之自動化生產設備，高扭矩永磁交流伺服馬達亦可應用於直接驅動系統，這其中包含了工業機器人、軍事武器高速射控系統、導控飛彈、飛航載具的致動器等等。本計畫將以永磁直流無刷馬達(具備 Resolver)為對象，發展以 DSP 為基礎的數位伺服適應控制技術。

直流無刷馬達亦為交流馬達之一種，而交流馬達控制的關鍵，主要在於扭矩的解耦控制。過去三十年來，發展出了各種不同控制架構的磁場向量控制方法[15]-[24]。總括而言，目前實際應用於工業界的主要有兩種控制方法：一是以電流向量控制為主的磁場導向控制(field-oriented vector control, FOC)；另一是以電壓向量控制為主的直接扭矩控制(direct torque control, DTC)。這兩種不同的控制方法均可達到快速且獨立的扭矩與激磁控制。

一個高性能的交流伺服驅動器，包含了多種複雜的控制運算，如脈寬調變控制、電流控制、磁場導向控制、弱磁控制、伺服迴路控制等等，這些控制法則一方面不易以硬體電路實現，另一方面又由於系統頻寬的需求，必須在極短的時間內完成，例如電壓向量控制一般必須在 40 微秒內完成，因此，一般的微控器不足以應付如此複雜的數值計算。所幸近年來 DSP 技術快速發展，使得 DSP 不但具有強大的計算能力(320C1x 約為 4-5MIPS、320C5x 約為 40-50 MIPS)[25]、同時價格(320C14 約為 150 NT、320C52 約為 540NT、320C50 約為 1900NT、320F240 約為 750NT)亦堪稱低廉，因此應用 DSP 於數位馬達與伺服控制，已成為目前的發展趨勢[25]-[38]。

德州儀器(Texas Instruments)公司於 1997 年推出了新一代的高性能單晶片 DSP 控制器(TMS320F240) [26]，此晶片的運算速度高達 20 MIPS (每秒執行二千萬個指令)，價格低廉(低於 20 元美金)，提供了完整的 I/O 埠，包括 A/D 轉換器、PWM 輸出、波形產生器、CAPTURE 輸入、數位輸入及輸出接腳、外部中斷等等。由與此單晶片 DSP 具備了一般微控器的 I/O 界，而同

時又具有強大的數值運算能力，因此非常適合應用於數位馬達與伺服控制。

應用於飛行裝置的伺服系統，由於外在環境劇烈的改變，因此系統將遭遇劇烈的參數變化與負載干擾，而且這些外在干擾通常無法以具體的數學模型描述，不論是確定的(deterministic)或是隨機的(stochastic)。面對這些實際存在的干擾與參數變化，傳統的強韌控制方法已無法應付[39]-[49]，因此有必要發展具有自調適應性(auto-tuning adaptive)的控制法則，使所設定的性能指標(performance index)在不同的操作環境均能達到最佳值，由於這些應用上的需求，適應控制已成為飛航系統伺服致動器重要的研究課題[50]-[57]。

1.2 研究步驟

本計畫擬以永磁直流無刷馬達(具備 Resolver)為基礎，研究適用於數位伺服致動控制系統的適應控制法則，並以 DSP 實現所設計出的控制法則，詳細之研究步驟說明如下：

1. IP與PI控制方法整數實現(integer realization)的比較分析

比例積分(proportional-integral)控制以廣泛應用於各種工業控制系統，其衍生的積分比例(integral-proportional)控制架構，由於具有先天的強韌特性，因此近年來逐漸受到工業界的重視與應用，本階段探討這兩種方法的 DSP 整數實現(integer realization)問題，並進行以 MATLAB 為基礎的分析與模擬。

2. 多迴路模型配適控制方法的分析與模擬

以多迴路控制架構為基礎，伺服迴路採用模型配適控制方法，分析與模擬當伺服系統在在劇烈負載變動下的動態響應。

3. 適應控制機制的比較分析

針對不同的適應控制機制與參數調整策略，探討具有實用價值的適應控制方法，如參考模型適應控制(model-reference adaptive control, MRAC)、自我調整適應控制(auto-tuning adaptive control, ATAC)等適應控制法則，本階段將針對這些較實用的適應控制方法，進行評估比較，並選擇或發展出一種新的方法，作為後續設計適應伺服控制系統的基礎。

4. 建立數位式伺服控制器的設計步驟

建立數位馬達控制板發展數位式伺服控制器的設計步驟，決定各控制參數的初值(initial value)，發展以 MATLAB 為基礎的控制器設計步驟，根據設計目標與系統之參數(馬達與功率放大器)，決定各控制參數的初值。

5. 多迴路模型配適控制器之設計

設計具有強韌性的多迴路模型配適控制器，並以 MATLAB 為基礎進行分析與模擬。

6. 多迴路模型配適控制器的DSP程式設計

發展以 DSP C-語言為主的模型配適控制程式，進行數位馬達控制的整合測試。

7. 發展自調式適應控制方法

根據實驗量測數據所得到的系統模型，發展控制參數自調法則。例如：根據所選擇的 PDFF 或 IPDFF 控制架構，決定各控制參數的調整順序(或同時調整)，發展具有收斂性的調適方法(tuning algorithm)，調適方法可以 MATLAB 或 C-語言實現。

8. 自調式適應伺服控制之電腦模擬

採用 SIMULINK 模擬軟體，進行此數位自調控制研究的電腦模擬。SIMULINK 可結合數位控制、電動機、與類比或數位控制模組，進行以數位控制為導向的電腦模擬，提供良好的人機介面，本計畫將以 SIMULINK 建立模擬分析的環境，並配合實驗結果進行比較分析。

9. 線上自調式適應伺服控制之整合

以個人電腦為基礎，經由 RS-232 介面，整合 DSP 控制卡、伺服驅動器、MATLAB、WINDOWS，完成一個以視窗環境為基礎的互動式數位伺服實驗系統。以永磁直流無刷馬達為對象，藉由此系統進行線上自調適應控制實驗，發展具有實用價值的自調式適應伺服控制方法。

10. 系統整合與測試

以永磁直流無刷馬達為對象，使控制器可獨立運作為標的，完成系統的實作驗證與性能評估。

第二章

適應伺服控制架構的介紹與比較

適應控制自從 1950 年代發展以來，出現許多不同形式的適應控制系統，其中較成熟且具實際應用實例的有四大類[50]，包括：增益排程(gain scheduling)、參考模型式適應控制(Model Reference Adaptive Control, MRAC)、自我校正調整器(Self-Tuning Regulators, STR)與自動調整控制器(Auto-Tuning Controller, ATC)，本章針對上述適應控制系統進行介紹。

2.1 增益排程

圖 2.1 為增益排程的適應控制系統方塊圖，包含控制器(controller)、受控體(plant)與增益排程器(gain schedule)等三個方塊，其中增益排程器藉由回授受控體的輸出訊號，來判斷受控體目前的工作狀態，然後選擇最適當的控制參數，以得到最佳響應。

由圖 2.1 可知，增益排程適應控制系統含有兩個控制迴路，一個迴路由控制器、受控體與回授訊號所構成，即一般含有回授之閉迴路控制系統；另一個迴路由控制器、受控體與增益排程三個功能方塊構成，根據受控體的工作狀態，更新控制器的控制參數。

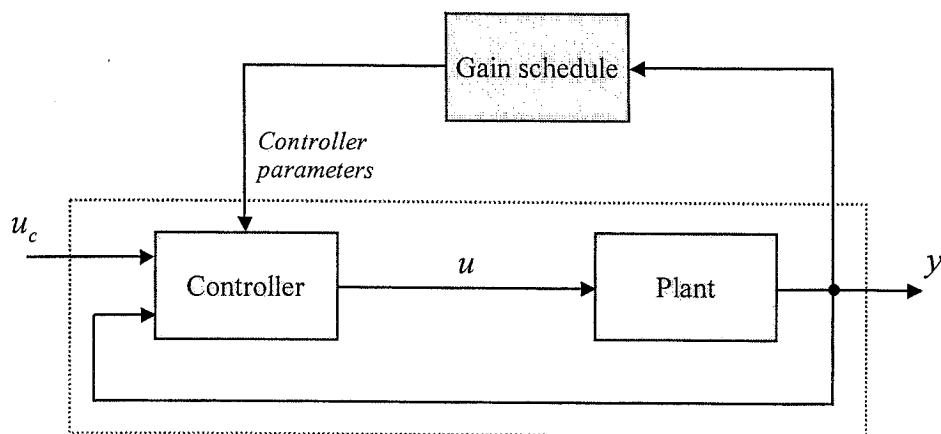


圖2.1 增益排程之適應控制系統方塊圖

2.2 參考模型式適應控制系統(MRAC)

圖 2.2 為參考模型式的適應控制系統方塊圖，包含控制器、受控體、可調整機構裝置(adjustment mechanism)與參考模型(reference model)等四個方塊，其中『參考模型』是根據系統響應要求所建立出的理想數學模型，用以描述當系統命令之輸入訊號為 u_c 時，能夠得到期望的系統響應 y_m ；『可調整機構裝置』則是根據參考模型的輸出訊號 y_m 、受控體的輸出訊號 y 與受控體的輸入控制訊號 u ，經由控制法則計算出一組相對應的控制器參數，可調整機構裝置的目的是用來消除受控體參數變化所造成的響應變動，以增加系統的強韌性，同時滿足系統要求的動態響應。可調整機構裝置的設計可以分為兩個部分：

- ◆ 量測受控體實際響應與參考模型的理想響應，並計算之間的誤差。
- ◆ 根據誤差量的大小，利用 sensitivity approach、MIT rule 或 fuzzy rule 等理論法則，計算出一組新的控制器參數。

由圖 2.2 可知，參考模型式適應控制系統含有兩個控制迴路，一個迴路由控制器、受控體與回授訊號所構成，即一般含有回授之閉迴路控制系統；另一個迴路由控制器、受控體、參考模型與可調整機構裝置等四個功能方塊

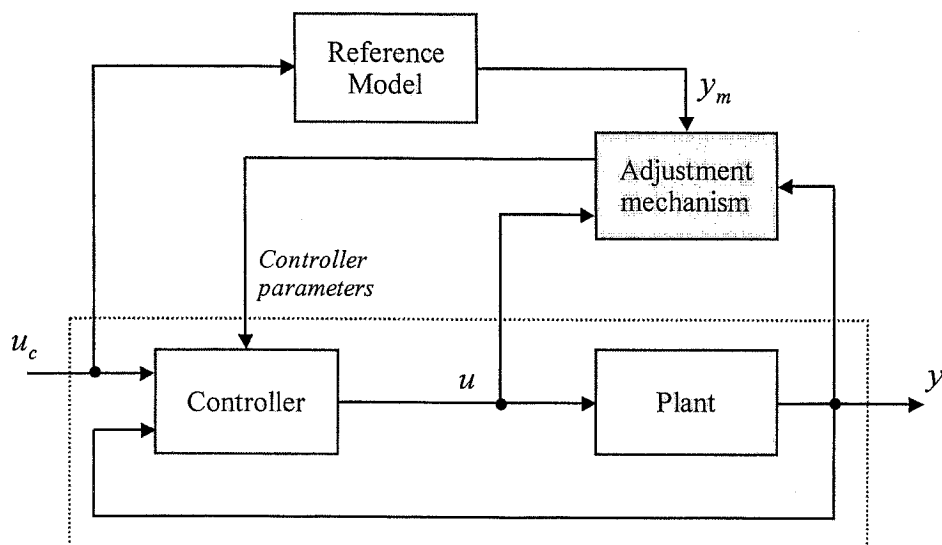


圖2.2 參考模型式之適應控制系統方塊圖

構成，根據外部的控制訊號與受控體的輸出訊號，經由計算產生一組新的控制器參數。

參考模型適應控制系統可分成兩類，一個是間接式參考模型適應控制系統(indirect model reference adaptive control system)，其方塊圖如圖 2.3 所示，首先必須先根據受控體的輸入輸出訊號，來估測受控體的參數，然後根據受控體的參數、實際響應與真實響應的誤差量，以及命令訊號，經由所設計的控制法則來計算並調整控制器參數，來得到最佳動態響應；另一個方式是直接式參考模型適應控制系統(direct model reference adaptive control system)，其方塊圖如圖 2.4 所示，與間接式參考模型適應控制系統不同的地方在於受控體的參數為已知，可以省去估測的動作，其餘的動作與原理則相同。

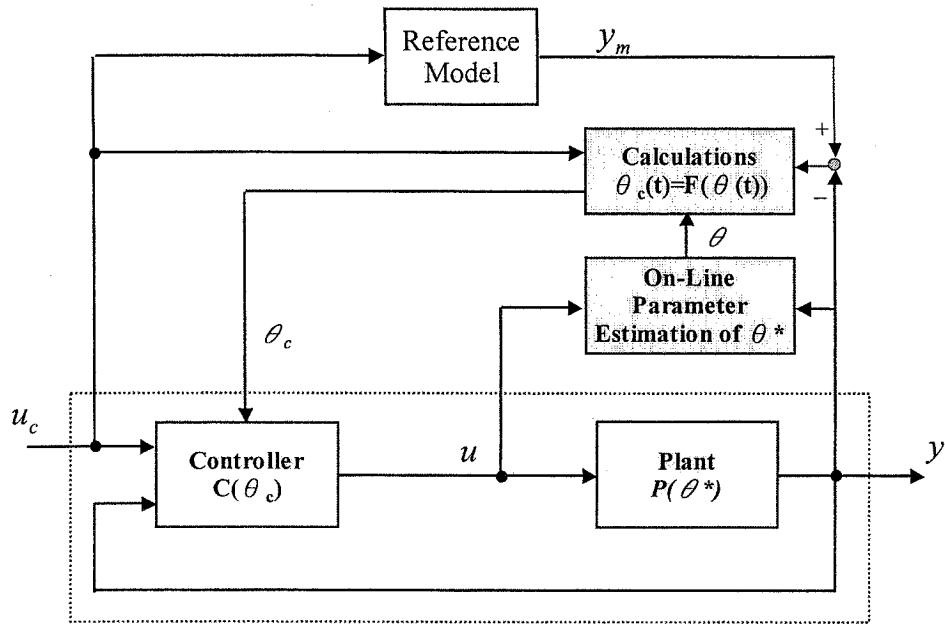


圖2.3 間接式參考模型適應控制系統

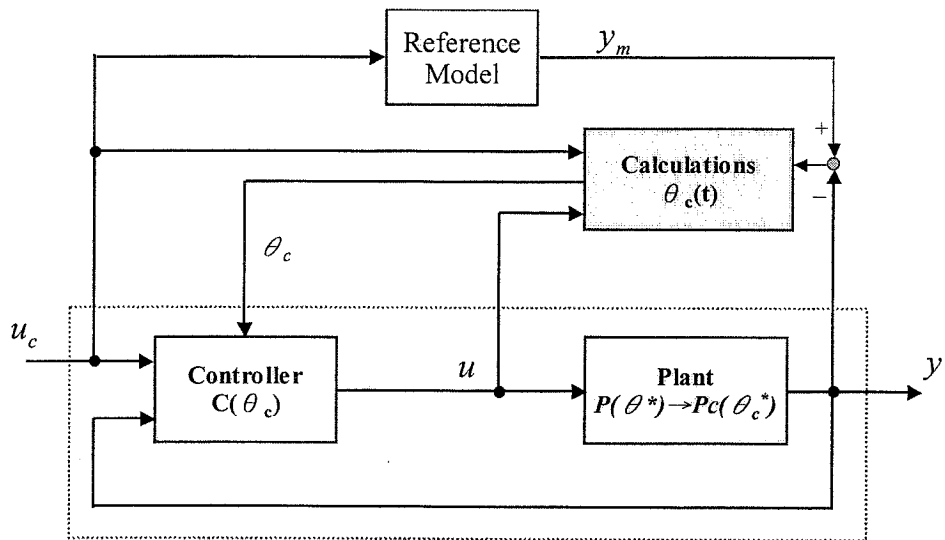


圖2.4 直接式參考模型適應控制系統

2-3 自我校正調整器(STR)

1958 年，Kalman 提出自我校正調整器(STR)的概念，探討以最小平方方法應用於控制器設計中，但由於當時適應控制理論尚未充分發展，以及缺乏適當的計算工具，Kalman 的想法未能進一步的發展與實現。直到 1970 年，隨著隨機控制理論、系統判別理論與計算機技術的發展，自我校正調整器的研究與應用迅速發展起來。1973 年 K. J. Astrom 與 B. Wittenmark 提出一個簡易可行的自我校正調節器機制，在這個機制中，用一個表示輸入輸出關係的線性差分方程式(可包含干擾項)作為系統預測數學模型(稱為可控自回歸滑動平均模型，縮寫為 CARMA)，用遞迴最小平方方法(recursive least square method)在線式(on-line)估計模型的參數，直接得到一個輸出平方差最小的自我校正調節器。K. J. Astrom 與 B. Wittenmark 於 1980 年又提出具有極點與零點配置功能的自我校正調節器，能夠在線式估測系統或控制器參數，使閉迴路系統的零點與極點配置到指定的位置，滿足系統動態響應要求。

圖 2.5 為一個典型的自我校正調整器的適應控制系統方塊圖，其中包含控制器、受控體、參數估測器(estimation)與控制器參數設計(controller design)等四個方塊，其中『參數估測器』是根據受控體的輸入控制訊號 u 與輸出訊號 y ，估測出受控體的參數；『控制器參數設計』則是根據設計者給定的規格與受控體參數，藉由控制法則計算得到一組新的控制器參數。

由圖 2.5 可知，自我校正調整器適應控制系統含有兩個控制迴路，一個迴路由控制器、受控體與回授訊號所構成，即一般含有回授之閉迴路控制系統；另一個迴路由控制器、受控體、參數估測與控制器參數設計等四個功能方塊所組成，根據規格要求與受控體的輸出訊號，經由計算產生一組新的控制器參數。

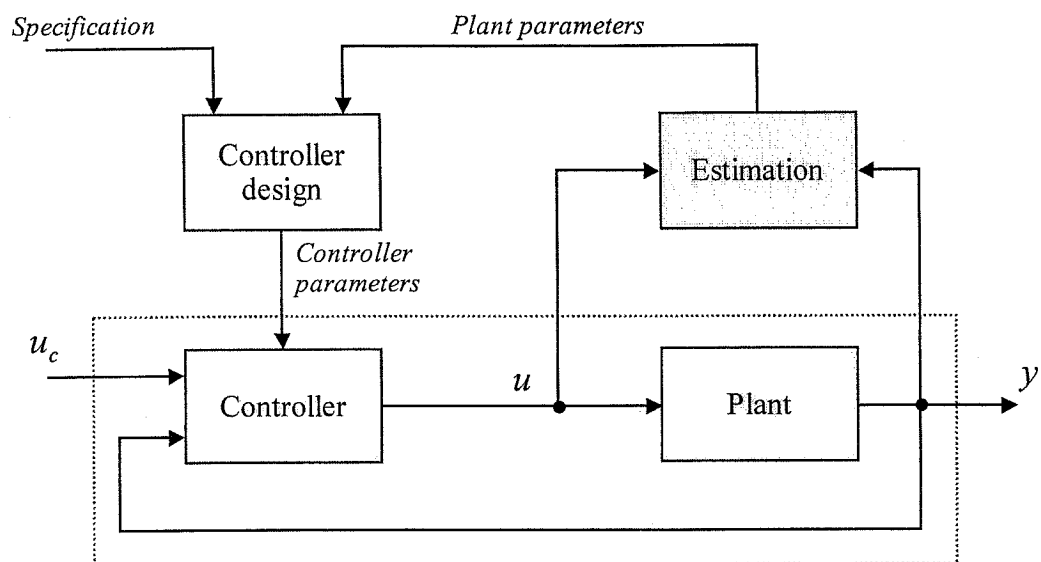


圖2.5 自我校正調整器的適應控制系統方塊圖

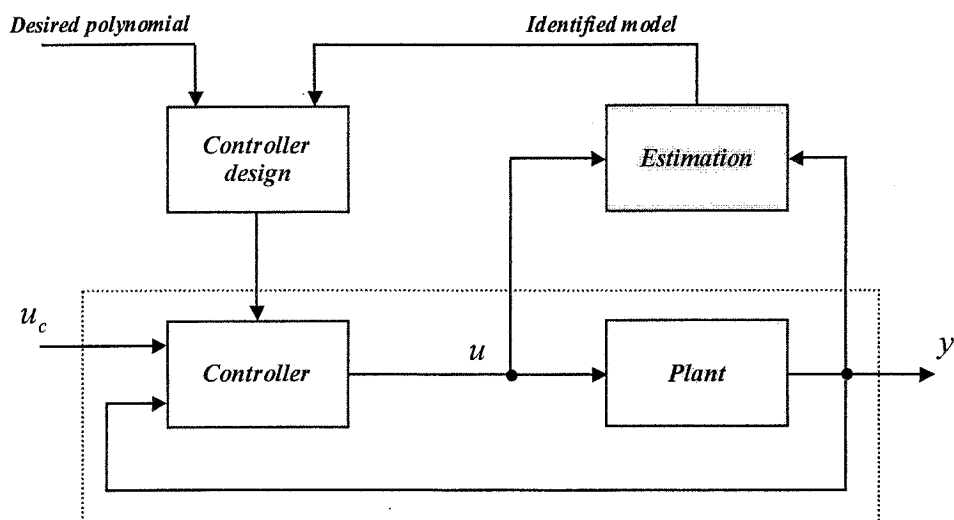


圖2.6 α -adaptive 極點配置的 STR 控制系統

L. Wang 和 D.H. Owens 於 1989 年提出 α -adaptive pole placement design 的 STR 控制系統，其方塊圖如圖 2.6 所示。『Estimation』方塊，以遞迴式最小平方法，估測受控體的模型，通常以離散時間之通式表式如下：

$$\begin{aligned}
G_p(k, z) &= \frac{b_1(k)z^{-1} + b_2(k)z^{-2} + \dots + b_n(k)z^{-n}}{1 + a_1(k)z^{-1} + a_2(k)z^{-2} + \dots + a_n(k)z^{-n}} \\
&= \frac{B(k, z^{-1})}{A(k, z^{-1})}
\end{aligned} \tag{2-1}$$

