

國立成功大學

電機工程學系

碩士論文

運動命令規劃之智慧技術元件設計與實現

Design and Implementation of Intellectual Properties
on Motion Command Planning

研究生：胡志寬

指導教授：鄭銘揚

中華民國九十五年七月

國立成功大學

碩士論文

運動命令規劃之智慧技術元件設計與實現

Design and Implementation of Intellectual Properties on
Motion Command Planning

研究生：胡志寬

本論業經審查及口試合格特此證明

論文考試委員

鄭銘揚

莊宏祥

洪正瑞

指導教授：鄭銘揚

系(所)主管：許渭州

中華民國九十五年七月十一日

**Design and Implementation of Intellectual Properties on
Motion Command Planning**

by

Chih-Kuan Hu

*A Thesis Submitted to the Graduate Division in Partial Fulfillment of the
Requirement for the Degree of*

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

NATIONAL CHENG KUNG UNIVERSITY

TAINAN, TAIWAN

REPUBLIC OF CHINA

July 11, 2006

Approved by

Ming-Yang Cheng _____

Hung-Shiang Chuang _____

Jeng-Ruey Horng _____

Advisor

Ming-Yang Cheng

Chairman :

Wei Chen Shu

運動命令規劃之智慧技術元件設計與實現

Design and Implementation of Intellectual Properties on Motion Command Planning

胡志寬* 鄭銘揚**

國立成功大學電機工程研究所

中文摘要

現今運動控制 IC 的開發可分為兩大方向，分別為整合型運動控制 IC 與專用型運動控制 IC，不論那一類型，在開發過程中皆可劃分為不同任務類型的區塊，這些區塊可將其設計成智慧技術元件 (Intellectual Properties；IP)。本論文之主要目的即為發展運動控制系統中關於運動命令規劃的各式 IP，並提供相關開發經驗給運動控制 IC 設計者，使其能快速整合各類 IP 以縮短開發時程，完成所需不同功能之運動控制 IC。

在運動命令規劃方面，加減速規劃、脈波命令輸出與 NURBS 曲線的產生皆為重要課題，於本論文中，將分別探討這三種任務類型區塊於運動命令規劃 IP 的設計與實現。首先，對於加減速規劃方面，採用數位迴旋積分方式實現具有梯型與 S Curve 兩種加減速規劃的 IP，相較於傳統多項式加減速規劃，本論文中所採用的方法具有容易規劃、零誤差的優點。對於脈波命令輸出方面，採用數位差分法 (Digital Difference Analyzer；DDA) 實現具有均勻脈波命令輸出的 IP，相較於傳統 DDA 脈波命令架構，本論文中修改之 DDA 脈波命令架構具有可變週期的優點。此外 NURBS (Non-Uniform Rational B-Spline) 參數式曲線，本論文採用去

零化簡運算之技巧予以實現，可大幅化簡運算的次數。相較於傳統樹狀結構運算法，本論文中所採用的方法具有高效能、省空間的優點。

基於開發測試的便利性與成本的考量下，本論文以現場可規劃閘陣列(Field Programmable Gate Array；FPGA)作為發展之平台，並以硬體描述語言(VHSIC Hardware Description Language；VHDL)的方式實現所設計之運動命令規劃 IP。

* 作者

** 指導教授

ABSTRACT

The motion control ICs can be divided into two categories — integrated and specific. Both types of ICs consist of several sub blocks. These sub blocks can be designed into Intellectual Properties (IP). The aim of this thesis is to develop different kinds of IP for motion planning. It is expected that, through this thesis, the development experience can be shared with the motion control designers so that the time duration for developing different motion control ICs can be shortened by integrating different kinds of IP more quickly.

It is known that acceleration/deceleration motion planning, pulse command generation, and generation of NURBS curves are all important subjects in motion planning. The design and implementation of IP for these subjects will be addressed in this thesis. First of all, the digital convolution technique is exploited to develop IP for trapezoid and S curve Acceleration/Deceleration motion planning. Compared with the conventional polynomial type methods, the proposed approach has advantages such as easy formulation and error free. For pulse command generation, the Digital Difference Analyzers (DDA) is used to implement the IP that can output the pulse commands smoothly. Compared with the conventional methods, the advantage of the proposed approach is that the sampling period of DDA is adjustable. In addition, a more efficient way to implement the NURBS (Non-Uniform Rational B-spline) parametric curve is employed to reduce the number of computation. Therefore, the proposed approach has advantages such as high efficiency and space saving compared with the commonly used tree structure-based approach.

In this thesis, considering the issues such as convenience and cost for

developing IP, the Field Programmable Gate Array (FPGA) is adopted as a platform for development. In addition, the developed IP for motion planning is implemented using the VHSIC Hardware Description Language (VHDL).

誌謝

在就讀碩士班的這兩年來，承蒙指導教授鄭銘揚博士的諄諄教誨，使畢業論文得以順利完成，感謝洪正瑞老師與莊宏祥老師在口試時對論文之指正，感謝實驗室科翰學長在研究上的協助，得以使本研究順利進行。並且感謝宏文、正虎等學長的提攜指導及待人處事上諸多意見的提供，感謝這兩年來同儕偉安、正偉、俊良、維志、昀傑在學習上的互相扶持及甘苦與共，此外還要感謝眾多學弟妹曙峰、盈惠、志豪、助彬、銘翔、章傑在生活上提供歡樂的心情，在此一並致上最誠摯的謝意。

最後感謝父母二十幾年來的辛苦栽培與親友們支持，如今得以完成學業並邁入人生的下一個階段。

僅以本論文

獻給父母家人及所有關愛我的人

目 錄

中文摘要.....	III
英文摘要.....	V
致謝.....	VII
目錄.....	VIII
表目錄.....	XI
圖目錄.....	XII
第一章 緒論.....	1
1-1 研究動機與目的.....	1
1-2 文獻回顧.....	3
1-3 本文架構.....	5
第二章 FPGA 簡介與 IP 設計流程.....	7
2-1 數位電路設計.....	7
2-2 FPGA 簡介.....	9
2-3 IP 的設計流程.....	10
2-4 VHDL 硬體描述語言.....	12
2-5 小結.....	12
第三章 加減速規劃 IP.....	13
3-1 加減速規劃架構.....	13
3-2 多項式之加減速規劃.....	14
3-2-1 梯型加減速規劃.....	14
3-2-2 S Curve 加減速規劃.....	16
3-3 數位迴旋積分之加減速規劃.....	17

3-3-1	梯型加減速規劃.....	18
3-3-2	S Curve 加減速規劃.....	19
3-3-3	定點數數位迴旋積分加減速規畫.....	20
3-3-4	定點數之數位迴旋積分餘數補償機制.....	25
3-3-4-1	數位迴旋積分之前的餘數補償.....	25
3-3-4-2	數位迴旋積分過程中的餘數補償.....	26
3-4	架構設計.....	28
3-5	小結.....	29
第四章	脈波命令輸出 IP.....	30
4-1	脈波命令與伺服驅動器.....	31
4-2	DDA 脈波命令功能.....	33
4-3	DDA 脈波命令演算法.....	34
4-4	DDA 脈波命令架構改良.....	36
4-5	小結.....	38
第五章	NURBS 曲線產生 IP.....	39
5-1	NURBS 參數式曲線簡介.....	39
5-2	NURBS 參數式曲線模型.....	40
5-3	NURBS 參數式曲線之樹狀結構運算.....	43
5-4	NURBS 參數式曲線運算化簡.....	44
5-5	樹狀結構與運算化簡之計算方法比較.....	47
5-5-1	運算次數比較.....	47
5-5-2	記憶體空間比較.....	51
5-6	架構設計.....	54
5-7	小結.....	54
第六章	實驗設備介紹與實驗結果.....	56

6-1	軟硬體設備.....	56
6-2	實驗結果.....	59
6-3	小結.....	71
第七章	結論與建議.....	72
參考文獻	73

表 目 錄

表 2-1 各類 IP 比較表.....	11
表 4-1 伺服驅動器位置模式脈波命令輸入分類訊號表.....	31
表 4-2 DDA 脈波命令演算表.....	35
表 5-1 不同方法之運算次數表.....	50
表 5-2 不同方法所需之記憶體空間表.....	53

圖 目 錄

圖 2-1	積體電路設計分類圖.....	7
圖 2-2	Stratix 系列內部架構圖.....	9
圖 2-3	邏輯單元架構圖.....	10
圖 2-4	IP 設計流程圖.....	11
圖 3-1	前加減速架構圖.....	14
圖 3-2	後加減速架構圖.....	14
圖 3-3	梯型參數式速度曲線與加速度曲線圖.....	15
圖 3-4	S Curve 參數式速度曲線與加速度曲線圖.....	17
圖 3-5	數位迴旋積分之梯型加減速示意圖.....	19
圖 3-6	數位迴旋積分之 S Curve 加減速示意圖.....	20
圖 3-7	速度輸出脈波數序列 $P[i]$	22
圖 3-8	未補償殘餘脈波數之梯型加減速序列 $P_2[k]$	22
圖 3-9	補償後之梯型加減速序列 $P_3[m]$	23
圖 3-10	未補償殘餘脈波數之 S Curve 加減速序列 $P_4[n]$	24
圖 3-11	補償殘餘脈波數於 $P_4[n]$ 序列的末端.....	24
圖 3-12	補償殘餘脈波數於 $P_4[n]$ 序列的加速及減速區域.....	24
圖 3-13	梯型加減速規劃數位迴旋積分過程中補償流程圖.....	27
圖 3-14	加減速 IP 硬體架構圖.....	29
圖 4-1	馬達驅動模式架構圖.....	30
圖 4-2	離散命令與實際目標示意圖.....	32
圖 4-3	馬達與 DDA 命令關係圖.....	33
圖 4-4	DDA 脈波命令基本架構圖.....	34

圖 4-5	觸發頻率與輸出脈波關係圖.....	36
圖 4-6	非對稱除頻器架構圖.....	37
圖 4-7	DDA 脈波命令改良架構圖.....	37
圖 5-1	NURBS 參數式曲線計算流程圖.....	43
圖 5-2	$p+1$ 階的基底函數 $N_{i,p}(u)$ 的樹狀結構圖.....	44
圖 5-3	第零階基底函數對非零階基底函數影響圖.....	45
圖 5-4	樹狀結構中非零區域圖.....	45
圖 5-5	化簡後之 NURBS 參數式曲線計算流程圖.....	46
圖 5-6	NURBS IP 硬體架構圖.....	54
圖 6-1	Altera Stratix EP1S40F780C5 FPGA 發展板.....	57
圖 6-2	史賓納 STC-VC33 DSP 控制卡.....	58
圖 6-3	MITSUBISHI AC 伺服馬達及驅動器.....	58
圖 6-4	梯型加減速曲線補償末端脈波輸出圖.....	60
圖 6-5	梯型加減速曲線補償加速及減速區脈波輸出圖.....	60
圖 6-6	S Curve 加減速曲線補償末端脈波輸出圖.....	60
圖 6-7	S Curve 加減速曲線補償加速及減速區脈波輸出圖.....	61
圖 6-8	不同條件下之梯型及 S Curve 加減速曲線圖.....	61
圖 6-9	點對點運動實驗驗證架構圖.....	62
圖 6-10	補償於末端之梯型加減速命令輸出位置.....	63
圖 6-11	補償於末端之梯型加減速馬達位置檢出圖.....	64
圖 6-12	補償於加速與減速區之梯型加減速命令輸出位置.....	64
圖 6-13	補償於加速與減速區之梯型加減速馬達位置檢出圖.....	65
圖 6-14	補償於末端之 S Curve 加減速命令輸出位置.....	65
圖 6-15	補償於末端之 S Curve 加減速馬達位置檢出圖.....	66
圖 6-16	補償於加速與減速區之 S Curve 加減速命令輸出位置.....	66

圖 6-17 補償於加速與減速區之 S Curve 加減速馬達位置檢出圖.....	67
圖 6-18 NURBS 曲線產生實驗結果一.....	68
圖 6-19 NURBS 曲線產生實驗結果二.....	69
圖 6-20 NURBS 曲線產生實驗結果三.....	70
圖 6-21 NURBS 曲線產生實驗結果四.....	71