而控制器的轉移函數如下：

$$\begin{aligned}
G_c(k, z) &= \frac{q_0(k) + q_1(k)z^{-1} + \dots + q_m(k)z^{-m}}{1 + p_1(k)z^{-1} + \dots + p_m(k)z^{-m}} \\
&= \frac{Q(k, z^{-1})}{P(k, z^{-1})}
\end{aligned} \tag{2-2}$$

故由(2-1)與(2-2)式可推得閉迴路轉移函數如下：

$$G(k, z) = \frac{B(k, z^{-1}) \cdot Q(k, z^{-1})}{A(k, z^{-1}) \cdot P(k, z^{-1}) + B(k, z^{-1}) \cdot Q(k, z^{-1})} \tag{2-3}$$

同時轉移函數分母的多項式，必須滿足 Diophantine equation，可得

$$A(k, z^{-1})P(k, z^{-1}) + B(k, z^{-1})Q(k, z^{-1}) = T(z^{-1}) \tag{2-4}$$

其中 $T(z^{-1})$ 是設計者所決定的穩定閉迴路多項式。

由(2-4)式可知，假設控制器的 $Q(k, z^{-1})$ 為 0， $P(k, z^{-1})$ 為 1， $T(z^{-1})$ 是 $A(k, z^{-1})$ 的時變函數，於是引入參數 α ，且將 $T(z^{-1})$ 表式如下：

$$\begin{aligned}
T(z^{-1}) &= A(k, z^{-1}) \\
&= 1 + \alpha \cdot a_1(k)z^{-1} + \alpha \cdot a_2(k)z^{-2} + \dots + \alpha \cdot a_n(k)z^{-n} \\
&= z^{-n} \prod_{i=1}^n [z - \alpha \cdot s_i(k)] \\
&= z^{-n} \prod_{i=1}^n [z - c_i(k)]
\end{aligned} \tag{2-5}$$

(2-5)式中之 $s_i(k)$ 表示估測模型 $\frac{B(k, z^{-1})}{A(k, z^{-1})}$ 的極點， $c_i(k)$ 表示設計者所決定之穩

定閉迴路多項式 $T(z^{-1})$ 的根，因此以數學式表示如下：

$$\alpha = \frac{c_i(k)}{s_i(k)} \quad (2-6)$$

α 通常介於 0、1 之間，因為設計者所希望的響應，會比實際的開迴路系統響應來的快一點。當 $\alpha=0$ 時，所有閉迴路的極點都位於複數平面的原點，得到的是一個高增益的適應控制器；當 $\alpha=1$ 時，所有閉迴路的極點都等於開迴路的極點，得到的是一個低增益的適應控制器。因此，設計者在設計的過程中，只要決定適當的 α 值，即可得到一個好的 STR 控制系統。

2.3 自調式適應控制系統

圖 2.7 為具有自動調整之適應控制系統方塊圖，包含控制器、受控體與自動調整控制器(ATC)等三個方塊，其中『自動調整控制器』藉由回授受控體的輸出訊號，然後根據回授訊號經過所設計的自調控制機制計算後，便會自動更新控制器的控制參數，以滿足期望的動態響應。

由圖 2.7 可知，增益排程適應控制系統含有兩個控制迴路，一個迴路由控制器、受控體與回授訊號所構成，即一般含有回授之閉迴路控制系統；另一個迴路由控制器、受控體與自動調整控制器三個功能方塊構成，根據受控體的輸出訊號，經由自動調整的機制，更新控制器的控制參數，達到適應控制的目的。

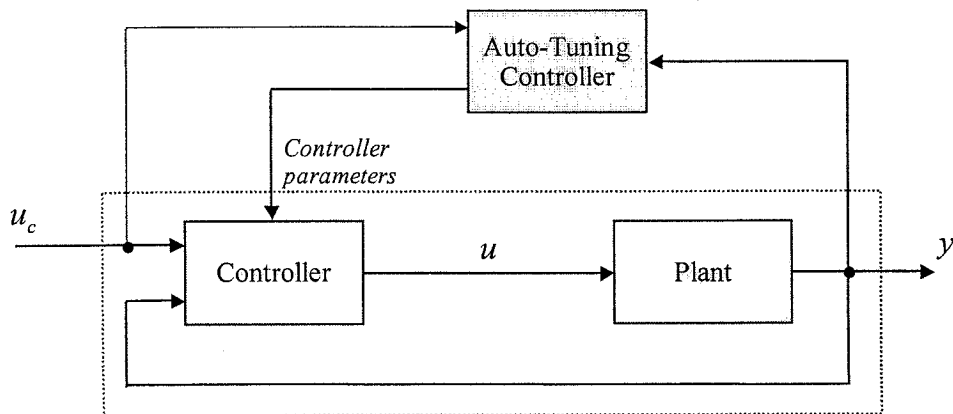


圖2.7 自動調整之適應控制方塊圖

目前應用最廣泛的控制器當屬於 PID 控制器，然而，如何在不同的負載變動之下，系統可以很快的調整 PID 控制器的參數，使得系統響應達到最佳化，這即是自動調整機制的研究重點，目前針對 PID 控制器所發展之自動調整機制，常見的有下列幾種方法：

◆ Ziegler-Nichols PID Tuning Method
Relay Feedback PID Tuning Method
Frequency Domain PID Tuning Method
Pole Placement PID Tuning Method
Optimization PI Tuning Method

下一章將針對自動調整之適應控制系統，及其自動調整器(ATC)做更詳盡的分析與介紹。

第三章

自調式適應伺服控制之介紹

3.1 簡介

自調式適應控制器是一種會根據要求響應及外部雜訊干擾來改變控制行為的一種控制器，簡單來說就是具有可調整的參數，以及調整的機制。由於在控制系統中，控制器參數設定的優劣，對系統的效能具有關鍵性的影響。但是在實際應用上，由於系統模型難以建立，造成了控制器參數在設定或調整時的困難，因此發展有系統並且有效的參數調整法則，對於實際控制系統的應用是十分重要的。

此外，因為 PID 控制器被廣泛的應用在實際系統上，故本章將針對適用於伺服控制之 PID 控制器架構、控制器參數設定法則及參數調整法則進行研究，圖 3.1 為典型 PID 控制架構方塊圖。

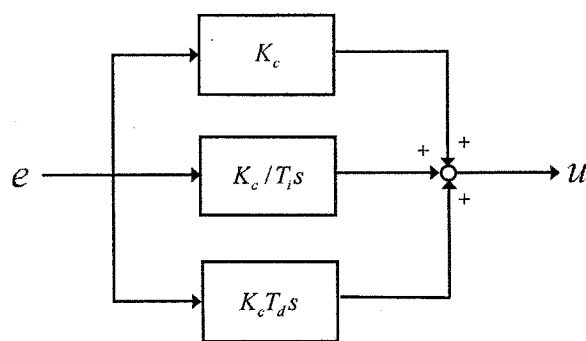


圖3.1 典型的PID控制器架構方塊圖

傳統在調整 PID 控制器參數時，都是利用試誤法 (trial and error method)，並依據表 3.1 的概念來進行調整，不但浪費許多時間，同時所調出來的參數也不是最佳值，因此無法得到最佳的響應。

表 3.1 PID 控制器參數與系統響應的關係

	響應	穩定度
K_c 增加	變快	降低
T_i 增加	變慢	增加
T_d 增加	變快	增加

在文獻方面，對於 PID 控制器架構所推導出來的參數設定法則可說是相當的完整[54]，如有名的 Ziegler-Nichols 設定法則[80]，即是透過實驗結果來歸納出一階時間延遲系統(first order system with time delay)所對應的 PID 控制器參數設定之經驗公式；還有針對 Ziegler-Nichols 設定法則作修正的方法[81]-[83]，期望得到更好的響應；此外還有一些方法是根據分析系統的頻率響應之增益邊界及相位邊界來建立參數設定法則[84]-[85]；以及分別對目標值(setpoint)、外部干擾(disturbance)與回授訊號(feedback)之誤差作運算，如絕對誤差積分(integrated absolute error, IAE)、誤差平方積分(integrated square error, ISE)與誤差最小平方法(least-square of the error, LSE)等，以得到最佳化參數設定的方法[79][86]，其中[79]為本計劃中所採用方法的依據。

3.2 PID 控制器自調法則

在前面一節已經簡單介紹過幾種自調方式，接下來將針對其中的幾個方法作更深入的描述。

3.2.1 Ziegler-Nichols Method

考慮一個一階時間延遲系統，其轉移函數可以表示為

$$G(s) = \frac{K}{1+sT} e^{-sL} \quad (3-1)$$

其中 K 為穩態值常數增益， L 為時間延遲，而 T 為時間常數，並定義一變數 a 表示為下式(3-2)。

$$a = K \frac{L}{T} \quad (3-2)$$

根據(3-1)式得到典型之步階響應圖如圖 3.2 所示。

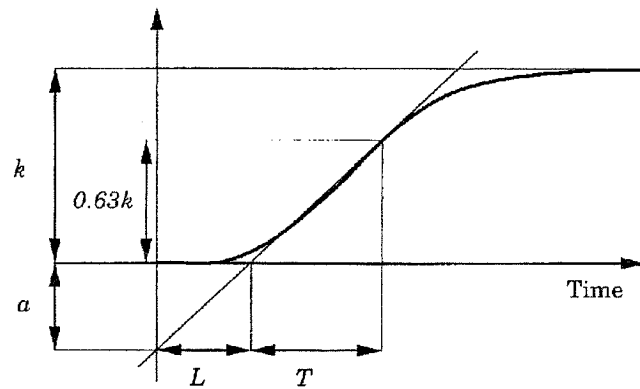


圖3.2 典型之步階響應圖

1942 年，Ziegler 和 Nichols 發展出一套經驗公式法則，根據開迴路步階響應即可適當決定 PID 控制器之參數，在圖 3.2 中，只需要 a 、 L 兩個變數即可以得到所要的參數值，規納出的 PID 控制器參數如表 3.2 所示。

表 3.2 Ziegler-Nichols 開迴路測試 PID 控制器參數

控制器	K_c	T_i	T_d
P	$1/a$	-	-
PI	$0.9/a$	$3L$	-
PID	$1.2/a$	$2L$	$L/2$

此方法仍有一些缺點，如在閉迴路步階響應時往往會造成一些不必要的振盪，而此缺點可藉由調整表 3-2 之數值即可改正，因此有許多類似的方法一樣是採用一個步階響應，並根據一個表格來得到適當的控制器參數 [81]-[82]，其中又以 [81] 最為常見。此外，在圖 3.2 中的 T 還可用來判斷 Ziegler-Nichols 方法是否適用，若是滿足 $0.1 < L/T < 0.6$ ，則此系統可用 Ziegler-Nichols 方法；若是 $L/T \geq 0.6$ ，則應該用其它可以補償延遲時間的自調法則；反之若是 $L/T \leq 0.1$ ，則應該採用較高階的控制補償器。

3.2.2 Relay Feedback Method

前面一節介紹了根據開迴路步階響應來決定控制器參數的方法，但是此種方法會造成系統在閉迴路時對於外部干擾的強韌性較差，因此有一些利用繼電器回授(relay feedback)控制的方法，其基本概念是觀察許多受控體有其極限週期振盪(limit cycle oscillations)。

考慮一繼電器回授控制系統，如圖 3.3 所示。其中 $G(s)$ 為受控體， y 為輸出訊號， u_c 為命令訊號，而 u 為輸入訊號。

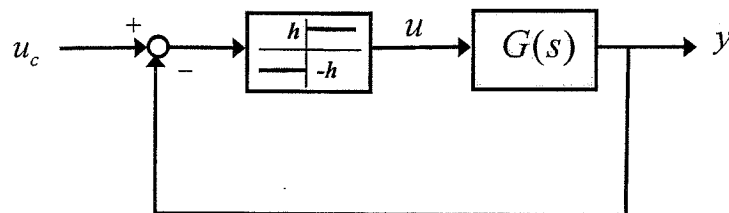


圖3.3 繼電器回授控制系統方塊圖

當命令訊號 $u_c = 0$ 時，輸入與輸出訊號波形會如圖 3.4 所示。輸入訊號 u 為一頻率 ω_u 的方波訊號，經由 Fourier 級數展開，可以將此輸入訊號表示為頻率為 ω_u 、 $3\omega_u$ 、 $5\omega_u$... 等的弦波訊號總和；而輸出訊號 y 可以忽略高次諧波，而近似為一弦波訊號，且頻率為 ω_u 。若令輸入方波的振幅為 h ，則其基本波

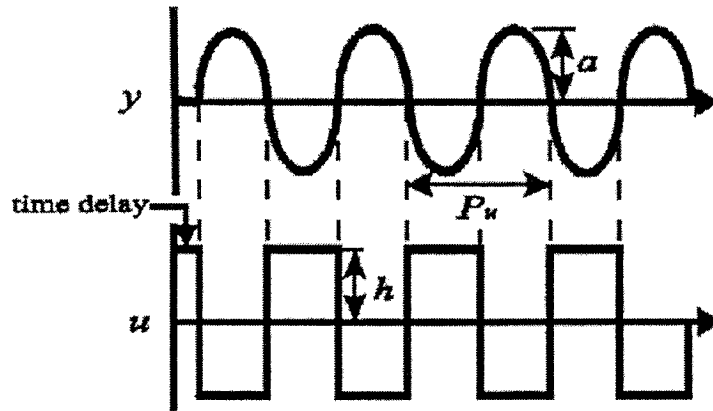


圖3.4 繼電器回授之輸入輸出波形

的振幅可以表示為 $4h/\pi$ ，故輸出訊號 y 的振幅 a 可表示為

$$a = \frac{4h}{\pi} |G(i\omega_u)| \quad (3-3)$$

同時輸入訊號的基本波與輸出訊號相位差為 180° ，故可推論在頻率為 ω_u 時，受控體會造成一相位落後 180° 的效果，其振盪可以表示為

$$\arg G(i\omega_u) = -\pi \quad (3-4a)$$

$$|G(i\omega_u)| = \frac{a\pi}{4h} = \frac{1}{K_u} \quad (3-4b)$$

其中 K_u 為系統的極限增益(ultimate gain)，亦為系統 $G(s)$ 在純增益控制下，滿足穩定度邊界條件的最大增益值；此外，可從 ω_u 得到極限週期(ultimate period) T_u 為

$$T_u = \frac{2\pi}{\omega_u} \quad (3-4c)$$

利用繼電器回授可以得到閉迴路控制響應，且 Ziegler 和 Nichols 也發展出一套經驗法則，根據 K_u 和 T_u 來適當決定 PID 控制器的參數值，如表 3.3 所示。然而 Ziegler-Nichols 方法對於某些系統還是會有缺點，如過大的超越量(overshoot)，因此有一些根據 Ziegler-Nichols 作改善的方法[82]-[83]，其 PID 控制器參數如表 3.4 所示。

表 3.3 Ziegler-Nichols 閉迴路測試之 PID 控制器參數

控制器	K_c	T_i	T_d
P	$K_u / 2$	-	-
PI	$K_u / 2.2$	$P_u / 1.2$	-
PID	$K_u / 1.7$	$P_u / 2$	$P_u / 8$

表 3.4 改良 Ziegler-Nichols 閉迴路測試之 PID 控制器參數

控制器	K_c	T_i	T_d
Z-N	$K_u / 1.7$	$P_u / 2$	$P_u / 8$
Some Overshoot	$K_u / 3$	$P_u / 2$	$P_u / 8$
No overshoot	$K_u / 5$	$P_u / 2$	$P_u / 8$

3.2.3 Frequency-Domain Method

此類方法乃是根據相位邊界(phase margin)及增益邊界(gain margin) 的概念來設計的。圖 3.5 為一簡單的閉迴路系統方塊圖。

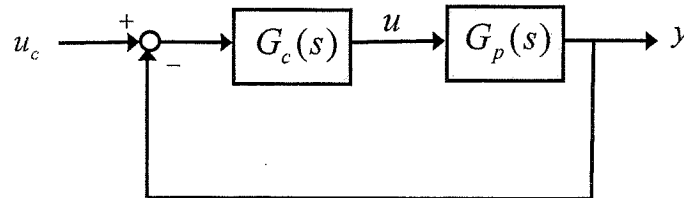


圖3.5 閉迴路控制系統方塊圖

其相位邊界 ϕ_m 可表示為

$$\phi_m = \pi + \arg G_p(i\omega_g)G_c(i\omega_g) \quad (3-5a)$$

其中 $|G_p(i\omega_g)G_c(i\omega_g)|=1$ ，且 ω_g 為增益交越頻率(gain cross-over frequency)；

而其增益邊界 A_m 為

$$A_m = \frac{1}{|G_p(i\omega_u)G_c(i\omega_u)|} \quad (3-5b)$$

其中 ω_u 為相位交越頻率(phase cross-over frequency)。

若是(3-4b)及(3-4c)中的 K_u 及 T_u 可以得到的話，則可藉由決定相位邊界 ϕ_m 的值，並透過一個簡單的式子來得到 PID 控制器的參數[84]，如

$$K_c = K_u \cos \phi_m \quad (3-6a)$$

$$T_i = \frac{T_u}{4\pi} \left(\tan \phi_m + \sqrt{1 + \tan^2 \phi_m} \right) \quad (3-6b)$$

$$T_d = T_i / 4 \quad (3-6c)$$

但這些式子只有在系統延遲時間 L 很小的時候才適用，否則會造成增益邊界很小。對於像(3-1)式之一階時間延遲系統而言，在[85]中介紹了一種可以選擇系統增益邊界 A_m 來得到 PI 控制器參數的方法，如

$$K_c = \frac{\pi T}{2A_m K L} \quad (3-7a)$$

$$T_i = T \quad (3-7b)$$

此方法同時也可以得到相位邊界 ϕ_m 為

$$\phi_m = \frac{\pi}{2} (1 - 1/A_m) \quad (3-8)$$

3.2.4 Pole Placement Method

考慮一個二階系統

$$G_p(s) = \frac{K_p}{(1+sT_1)(1+sT_2)} \quad (3-9)$$

此系統共有三個變數，由於 PID 控制器亦包含三個可控制的參數，因此應用 PID 控制器可以隨意決定此閉迴路系統的極點，若其 PID 控制器之轉移函數 $G_c(s)$ 為

$$G_c(s) = \frac{K(1+sT_i+s^2T_iT_d)}{sT_i} \quad (3-10)$$

且其方塊圖如圖 3.5 所示，則可得到其特徵方程式為

$$s^3 + s^2 \left(\frac{1}{T_1} + \frac{1}{T_2} + \frac{K_p K T_d}{T_1 T_2} \right) + s \left(\frac{1}{T_1 T_2} + \frac{K_p K}{T_1 T_2} \right) + \frac{K_p K}{T_1 T_1 T_2} = 0 \quad (3-11)$$

假設為一標準三階系統，則其閉迴路特徵方程式可表示為

$$(s + \alpha\omega)(s^2 + 2\zeta\omega s + \omega^2) = 0 \quad (3-12)$$

結合(3-11)與(3-12)式，可以得到

$$\frac{1}{T_1} + \frac{1}{T_2} + \frac{K_p K T_d}{T_1 T_2} = \omega(\alpha + 2\zeta) \quad (3-13a)$$

$$\frac{1}{T_1 T_2} + \frac{K_p K}{T_1 T_2} = \omega^2(1 + 2\zeta\alpha) \quad (3-13b)$$

$$\frac{K_p K}{T_1 T_1 T_2} = \alpha\omega^3 \quad (3-13c)$$

經過化簡可以得到 PID 控制器的參數分別為

$$K = \frac{T_1 T_2 \omega^2 (1 + 2\zeta\alpha) - 1}{K_p} \quad (3-14a)$$

$$T_i = \frac{T_1 T_2 \omega^2 (1 + 2\zeta\alpha) - 1}{T_1 T_2 \alpha \omega^3} \quad (3-14b)$$

$$T_d = \frac{T_1 T_2 \omega (\alpha + 2\zeta) - T_1 - T_2}{T_1 T_2 (1 + 2\zeta\alpha) \omega^3 - 1} \quad (3-14c)$$

因此可以透過選擇 α 、 ω 及 ζ ，和一些系統鑑別的方法，來適當決定PID控制器參數；值得注意的是，當 $\omega = \omega_c = \frac{T_1 + T_2}{T_1 T_2 (\alpha + 2\zeta)}$ 時， $T_d = 0$ ，即PID控制器變成一個純PI控制器；且若是 $\omega < \omega_c$ ，則 $T_d < 0$ ，因此頻率 ω_c 為頻寬之最小邊界；故 ω 的選擇便十分重要。

3.2.5 Optimization Method

此類方法是定義一些誤差的目標函數(cost function),如 IAE、ISE 與 LSE 等,並且在合理的範圍內去尋找最小的目標函數,以得到 PID 控制器的最佳解。在此提出一個可同時調整 PI 控制器兩個參數的方法[79],並以加上權重因子(weighting factor)的 IAE 作為目標函數,圖 3.6 為改良式雙自由度(Modified Two-Degree-Of-Freedom, M2DOF)的調整法則示意圖。

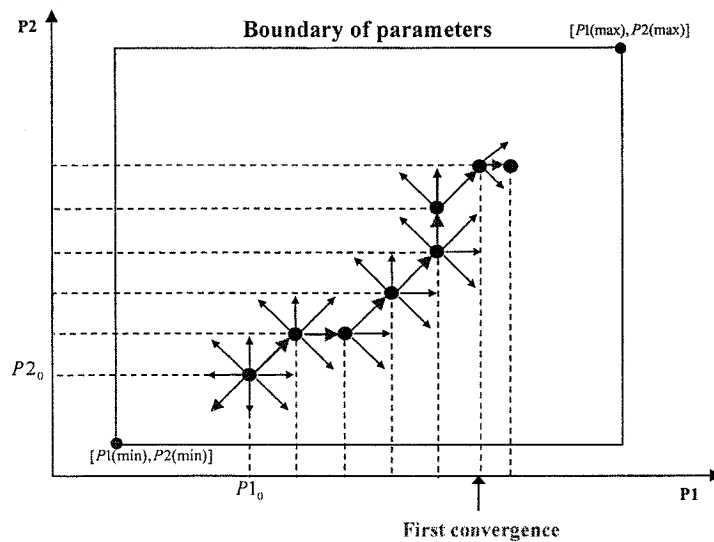


圖3.6 M2DOF之自調法則

首先決定一組 PI 控制器參數的初始值,接著根據回授與命令訊號所造成的誤差,計算出一個目標函數值,即當作此組控制器參數的性能指標(performance index),接下來再改變 PI 控制器參數,以量測另一組的性能指標,依序作完 9 組資料來作比對,求出最小的性能指標即為此 9 組內的相對最佳值;將之前參數組所移動的軌跡記錄下來,接下來會根據此軌跡趨勢去尋找相對應的最佳解,以減少測試組的數目,增快搜尋速度;持續上述步驟,一旦發現連續兩次找到同一組參數為相對最佳參數時,即將此組參數視為絕對最佳參數,此時會達到第一次收斂;接下來縮小控制器參數變動範圍,再重複作第二次的測試,以得到更精確的最佳參數解。

第四章

參考模型式數位適應伺服控制

本章主要分為四個部分，第一部份是推導永磁式交流馬達的數學模型；第二部分是說明所採用的參考模型以及如何決定參考模型的轉移函數；第三部分是適應控制機制的推導；第四部分是根據所推導的適應控制機制，進行數位式適應伺服控制的系統模擬。

4.1 永磁式交流馬達數學模型推導

根據前期計畫第二章所推導之永磁式交流馬達數學模型方程式，可建立如圖 4.1 永磁式交流馬達動態模型方塊圖。當馬達在定轉矩區域操作時，不需提供磁場電流，即 $i_{ds} = 0$ ， $\bar{i}_s = i_{qs}$ ，將此條件與圖 4.1 永磁式交流馬達動態模型方塊圖結合，可以推得完全解耦時永磁式交流馬達動態模型，如圖 4.2 所示。由圖 4.2 發現，永磁式交流馬達藉由座標轉換的技巧得到 $d-q$ 軸座標系的動態模型，相當於永磁式直流馬達的模型，因此，在本章的分析將永磁式交流馬達的簡化模型視為一個永磁式直流馬達，藉以簡化原先複雜的多迴路控制系統，進而分析並設計數位式適應伺服控制器。表 4.1 為所採用之永磁式交流馬達的重要參數表。

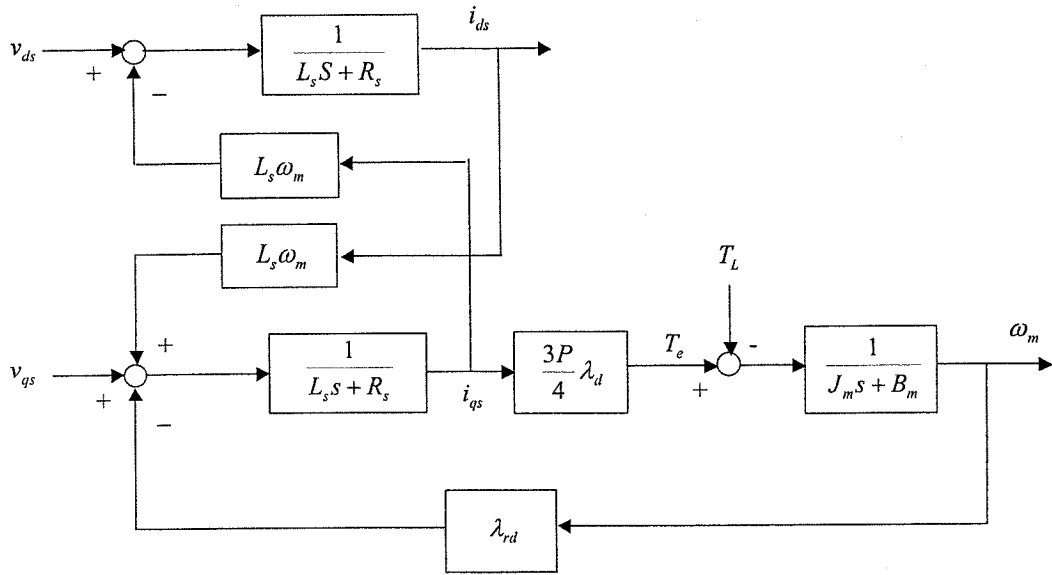


圖4.1 永磁式交流馬達動態模型

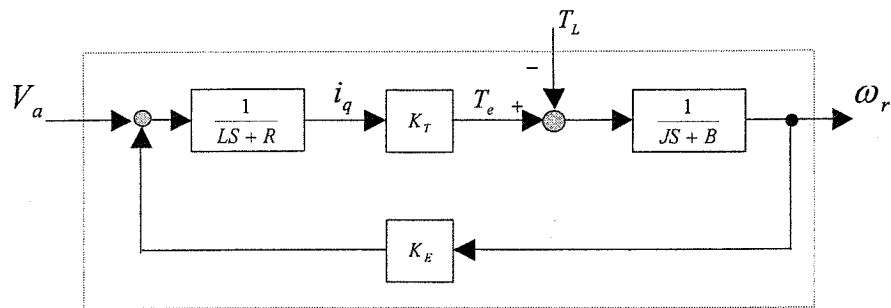


圖4.2 完全解耦永磁式交流馬達動態模型方塊圖

表4.1 馬達參數對照表

參數名稱	參數說明	數值
L_a	定子線圈電感	0.7 (mH)
R_a	定子線圈電阻	0.724 (Ω)
J	轉動慣量	8.05×10^{-5} ($\text{kg} \cdot \text{m}^2$)
B	黏滯摩擦係數	1×10^{-5} (N-m-sec)
K_T	扭矩常數	0.18 (N-m/amp)
K_E	反抗電動勢常數	0.18 (V/(rad/sec))

4.2 參考模型推導

參考模型的響應即為整體控制系統所希望的響應，理論上較佳的參考模型可以得到較佳的系統響應，但實際的控制系統響有其最佳的響應限制，所以，在決定參考模型時，需要考量實際系統所能表現出來的最佳響應，才能得到一個最適當的參考模型。

本計畫選用標準的二階轉移函數作為系統的參考模型架構，其閉迴路轉移函數如下

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4-1)$$

圖 4.3 為標準二階轉移函數的步階響應圖，其中最大超越量發生的時間為

$$t_{\max} = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \quad (4-2)$$

最大超越量數值為

$$\sigma_m = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \quad (4-3)$$

從 10% 上升到 90% 所需的時間為

$$t_r = \frac{0.8 + 2.5\zeta}{\omega_n} \quad 0 < \zeta < 1 \quad (4-4)$$

進入穩態的時間為(小於 5% 誤差量)

$$\begin{aligned} t_s &= \frac{4.5\zeta}{\omega_n} & \zeta > 0.69 \\ t_s &\cong \frac{3.2}{\zeta\omega_n} & 0 < \zeta < 0.69 \end{aligned} \quad (4-5)$$

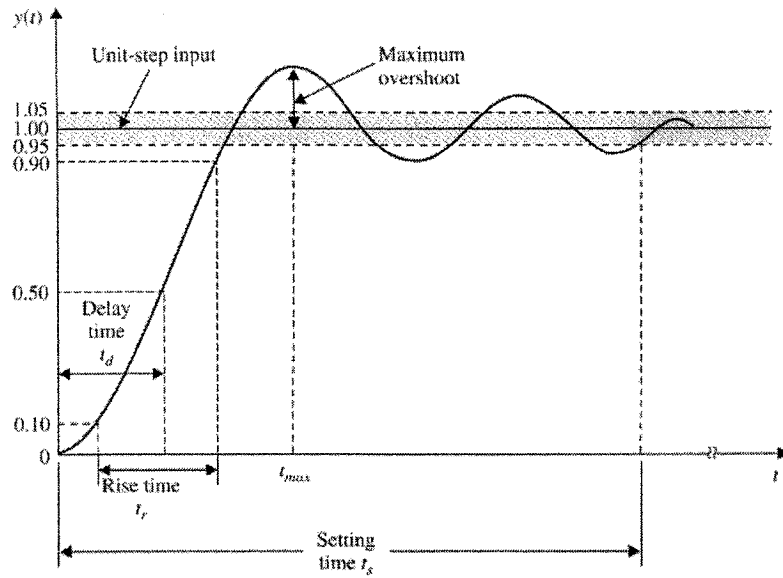


圖4.3 標準二階轉移函數之步階響應圖

一般而言，永磁式交流馬達的工作電壓、電流有一定的限制，根據伺服馬達的額定規格表，本計畫所採用之最大工作電壓為 150 伏特，最大工作電流為 10 安培，當 q 軸電流在 10 安培的限制之下，可推導出馬達的最快加速曲線，做為參考模型最大上升時間的限制。

根據圖 4.2 簡化之永磁式交流馬達動態模型，可將電流與轉速的關係式表示為

$$\frac{\omega_r}{i_q^*} = \frac{K_T}{JS + B} \quad (4-6)$$

假設 q 軸電流為 10 安培的步階響應，可將上式表示為

$$\omega_r(S) = \frac{10}{S} \cdot \frac{\frac{K_T}{J}}{S + \frac{B}{J}} \quad (4-7)$$

將上式轉換成時域(time domain)，可得

$$\omega_r(t) = \frac{10 \cdot K_T}{B} (1 - e^{-\frac{B}{J}t}) \quad (4-8)$$

將上式的 exponential 項展開，並將轉速轉換成 RPM，可得

$$\omega_r(t) = \frac{60}{2\pi} \cdot \frac{10 \cdot K_T}{J} \cdot t \quad (4-9)$$

根據上式可得到如圖 4.4 馬達的最佳加速曲線圖，作為選擇參考模型的標準。

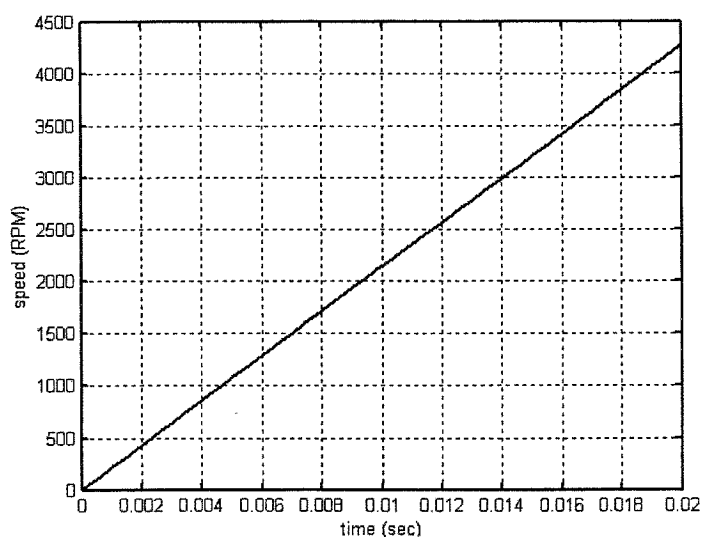


圖4.4 在10A電流限制下馬達的最佳加速曲線圖

假設馬達的目標轉速為 1500RPM，由圖 4.4 或式子(4-9) 可知，在 q 軸電流限制為 10 安培的條件限制下，約需 7ms 的時間，馬達方能從靜止加速到 1500RPM，由於此條件是最佳的狀況，在考慮 PWM 開關切換、數位取樣..等情況，假設參考模型的 t_r 為 10ms。另外，根據經驗法則可知，標準二階轉移函數的 ζ 值，若為 0.707，則系統的實際響應約有 5% 的超越量；若為 1，則系統的實際響應不會有超越量發生，在馬達速度控制系統，希望系統的響應有微量的超越量，因此，假設參考模型的 ζ 值為 0.9，使得系統響應的最大超越量小於 1%。

將 $t_r=10\text{ms}$ 、 $\zeta=0.9$ 代入式子(4-4)，得到 $\omega_n=305$ ，接著將 ζ 與 ω_n 代入式子(4-1)，可得參考模型之標準二階轉移函數為

$$\frac{\omega_r(s)}{\omega_r^*(s)} = \frac{93025}{s^2 + 549s + 93025} \quad (4-10)$$

取樣時間為 1ms，數位化後轉移函數為

$$\frac{\omega_r(z)}{\omega_r^*(z)} = \frac{0.0388z + 0.0323}{z^2 - 1.5065z + 0.5775} \quad (4-11)$$

4.3 適應控制機制推導

圖 4.5 為本計畫所採用的參考模型式數位伺服控制系統的系統方塊圖，速度控制命令 ω_r^* ，經過標準二階的參考模型，得到期望之伺服控制系統速度響應 ω_m ；速度命令 ω_r^* ，經過 PI 速度控制器產生 q 軸電流控制訊號 i_{qs}^* ，並與 MIT rule based adaptation mechanism 輸出之 q 軸電流補償訊號 i_{qc}^* 相加，得到真正的 q 軸電流控制命令 i_q^* ，然後在馬達的輸出得到實際之速度響應 ω_r 。

其中，MIT rule based adaptation mechanism 根據速度命令 ω_r^* 、參考模型速度響應與實際速度響應的誤差量，以及速度控制器產生之 q 軸電流控制訊號 i_{qs}^* ，藉由 MIT rule based 計算，產生一個 q 軸電流的補償訊號 i_{qc}^* ，使得馬達控制系統之實際速度響應與期望之速度響應相同。

在推導適應控制機制的過程中，首先定義誤差函式(error function)為

$$e = \omega_m - \omega_r \quad (4-12)$$

而其目標函式(cost function)則定義為

$$J(\theta) = \frac{1}{2} \cdot e^2 \quad (4-13)$$

接著根據 MIT rule，推導出其 update equation 為

$$i_{qc}^* = -\gamma \cdot \text{sign} \left(\frac{\partial J}{\partial i_{qc}^*} \right) \cdot e \quad (4-14)$$

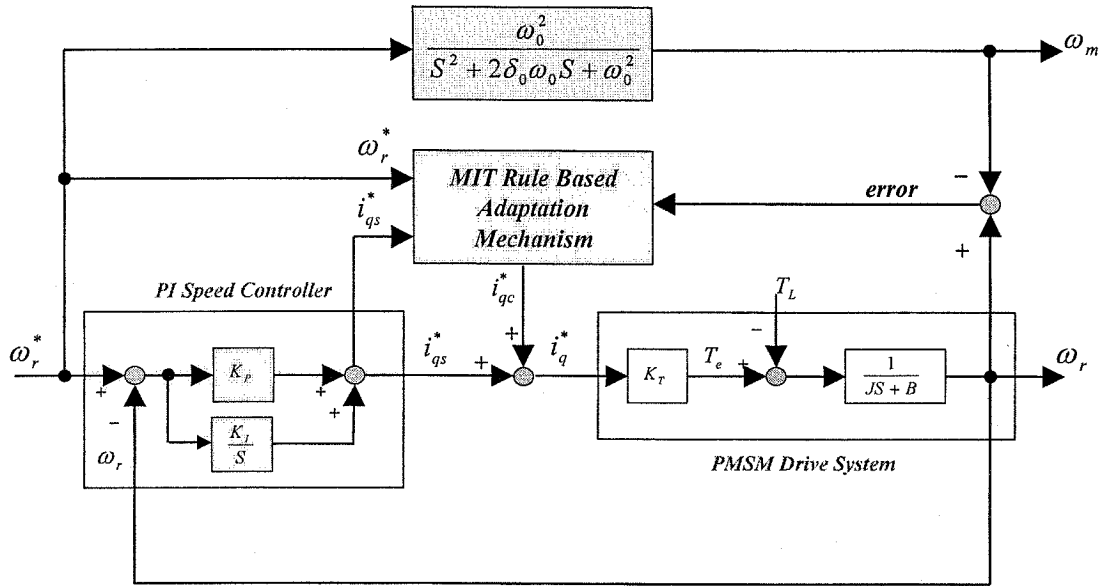


圖4.5 模型配適伺服控制系統方塊圖

其中 γ 稱為學習因子(learning factor)，將其表示為

$$\gamma = [a1 \quad a2] \cdot \begin{bmatrix} \omega_r^* \\ i_{qs}^* \end{bmatrix} \quad (4-15)$$

(4-12)-(4-15)式即為所設計出的適應控制法則，下一節將根據所推導的適應控制機制，進行整個數位式適應伺服控制系統的模擬，驗證其適應控制機制是否能夠滿足要求的響應。

4.4 適應控制系統模擬

圖 4.6 是根據本計畫提出之參考模型式伺服控制系統所建立之全數位 SIMULINK 模擬方塊圖，其中包含了數位化之參考模型、數位化之馬達模型、電流控制器、速度控制器，以及利用 MATLAB 函式所實現之適應控制機制。圖 4.7 是以 SIMULINK 實現之數位速度控制器的模擬方塊圖，採用 PI 控制器，取樣時間為 1ms；圖 4.8 是以 SIMULINK 實現之數位式電流控制器模擬方塊圖，採用 P 控制器，取樣時間為 100us；圖 4.9 為數位化之馬達模型的 SIMULINK 模擬方塊圖，取樣時間為 100us。