第一章 緒論

1-1 研究動機與目的

在工業自動化的發展過程中，造就了工具機(machine tools)相關產業如雨後春筍般的蓬勃發展，且隨著科技的進步，對於高品質加工的需求也隨之增高，現代的加工機也被相對地要求能提供更高速度及更精密的製造技術。因此，為了提升加工精度，除了以改良機械本體架構外，主宰整合工具機軟硬體間運作的運動控制器設計相形之下更顯得重要。近年來，國內外製造廠商所發展的控制器大多為搭配電腦使用的運動控制卡為主，此種控制卡架構大致上可分為兩種不同的類型。第一種為開放式架構(open-structure)，控制卡上包含 A/D、D/A 和 I/O...等電路，使用者可利用運動控制卡製造商所提供之函式庫或是自行規劃的程式於 PC 端進行控制。然而隨著控制法則的演進使得運算量大增，PC 端中央處理器的負荷加重，加上與運動控制卡之溝通介面在時間上的損失，使得即時(real-time)控制之需求便無法滿足，進而發展出以 DSP 晶片為主的 DSP-Based (Digital Signal Processor)運動控制卡，使用者可自行撰寫軟體於 DSP 上執行，主要的運算方面則由 DSP 執行，處理速度因此大幅提升且可以達到即時控制的需求。第二種為封閉式架構(closed-structure)，控制卡上除 A/D、D/A 和 I/O...等電路外，其計算核心上為特用的運動控制晶片，此晶片通常具有產生點對點運動命令規劃(point-to-point motion planning)、直線插值(linear interpolation)、圓弧插值(circular interpolation)...等的功能，使用者僅需於 PC 端依照運動控制卡所提供之參數進行調整與設定即可。兩種控制卡架構各有優缺點，前者雖然具擴充性高，但普遍

來說，其價位偏高；後者價位上雖然較為便宜，但整體使用上卻被限定在某些特定功能運作上，使用者擴充性不高。

在運動控制卡的開發上，除了必須考慮運動控制卡軟硬體架構配置的特性與設計優良的控制器外，運動控制命令規劃的合適為精密製造的重要因素之一，因此，控制卡設計上也須具備能提供完善的運動控制命令規劃。且隨著近年來，積體電路製程技術的成熟，使得系統單晶片(System on a Chip; SoC)成為普遍的發展趨勢。因此，國內外製造廠商在開發運動控制卡時，多半會結合相關技術整合開發特用積體電路晶片(Application Specific Integrated Circuits; ASIC)於運動控制卡上。有鑑於此，於本論文中，將以封閉式架構為參考依據，發展運動控制相關功能性區塊，分別為 1.加減速規劃，2.脈波命令輸出規劃，3.NURBS 曲線產生，等運動命令規劃。並以將不同功能性區塊整合成各自智慧技術元件(Intellectual Properties; IP)的方式，實現於現場可程式閘陣列(Field Programmable Gate Array; FPGA)。由於使用 IP 整合的積體電路開發方式，具有開發速度快、選擇多樣化與節省成本等優點，對於發展特用晶片的廠商來說，相當有利於提供開發整合及運動控制晶片之相關經驗。

本論文中，共規劃與實現加減速規劃 IP、脈波輸出 IP 與 NURBS (Non-Uniform Rational B-Spline)曲線產生 IP 等三種 IP 型態。對於加減速規劃方面，相較於傳統的多項式加減速規劃方式的繁雜，本論文採用數位迴旋積分方式實現具有梯型與 S Curve 兩種加減速規劃的 IP。對於脈波命令輸出方面，本論文使用數位差分分析法(Digital Difference Analyzer; DDA)來實現具有均勻脈波命令輸出的 IP。此外，本論文亦完成 NURBS 曲線產生的 IP。

1-2 文獻回顧

本論文之發展方向為運動控制領域中之加減速規劃、脈波命令產生與 NURBS 曲線的產生。其中，加減速規劃方面，對於工具機而言，速度命令的加減速規劃會影響機台實際的進給率，因此，加減速規劃合宜與否將對於刀具、工件或機台有很大的影響。一般常見的加減速規劃以梯型加減速規劃(Trapezoidal ACC/DEC Planning)、S Curve 加減速規劃(亦稱 Bell Shape ACC/DEC Planning)...等為主。傳統有加減速曲線多半透過多項式參數曲線來描述，雖然方法簡單易懂，但計算上卻較為繁雜，不利於以硬體方式實現。另一方面，Kim 及 Jeon 以軟體加減速規劃(software ACC/DEC Planning)的方式應用數位迴旋積分的觀念來實現梯型與 S Curve 加減速規劃[5]。爾後，為了考慮急跳度(jerk)所可能帶來的影響，Jeon 等學者更進一步發展出具非對稱性之平滑加減速規劃曲線[9][10]。Chen 及 Lee 等學者以有限脈衝數位濾波器(Digital Finite Impulse Response filter)的方式規劃後加減速曲線，實現出較 S Curve 更平滑之加減速曲線 [3]。

在脈波命令產生方面，數位差分分析法(Digital Difference Analyzer；DDA) 為一種常應用在工具機上的脈波命令產生方法。此一方法由美國密西根大學知名學者 Koren 所提出[23]，其特色為可使命令脈波均勻的輸出。

至於在空間中自由曲線描述方面，常見的表示方式為以 Bezier、B-Spline、Rational B-Spline 或 NURBS 等自由曲線表示方式。例如: Tsay

等學者於 1993 年將 Rational B-Spline 應用於凸輪的描述[7]，並於 1996 年將 Nonparametric Rational B-Spline 應用於柱狀凸輪的描述[6]。Piegl 將 Bézier、B-Spline、NURBS 等曲線的描述提供以完整的數學架構，提供後人設計空間中曲線及曲面的方法。近年來，許多工具機控制器設計廠商，如日商 Fanuc、Mistubishi 及德商 Siemens 等大廠均有發展整合自由曲線描述方式之控制器。

在運動控制相關的研究中，國內外學者大多以 FPGA 作為快速驗證平台開發特用原型(prototype)晶片，相關研究整理如下：

1.) 控制器方面的研究

對於控制器設計方面，早期控制器大多由 PC 或 DSP 主宰，近年來，在系統晶片化(SoC)的帶動下，實現演算法及控制架構的任務則由能多工運作的 FPGA 擔任。相關實現於 FPGA 的控制器架構如：PID 控制器[15][44]、模糊(Fuzzy)控制器[42][43]、模糊 PID 控制器[2][31]、強健(robust)控制器[24]、前饋(feed-forward)控制器[25]...等。

2.) 運動命令產生方面的研究

傳統的曲線描述方式多半為依賴電腦輔助設計(Computer Aid Design；CAD)軟體產生，雖然規劃使用上容易，然而受限於套裝軟體的價位及開發上的便利性。因此，近年來，漸漸有學者以 FPGA 來實現硬體架構，藉此產生自由曲線規劃運動命令。如學者 Ou 及 Hsu 以 FPGA 為發展平台發展 NURBS 曲線[36]，學者 Gopi 與 Manohar[16]提出以簡化曲線演算法的方式以降低硬體計算量開發 NURBS 曲線產生晶片，爾後 Yau 及 Yuan 等學者開發於運動控制中不可獲缺的 NURBS 插值器實現於

FPGA [8][35]。對於加減速產生部份，一般市售的運動控制晶片大多具有簡單的加減速規劃命令，如 Performance Motion Devices(PMD)[21]及 NOVA Electronics[20]等公司所研發的運動控制晶片。於學界研發部分，Jeon 及 Kim 等學者則將前面所提及非對稱平滑曲線以 FPGA 加以實現加減速規劃[11]，Carrica 等學者於 2003 年提出一脈波輸出的演算法根據所規劃的速度曲線，輸出脈波於步進馬達驅動[4]。李文猶於 2005 年提出以估測的方式，實現梯型與 S Curve 加減速以 DDA 輸出的方式產生脈波[27]。其他有關以 FPGA 實現 DDA 輸出脈波的運動控制命令之研究如[26][38][39][41][45]。此外，由於客制化驅動器的盛行，因此，FPGA 也被廣泛地應用於實現脈寬調變輸出驅動，如實現以弦波脈寬調變(Sinusoidal Pulse-Width Modulation; SPWM)[17][22][33]驅動馬達，以三相方波(square wave of three phases)命令驅動主軸馬達[28]以及空間向量脈寬調變(Space Vector Pulse-Width Modulation; SVPWM)[32][40][44]實現等研究。

1-3 本文架構

本論文將針對運動控制命令中之加減速規劃、脈波命令產生與 NURBS 曲線的產生進行研究與分析，並將上述運動控制命令以 IP 的型態規劃成特用任務區塊於 FPGA 內。

本論文共分七章，其中第二章將對 FPGA 與 IP 做基本的介紹；第三章針對加減速做探討並設計一數位迴旋積分加減速 IP；第四章針對脈波命令做探討並設計一數位差分分析脈波輸出 IP；第五章將簡介 NURBS

曲線並設計一 NURBS 曲線產生 IP；第六章則為實驗設備的簡介與實驗結果的展示；第七章則為本論文的結論與未來研究之建議。

第二章 FPGA 簡介與 IP 設計流程

科技的發展日新月異，自從西元 1959 年德州儀器與 Fairchild 公司共同推出了世界上的第一顆積體電路(Integrated Circuit ; IC)開始，IC 產品迅速的融入各種電子設備中，使得各種電子設備變的更小、更便宜、更省電，也因為 IC 的受到歡迎，使得 IC 設計產業日益蓬勃發展。以下的章節將針對數位電路設計、FPGA、IP 及本論文所開發所使用的硬體描述語言 VHDL 作介紹。

2-1 數位電路設計

由於電子產品的需求大增，造就了各種不同的積體電路晶片設計技術發展，其相關技術設計分類圖表如圖 2-1 所示[30]：

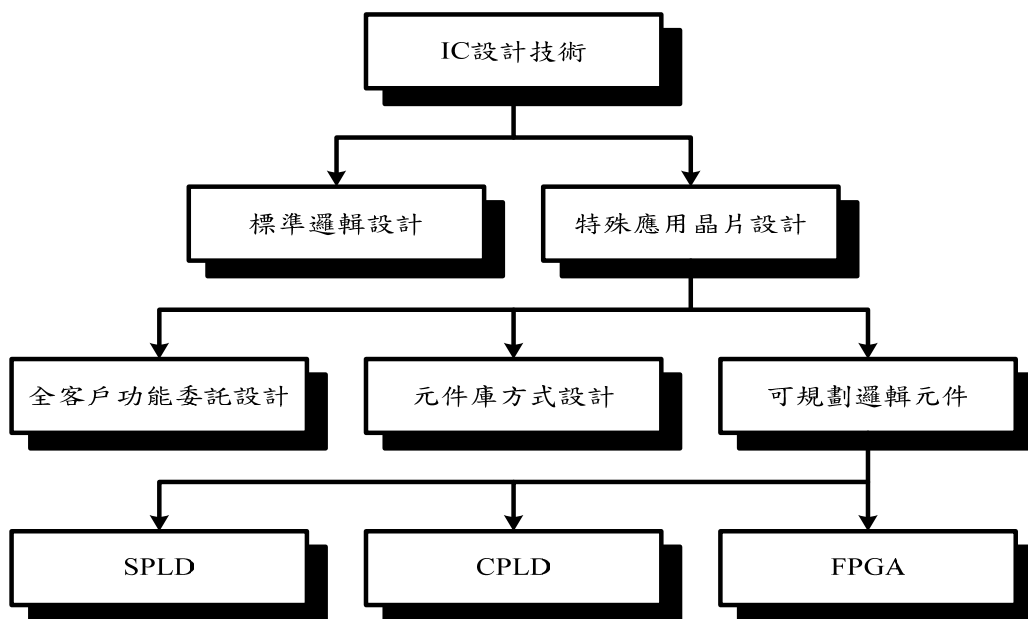


圖 2-1 積體電路設計分類圖

圖 2-1 中之各部詳細描述如下：

◆ 標準邏輯設計：

為傳統式的離散電路設計方式，使用目前市面上已經有的邏輯元件 IC，以人工佈線的方式將其整合而成以達到所規劃之目的。這種設計方法會有成本過高、佈線複雜、雜訊干擾、維修不易等缺點。

◆ 全客戶功能委託設計：

為全訂製電路(full custom circuit design)的 IC 設計方法，這種方法只需將所需之目的告知廠商，其設計到製程皆由廠商完成。這種設計方法會有成本過高、所需時間太長等缺點[47]。

◆ 元件庫方式設計：

此種設計方法由已設計完成的元件庫中取得元件，基本上，元件庫中具有一些通用且常見的元件，例如邏輯閘、計數器、正反器等，通常提供元件庫的廠商為半導體製程公司，例如台積電、聯電等。這種設計方法會有所需時間太長、元件庫取得不易等缺點。

◆ 可規劃邏輯元件：

此種設計方法由廠商先將一些基本的電路元件整合成晶胞(cell)的形式，再將其以陣列的方式排列於一晶片上，使用者可依照所需之目的規劃出晶胞間的連接狀況，具有可重複使用與即時驗證的特性，但這種設計方法也有 IC 體積過大、系統執行效能較低等缺點。

2-2 FPGA 簡介

現場可程式閘陣列(FPGA)是屬於特殊型態之應用積體電路(ASIC)的一種，由於製程技術的進步，廠商對 FPGA 的設計除了規劃邏輯容量越來越大外，更加入了許多實用的功能。本論文中所使用的晶片為 Altera 公司於 2002 年所發表的 Stratix 系列 FPGA，該晶片即具有容量大的優點及內建特殊功能區塊，以下為以此系列晶片為例之 FPGA 簡介。

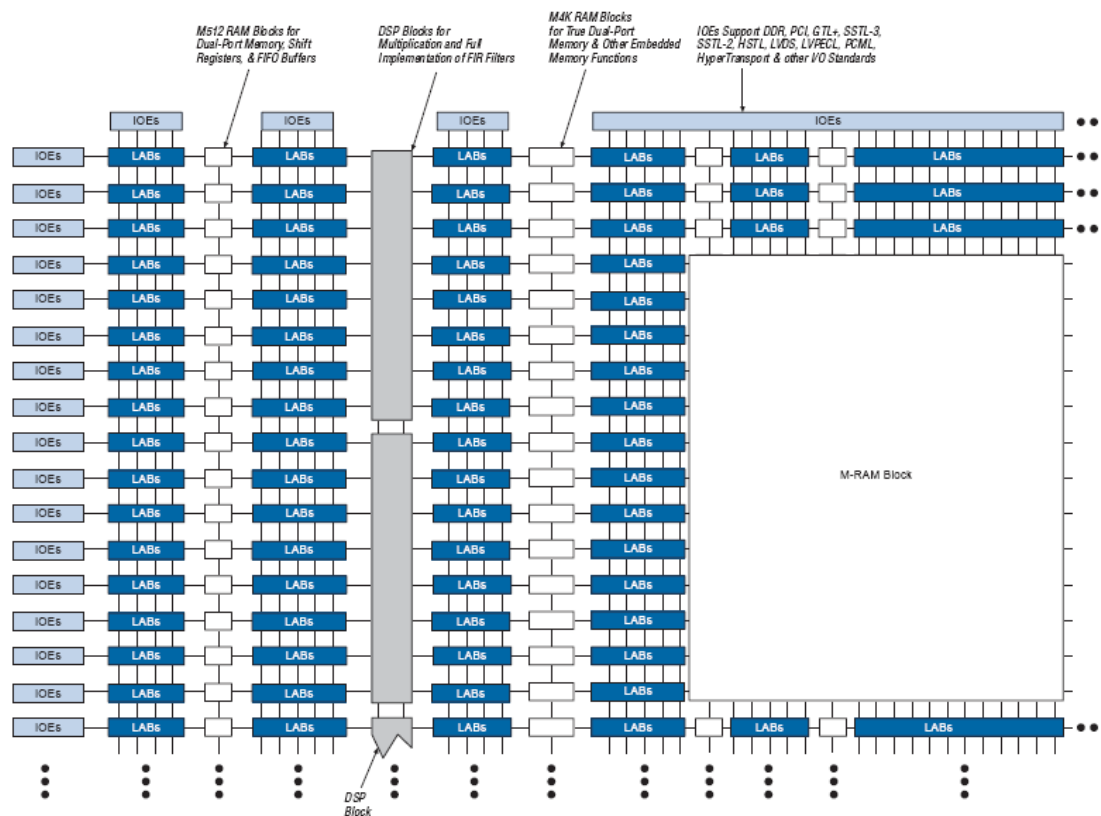


圖 2-2 Stratix 系列內部架構圖

Altera 所出產的 Stratix 系列的 FPGA 採用 1.5V 的核心電壓，0.13 μ m 的全銅製程，其邏輯單元(Logic Elements; LE)的數量高達 10,570 到 79,040 個，並具有高達 7Mbits 的內部記憶體，相較於傳統 FPGA 更加入了數位

信號處理區塊(DSP block)、多種差分 I/O 接腳、鎖相迴路(PLL)時脈管理等實用的功能。圖 2-2 為 Stratix 系列 FPGA 的內部架構，其中最小的單元為邏輯閘陣列(Logic Array Blocks ; LAB)，而每一個邏輯閘陣列是由十個邏輯單元所構成，邏輯單元的架構如圖 2-3 所示[1]。

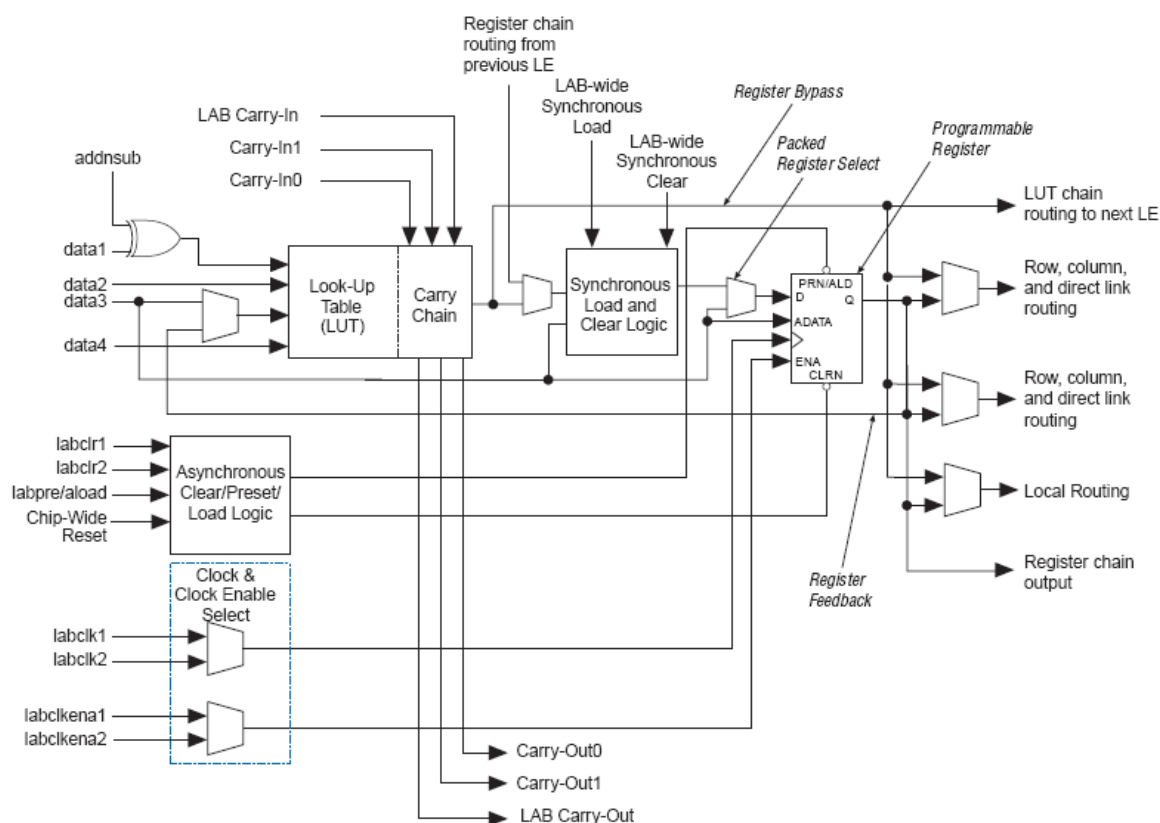


圖 2-3 邏輯單元架構圖

總和來說，不論 FPGA 內部是否提供特殊功能的架構來開發所需功能之特用晶片，使用者本身亦可以選擇應用可程式通道連接的方式，來連接 FPGA 內部的邏輯單元與各式區塊以達到所需目的，此為其應用上之最基本架構。

2-3 IP 的設計流程

智慧技術元件(IP)為經過設計、驗證且具備特定功能的積體電路設計[49]。依照設計流程的不同 IP 又可分為(1)軟核 IP(soft IP)、(2)韌核 IP(firm IP)與(3)硬核 IP(hard IP)等三種，表 2-1 為各種 IP 特性之列表之比較，開發者可以依據本身需求選擇合適的 IP 種類。

表 2-1 各類 IP 比較表

項目	應用彈性	製程彈性	修改彈性	整合風險
軟核 IP	高	高	高	高
韌核 IP	中	中	中	中
硬核 IP	低	低	低	低

IP 的設計流程較 IC 的設計流程簡單，由圖 2-4 可以看出 IP 的設計流程僅為 IC 的設計流程的分支，而不同種類的 IP 其流程分支與開發時間的長短皆不相同，但基本上還是遵循了由上而下(top-down)的設計方法[29]，目前對於 IP 的規範都來自 VSIA(Virtual Socket Interface Alliance)聯盟，此聯盟致力於 IP 公開標準的制定，以確保設計出的 IP 具有一致性。

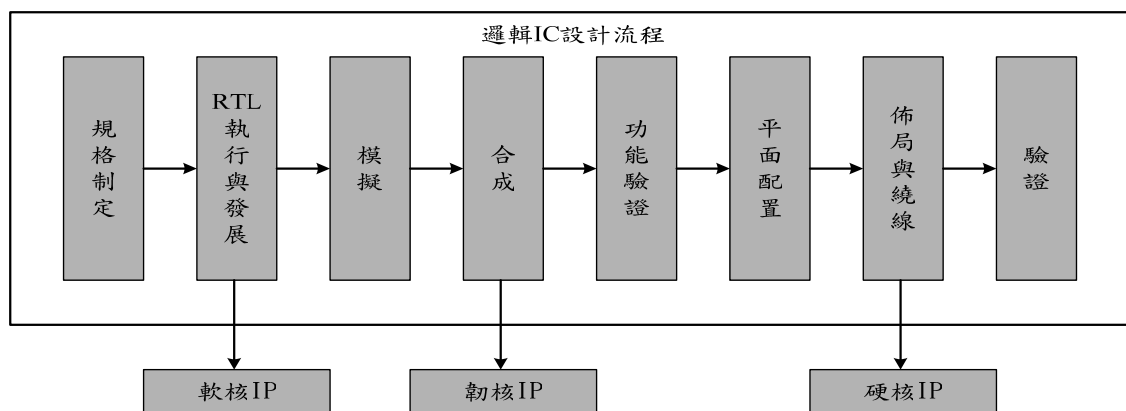


圖 2-4 IP 設計流程圖

2-4 VHDL 硬體描述語言

VHDL (VHSIC Hardware Description Language)原為美國國防部在1980年代為了開發戰鬥機中的積體電路而特別制定的一種硬體描述語言，其設計本意為將電子電路以文件形式及軟體方式撰寫及保存下來。而 VHSIC 為“Very High Speed Integrated Circuits Program”之簡稱，主要為提供一個能快速驗證的語言工具來發展涵蓋電路描述、合成以及模擬等設計步驟。1987年美國軍方轉移美國電機電子學會(IEEE)後，正式將 VHDL 制定成 IEEE 1076 標準。

為了縮短開發時程以因應需求，VHDL 也被廣泛地應用於商用 IC 產品的開發上，1993 年被加以增修制定成 IEEE 1164 標準，1996 年再將電路合成的規格及程序加入成為 IEEE 1076.3 標準。透過這些製定的標準規範，使用者在開發上可依據這些標準規格開發通用的 IC 描述格式。

2-5 小結

本章節為對於開發 IP 的流程與所需的軟硬體如 FPGA 與 VHDL 等作一簡介，主要目的為提供欲開發 IP 之研發者一些簡單的切入觀點，期能提供對於 IP 開發設計的初步了解與認識，如需進一步了解硬體描述語言的撰寫與使用可參考[30][37]。

第三章 加減速規劃 IP

由於機構及致動器的特性，大多數機台會有速度、加速度和急衝度(jerk)之限制，因此機台在移動時運轉需平順、降低震動，否則過大的運動速度或加速度會造成機台的損壞。當致動器為電動馬達時，若其電流與速度命令是連續的，則機台較不會有急遽的變化發生。因此機台的控制成功與否和速度命令的加減速規劃是否合宜息息相關。本章節中將介紹加減速規劃架構、傳統多項式加減速規劃及本文所採用的數位迴旋積分加減速規劃。