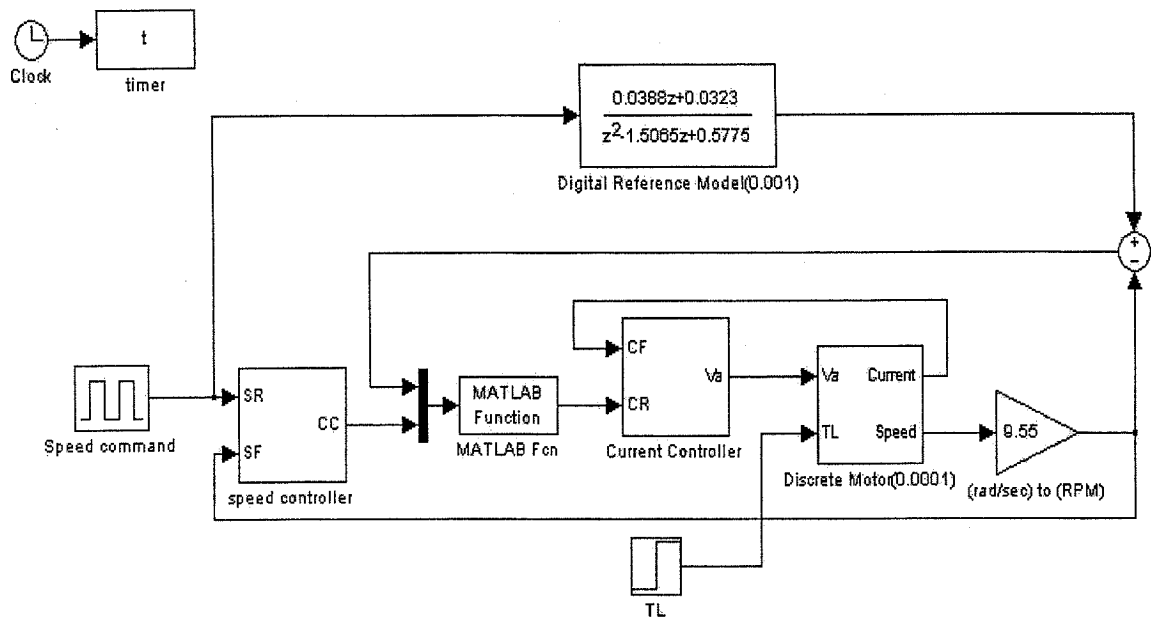


圖4.6 參考模型式適應伺服控制系統之模擬方塊圖

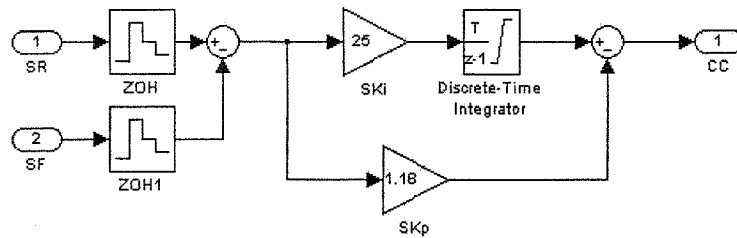


圖4.7 數位式速度控制迴路之模擬方塊圖

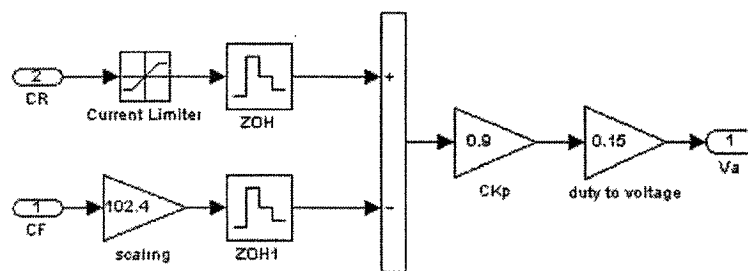


圖4.8 數位式電流控制迴路之模擬方塊圖

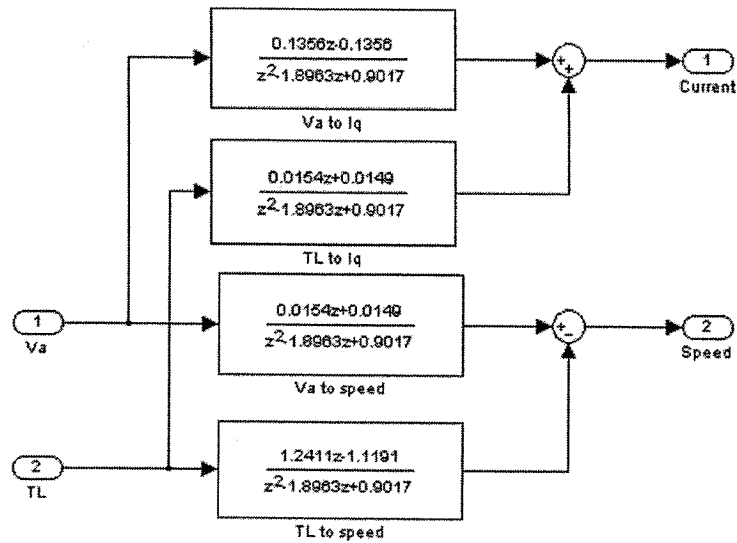


圖4.9 馬達數位化之模擬方塊圖

圖 4.10 是僅採用傳統 PI 控制器而未加上適應控制機制的模擬結果，由模擬結果可知，由於在傳統控制架構上並沒有根據參考模型作適當的補償控制，故實際的速度響應往往無法滿足要求的速度響應，必須使用試誤法的方式調整 PI 控制器的控制參數，但這樣的過程需要耗費許多時間。

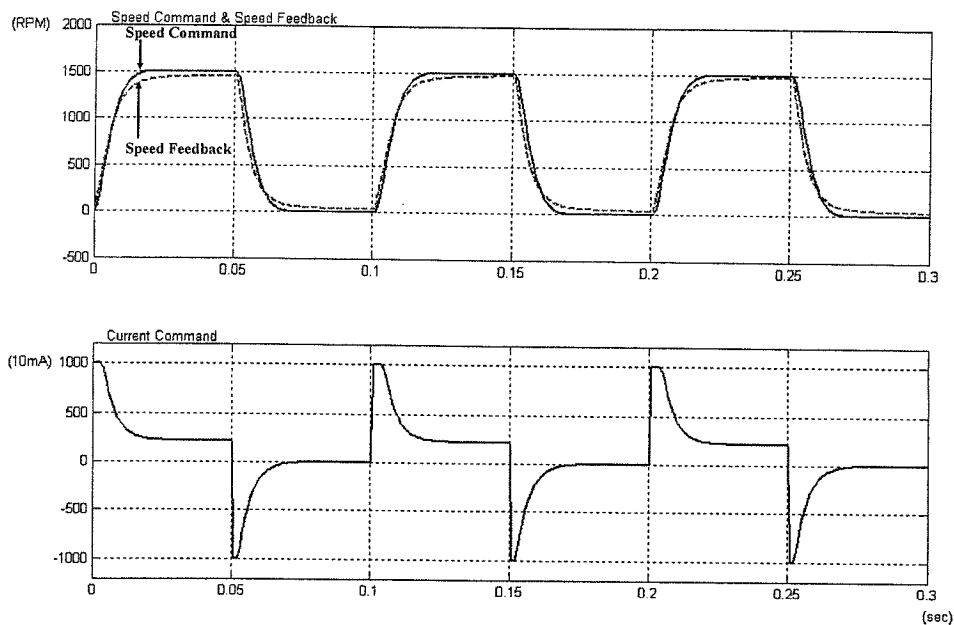


圖4.10 PI伺服控制系統之模擬結果

圖 4.11 是加上適應控制機制的伺服控制模擬結果，與圖 4.10 的速度響應結果比較，發現藉由適應控制機制，可以有效的補償實際速度響應與預期速度響應讀誤差量，使其無論馬達參數變動或是負載變動都能夠準確的追隨參考模型的理想響應。圖 4.11 的第三個波形圖是適應控制機制所計算出來的控制補償力(control effort)。

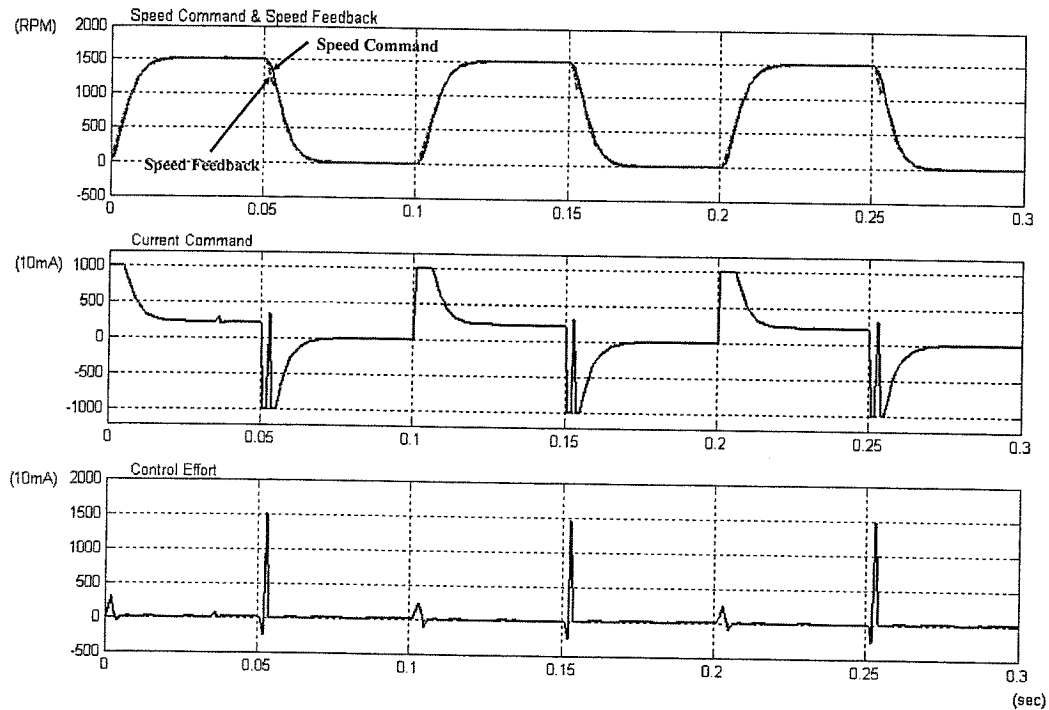


圖4.11 適應控制機制加入PI伺服控制系統模擬結果(無負載)

圖 4.12 是測試加上適應控制機制的伺服控制系統對於負載變動的模擬結果，在 0.1 秒時加入一個 0.2N-m 的負載扭矩，由圖可知適應控制機制的控制補償力具有學習的機制，可在第三個週期明顯看出速度響應已修正到與預期之速度響應相同。

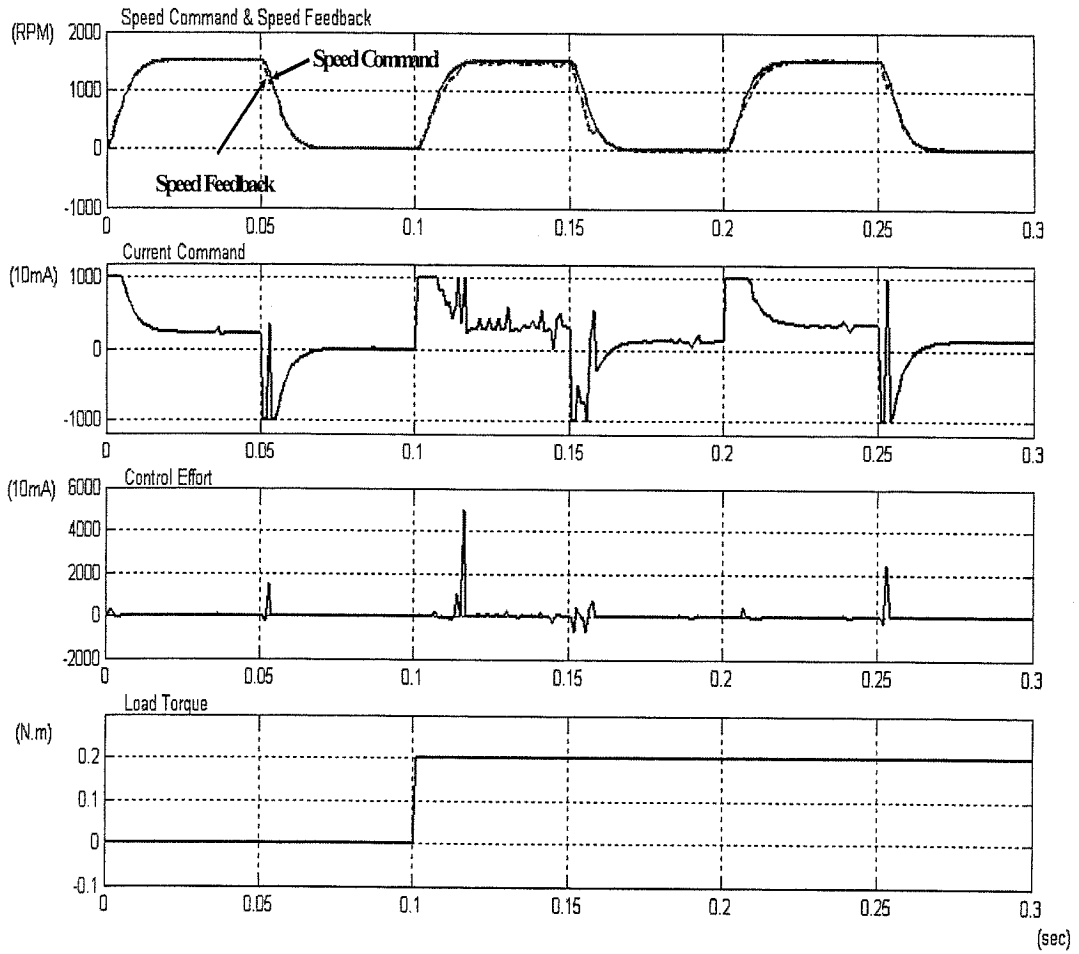


圖4.12 適應控制機制加入PI伺服控制系統模擬結果
(0.1sec加入0.2 N-m的扭矩負載)

第五章

MATLAB 參數自調控制方法

5.1 原理說明

在第三章介紹了許多 PID 參數自動調整的方法，其中本計劃所採用的方法是屬於「最佳化方法(optimization methods)」，此自調控制器的設計中包含了幾個重點：首先是選定需進行自動調整的參數值，並且根據所要求的規格定出這些參數值的合理邊界值，以及需調整的解析度；接著送出測試訊號，可用一連串的步階訊號輸入，得到相對應的響應時間及誤差後，利用所定義的目標函數(cost function)來算出在此參數組下的性能指數(performance index)，並配合一個可以同時調整兩個參數的調整法則(M2DOF)決定下一組測試參數值，透過 RS-232 網路傳輸，來進行線上自調，一直反覆進行直到獲得最小的性能指數值，即表示所得到的參數組為某個範圍內的「相對最佳值」。圖 5.1 為包含自調控制器之永磁交流馬達伺服控制方塊圖，在本計畫中將針對速度控制迴路進行設計自調控制器，以下將逐一介紹以 MATLAB 為基礎的設計流程。

5.1.1 參數邊界值及初始值的決定

此為進行自調控制前的第一步驟。在決定了需進行自動調整的參數後，設計者需根據各個硬體的規格，以及響應的要求來定出各個參數的合理邊界範圍。此範圍的決定為最基本卻也最重要的，因為邊界值決定的不好，可能

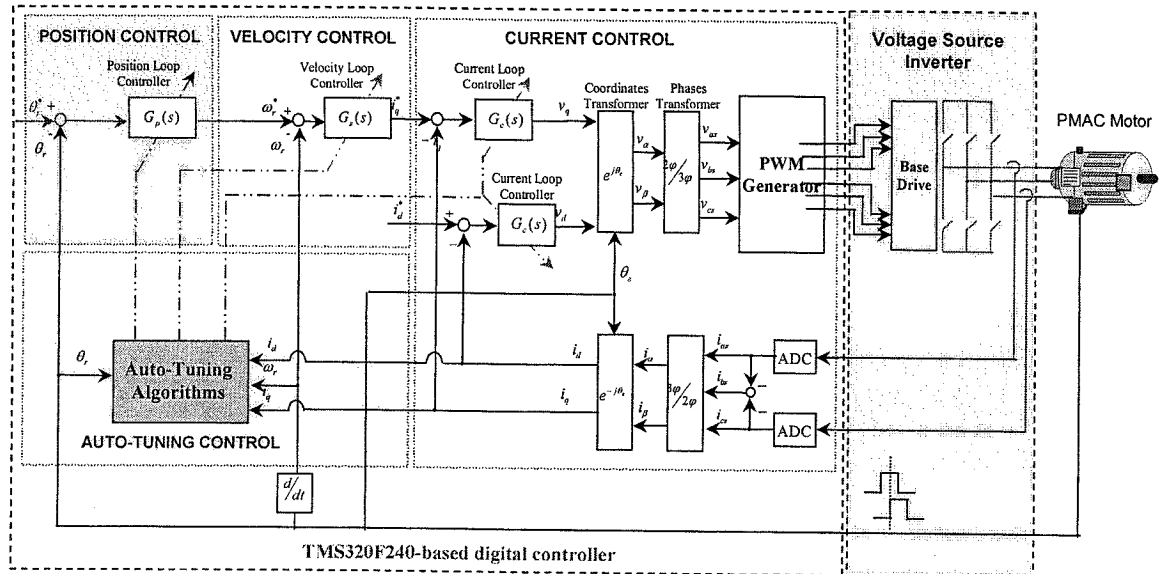


圖5.1 自調式之永磁交流馬達伺服控制方塊圖

會影響整個自調控制器的性能，舉例來說，邊界值定的太窄，可能無法找到真正的理想參數值；相反的，定的太寬，會增加自調完成的時間，甚至可能造成過大的增益，使得整個系統不穩定。

一旦將邊界值決定了後，便要設定一組各個參數的初始值，此初始值的可根據分析系統頻寬及控制器特性所得到的結果決定，或是隨意選取符合邊界條件的任何一點，設計者也可利用這個初始值來檢驗參數合理邊界範圍。同樣的，此初始值也會影響自調控制器的整體性能及收斂所需時間。

5.1.2 測試訊號的產生

設計者可自行選擇輸入的測試訊號，例如可產生一些隨機的訊號，再利用隨機搜尋的技巧進行最佳化，通常是選擇一連串週期性的步階訊號作為測試輸入訊號，如此即可利用簡單的計算得到步階響應的響應值，如上升時間、暫態誤差、穩態誤差等，接下來就可以進行目標函數的計算。值得注意的是，由於受到 RS-232 傳輸非同步與非即時性的影響，加上每筆傳輸的資料量也都是固定的，故在進行資料擷取的時候，必須要能夠保證所擷取的資料包含了一個完整的步階響應，因此以責任比為 50% 的週期性步階命令而

言，每次至少需要擷取 1.5 個週期，如圖 5.2(a)所示；使用者亦可定義責任比非 50%的週期性步階命令，如圖 5.2(b)所示，希望能夠在同樣的資料量下，得到較長的穩態響應，而不會因為週期過度頻繁的命令使得馬達產生振盪。

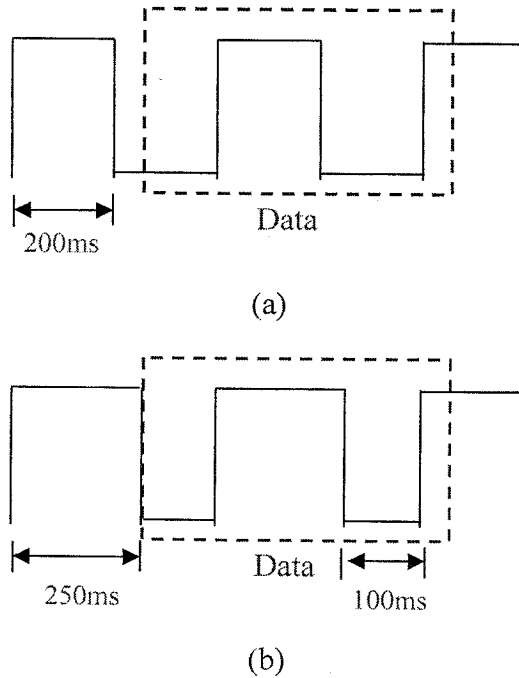


圖5.2 測試訊號與資料擷取波形圖 (a)責任比為50% (b)責任比不為50%

5.1.3 目標函數的定義

目標函數可根據設計者的要求自行定義，在本計劃中所定義的目標函數為：

$$f(t) = w_1(e(t)) \int_{t_1}^{t_2} |e(t)| dt + w_2 \int_{t_2}^{t_3} |e(t)| dt \quad (5-1a)$$

$$\begin{aligned} w_1(e(t)) &= w_{11} \text{ (high gain)} && \text{for } e(t) < 0 \text{ (overshoot)} \\ &= w_{12} \text{ (low gain)} && \text{for } e(t) > 0 \end{aligned} \quad (5-1b)$$

其中 $f(t)$ 為目標函數， $e(t)$ 為誤差函數， $t_1 \sim t_2$ 為暫態響應積分之起始及結束時間， $t_2 \sim t_3$ 為進入穩態響應後積分之起始及結束時間， $w_1(e(t))$ 為一隨著誤差改變之權重函數(weighting function)，當響應產生過衝現象(overshoot)