3-1 加減速規劃架構

一般而言，加減速規劃有兩種架構——前加減速規劃及後加減速規劃[19][48]，以下為這兩種加減速架構的簡介。

◆ 前加減速架構

前加減速顧名思義就是進給率資料與曲線資料在進入插值器之前先進行加減速曲線規劃，如圖 3-1 所示。其優點就是不會有路徑命令誤差的出現，缺點為計算上較為複雜。

◆ 後加減速架構

後加減速為進給率資料與曲線資料在進入插值器之後再由各軸自行進行加減速曲線規劃，如圖 3-2 所示。其優點為實現容易，但其缺點為會有路徑命令誤差存在的情況。

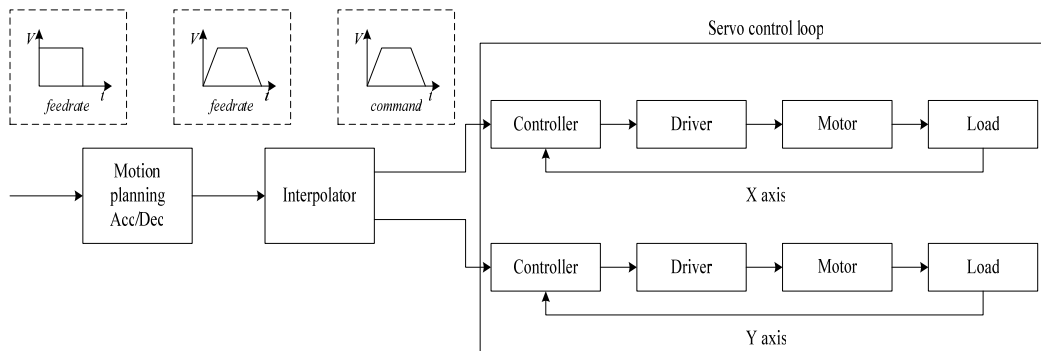


圖 3-1 前加減速架構圖

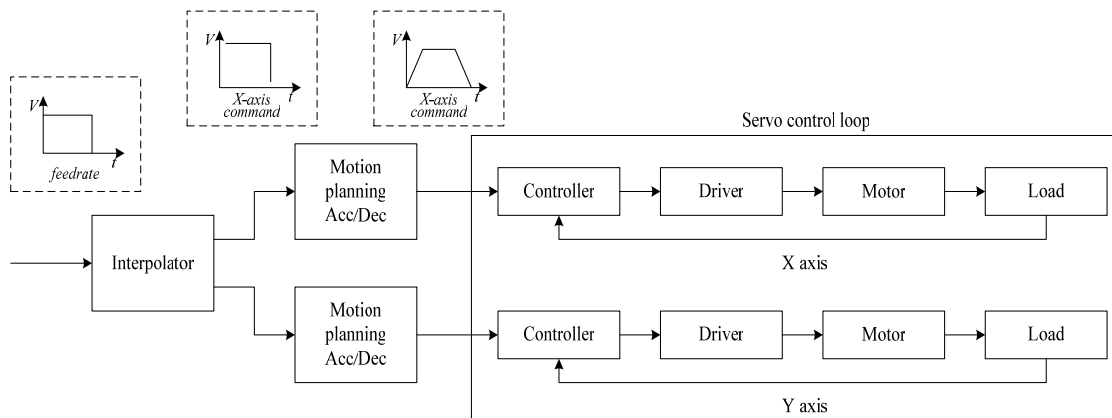


圖 3-2 後加減速架構圖

3-2 多項式之加減速規劃

常見的加減速規劃方式為以多項式來描述加減速曲線，透過多項式可以描述不同的速度曲線，如梯型、S Curve、鐘型、指數型等等。梯型與 S Curve 兩種加減速規劃為最常使用被使用的速度曲線，因此，於下一小節中將對於這兩種加減速曲線作簡單的介紹。

3-2-1 梯型加減速規劃

梯型加減速曲線是由二段一次曲線與一段零次曲線所組成，其速度曲線與加速度曲線如圖 3-3 所示，其參數式表示為由(3.1)至(3.3)所示。從公式中可以得知，梯型加減速曲線主要由以下三個參數決定：

1. 加速度(A)
2. 最大速度(V_{max})
3. 移動總距離(S)

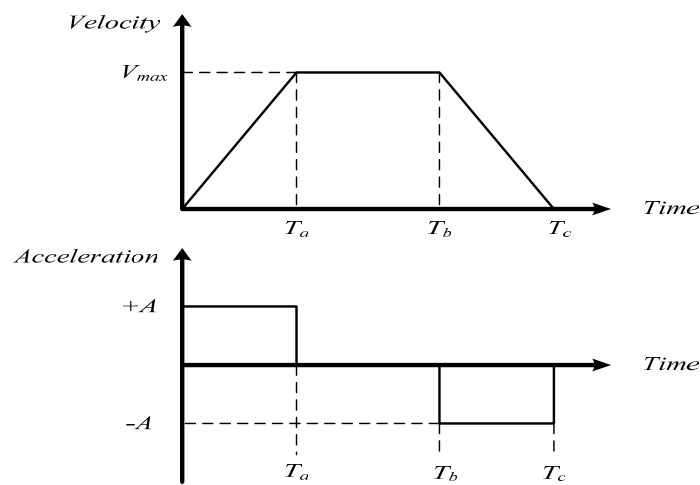


圖 3-3 梯型參數式速度曲線與加速度曲線圖

$$T_a = \frac{V_{\max}}{A} \quad (3.1)$$

$$T_b = \frac{S}{V_{\max}} \quad (3.2)$$

$$T_c = T_a + T_b \quad (3.3)$$

公式中的時間參數為：

T_a ：定加速度時間 $T_b - T_a$ ：最大速度時間

$T_c - T_b$ ：定減速度時間

梯型加減速曲線有一限制條件，其最大速度時間不可以小於零，當最大速度時間等於零時，加減速曲線由梯型變為三角型，而小於零時則代表無法使馬達加速至最大速度。

3-2-2 S Curve 加減速規劃

S Curve 加減速曲線是由四段二次曲線、二段一次曲線與一段零次曲線所組成，其速度曲線與加速度曲線如圖 3-4 所示，其參數式由(3.4)至(3.7)所示。由公式可知，S Curve 加減速曲線主要由以下四個參數決定：

1. 最大加速度(A_{max})
2. 最大速度(V_{max})
3. 移動總距離(S)
4. 平均加速度(A_{avg})

$$T_a = \frac{T_c - (T_b - T_a)}{2} \quad (3.4)$$

$$T_b - T_a = \frac{2 \cdot V_{max}}{A_{max}} - T_c \quad (3.5)$$

$$T_c = \frac{V_{max}}{A_{avg}} \quad (3.6)$$

$$T_d - T_c = \frac{S - V_{max} \cdot T_c}{V_{max}} \quad (3.7)$$

公式中的時間參數為：

T_a ：變加速度時間

$T_b - T_a$ ：最大加速度時間

T_c ：加速度時間

$T_d - T_c$ ：最大速度時間

S Curve 加減速曲線具有兩限制條件如公式(3.8)(3.9)所示，其中公式(3.8)是為了避免最大加速度時間與最大減速度時間小於零或重疊變加速度時間。而公式(3.9)則是避免馬達尚未到達最大速度就已到達目標位置所設的限制。

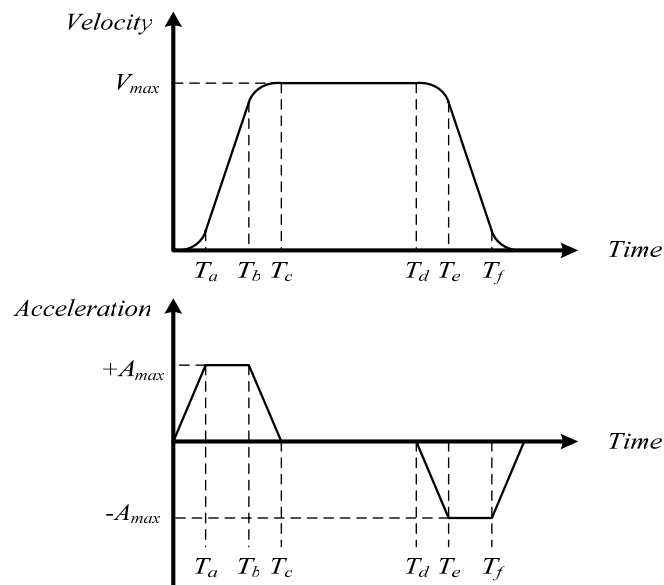


圖 3-4 S Curve 參數式速度曲線與加速度曲線圖

$$0.5 \cdot A_{\max} \leq A_{\text{avg}} \leq A_{\max} \quad (3.8)$$

$$S \geq \frac{2 \cdot V_{\max}^2}{A_{\max}} \quad (3.9)$$

3-3 數位迴旋積分之加減速規劃

由上一小節得知，使用多項式來描述加減速曲線雖然對於曲線的設

計上較有彈性及多變性，但由於運算式的繁雜導致計算上的負擔增加，並且所需的參數亦繁多，相當不利於硬體運算。有鑑於此，本論文參考 Jeon 等學者所提出的數位迴旋積分之加減速規劃[5]實現於 FPGA 內，來完成加減速規劃 IP。此種方法運算過程簡單，並且使用相同的硬體設計與四個相同的參數就可以同時達成梯型與 S Curve 的加減速規劃。於開發晶片上，採用此種方法規劃加減速曲線可以達到節省邏輯單元與提升整體運算效能，因此，十分適合用於發展運動控制命令所需之加減速規劃 IP 的設計。於本小節中將介紹如何以數位迴旋積分進行梯型與 S 型加減速規劃，並對於本論文所設計之加減速規劃作一深入的探討。

3-3-1 梯型加減速規劃

首先考慮梯型加減速曲線部分，其所需給定之參數如下所示：

1. 移動總距離(S)
2. 最大速度(V_{max})
3. 梯型加減速參數(N_l)

基於最大速度 V_{max} 與移動總距離 S ，可以得到一速度序列 $X[i]$ ，其公式如下所示：

$$N = \frac{S}{V_{max}} \quad (3.10)$$

經由梯型加減速參數 N_l ，可產生一數位迴旋積分參數序列 $Y_1[j]$ ，為

避免最大速度時間小於或等於零，則必須選擇限制條件如下所示：

$$N_1 < 0.5 \times N \quad (3.11)$$

利用速度序列 $X[i]$ 與數位迴旋積分參數序列 $Y_1[j]$ ，經一次數位迴旋積分後，產生之梯型加減速序列 $X_1[k]$ ，如圖 3-7 所示，而數位迴旋積分公式如(3.12)所示。

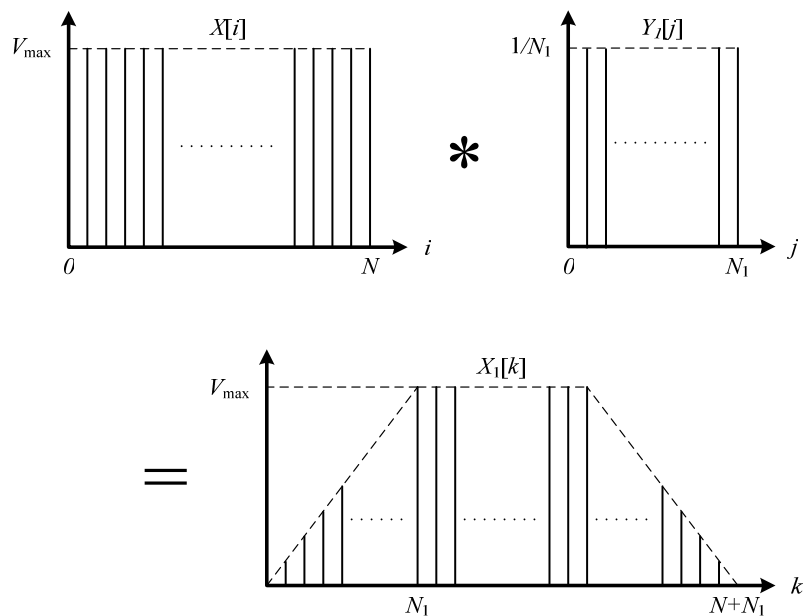


圖 3-5 數位迴旋積分之梯型加減速示意圖

$$X_1[k] = \frac{X[k] - X[k - N_1]}{N_1} + X_1[k - 1] \quad (3.12)$$

3-3-2 S Curve 加減速規劃

S Curve 加減速規劃是將梯型加減速規劃後的速度曲線再次進行數

位迴旋積分所產生的加減速曲線，其所需參數除了梯型加減速規劃的三個參數外，必須再加上 S Curve 加減速參數 N_2 。為避免最大加速度時間與最大減速度時間小於零或重疊變加速度時間， N_2 必須滿足條件 $N_2 < N_1$ ，經由 S Curve 加減速參數 N_2 可產生一數位迴旋積分參數序列 $Y_2[l]$ 。

利用梯型加減速序列 $X_1[k]$ 與數位迴旋積分參數序列 $Y_2[l]$ ，經一次數位迴旋積分後，產生之 S Curve 加減速序列 $X_2[m]$ ，如圖 3-6 所示。

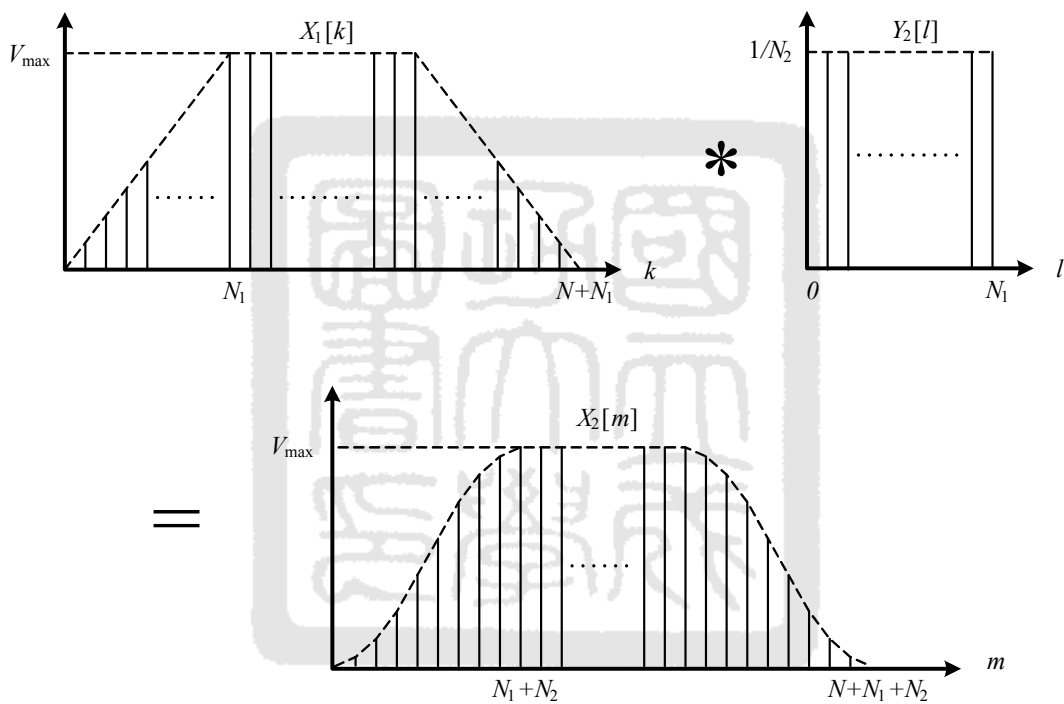


圖 3-6 數位迴旋積分之 S Curve 加減速示意圖

3-3-3 定點數數位迴旋積分加減速規劃

由於 Jeon 等學者所提出的數位迴旋積分實現上為以軟體程式設計實現，但若以 FPGA 實現時，使用浮點數(Floating point)設計的方式會造成

小數點誤差的累積。相較於使用定點數(Fixed point)運算的處理方式，在硬體的設計上，除了運算單元不易實現外，也會導致運算時間較長。為避免此一缺失，本論文將浮點數運算之數位迴旋積分加減速規劃改為以全定點數運算的方式，目的為完成一零誤差之加減速規劃 IP，詳細設計過程如下。

本文首先考慮梯型加減速規劃，相較於浮點數數位迴旋積分加減速規劃的參數，除梯型加減速參數 N_1 為整數無須修改外，最大速度 V_{max} 修改為最大輸出脈波數 P_{max} ，移動總距離 S 修改為輸出總脈波數 T_p ，如此一來，所有參數皆為整數型式。

基於最大輸出脈波數 P_{max} 與輸出總脈波數 T_p 等等設定參數，可以得到一速度脈波數序列 $P[i]$ ，如圖 3-7 所示。且由於使用定點數運算的方式，因此將公式(3.10)修改為公式(3.13)，其中 N 為最大輸出脈波數序列的個數， R 為殘餘脈波數，由脈波數序列 $P[i]$ 中可分割出最大輸出脈波數序列 $P_1[i]$ ，其定義如公式(3.14)所示：

$$T_p = N \cdot P_{max} + R \quad 0 \leq R < N \quad (3.13)$$

$$P_1[i] = \begin{cases} P_{max} & 1 \leq i \leq N \\ 0 & otherwise \end{cases} \quad (3.14)$$

經過以上的設計步驟後，所選定之梯型加減速參數 N_1 所產生數位迴旋積分參數序列 $Y_1[j]$ ，將維持不變，其限制條件也不變。

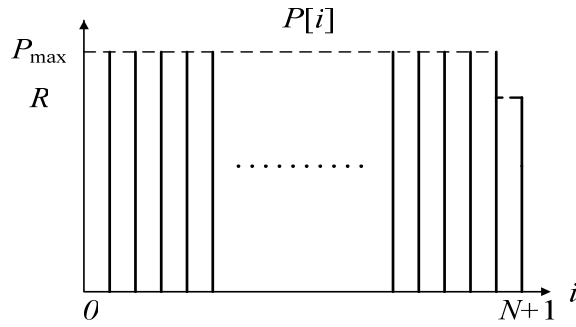


圖 3-7 速度輸出脈波數序列 $P[i]$

將最大輸出脈波數序列 $P_1[i]$ 與數位迴旋積分參數序列 $Y_1[j]$ 經一次數位迴旋積分後，產生未補償殘餘脈波數之梯型加減速序列 $P_2[k]$ ，如圖 3-8 所示。爾後，將討論餘數補償部分，首先，將殘餘脈波數插入 $P_2[k]$ 序列中作補償，進而產生補償後之梯型加減速序列 $P_3[m]$ 。本論文所規劃的殘餘脈波數補償方式有兩種：1). 將殘餘脈波數補償於 $P_2[k]$ 序列的末端，如圖 3-9(a) 所示。 2). 將殘餘脈波數補償於 $P_2[k]$ 序列的加速及減速區域，如圖 3-9(b) 所示，詳細的補償方法將陳述於 3-3-4-1 小節。而在數位迴旋積分過程中因定點數除法而產生的餘數，其詳細的補償方法將陳述於 3-3-4-2 小節。

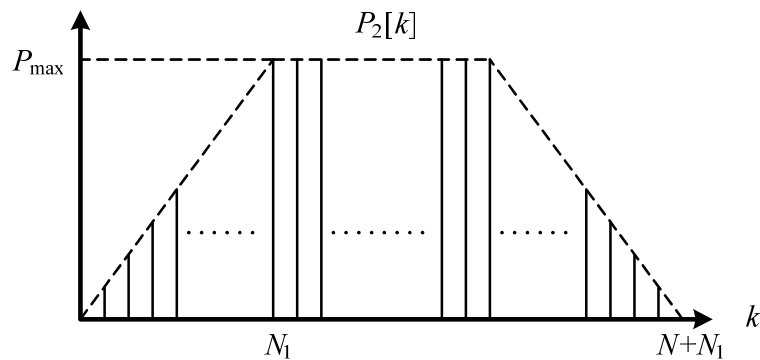
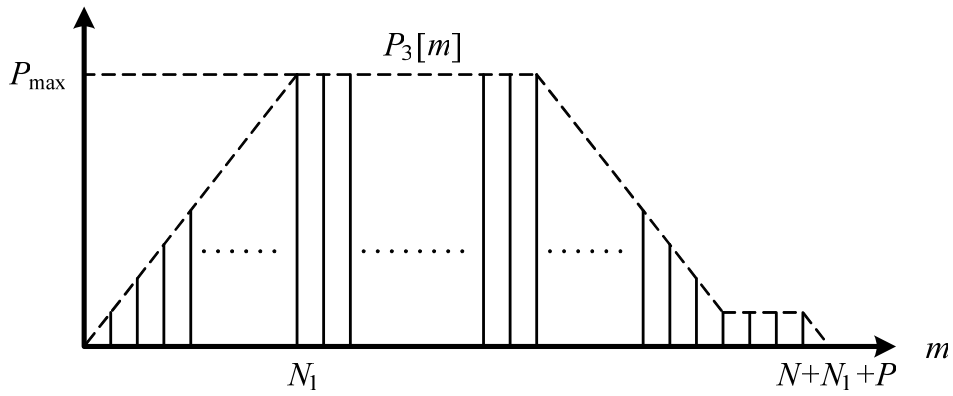
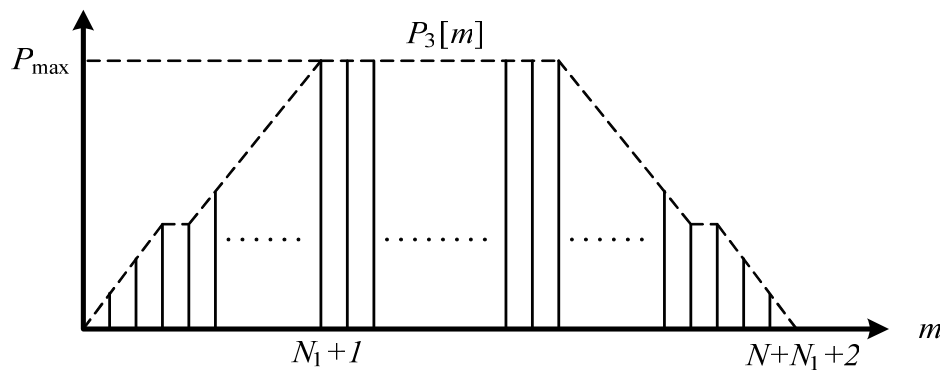


圖 3-8 未補償殘餘脈波數之梯型加減速序列 $P_2[k]$



(a)



(b)

圖 3-9 補償後之梯型加減速序列 $P_3[m]$ (a). 殘餘脈波數補償於 $P_2[k]$ 序列的末端 (b). 殘餘脈波數補償於 $P_2[k]$ 序列的加速及減速區域

至於在 S Curve 加減速規劃方面，因加入之 S Curve 加減速參數 N_2 為整數，所以不會影響其定點數運算，其限制條件也不變，故利用其所產生之位迴旋積分參數序列 $Y_2[l]$ 也不變。將未補償之梯型加減速序列 $P_2[k]$ 與數位迴旋積分參數序列 $Y_2[l]$ 經一次數位迴旋積分後，即產生未經補償殘餘脈波之 S Curve 加減速序列 $P_4[n]$ ，如圖 3-10 所示。將殘餘脈波插入 $P_4[n]$ 序列中作補償，進而產生補償後之 S Curve 加減速序列 $P_5[o]$ ，補償後之序列如圖 3-11 及圖 3-12 所示。其中圖 3-11 為將殘餘脈波數補償於 $P_4[n]$ 序列的末端，圖 3-12 為將殘餘脈波數補償於 $P_4[n]$ 序列的加速