時，利用一個很大的增益來增加目標函數的值， w_2 為穩態時的權重因子 (weighting factor)，設計者亦可根據要求規格自行調整不同的權重因子，以得到最佳的自調控制結果。

5.1.4 調整法則

在第三章中已經簡單的介紹過本計劃所採用的調整法則 M2DOF，如圖 5.3 所示。在初始化設定時已經分別選定了兩個參數調整的間格，此間格表示參數調整的解析度，若間格越大，自調收斂時間會減少，但是參數的精準度會降低；相反的，若是間格越小，雖然參數調整的精準度會增加，但卻會增加自調收斂時間。因此將自調過程分為兩段，第一段先將調整間格放大，以盡快得到參數收斂區間，接著第二段再將間格變小，進行參數細部微調，以彌補參數精確度不足的缺點。此方法可有效減少參數自調收斂時間，同時亦可更精確的得到參數最佳值。此外此調整法則亦可記錄前一次調整參數的趨勢，來判斷下一次參數搜尋軌跡，以減少自調收斂時間。

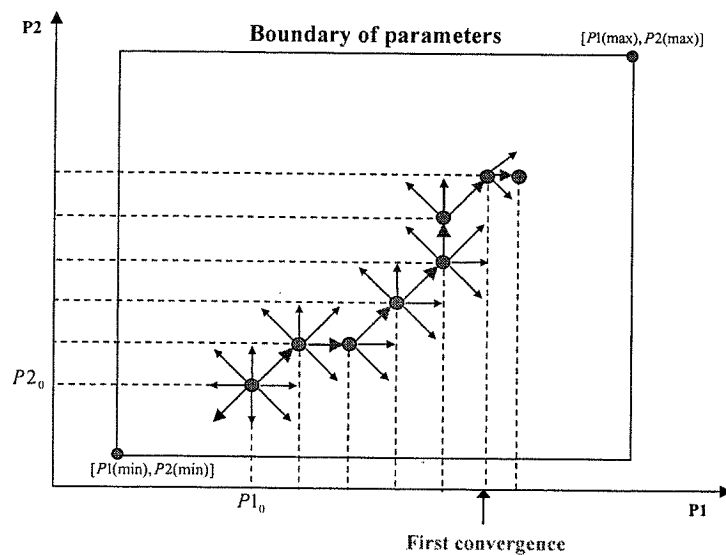


圖5.3 M2DOF的自調法則示意圖

5.2 MATLAB軟體實現

利用軟體設計的方式來實現整個自調控制的機制有許多優點，例如透過模組化的設計，設計者可在日後輕易替換整個機制中的某一部份，而不需重新設計，整個自調控制器設計流程圖如圖 5.4 所示。

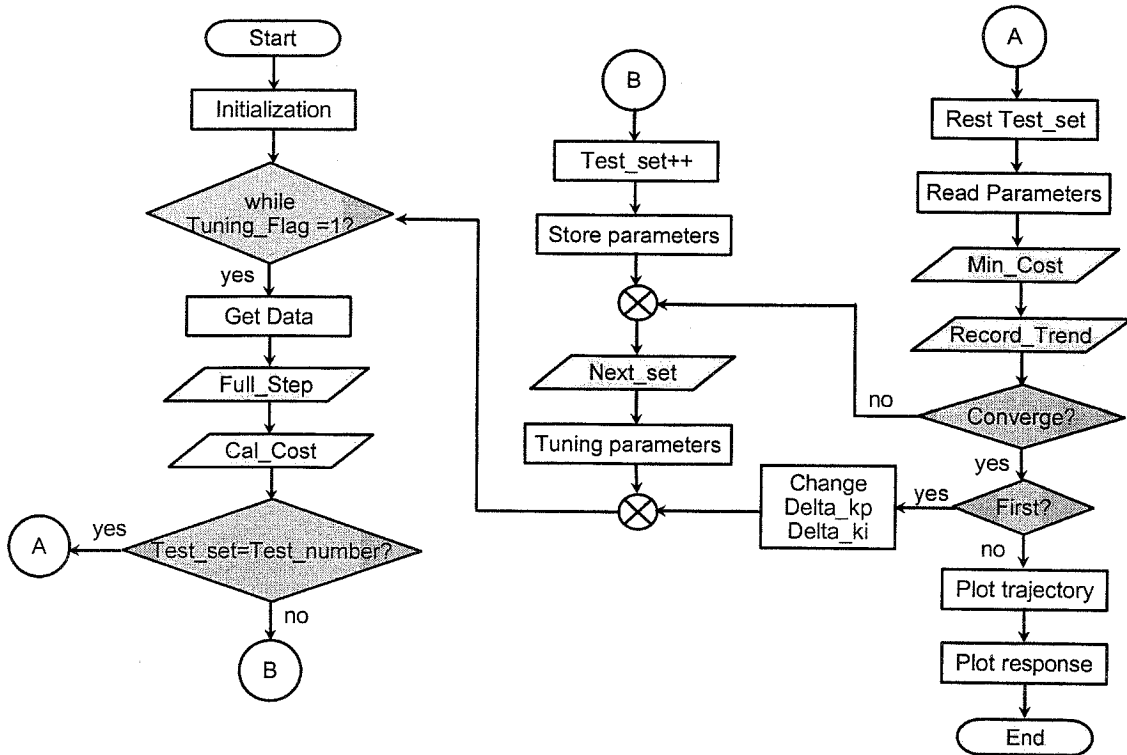


圖5.4 MATLAB自調控制器設計流程圖

5.2.1 主程式說明

此為本計劃所發展之自調控制器的主程式，將前面一節所提出的原理用軟體實現，並採用副函式的概念將程式模組化，接下來將說明各個副函式的程式內容。

```

Init; %初始化設定
tic; %開始計算程式執行時間

while Tuning_flag == 1 %開始自調

    Flag_old = Flag_new; %判斷資料存放在哪一暫存區內
    Flag_new = Read_flag(buf_sel);
    while Flag_new == Flag_old
        Flag_new = Read_flag(buf_sel);
    end
    if Flag_new==0 %擷取命令及響應資料
        Command=Read_data(1,3);
        Feedback=Read_data(1,4);
    elseif Flag_new==1
        Command=Read_data(0,3);
        Feedback=Read_data(0,4);
    end

    [Start_index,End_index]=Full_step(Command); %擷取一完整步階響應

    if Start_index(1) > End_index(1)
        if length(End_index)<2
            End_index = 300;
        else
            End_index(1) = End_index(2);
        end
    end
end

Cost = Cal_cost(Command,Feedback,Start_index(1),End_index(1)); %計算Cost值%
Cost_buf(Test_set)=Cost(1);

    if Test_set == 1 %將初始值存到一陣列中%
        Kp_buf(Test_set)=Kp(1);
        Ki_buf(Test_set)=Ki(1);
        Command_buf(Test_set)=Command(1);
        Feedback_buf(Test_set)=Feedback(1);
    end
end
if Test_set < Test_number
    [Kp_new,Ki_new] = Next_set(Kp,Ki,Test_set,Test_number,Tuning_trend); %尋找下組測試點%
    if Kp_new > Kp_limit %判斷是否已到boundart condition%
        Kp_new = Kp_limit;
    elseif Kp_new < 1
        Kp_new = 1;
    end
    if Ki_new > Ki_limit
        Ki_new = Ki_limit;
    elseif Ki_new < 1
        Ki_new = 1;
    end
end
end

```

```

Write(kvp,Kp_new); %送出下一組參數值
Write(kvi,Ki_new);
Write_flag(para_tune,1);
Test_set = Test_set + 1;
Kp_buf(Test_set) = Kp_new;
Ki_buf(Test_set) = Ki_new;
Command_buf(Test_set) = Command(1);
Feedback_buf(Test_set) = Feedback(1);
else
Kp_plot(Main_index) = Kp; %紀錄Kp,Ki變化軌跡
Ki_plot(Main_index) = Ki;
Main_index = Main_index + 1;
[Kp,Ki,Cost_min]=Min_cost(Kp_buf,Ki_buf,Cost_buf); %找出最小之Cost值相對應的參數組
Kp_plot(Main_index) = Kp; %紀錄Kp,Ki變化軌跡
Ki_plot(Main_index) = Ki;

[Test_number,Tuning_trend]=Record_trend(Kp_plot(Main_index),Kp_plot(Main_index-1),Ki_plo
t(Main_index),Ki_plot(Main_index-1));

%畫圖
figure(1);
plot(Kp_plot,Ki_plot);
grid;
figure(2);
plot(t,Feedback,t,Command);
grid;

%將參數重置
Test_set = 1;
Cost_buf=0;
Kp_buf=0;
Ki_buf=0;

%判斷是否收斂
Finish = Converge(Kp_plot,Ki_plot,Kp,Ki,Main_index);
if Finish == 1
if First == 1 %第一次收斂
First = 0;
Delta_Kp = 2; %改變參數變化間格
Delta_Ki = 20;
else
Tuning_flag = 0; %第二次收斂
end
end
end
end
end
toc; %結束計算程式執行時間

```

5.2.2 Full_step 副函式

此函式主要是從擷取到的資料中，找出一個完整的步階響應，以進行誤差計算，圖 5.5 為此函式的輸入輸出方塊圖。

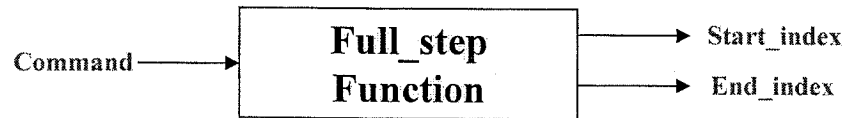


圖5.5 Full_step副函式輸入輸出方塊圖

```
%副函式-Full_step:擷取一個完整的step%
function [Full_step_out1,Full_step_out2] = Full_step( Full_step_in )
j = 1;
k = 1;
for i=1:299
    if (Full_step_in(i+1) - Full_step_in(i)) > 50
        Full_step_out1(j) = i+1;
        j = j + 1;
    end
    if (Full_step_in(i+1) - Full_step_in(i)) < -50
        Full_step_out2(k) = i+1;
        k = k + 1;
    end
end
end
```

5.2.3 Cal_cost 副函式

此函式主要是根據所定義的目標函數，來計算出此時的性能指標，圖 5.6 為此函式的輸入輸出方塊圖。

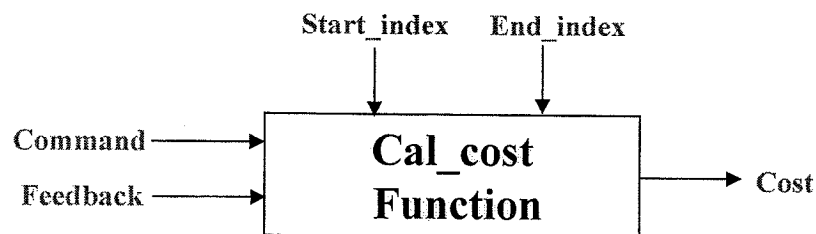


圖5.6 Cal_cost副函式輸入輸出方塊圖

```

%副函式-Cal_cost:計算IAE值%
function Cal_cost_out = Cal_cost( Cal_cost_in1, Cal_cost_in2, Start_addr, End_addr )
Cal_cost_out = 0;

for i=(Start_addr):(Start_addr+10)    %暫態響應%
    if Cal_cost_in2(i) > Cal_cost_in1(i) % overshoot %
        Cal_cost_out = Cal_cost_out + 100*( Cal_cost_in2(i) - Cal_cost_in1(i) );
    else
        Cal_cost_out = Cal_cost_out + 10*( Cal_cost_in1(i) - Cal_cost_in2(i) );
    end
end

for i=(Start_addr+11):(End_addr)    %穩態響應%
    Cal_cost_out = Cal_cost_out + abs((Cal_cost_in1(i) - Cal_cost_in2(i)));
end
Cal_cost_out = round(Cal_cost_out);

```

5.2.4 Min_cost 副函式

此函式主要是根據各組參數所求得之目標函數，找尋其中最小的性能指標，以及其相對應的參數解。圖 5.7 為此函式的輸入輸出方塊圖。

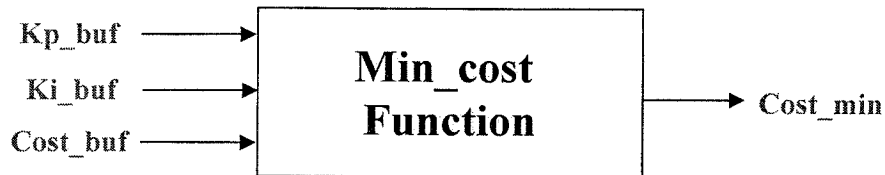


圖5.7 Min_cost副函式輸入輸出方塊圖

```

%副函式-Min_cost:找尋最小的cost值及其相對應參數值%
function [Min_cost_out1, Min_cost_out2, Min_cost_out3] = Min_cost ( Min_cost_in1,
Min_cost_in2, Min_cost_in3 )
Cost_min = 0;
index = 1;
[Cost_min,index]=min( Min_cost_in3 );
Min_cost_out1 = Min_cost_in1( index ); % Kp = Kp_buf( index ) %
Min_cost_out2 = Min_cost_in2( index ); % Ki = Ki_buf( index ) %
Min_cost_out3 = Cost_min;

```

5.2.5 Record_trend 副函式

此函式主要是根據上一次參數自調的趨勢，來決定下一次參數搜尋的軌跡，以減少不必要的搜尋，增加自調搜尋效率。圖 5.8 為此函式的輸入輸出方塊圖。

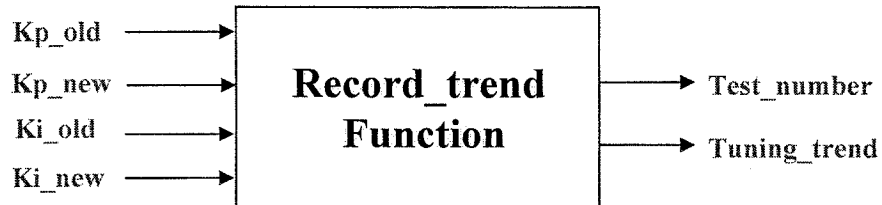


圖5.8 Record_trend副函式輸入輸出方塊圖

```
*副函式-Tuning_trend:參數自調趨勢*
function [Record_trend_out1,Record_trend_out2] = Record_trend(P1_new,P1_old,P2_new,P2_old)

if (P1_new-P1_old)~=0 & (P2_new-P2_old)~=0
    Record_trend_out1 = 6;
    if (P1_new-P1_old)>0 & (P2_new-P2_old)>0
        Record_trend_out2 = 1;
    elseif (P1_new-P1_old)>0 & (P2_new-P2_old)<0
        Record_trend_out2 = 2;
    elseif (P1_new-P1_old)<0 & (P2_new-P2_old)<0
        Record_trend_out2 = 3;
    elseif (P1_new-P1_old)<0 & (P2_new-P2_old)>0
        Record_trend_out2 = 4;
    end
end
elseif (P1_new-P1_old)~=0 | (P2_new-P2_old)~=0
    Record_trend_out1 = 4;
    if (P1_new-P1_old)>0
        Record_trend_out2 = 1;
    elseif (P2_new-P2_old)<0
        Record_trend_out2 = 2;
    elseif (P1_new-P1_old)<0
        Record_trend_out2 = 3;
    elseif (P2_new-P2_old)>0
        Record_trend_out2 = 4;
    end
end
end
```

5.2.6 Next_set 副函式

此函式主要是根據 Record_trend 所得到的參數變化趨勢，來決定接下來參數搜尋的方式。圖 5.9 為此函式的流程圖。

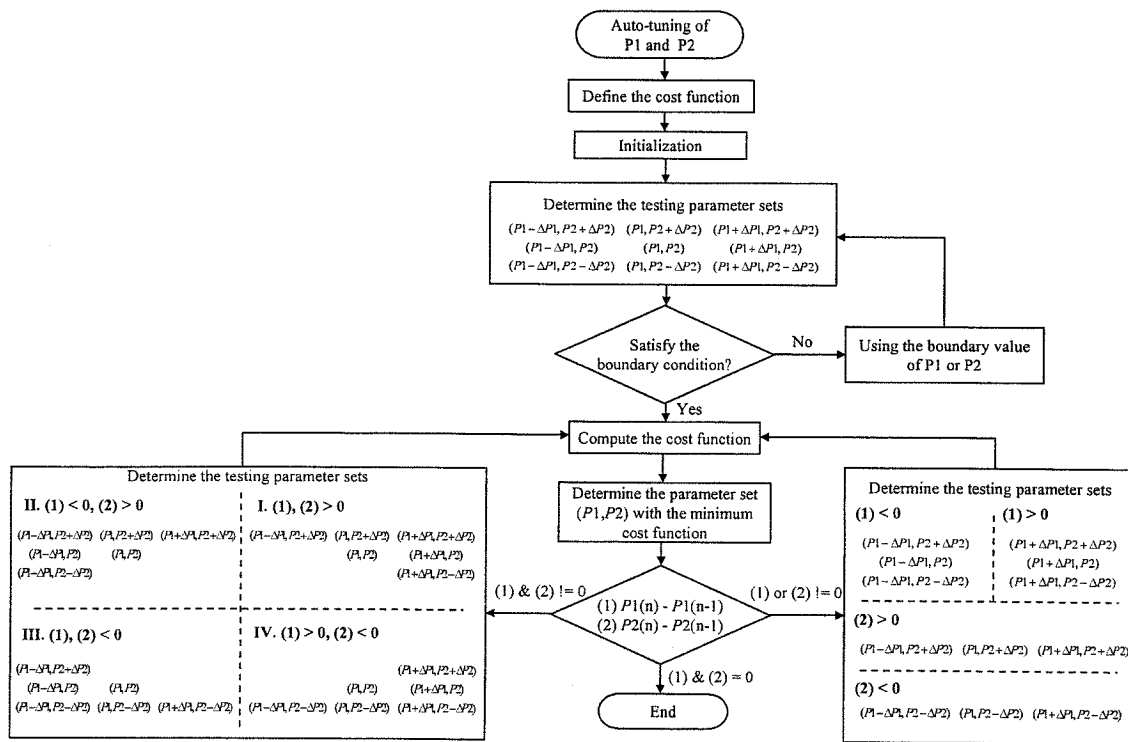


圖 5.9 Next_set 副函式流程圖

%副函式-Next_set:下一組參數測試組%

```
function [Next_set_out1,Next_set_out2] = Next_set( Next_set_in1, Next_set_in2, Set, Number, Trend )
```

```
if Number==9
```

```
    switch Set % Clockwise %
```

```
        case 1
```

```
            Next_set_out1 = Next_set_in1 + Delta_Kp;
```

```
            Next_set_out2 = Next_set_in2 + Delta_Ki;
```

```
        case 2
```

```
            Next_set_out1 = Next_set_in1 + Delta_Kp;
```

```
            Next_set_out2 = Next_set_in2;
```

```
        case 3
```

```
            Next_set_out1 = Next_set_in1 + Delta_Kp;
```

```
            Next_set_out2 = Next_set_in2 - Delta_Ki;
```

```
        case 4
```

```
            Next_set_out1 = Next_set_in1;
```

```
            Next_set_out2 = Next_set_in2 - Delta_Ki;
```

```

case 5
    Next_set_out1 = Next_set_in1 - Delta_Kp;
    Next_set_out2 = Next_set_in2 - Delta_Ki;
case 6
    Next_set_out1 = Next_set_in1 - Delta_Kp;
    Next_set_out2 = Next_set_in2;
case 7
    Next_set_out1 = Next_set_in1 - Delta_Kp;
    Next_set_out2 = Next_set_in2 + Delta_Ki;
case 8
    Next_set_out1 = Next_set_in1;
    Next_set_out2 = Next_set_in2 + Delta_Ki;
end
elseif Number==6
    if Trend==1 % P1(n)-P1(n-1)>0 && P2(n)-P2(n-1)>0 %
        switch Set
            case 1
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2 + Delta_Ki;
            case 2
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2;
            case 3
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2 - Delta_Ki;
            case 4
                Next_set_out1 = Next_set_in1 - Delta_Kp;
                Next_set_out2 = Next_set_in2 + Delta_Ki;
            case 5
                Next_set_out1 = Next_set_in1;
                Next_set_out2 = Next_set_in2 + Delta_Ki;
        end
    elseif Trend==2 % P1(n)-P1(n-1)>0 && P2(n)-P2(n-1)<0 %
        switch Set
            case 1
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2 + Delta_Ki;
            case 2
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2;
            case 3
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2 - Delta_Ki;
            case 4
                Next_set_out1 = Next_set_in1;
                Next_set_out2 = Next_set_in2 - Delta_Ki;
            case 5
                Next_set_out1 = Next_set_in1 - Delta_Kp;
                Next_set_out2 = Next_set_in2 - Delta_Ki;
        end
    elseif Trend==3 % P1(n)-P1(n-1)<0 && P2(n)-P2(n-1)<0 %
        switch Set