及減速區域。

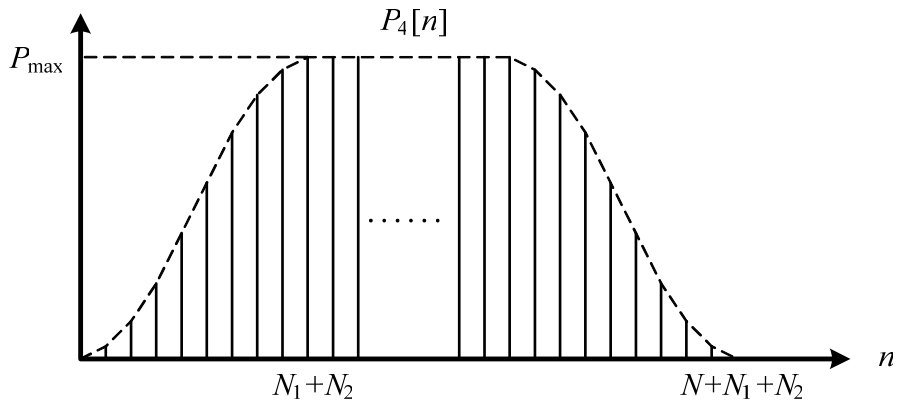


圖 3-10 未補償殘餘脈波數之 S Curve 加減速序列 $P_4[n]$

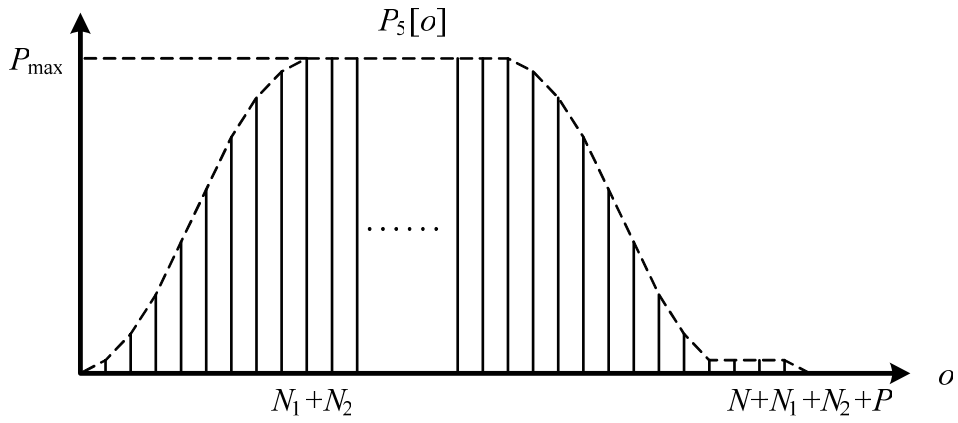


圖 3-11 補償殘餘脈波數於 $P_4[n]$ 序列的末端

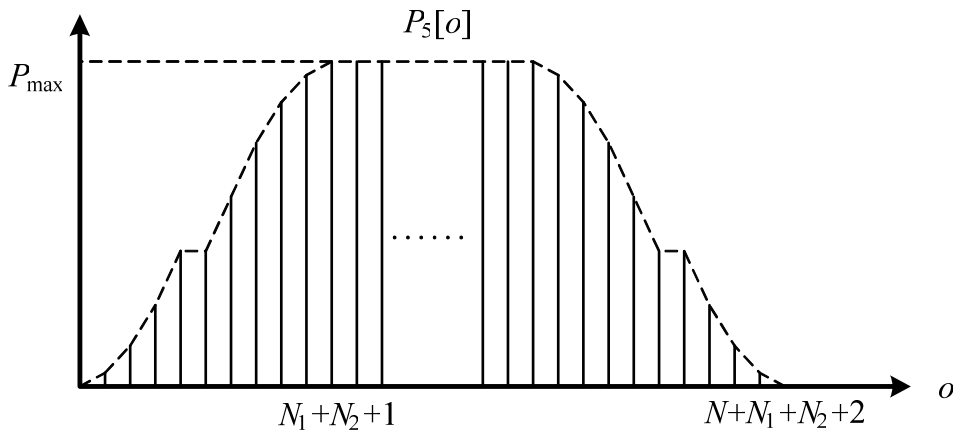


圖 3-12 補償殘餘脈波數於 $P_4[n]$ 序列的加速及減速區域

3-3-4 定點數之數位迴旋積分餘數補償機制

為提昇整體運作效能，本論文所發展之加減速規劃於硬體架構實現上採用定點數運算型態，然實現數位迴旋積分運算式時，必須考慮除法器的使用上會有餘數產生的情形。為了克服餘數所可能衍生出的誤差問題，本論文提出餘數補償的機制，利用補償計算過程所產生的餘數方式達到零誤差之精度。於本論文的研究中，可以發現不論在梯型加減速規劃或 S Curve 加減速規劃的過程，皆會有餘數產生的問題需要考慮。為了避免因為這些餘數的問題而產生命令誤差，本小節中將探討當餘數產生時之補償機制。

由前面章節的討論可以得知，本論文所探討之餘數產生的情況有兩種：第一種餘數產生於數位迴旋積分之前，為產生脈波序列 $P[i]$ 而出現之餘數，可由公式(3.13)所得知；第二種餘數為於數位迴旋積分運算的過程中產生，可由公式(3.12)得知。

3-3-4-1 數位迴旋積分之前的餘數補償

於本小節中，我們將詳細介紹兩種數位迴旋積分的餘數補償方法。

第一種補償的方法為使用公式(3.13)中的餘數 R ，插入未補償加減速規劃曲線的末端。根據餘數 R 及除數 V ，可以得到加減速規劃之補償公式如(3.15)所示，其中定義 V 為未補償加減速規劃曲線的最後一個脈波數，這一個脈波數在梯型加減速規劃中就是 $P_2[N+N_1-1]$ 序列，在 S Curve

加減速規劃中就是 $P_4[N+N_1+N_2-1]$ 序列，而定義 A 為所需插入 V 值的序列次數，定義 B 為最後插入的序列值。其示意圖如圖 3-9(a)及圖 3-11 所示。圖中 P 值表示所需補償插入的次數，倘若所剩餘的最後序列值為零，則 P 等於 A ，其公式如(3.16)所示。

$$R = A \cdot V + B \quad (3.15)$$

$$P = \begin{cases} A & B = 0 \\ A + 1 & otherwise \end{cases} \quad (3.16)$$

第二種補償的方法為使用公式(3.13)中的餘數 R ，分別插入未補償加減速規劃曲線的加速與減速區間，其補償方式為將(3.13)式中的餘數 R ，取其對半之序列值分別插入前加減速區間(E 為加速區間， F 為減速度區間。若式 3-17 之餘值 D 為非零，則 F 則為商序列值 C 加 1)，其加減速規劃之補償公式如(3.17)所示，於加速區間插入的值 E 與減速區間插入的值 F 之公式如(3.18)(3.19)所示，詳細的補償示意圖如圖 3-9(b)及圖 3-12 所示。

$$R = 2 \cdot C + D \quad (3.17)$$

$$E = C \quad (3.18)$$

$$F = \begin{cases} C & D = 0 \\ C + 1 & otherwise \end{cases} \quad (3.19)$$

3-3-4-2 數位迴旋積分過程中的餘數補償

由公式(3.12)得知於數位迴旋積分的過程中，因除法器的使用且由於硬體實現使用定點數運算的原因，仍會有餘數產生的問題。因此，本節

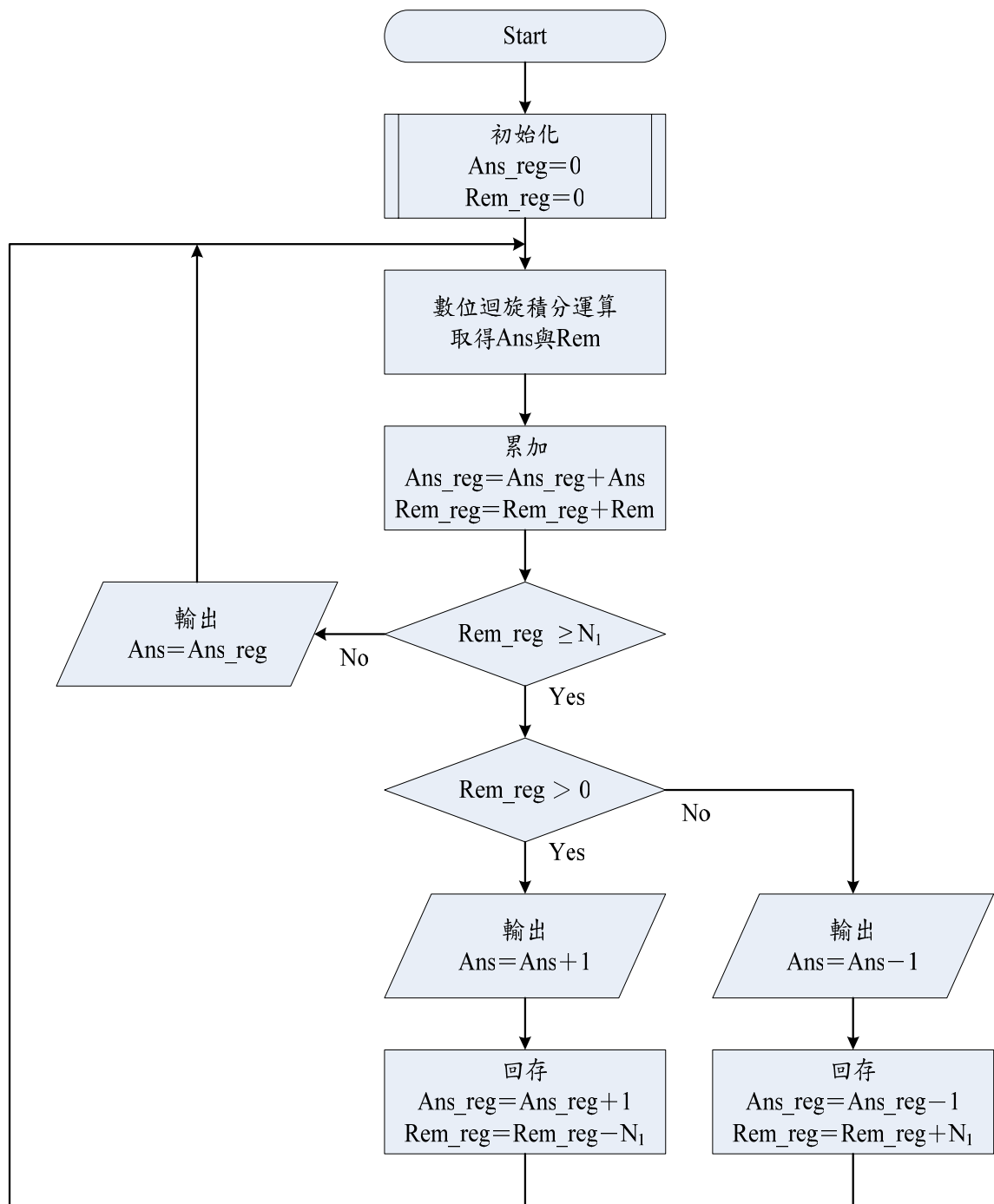


圖 3-13 梯型加減速規劃數位迴旋積分過程中補償流程圖

將探討於實現數位迴旋積分過程中之餘數補償機制。其詳細的補償步驟解釋如下(以產生梯型加減速為例)，整體流程圖如圖 3-13 所示。其中 Rem_reg 表示餘數暫存器，Ans_reg 表示商數暫存器，Rem 表示目前餘數

值，Ans 表示目前商數值。

步驟一、暫存器值初始化。

步驟二、計算數位迴旋積分並更新計算後之商值及餘數值。

步驟三、判斷餘數絕對值是否大於梯型(S 型)加減速參數 N_1 (N_2)，若是則執行步驟四，若不是則回到步驟二。

步驟四、加速或減速曲間判斷及輸出補償。

步驟五、計算並更新餘數暫存器，更新商數暫存器，回到步驟二。

圖 3-13 為梯型加減速規劃於數位迴旋積分過程中之餘數補償的流程圖，若要實現 S 型加減速規劃及探討餘數補償機制，只需將 N_1 改寫為 N_2 ，即可表示成 S Curve 加減速規劃於數位迴旋積分過程中餘數補償的流程圖。

3-4 架構設計

本節介紹加減速 IP 之硬體架構設計，其架構如圖 3-14 所示，其中共分為八個主要區塊，其功能如下。Input Block：栓鎖輸入之加減速參數。FSM：為有限狀態機負責控制協調各區塊間的運作。Debounce Block：由於 FSM 在狀態改變時輸出會有彈跳現象產生導致輸出錯誤故使用此區塊改善。Arith Block：資料運算之區塊。RAM：儲存梯型資料以供 S curve 運算。Address Counter：記憶體位置計數。Sel Block：資料流之切換區塊。Make_up Block：餘數補償區塊。