```



```

    case 1
        Next_set_out1 = Next_set_in1 + Delta_Kp;
        Next_set_out2 = Next_set_in2 - Delta_Ki;
    case 2
        Next_set_out1 = Next_set_in1;
        Next_set_out2 = Next_set_in2 - Delta_Ki;
    case 3
        Next_set_out1 = Next_set_in1 - Delta_Kp;
        Next_set_out2 = Next_set_in2 - Delta_Ki;
    case 4
        Next_set_out1 = Next_set_in1 - Delta_Kp;
        Next_set_out2 = Next_set_in2;
    case 5
        Next_set_out1 = Next_set_in1 - Delta_Kp;
        Next_set_out2 = Next_set_in2 + Delta_Ki;
end
elseif Trend==4 % P1(n)-P1(n-1)<0 && P2(n)-P2(n-1)>0 %
    switch Set
        case 1
            Next_set_out1 = Next_set_in1 + Delta_Kp;
            Next_set_out2 = Next_set_in2 + Delta_Ki;
        case 2
            Next_set_out1 = Next_set_in1 - Delta_Kp;
            Next_set_out2 = Next_set_in2 - Delta_Ki;
        case 3
            Next_set_out1 = Next_set_in1 - Delta_Kp;
            Next_set_out2 = Next_set_in2;
        case 4
            Next_set_out1 = Next_set_in1 - Delta_Kp;
            Next_set_out2 = Next_set_in2 + Delta_Ki;
        case 5
            Next_set_out1 = Next_set_in1;
            Next_set_out2 = Next_set_in2 + Delta_Ki;
    end
end
elseif Number==4
    if Trend==1 % P1(n)-P1(n-1)>0 && P2(n)-P2(n-1)=0 %
        switch Set
            case 1
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2 + Delta_Ki;
            case 2
                Next_set_out1 = Next_set_in1 + Delta_Ki;
                Next_set_out2 = Next_set_in2;
            case 3
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2 - Delta_Ki;
        end
    elseif Trend==2 % P1(n)-P1(n-1)=0 && P2(n)-P2(n-1)<0 %
        switch Set
            case 1
                Next_set_out1 = Next_set_in1 + Delta_Kp;
                Next_set_out2 = Next_set_in2 - Delta_Ki;
        end
    end

```

```

        case 2
            Next_set_out1 = Next_set_in1;
            Next_set_out2 = Next_set_in2 - Delta_Ki;
        case 3
            Next_set_out1 = Next_set_in1 - Delta_Kp;
            Next_set_out2 = Next_set_in2 - Delta_Ki;
        end
elseif Trend==3 % P1(n)-P1(n-1)<0 && P2(n)-P2(n-1)=0 %
    switch Set
        case 1
            Next_set_out1 = Next_set_in1 - Delta_Kp;
            Next_set_out2 = Next_set_in2 - Delta_Ki;
        case 2
            Next_set_out1 = Next_set_in1 - Delta_Kp;
            Next_set_out2 = Next_set_in2;
        case 3
            Next_set_out1 = Next_set_in1 - Delta_Kp;
            Next_set_out2 = Next_set_in2 + Delta_Ki;
        end
elseif Trend==4 % P1(n)-P1(n-1)=0 && P2(n)-P2(n-1)>0 %
    switch Set
        case 1
            Next_set_out1 = Next_set_in1 + Delta_Kp;
            Next_set_out2 = Next_set_in2 + Delta_Ki;
        case 2
            Next_set_out1 = Next_set_in1 - Delta_Kp;
            Next_set_out2 = Next_set_in2 + Delta_Ki;
        case 3
            Next_set_out1 = Next_set_in1;
            Next_set_out2 = Next_set_in2 + Delta_Ki;
    end
end

```

5.2.7 Converge 副函式

此函式主要是判斷參數自調是否已經收斂。

```

%副函式-Converge:判斷是否已經收斂
function Converge_out = Converge ( Converge_in1, Converge_in2, Converge_in3, Converge_in4,
Plot_index )
if (( Converge_in1( Plot_index - 1 ) == Converge_in3 ) & ( Converge_in2 ( Plot_index - 1 )
== Converge_in4 ))
    Converge_out = 1;
else
    Converge_out = 0;
end
end

```

第六章

實驗結果與分析

6.1 實驗系統架構

本計畫採用一顆德州儀器(Texas Instruments, TI)的數位訊號處理器(digital signal processor, DSP)-TMS320F240 作為控制板之核心，再加上一個功率轉換級，即可構成一個全數位的永磁式交流馬達驅動器，同時此驅動器可以透過串列通訊埠(RS-232)之傳輸協定與電腦溝通，用以進行線上參數自動調整，與觀測馬達動態響應，圖 6.1 為 DSP 與 MATLAB 通訊架構說明圖。

本計畫主要是針對速度控制迴路進行線上控制參數自調實驗，圖 6.2 為速度控制迴路自調控制方塊圖，其中包含了 PI 控制器與自調控制器(ATC)。此線上自調控制實驗是在 DSP 內實現全數位速度迴路 PI 控制器，並利用 RS-232 傳輸與 MATLAB 做溝通，藉由所設計的 MATLAB 自調控制程式來調整速度迴路之 PI 控制參數值，以期達到最佳速度控制響應，完成整個網路式線上自調控制系統。以下之實驗結果將分別設定不同之 PI 控制參數初始值，藉由 MATLAB 自調控制機制能調整至最佳響應之控制器參數值，並且驗證所計算出的參數組是否確實為最佳參數解。值得注意的是，由於所採用的是定點數(fixed-point)DSP，因此為了增加控制參數的解析度，本實驗將 P 控制器參數表示為 Q6 格式，而 I 控制器參數則表示為 Q8 格式。

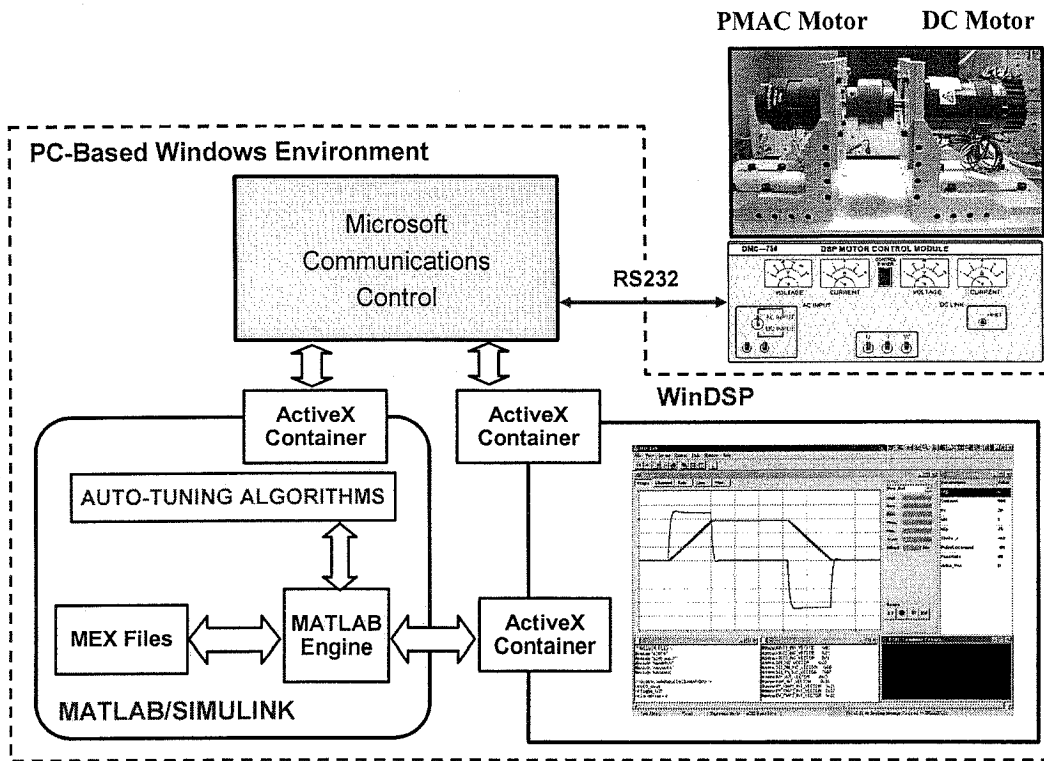


圖6.1 DSP與MATLAB通訊架構說明圖

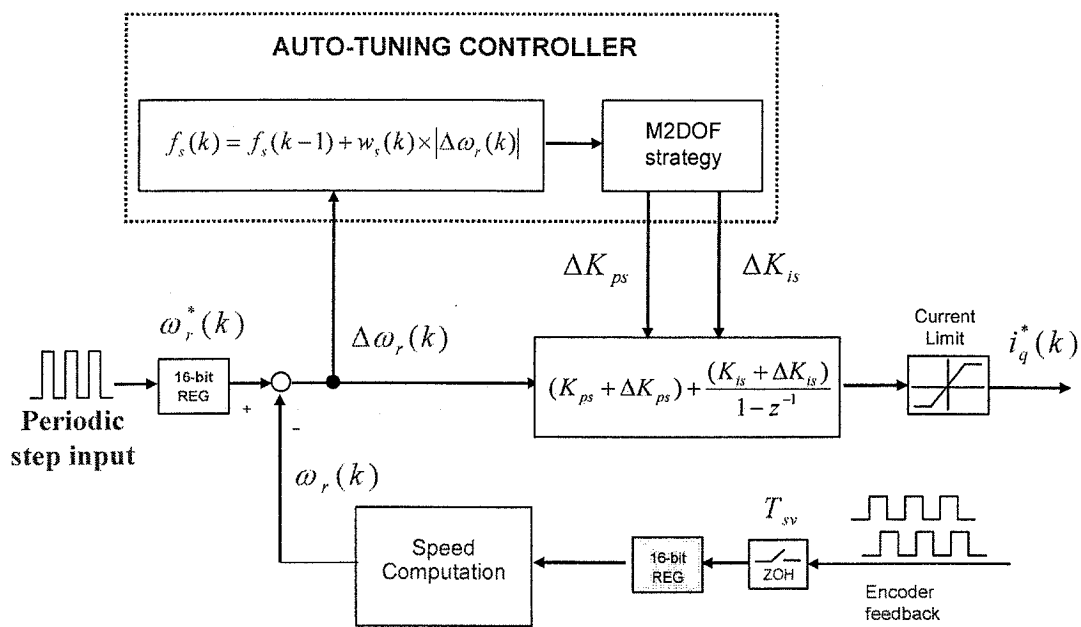


圖6.2 速度控制迴路自調控制方塊圖

6.2 實驗結果

為了進行參數自動調整實驗，必須要輸入測試訊號，在本實驗中所採用的測試訊號為1000rpm的速度步階訊號，圖6.3為速度步階響應圖之自調過程圖，其PI控制器參數之初始值設為 $K_p=100$ ， $K_i=0$ ，經由所設計的MATLAB參數自調控制器調整後，可看出速度步階響應調整之過程，確實有逐漸收斂至最佳響應的情形。圖6.4為根據所設計的參數自調控制搜尋方式，所得到PI控制器參數與目標函數之收斂過程，最後PI速度控制器參數將收斂至 $K_p=45$ ， $K_i=30$ 。整個參數自調過程所耗費的時間約為168.4秒。

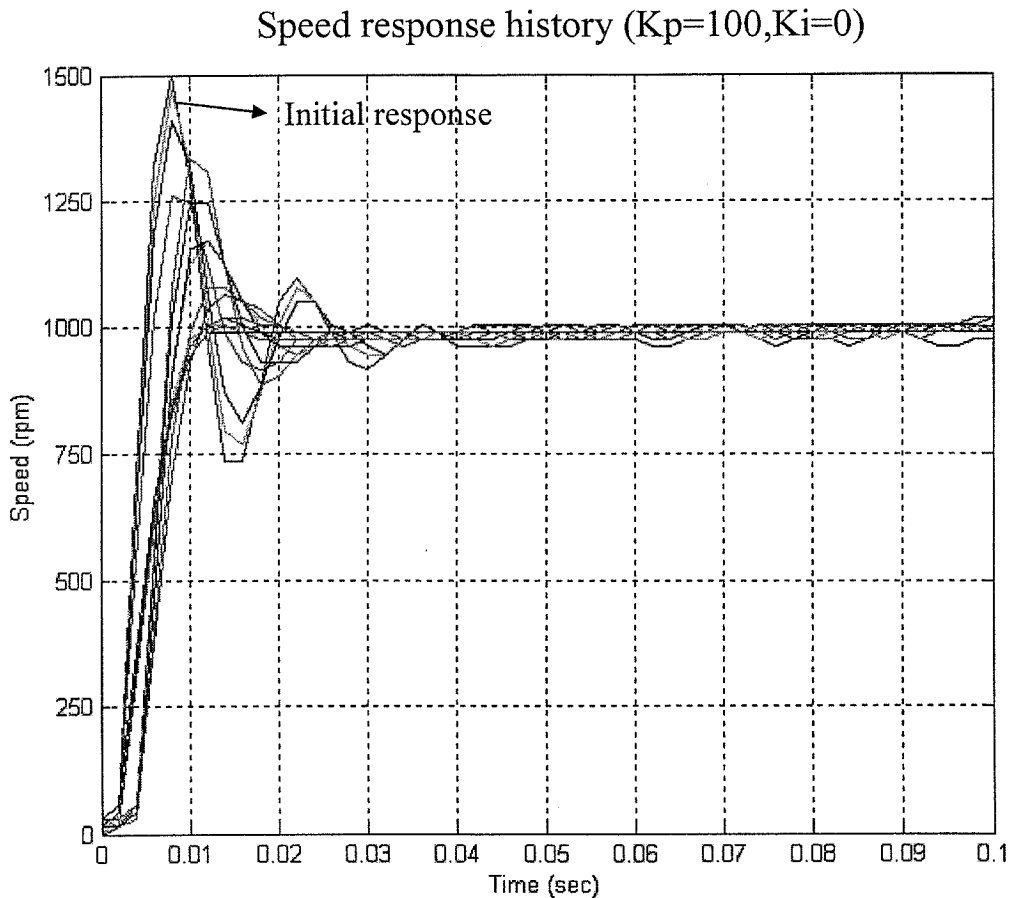


圖6.3 速度步階響應之自調過程圖 ($K_p=100, K_i=0$)

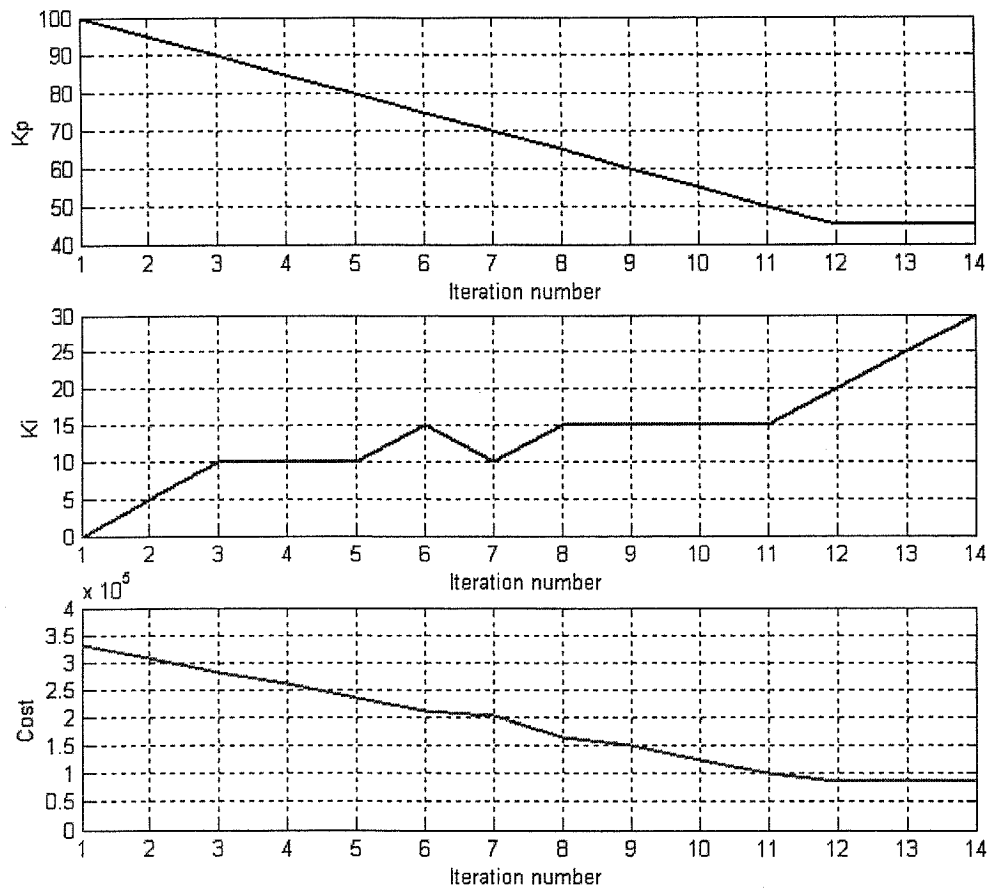


圖6.4 PI控制器參數之收斂過程圖

圖6.5為速度步階響應圖之自調過程圖，其PI控制器參數之初始值設為 $K_p=100$ ， $K_i=35$ ，經由所設計的MATLAB參數自調控制器調整後，可看出速度步階響應調整之過程，確實有逐漸收斂至最佳響應的情形。圖6.6為根據所設計的參數自調控制搜尋方式，所得到PI控制器參數與目標函數之收斂過程，最後PI速度控制器參數將收斂至 $K_p=40$ ， $K_i=25$ 。整個參數自調過程所耗費的時間約為125.8秒。

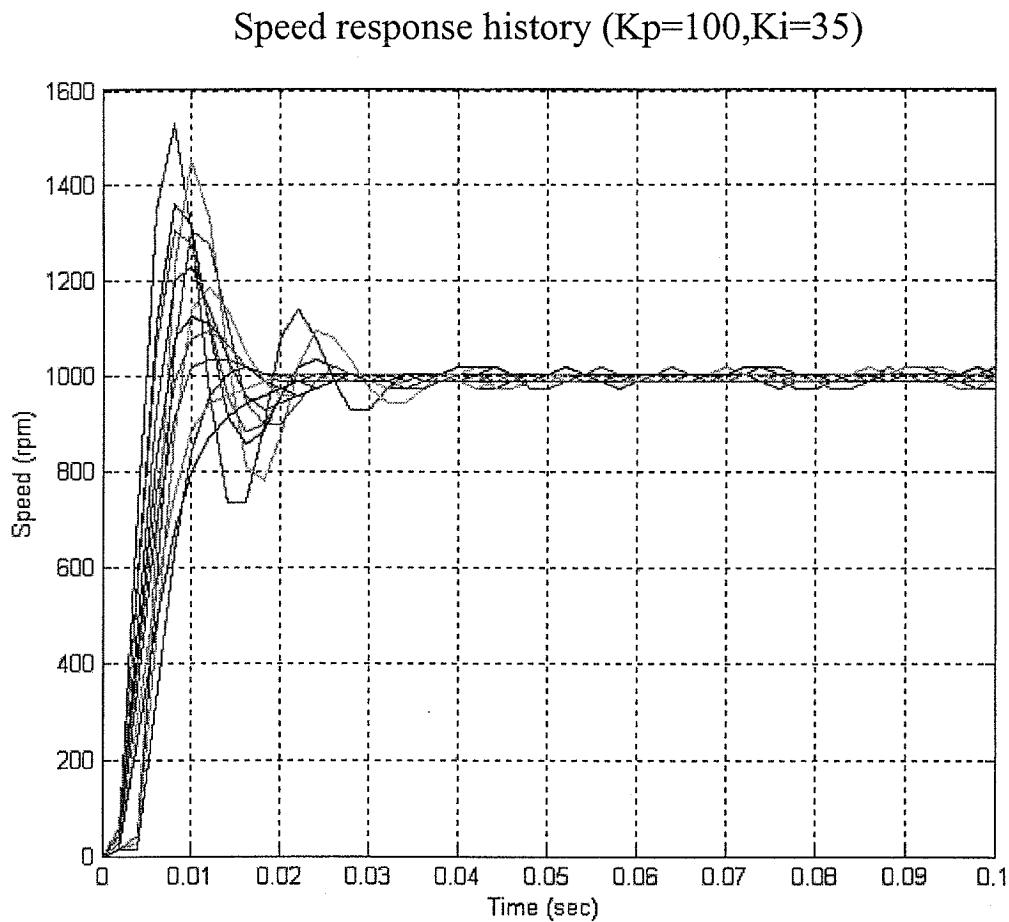


圖6.5 速度步階響應之自調過程圖 ($K_p=100, K_i=35$)