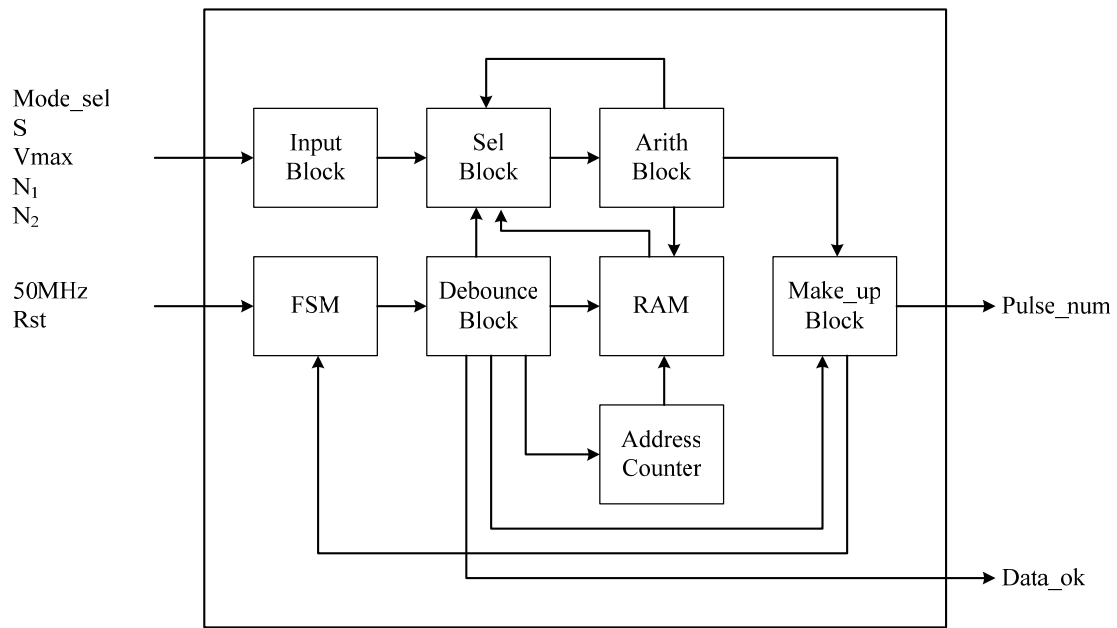


圖 3-14 加減速 IP 硬體架構圖

3-5 小結

本章介紹了常見的參數式梯型與 S Curve 加減速曲線規劃方式，但基於硬體實現的考量，本論文使用定點數數位迴旋積分的方式實現一零誤差之加減速規劃 IP。

第四章 脈波命令輸出 IP

工具機主要的動力來源為馬達，常見於工具機上使用的 AC/DC 伺服馬達，使用者可以依自己的需求設定驅動器所提供的位置模式、速度模式與轉矩模式等三種操作模式，模式架構圖如圖 4-1 所示。這三種模式的輸入命令格式有不同的形式，如位置模式為脈波命令輸入的形式，速度模式與轉矩模式則為電壓命令輸入形式。

在點對點運動中，常使用的伺服馬達操作模式為脈波命令輸入的方式。操作此種模式下，其優點在於使用者無須自行設計控制器，而是使用驅動器內建的控制迴路架構，使用者僅需調整伺服驅動器參數即可。也因為如此，在此種模式下，精密定位除了要仰賴前面所探討之合適加減速規劃外，另外一個重要的探討因素為是否有準確的脈波命令提供。因此，本論文所研究及發展之脈波命令輸出 IP，主要目的為發展零誤差之脈波輸出產生器。本論文將以具備均勻輸出脈衝的特點之 DDA 為發展藍圖架構，並以硬體描述語言於 FPGA 內實現。

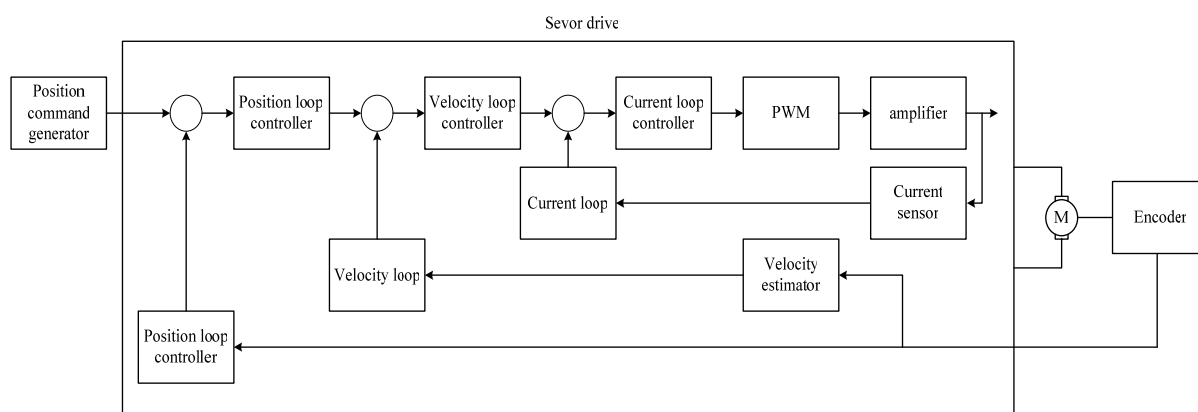


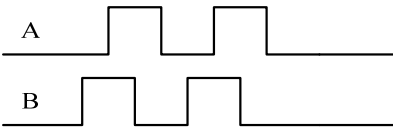
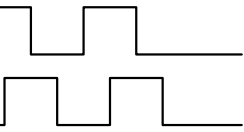
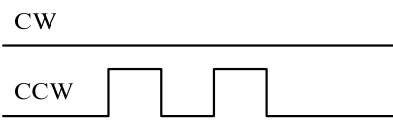
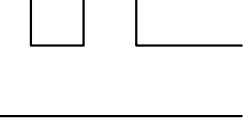
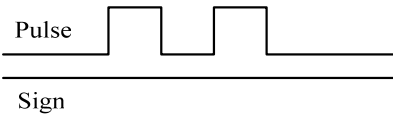
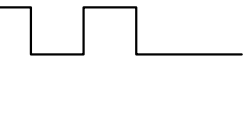
圖 4-1 馬達驅動模式架構圖

4-1 脈波命令與伺服驅動器

於前一小節中，曾提到當驅動器設定在位置模式下時，由於僅需輸入脈波命令，其餘所有的控制迴路皆使用伺服驅動器中的內建的控制器，驅動器使用者僅需透過參數調整設定使系統達到最佳狀態即可，使用上十分的方便。

一般伺服驅動器接收脈波命令的格式可分為三種，如表 4-1 所示 [18]。除此之外，驅動器對於所接受的脈波寬度也有所限制，因此，在設計輸出脈波的寬度上，必須依據所使用之伺服驅動器的規格，來決定所規劃的脈波寬度是否可達到伺服驅動器可以接受的標準而定。在本論文中，所規劃的驅動命令格式採用 Pulse/Sign 的格式實現脈波命令輸出 IP 之脈波輸出。

表 4-1 伺服驅動器位置模式脈波命令輸入分類訊號表

格式	CCW command	CW command
A/B phase		
CW/CCW		
Pulse/Sign		

在伺服驅動器的命令接收上，不論設定於何種工作模式，皆是以離散的方式輸入，如圖 4-2 所示。橫軸為時間，縱軸為馬達轉動量，分別表示在每一個 Δt 時送出一命令值，再將命令值以直線段予以連接，這樣可取得一直線片段連續曲線。此時，若 Δt 長度越小，則直線片段連續曲線就越逼近目標軌跡，理論上 $\Delta t \rightarrow 0$ 則直線片段連續曲線等於目標軌跡。但由於伺服驅動器的響應速度有所限制，並由一般的伺服控制中所獲得的經驗法則而知，當操作於在位置模式時，輸入伺服驅動器的命令下達時間(命令週期) 通常會設定在 $\Delta t = 2 \sim 5ms$ ，以獲得整體伺服控制系統的最佳效能。

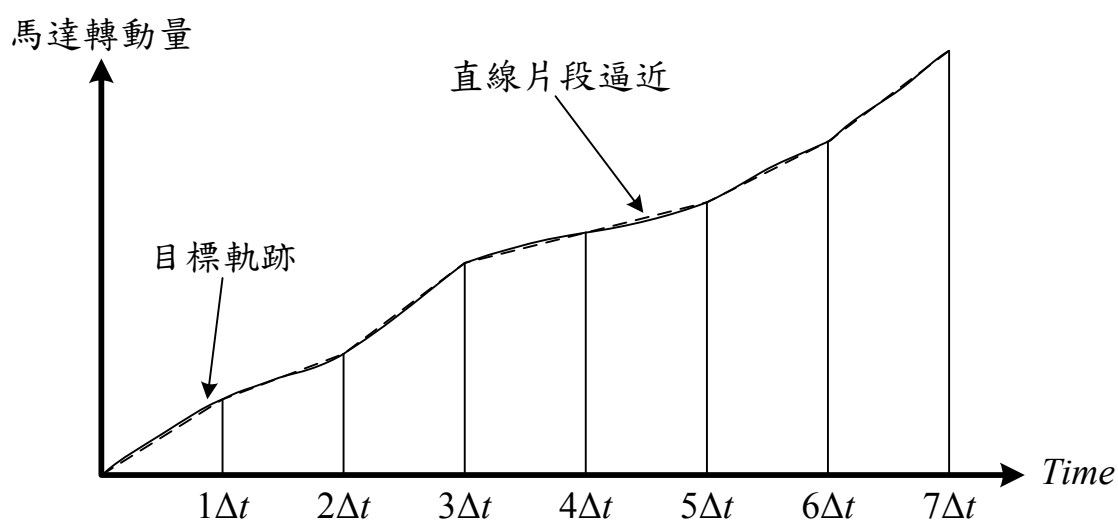


圖 4-2 離散命令與實際目標示意圖

觀察圖 4-2 中的每一個 Δt 片段，由於使用直線片段逼近，故其斜率相近於馬達的角速度。如此一來，馬達是以等速方式前進，也由於在每一片段中馬達需要以等速方式前進，所以必須將脈波命令在 Δt 時間內均勻地輸入至伺服驅動器。

4-2 DDA 脈波命令功能

脈波命令輸出方面，本論文採用業界常見的 DDA 脈波輸出方式，其優點為能在命令週期時間內均勻地輸出命令脈波訊號。使用 DDA 來實現脈波命令的輸出，所需的輸入參數為定位命令(即所需輸出脈波數)與命令週期時間(即 Δt)。經過 DDA 運算處理後，即可將規劃所需輸出的脈波數於 Δt 時間內均勻地送出，確保伺服馬達或步進馬達在 Δt 時間內走完定位命令下達之行走步數。

如圖 4-3 所示，可看出馬達轉動步數與 DDA 命令的關係，其中橫軸為時間，縱軸為馬達轉動步數。在第一區間中 DDA 命令為正轉 500 步， $500 \text{ 步}/\Delta t$ 為其速度；第二區間中馬達停止不動；第三區間中 DDA 命令為正轉 1000 步， $1000 \text{ 步}/\Delta t$ 為其速度；第四區間馬達停止不動；第五區間 DDA 命令為反轉 1500 步，而 $-1500 \text{ 步}/\Delta t$ 為其速度。

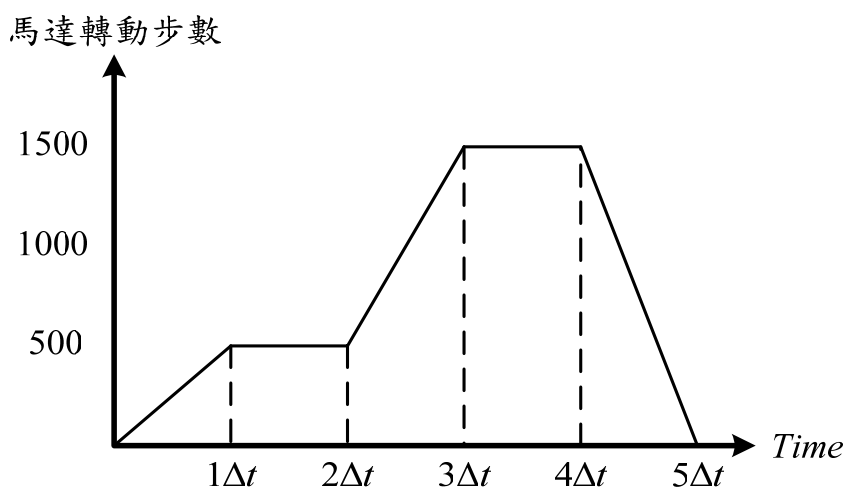


圖 4-3 馬達與 DDA 命令關係圖

4-3 DDA 脈波命令演算法

DDA 脈波命令的演算法，是使用累加與溢位來完成，Koren 等學者所採用之 DDA 其基本的架構如圖 4-4 所示[23]。

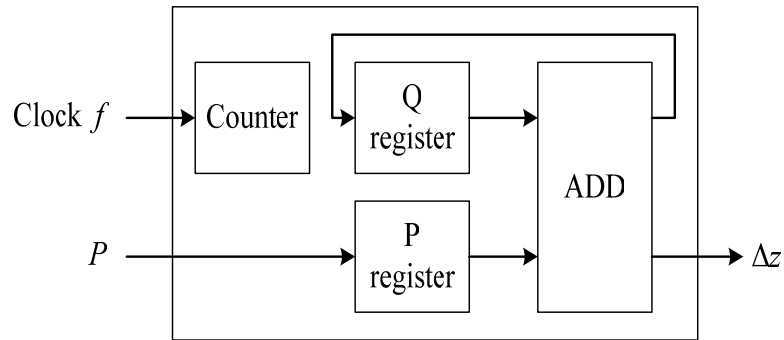


圖 4-4 DDA 脈波命令基本架構圖

圖 4-4 中，P 暫存器、Q 暫存器與加法器的大小都為 n Bits，計數器大小為 $n+1$ Bits，Clock f 為觸發頻率， P 為在 Δt 時間內期望之輸出脈波數， Δz 為加法器溢位輸出，其中 DDA 週期與期望之輸出脈波數 P 的公式如下：

$$\Delta t = \frac{2^n}{f} \quad (4.1)$$

$$0 \leq P \leq 2^n \quad (4.2)$$

DDA 脈波命令演算法的運作步驟如下所示：

步驟一、Q 暫存器與計數器皆清空為零，P 暫存器栓鎖住輸入的 P 值。

步驟二、判斷 Clock f 是否觸發，是則計數器加一並跳到步驟三，否則繼續步驟二的判斷。

步驟三、計數器是否等於 $2^n + 1$ ，是則跳到步驟五，否則跳到步驟四。

步驟四、Q 暫存器加上 P 暫存器並回存到 Q 暫存器，並跳回步驟二。

步驟五、結束。

假設在 $n = 3$ Bits、 $P = 6$ 與 $\text{Clock } f = 1 \text{ kHz}$ 的情形下，DDA 所產生的脈波命令如下所示：

$$\Delta t = \frac{2^3}{1\text{kHz}} = 8\text{ms}$$

表 4-2 DDA 脈波命令演算表

Counter	P reg	Q reg	ADD ans	Δz
1	6	0	6	0
2	6	6	4	1
3	6	4	2	1
4	6	2	0	1
5	6	0	6	0
6	6	6	4	1
7	6	4	2	1
8	6	2	0	1