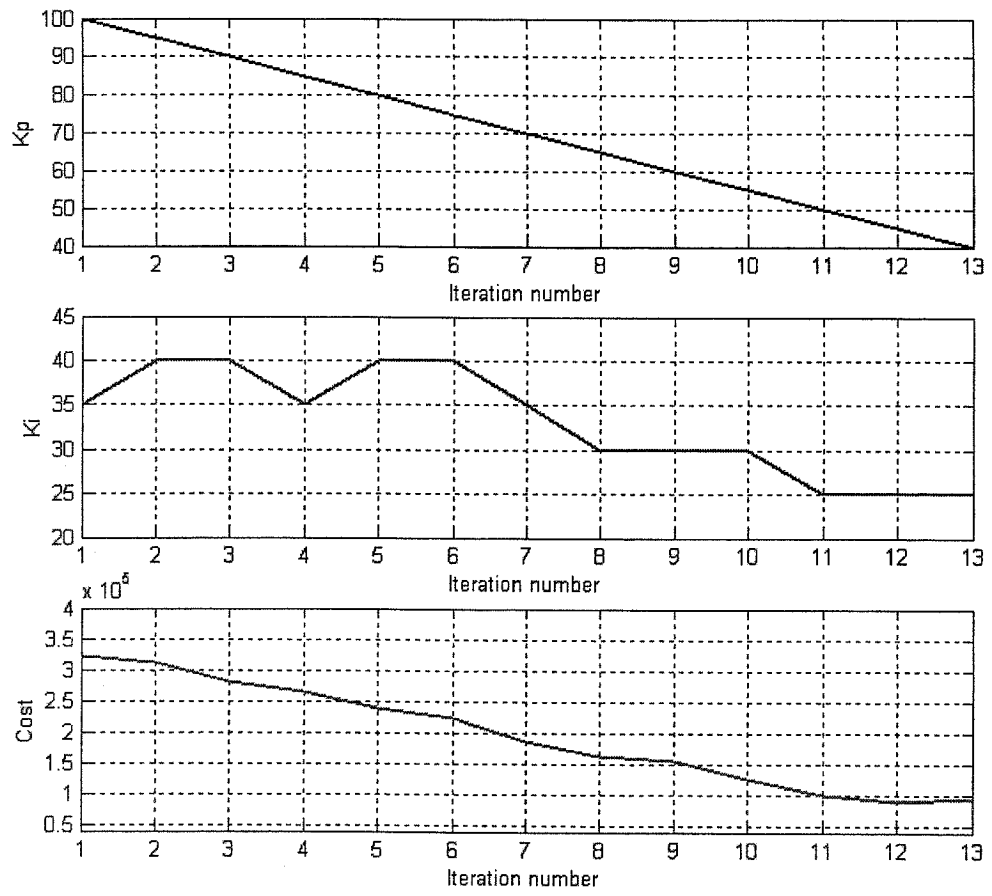


圖6.6 PI控制器參數之收斂過程圖

圖6.7為速度步階響應圖之自調過程圖，其PI控制器參數之初始值設為 $K_p=10, K_i=0$ ，經由所設計的MATLAB參數自調控制器調整後，可看出速度步階響應調整之過程，確實有逐漸收斂至最佳響應的情形。圖6.8為根據所設計的參數自調控制搜尋方式，所得到PI控制器參數與目標函數之收斂過程，最後PI速度控制器參數將收斂至 $K_p=45, K_i=30$ 。整個參數自調過程所耗費的時間約為102.4秒。約為40.9秒。

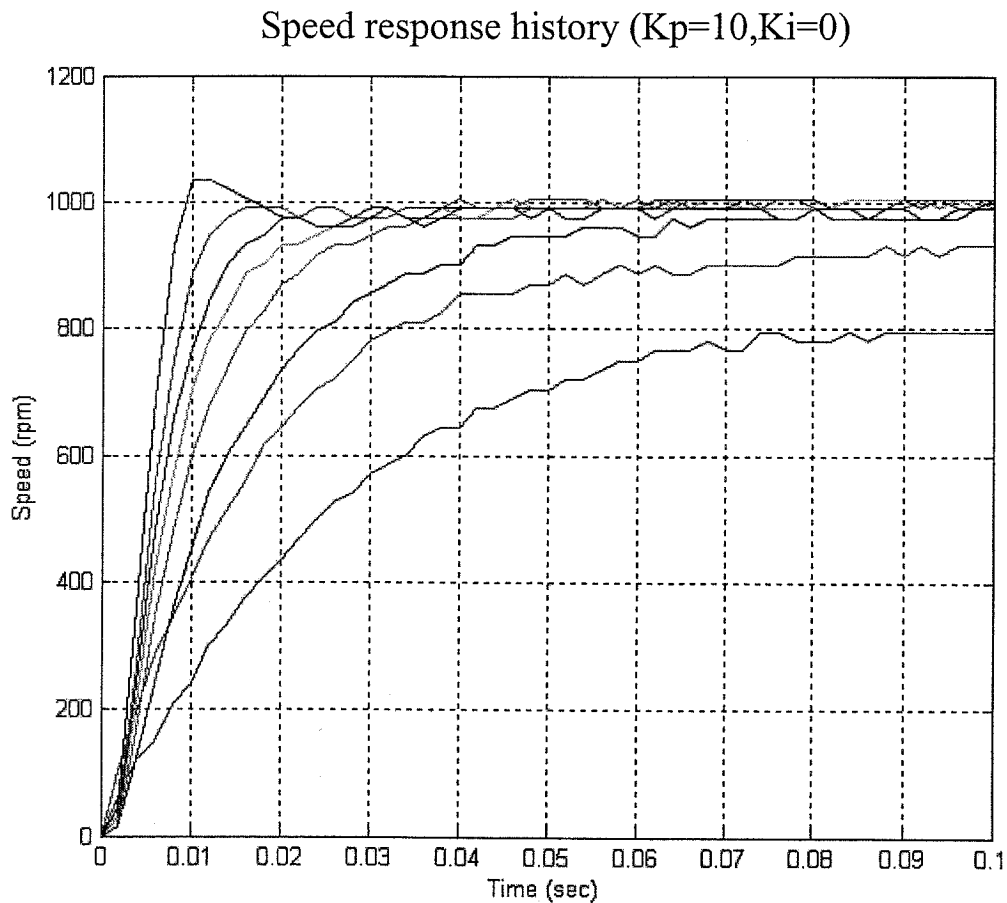


圖6.7 速度步階響應之自調過程圖 ($K_p=10, K_i=0$)

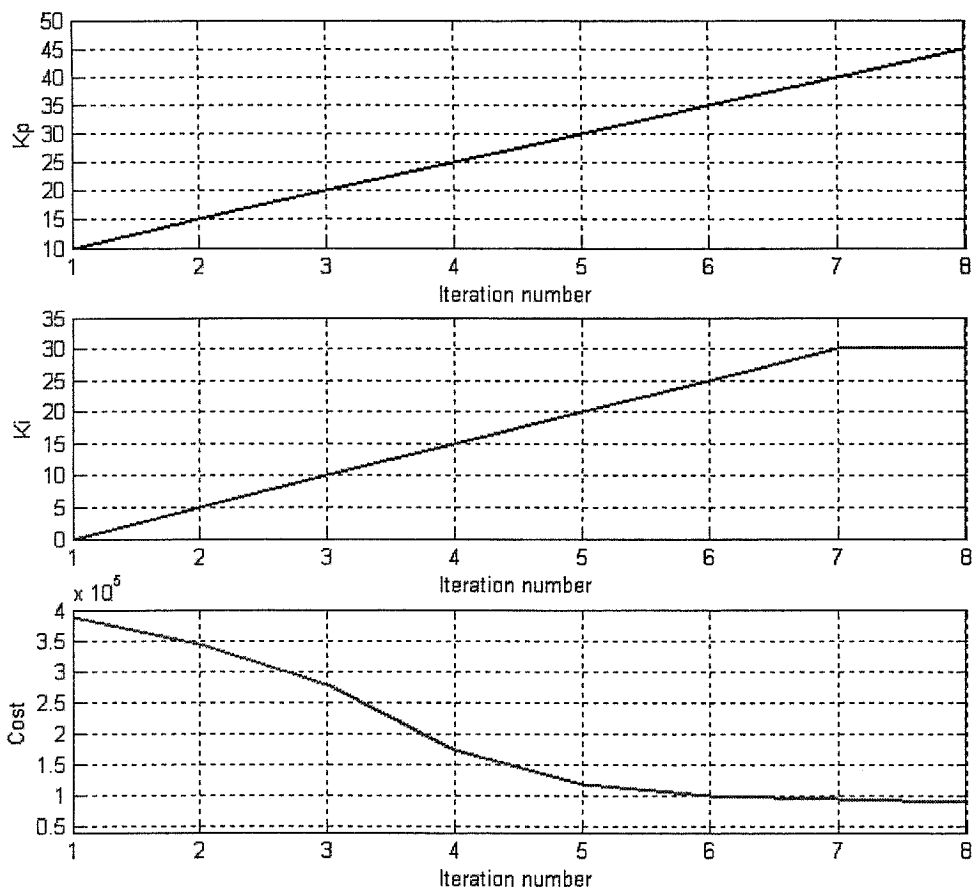


圖6.8 PI控制器參數之收斂過程圖

圖6.9為速度步階響應圖之自調過程圖，其PI控制器參數之初始值設為 $K_p=10$ ， $K_i=35$ ，經由所設計的MATLAB參數自調控制器調整後，可看出速度步階響應調整之過程，確實有逐漸收斂至最佳響應的情形。圖6.10為根據所設計的參數自調控制搜尋方式，所得到PI控制器參數與目標函數之收斂過程，最後PI速度控制器參數將收斂至 $K_p=45$ ， $K_i=30$ 。整個參數自調過程所耗費的時間約為110.7秒。

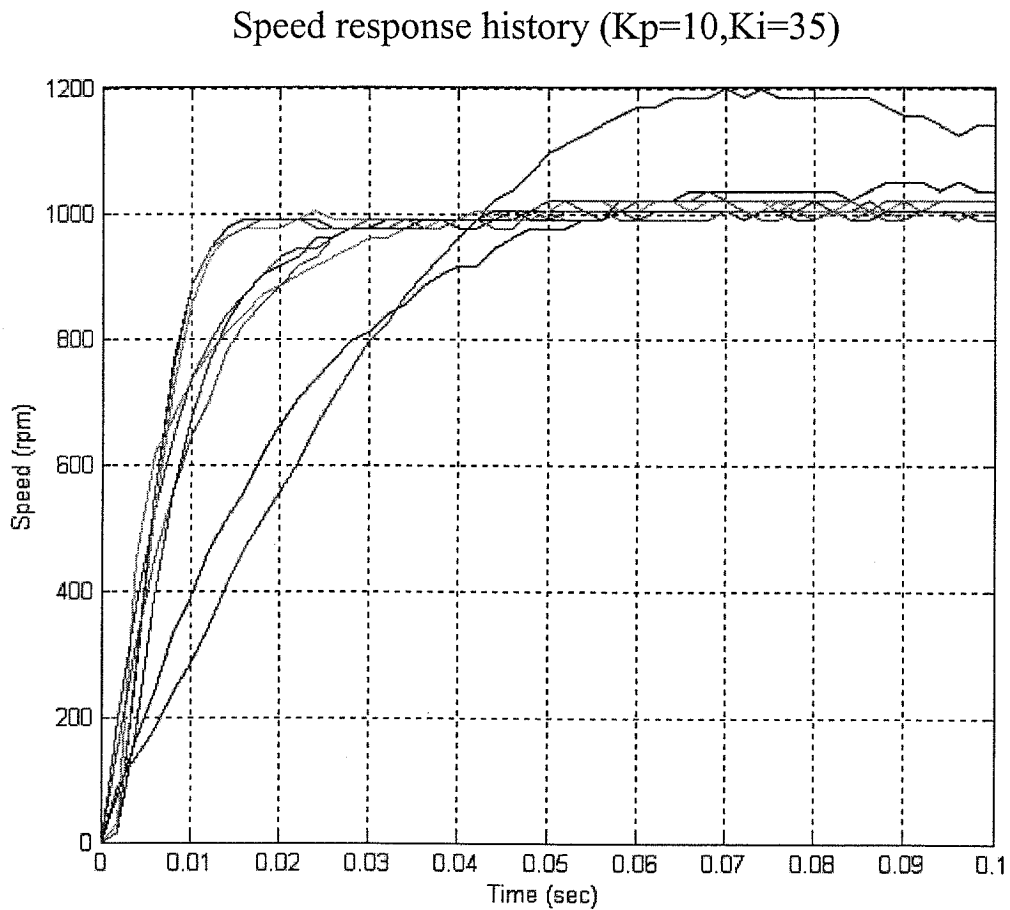


圖6.9 速度步階響應之自調過程圖 ($K_p=10, K_i=35$)

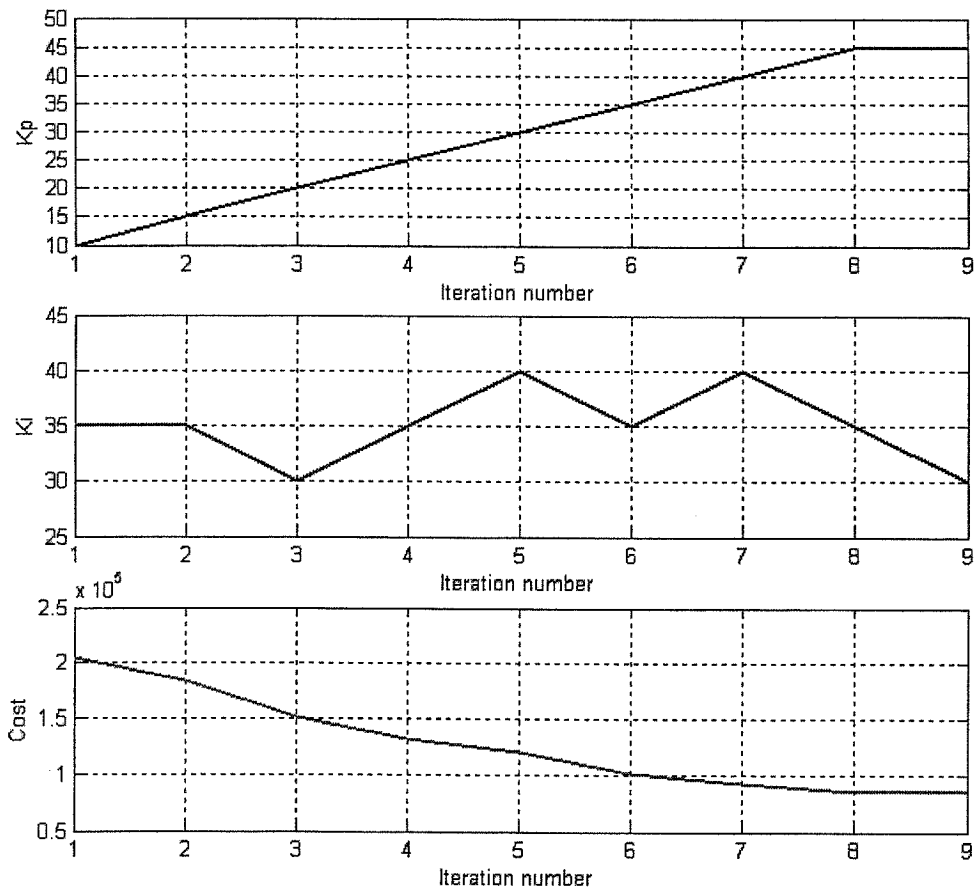


圖6.10 PI控制器參數之收斂過程圖

圖 6.11 為四組不同初始的 PI 控制參數自動調整收斂圖，由圖可知利用所設計的自動調整控制機制，不論 PI 控制器參數的初始值為何，都可將其調整到一個最佳化區間(optimum region)內，以得到最佳化的速度步階響應，圖 6.12 為經由自調控制器所得到的最佳速度步階響應與其相對應的靜止兩軸($\alpha\beta$ 軸)電流命令與回授響應圖。此外，為了驗證所調整出的 PI 控制器參數在整個參數空間內是最佳解，故本實驗將所有 PI 控制器參數組與其相對應的目標函數都作了一次測試，如圖 6.13 所示，由圖可知道 PI 控制器參數的最佳化區間分別為 $K_p:[40,50]$ 、 $K_i:[25,35]$ ，故可驗證本計畫所設計之自調控制機制確實能夠將參數收斂到一個最佳化區間內。

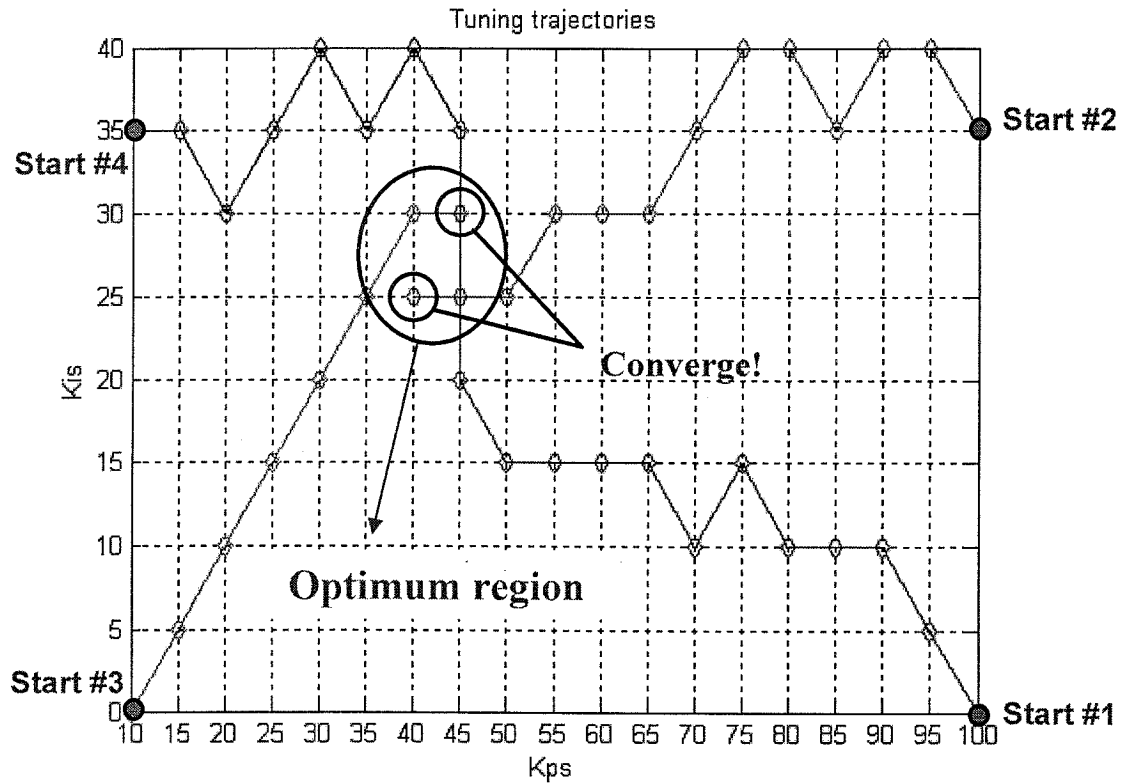


圖6.11 四組PI控制參數自動調整收斂圖

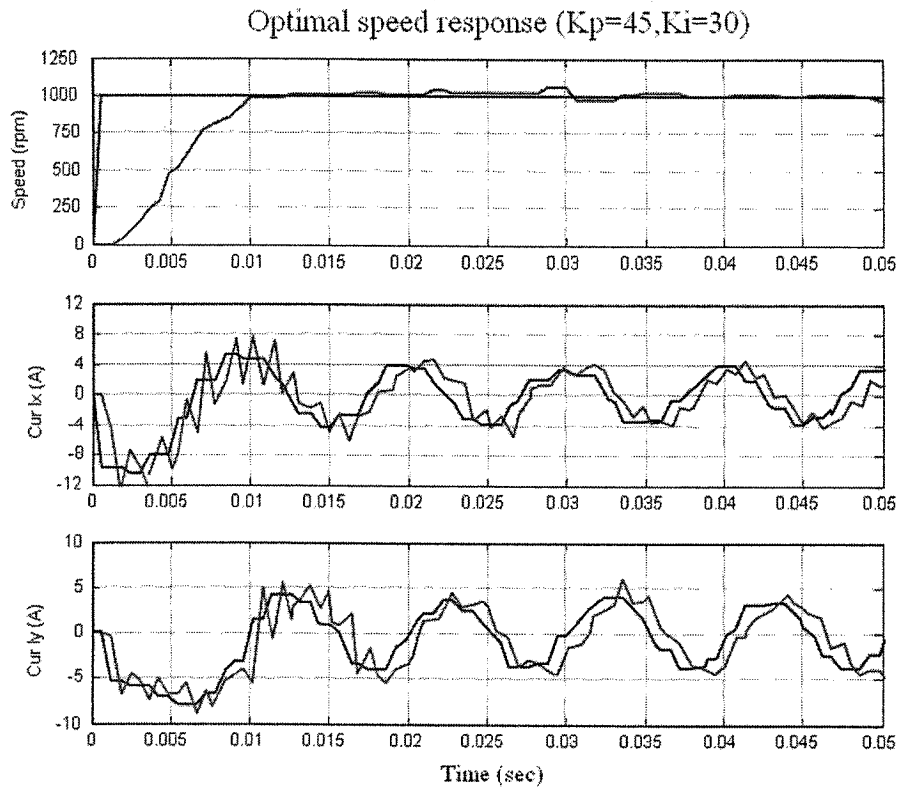


圖6.12 最佳速度步階與靜止兩軸電流之響應圖

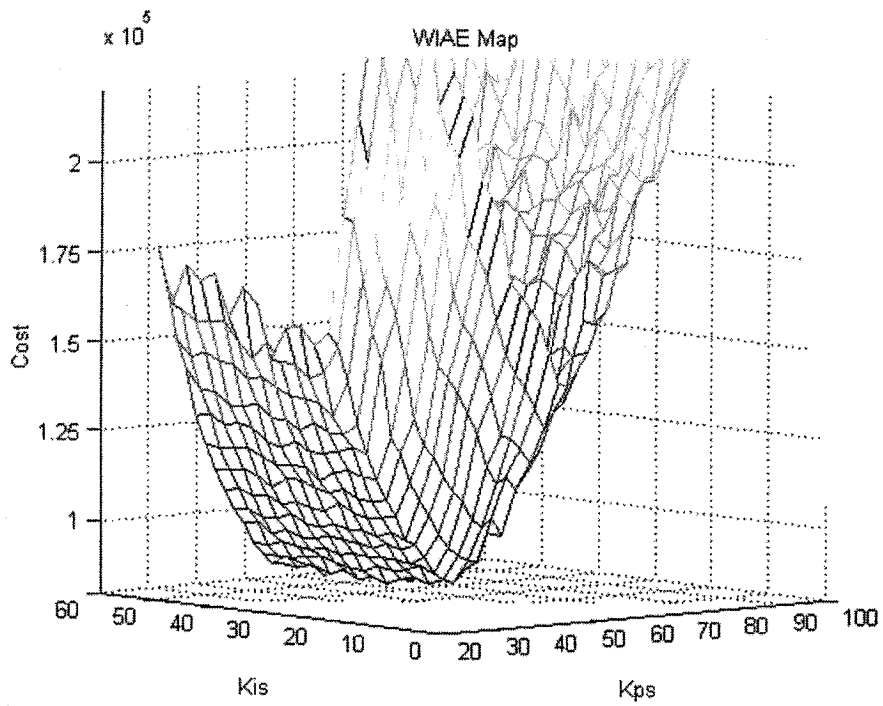


圖6.13 PI控制器參數與目標函數的對應圖

第七章

結論

本計畫針對永磁式交流馬達研究並發展各種適應型控制法則，並將其整合應用於伺服控制系統內，用以增加伺服系統對於外在干擾以及參數變動的強韌性，以期達到最佳化控制響應。在硬體方面，本計畫採用德州儀器公司的 DSP 晶片 TMS320F240 為核心，並配合功率驅動級作為全數位永磁交流馬達驅動器；而在軟體方面，所有的控制程式都是利用 DSP 以及 MATLAB 所開發的，不但可節省硬體成本，亦可增加修改彈性。

在自調式適應伺服控制的整合實驗方面，本計畫採用 RS-232 作為傳輸介面，並且自訂網路通訊協定，以建立 DSP 與 MATLAB 的通訊介面，並在此系統下，發展出一套網路式的自調適應控制技術應用在伺服驅動系統。藉由 MATLAB 之數學運算能力開發出自動調整控制器(ATC)，在不知道馬達的參數以及負載的情形下，能夠在短時間內完成伺服控制器參數的最佳設定，以達到伺服系統最佳之動態響應。本計畫成功的建立以 DSP 為基礎之自調式適應控制系統，未來可利用此系統進行更深入的研究，希望能夠朝向全數位智慧型永磁式交流馬達驅動器的目標努力。

參考文獻

Introduction to AC Drive Technology

- [1] 鄒應嶼, 交流伺服驅動導讀, 技術報告 TR-022, 交通大學 電力電子與運動控制實驗室, Jan. 8, 1997.
- [2] W. Leonhard, "Microcomputer control of high dynamic performance ac-drive - a survey," *Automatica*, vol. 22, no. 1, pp. 1-19, 1986.
- [3] W. Leonhard, Adjustable-Speed AC Drives, *Proc. IEEE*, vol. 76, pp. 455-471, April 1988.
- [4] B. K. Bose, "Adjustable speed AC drives - a technology status review," *Proc. IEEE*, vol. 70, no. 2, pp. 116-196, Feb. 1982.
- [5] B. K. Bose, *Power Electronics and Variable Frequency Drives*, IEEE Press, 1997.
- [6] M. A. Rahman, "Permanent magnet synchronous motors - a review of the state of design art," *Proc. Int. Conference on Electrical Machines*, Athens, pp. 312-319, 1980.
- [7] T. Kume. "High-performance vector-controlled ac motor drives: applications and new technologies," *IEEE Trans. on Ind. Appl.*, vol. 23, Sept./Oct. 1987.
- [8] R. Lessmeier, W. Schumacher, and W. Leonhard, "Microprocessor-controlled ac-servo drives with synchronous or induction motors: which is preferable?," *IEEE Trans. on Ind. Appl.*, vol. 22, no. 5, pp. 812-819, Sep./Oct. 1986.
- [9] Y. Dote and S. Kinoshita, **Brushless Servomotors**, Clarendon Press, Oxford. 1990.
- [10] Duane C. Hanselman, **Brushless Permanent-Magnet Motor Design**, McGraw-Hill, Inc., 1994.
- [11] P. Pillay and R. Krishnan, "Application characteristics of permanent magnet synchronous and brushless DC motors for servo drives," *IAS Annual Meeting*, Atlanta, pp. 380-390, 1987.
- [12] D .P. M. Cahill and B. Adkins, "The permanent magnet synchronous motor," *Proc. IEE*, vol. 109, pt. A Dec., pp. 483-491, 1962.
- [13] M. Bilewski, A. Fratta A. Vagati, F Villata, "Control of high performance permanent magnet synchronous motor drives", *IEEE IAS Annual Meeting*, pp. 531-538, 1990.
- [14] T. M. Jahns, "Motion control with permanent-magnet ac machines," *IEEE Proc.*, vol. 82, no. 8, pp. 1241-1252, Aug. 1994.

FOC and DTC for AC Drives

- [15] F. Blaschke, "The principle of field orientation as applied to the new TRANSVECTOR closed loop control system for rotating field machines," *Siemens Rev.*, vol. 34, pp. 217-220, 1972.
- [16] K. H. Bayer, H. Waldmann, and M. Weibelzahl, "Field-oriented closed-loop control of a synchronous machine with new transvector control system," *Siemens Rev.*, vol. 39, no. 5, pp. 220-223, 1972.
- [17] D. W. Novotny and T. A. Lipo, **Vector Control and Dynamics of AC Drives**, Clarendon Press, Oxford, 1996.
- [18] R. D. Doncker and D. W. Novotny, "The universal field oriented controller," *IEEE Trans. Ind. Appl.*, pp. 450-456, Oct. 1988.
- [19] E. Y. Y. Ho and P. C. Sen, "Decoupling control of induction motor drives," *IEEE Trans. on Ind. Electron.*, vol. IE-35, no. 2, pp. 253-262, 1988.
- [20] N. C. Enslin and F. S. Van Der Merve, "Improving the open-loop torque step response of induction motors," *IEE Proc. B*, vol. 134 no. 6, pp. 317-323, 1988.
- [21] S. Ogasawara, H. Akagi, and A. Nabae, "The generalized theory of indirect vector control for ac machines," *IEEE Trans. Ind. Appl.*, vol. 24, no.3, pp.470-478, May/June 1988.
- [22] M. Depenbrock, "Direct self-control (DSC) of inverter-fed induction machine," *IEEE Tran. Power Electron.*, vol.3, no. 4, pp. 420-429, Oct. 1988.
- [23] G. Diana and R. G. Harley, "An aid for teaching field oriented control applied to induction machine," *IEEE Trans.*, vol. PWRS-4, no. 3, pp. 1258-1261, 1990.
- [24] A. Hughes, J. Corda, and D. A. Andrade, "Vector control of cage induction motors: a physical insight," *IEE Proc.*, vol. 143, no. 1, pp. 59-68, 1996.

DSP for Digital Motor Control

- [25] 鄒應嶼, DSP數位馬達控制技術, 技術報告 TR-019, 交通大學 電力電子與運動控制實驗室, Feb. 4, 2000.
- [26] **TMS320C24x User's Guide**, Texas Instruments, 1997.
- [27] Y. Dote, **Servo Motor and Motion Control Using Digital Signal Processors**, Prentice Hall and Texas Instruments Digital Signal Processing Series, 1990.
- [28] P. Pillay, C. R. Allen and R. Budhabhathi, "DSP-based vector and current controllers for a permanent magnet synchronous motor drive", *IEEE IAS Annual Meeting*, pp. 539-544, 1990.
- [29] M. E. Ahmed and P.R. Belanger, "Scaling and round-off in fixed-point implementation of control algorithms," *IEEE Trans. on Ind. Electron.*, vol. 31, no. 3, pp. 228-234, August 1984.

- [30] C. P. Diduch and R. Doraiswami, "Design of a microcomputer based servomechanism controller," *Comput. & Elect. Engng.*, vol. 13, no. 1, pp. 29-40, 1987.
- [31] H. Hanselmann, "Implementation of digital controllers — a survey," *Automatica*, vol. 23, no. 1, pp. 7-32, 1987.
- [32] Hsiang-Jui Wu and Ying-Yu Tzou (鄒應嶼), "Design and implementation of a digital ac induction servo drive," *Proceedings of the 12th National Symposium on Automatic Control*, ROC, pp. 517-528, 1988.
- [33] M. F. Rahman, T. S. Low, and L. B. Wee, "Development of a digitally controlled permanent magnet brushless DC drive system," *Conference on Applied Motion Control*, pp. 283-288, 1986.
- [34] DSP Solutions for BLDC Motors, Literature Number: BPRA055, Texas Instruments, March 1997.
- [35] Implementation of a speed controlled brushless DC drive using TMS320F240, Literature Number: BPRA064, Texas Instruments Europe, July 1997.
- [36] Implementation of a sensorless speed controlled brushless DC drive using TMS320F240, Literature Number: BPRA072, Texas Instruments Europe, November 1997.
- [37] S. Ogasawara and H. Akagi, "An approach to position sensorless drive for brushless DC motor," *IEEE IAS Annual Meeting Conf. Rec.*, pp. 443-447, 1990.
- [38] N. Matsui and M. Shigyo, "Brushless DC motor control without position and speed sensors," *IEEE Trans. on Ind. Appl.*, vol. 28, no. 1, pp. 120-127, 1992.

Robust Servo Control

- [39] 鄒應嶼, 直流伺服的多迴路控制, 技術報告TR-027, 交通大學 電力電子與運動控制實驗室, Feb. 5, 2000.
- [40] Ying-Yu Tzou (鄒應嶼) and Hsiang-Jui Wu, "LQG/LTR control of an ac induction servo drive," *IEEE Trans. on Power Electronics*, pp., 214-221, March 1995.
- [41] Ying-Yu Tzou (鄒應嶼), "DSP-based robust control of an ac induction servo drive for motion control," *IEEE Trans. on Control Systems Technology*, vol. 4, no. 6, pp. 614-626, Nov. 1996.
- [42] Hsiang-Jui Wu, Yao-Tsung Tseng, and Ying-Yu Tzou (鄒應嶼), "Application of H-infinity control theory to servo controller design for ac induction motor drives," *First International Power Electronics and Motion Control Conference*, Beijing, China, June 27-30, 1994.
- [43] Ying-Yu Tzou (鄒應嶼) and Hsiang-Jui Wu, "Multimicroprocessor-based robust control of an ac induction servo motor," *IEEE Transactions on Industry Applications*, vol. 26, no. 3, pp. 441-449, May/June 1990.
- [44] H. Gross, Electrical Feed Drives for Machine Tools, John Wiley & Sons, 1983.

- [45] DSP-Based Robust Model Matching Control of a Load Insensitive DC Drive, Research Study, PEMLAB, Aug. 10, 1996.
- [46] E. J. Davison and A. Goldenberg, "Robust control of a general servo-mechanism problem: the servo compensator," *Automatica*, vol. 11, pp. 461-471, 1975.
- [47] C. P. Diduch and R. Doraiswami, "Robust servomechanism controller design for digital implementation," *IEEE Trans. on Ind. Electronics*, vol. 34, no. 2, pp. 172-179, May 1987.
- [48] I. M. Horowitz, *Synthesis of Feedback Systems*, New York and London: Academic Press, 1963.
- [49] B. A. Francis and W. M. Wonham, "The Internal model principle of control theory," *Automatica*, vol. 12, pp. 457-465, 1976.

Adaptive Control

- [50] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Addison Wesley, 1989.
- [51] Ming-Fa Tsai and Ying-Yu Tzou, "A transputer-based adaptive speed controller for ac induction motor drives with load torque estimation," *IEEE Trans. on Ind. Appl.*, vol. 33, pp. 558-566, March/April, 1997.
- [52] B. Courtiol and I. D. Laudan, "High speed adaptation system for control electrical drives," *Automatica*, vol. 11, pp. 119-127, 1975.
- [53] K. J. Astrom, "Tuning and adaptation," *IFAC Conf. Rec.*, pp. 1-18, 1996.
- [54] K. J. Astrom and T. Hagglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for PID controllers - a survey," *Control Engineering Practice*, vol.1, no. 4, pp. 699-714, 1993.
- [55] Shi-Zhong He, Shaohua Tan, Feng-Lan Xu, and Pei-Zhuang Wang, "Fuzzy self-tuning of PID controller," *Elsevier Science Publishers*, pp. 37-46, 1993.
- [56] M. Zhuang and D. P. Atherton, "Automatic tuning of optimum PID controllers," *IEE Proceedings-D*, vol. 140, no. 3, pp. 216-224, May, 1993.
- [57] K. H. Gurubaszvarzj, "Implementation of a self-tuning control using digital signal processor chips," *IEEE Control Systems Magazine*, pp. 38-42, June, 1989.

IP Control

- [58] P. K. Nandam and P. C. Sen, "Analog and digital speed control of DC drives using proportional-integral and integral-proportional control techniques," *IEEE Trans. on Ind. Electron.*, pp. 227-233, May 1987.
- [59] G. A. Perdikaris, *Computer Controlled Systems: Theory and Applications*, Kluwer Academic Publishers, 1991.
Chap. 6: Computer Controlled Systems: Computer Algorithms for Deadbeat Response
Chap. 7: Computer Controlled Systems: PID, PDFF, Phase-Lead and Other Control Algorithms

- [60] K. J. Astrom and H. Steingrimsson, "Implementation of a PID controller on a DSP," from: **Digital Control Applications with the TMS320 Family: Selected Application Notes**, Texas Instruments, 1991.
- [61] I. Ahmed, "Implementation of PID and deadbeat controllers with the TMS320 family," Texas Instruments, Application Note.

Complementary & Model Matching Control

- [62] T. Eisaka, Y. Zhong, S. Bai, R. Tagawa, "Evaluation of robust model matching for the control of a DC servo motor," *Int. J. Control*, vol. 50, no. 2, pp. 479-493, 1989.
- [63] J. Caro, J. Dufour, and A. Jakubowicz, "A microprocessor-based position control of a DC drive taking account the load's variations," *IEEE IAS Annual Meeting*, pp. 680-685, 1984.
- [64] K. Tamaki, K. Ohishi, K. Ohnishi, and K. Miyachi, "Microprocessor-based robust control of a DC servo motor," *IEEE Control System Magazine*, pp. 30-35, Oct. 1986.
- [65] S. Bai and R. Tagawa, "Load insensitive electrical position servo: a new design method and its experimental evaluation," *IEEE IECON Conf. Rec.*, pp. 600-605, 1984.
- [66] R. Kelly, "A linear-state feedback plus adaptive feedforward control for DC servomotors," *IEEE Trans. Ind. Electron.*, vol. 34, no. 2, pp. 153-157, May 1987.
- [67] T. C. Hsia, "A new technique for robust control of servo systems," *IEEE Trans. on Ind. Electronics*, vol. 36, no. 1, pp. 1-7, Feb. 1989.

Auto-Tuning of IP & Model Matching Control

- [68] K. Ohishi, M. Nakao, K. Ohnishi, and K. Miyachi, "Adaptive DC servo drive control taking force disturbance suppression into account," *IEEE Trans. on Ind. Appl.*, vol. 24, no. 1, pp. 171-176, Jan./Feb. 1988.
- [69] B. Courtiol and I. D. Laudan, "High speed adaptation system for control electrical drives," *Automatica*, vol. 11, pp. 119-127, 1975.
- [70] C. D. Walrath, "Adaptive bearing friction compensation based on recent knowledge of dynamic friction," *Automatica*, vol. 20, no. 6, pp. 717-727, 1984.
- [71] S. D. Kraft and E. T. Wall, "Experimental microprocessor-based adaptive control system," *IEEE Contr. Syst. Magn.*, vol. 6, no. 1, Feb. 1986.
- [72] Qingding Guo, Yue Zhou, and Wei Guo, "Neural network real-time IP position controller online design for permanent magnet linear Synchronous motor," *IEEE IPENC Conf. Rec.*, vol. 2, pp. 996-999, 2000.
- [73] Bin Zhang, Yaohua Li, and Yansheng Zuo, "A DSP-based fully digital PMSM servo drive using on-line self-tuning PI controller," *IEEE IPENC Conf. Rec.*, vol. 2, pp. 1012-1017, 2000.
- [74] K. H. Gurubasavaraj, "Implementation of a self-tuning controller using digital signal processor chips," *IEEE Control Systems Magazine*, June 1989.

- [75] Hsiang-Jui Wu and Ying-Yu Tzou (鄒應嶼), "Design of an automated parameter measurement system for ac induction servo drives," *Proceedings of the 11th Symposium on Electrical Power Engineering*, ROC, pp. 420-427, 1990.
- [76] H. J. Wu, M. F. Tsai, and Ying-Yu Tzou (鄒應嶼), "Transputer-based fully digital adaptive control system for high-performance ac induction motor drives," *IEEE Power Electronics Specialists Conf. Rec.*, Taipei, Taiwan, ROC, pp. 1250-1256, June 20-24, 1994.
- [77] Ying-Yu Tzou (鄒應嶼), "DSP-based self-tuning current vector control of a PWM inverter for ac motor drive," *IEEE PESC Conf. Rec.*, June 18-22, 1995.
- [78] M. F. Tsai and Ying-Yu Tzou (鄒應嶼), "A transputer-based adaptive speed controller for ac induction motor drives with load torque estimation," *Int. Auto. and Contr. Conf.*, June, 1995.
- [79] Ying-Yu Tzou (鄒應嶼) and Yuh-Farn Lin, "Auto-tuning control of self-commissioning electric drives," *IEEE IECON Conf. Rec.*, New Orleans, USA, pp. 483-487, Nov. 9-14, 1997.
- [80] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. Amer. Soc. Mech. Eng.*, vol. 64, pp. 759-768, 1942.
- [81] G. H. Cohen and G. A. Coon, "Theoretical consideration of retarded control," *Trans. Amer. Soc. Mech. Eng.*, vol. 75, pp. 827-834, 1953.
- [82] D. E. Seborg, T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control*. New York: Wiley, 1989.
- [83] C. C. Hang, K. J. Astrom, and W. K. Ho, "Refinements of the Ziegler-Nichols tuning formula," *Inst. Elect. Eng. Proc., Pt. D*, vol. 138, no. 2, pp. 111-118, 1991.
- [84] K. J. Astrom and T. Hagglund, *Automatic Tuning of PID Controllers*, Instrument Soc. Amer., Research Triangle Park, NC, 1988.
- [85] K. W. Ho, C. C. Hang and J. H. Zhou, "Performance and gain and phase margins of well-known PI tuning formulas," *IEEE Trans. Control System Technology*, vol. 3, pp. 245-248, June., 1995.
- [86] F. G. Shinskey, *Process-Control Systems. Application, Design, and Tuning*. New York: McGraw-Hill, 1988.