4-4 DDA 脈波命令架構改良

在前幾節中，介紹了 DDA 理論的基礎，但事實上於硬體實現時會有些許的不同與需注意的地方。首先由公式(4.1)可算出 DDA 的週期 Δt ，而改變 Clock f 就可改變 DDA 的週期 Δt ，並由於各種伺服驅動器的響應不同，所以下達的命令週期(DDA 的週期 Δt)也不同，故本文中設計一除頻器使 Clock f 有所改變，使得所規劃之 DDA 為可變週期(Δt)之型態。

由表 4-2 所示的結果可知，其中溢位位元 Δz ，為實際脈波命令輸出腳，在連續兩次時脈觸發的累加都有溢位時，但並不會連續輸出兩個脈波，而是輸出一個長達兩倍時脈週期的脈波。其原因為使用加法器會造成延遲的情況產生，導致使用觸發脈波的正向與溢位位元 Δz 做 AND 邏輯運算時會使輸出脈波錯誤，故本論文將觸發脈波的反向 $\overline{\text{Clock } f}$ 與溢位位元 Δz 做 AND 邏輯運算，確保脈波能正確輸出。其相關訊號與時脈關係圖如圖 4-5 所示：

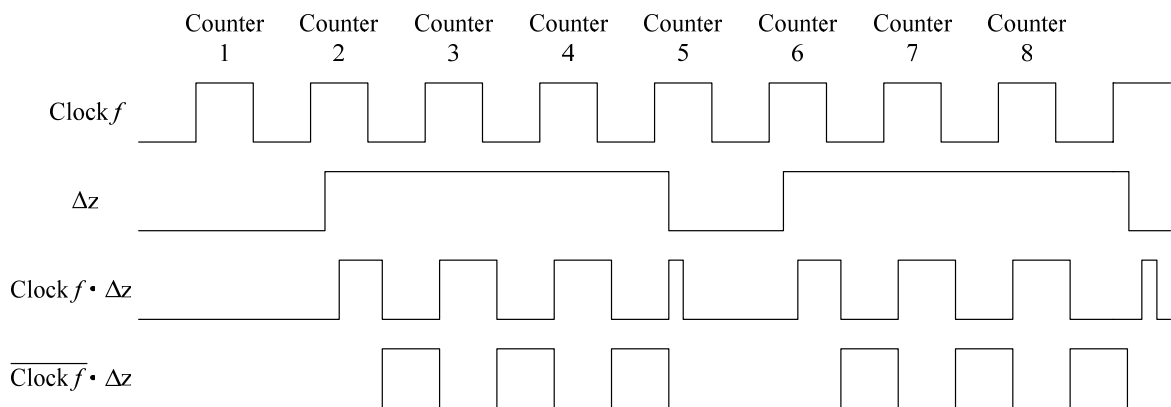


圖 4-5 觸發頻率與輸出脈波關係圖

由於本論文中所設計的除頻器是以外部輸入頻率 f_{in} 與除頻值 m 來

決定 Clock f ，而其架構為計數器與比較器，如圖 4-6 所示。故其輸出 Clock f 的頻率為屬於非對稱(asymmetrical)型式，其 DDA 輸出脈波寬度 w 如公式(4.3)所示。此外，由於本論文採用 Pulse/Sign 的格式實現脈波命令輸出 IP，在硬體運算中則以二補數表示負數，故規劃的輸出脈波數 P 之最高位元可代表其正負號，也就是表示為正反轉信號 Sign。總和以上的修正及改良後，本文所建構之 DDA 脈波命令架構如圖 4-7 所示，其中共分為六個主要區塊，其功能如下，Div f ：非對稱除頻器。Counter：計數是否完成所需之輸出。Abs：決對值運算區塊。Q register 與 P register：儲存 Q 與 P 值之記憶體。ADD：加法器。

$$w = \frac{1}{f_{in}} \times (m - 1) \quad (4.3)$$

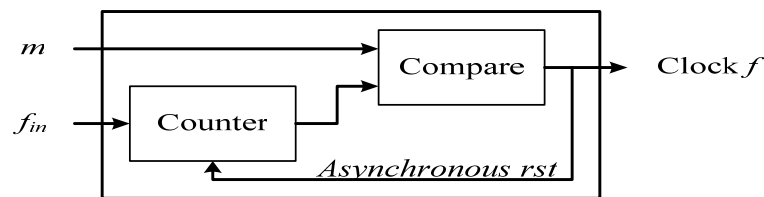


圖 4-6 非對稱除頻器架構圖

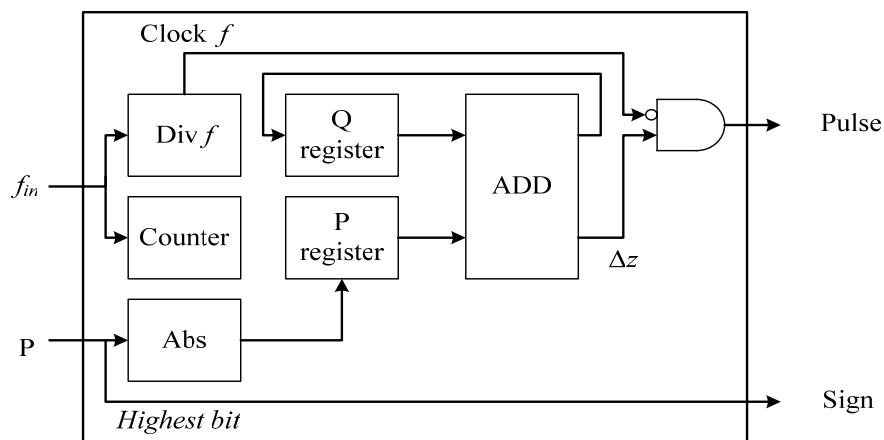


圖 4-7 DDA 脈波命令改良架構圖

4-5 小結

本章節所發展之脈波命令輸出 IP 為以傳統 DDA 架構為發展藍圖，修正實際硬體輸出時所產生的誤差情形，改良成具零誤差輸出且可變週期之 DDA 脈波輸出 IP，期能提供運動控制設計廠商開發脈波產生器時一個參考的依據。

第五章 NURBS 曲線產生 IP

電腦輔助設計與製造系統(CAD/CAM)為現代工業中不可或缺的一環，尤其在模具加工業及汽機車及船舶外型設計相關產業更是如此。在這些產業中，能否完整且準確地呈現出使用者所要的形狀，往往取決於 CAD 系統所提供的 2D 或 3D 的設計軟體工具其函式庫中對曲線/曲面的建構能力。在這些軟體函式庫中，基本上皆具備了以隱函式(implicit form)及參數式(parametric form)的各種曲線/曲面表示式以幫助使用者建構所需的曲線/曲面。對於空間中自由曲線/曲面表示而言，大多是以 Bezier、B-Spline 或 NURBS (Non-Uniform Rational B-Spline)等參數式呈現。

近年來，NURBS 曲線/曲面之應用更廣泛的使用於工具機產業上，目前在業界已有日本 FANUC、Mitsubishi 與德國 SEMENS 等公司，開發出可直接接收由 CAD/CAM 所傳來的 NURBS 曲線/曲面並進行加工之運動控制器，NURBS 曲線/曲面描述已成為 CAD/CAM 系統中加工程式命令(G Code、M Code)的標準格式之一。有鑑於此，本論文研究如何產生 NURBS 曲線並以硬體電路實現於 FPGA 內。並透過規劃成 IP 的功能，提供使用者可以依據所規劃之輸入參數，產生由 CAD/CAM 所設計之 NURBS 曲線。除此之外，本論文並以探討演算法簡化的方式，用以降低 NURBS 的計算量與硬體邏輯單元數，並藉此提升整體演算速度。

5-1 NURBS 參數式曲線簡介

從 70 年代開始的文獻可以發現，NURBS 被用來表示圓錐曲線，80 年代開始則逐漸被使用於 CAD/CAM 電腦繪圖[6][7]，其優點如下所示 [13]：

1. 提供一個共同的數學模型來表示可解析曲線(如：conics, quadrics ,etc)或自由型曲線(free-form curves)。
2. 以 NURBS 來描述不同的曲線或是多變形圖形，往往只需調整控制點(control point)與權重值(weight)即可，提供了更具彈性的輪廓設計方法。
3. 對於座標間的轉換、縮放、旋轉、平移，NURBS 都具有不變 (invariant)的特性。

但是 NURBS 參數式曲線也有下列缺點：

1. 需要更多的資料來定義傳統的曲線，例如要描述一個二維圓形曲線，傳統的描述只要決定圓心與半徑，但是 NURBS 參數式曲線描述則要 7 個控制點、7 個權重值、10 個節點向量與 1 個級數。
2. 權重值不當的選擇將會導致極不好的參數化結果。
3. 傳統的曲線描述在幾何的處理上較 NURBS 簡單，例如傳統的曲線描述對於判斷兩曲線是否相交比較容易。

5-2 NURBS 參數式曲線模型

在討論 NURBS 參數式曲線模型之前，先介紹此模型所需之參數如下所示：

◆ 參數值 u ：

參數值 u 分為正規化與非正規化，在本論文中使用正規化參數值，所以參數值 u 的範圍為 0 到 1。

◆ 級數 p (degree)：

級數 p 為基底函數階次減一，所以級數與階次擇其一為所需參數，在本論文中使用級數 p 。

◆ 控制點(control point)：

控制點用來決定曲線位置，其型態為一陣列，在本論文中使用二維陣列，如公式(5.1)所示，其中分別表示 X 軸與 Y 軸的位置，長度為 $n+1$ 。

$$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = \begin{bmatrix} P_{x_0} & P_{x_1} & \cdots & P_{x_{n-1}} & P_{x_n} \\ P_{y_0} & P_{y_1} & \cdots & P_{y_{n-1}} & P_{y_n} \end{bmatrix} \quad (5.1)$$

◆ 權重值(weight)：

權重值用來決定曲線朝控制點接近的傾向程度，其型態為一維陣列，長度與控制點陣列相同，如下所示：

$$\omega = [\omega_0 \quad \omega_1 \quad \cdots \quad \omega_{n-1} \quad \omega_n] \quad (5.2)$$

◆ 節點向量(knot vector)：

節點向量基於參數 u 來決定曲線的變化，其型態為一維陣列，長

度由級數 p 與控制點陣列長度 $n+1$ 來決定，如下所示：

$$u = [u_0 \quad u_1 \quad \cdots \quad u_{p+n} \quad u_{p+n+1}] \quad (5.3)$$

介紹完所需參數後，NURBS 的數學模型如下所示[14]：

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u) \omega_i P_i}{\sum_{j=0}^n N_{j,p}(u) \omega_j} = \sum_{i=0}^n P_i R_{i,p}(u) \quad (5.4)$$

NURBS 對於空間中曲線描述由公式(5.4)所示，其中符號的定義如下所示：

- $R_{i,p}(u)$: 單一有理式 B 型曲線 (single rational B-spline)
- $N_{i,p}(u)$: 基底函數 (basis function)
- P_i : 控制點 (control point)
- ω_i and ω_j : 權重值 (weight)
- p : 級數 (degree)
- $n+1$: 控制點的個數

其中單一有理式 B 型曲線定義如公式(5.5)所示，基底函數的定義如公式(5.6)與(5.7)所示：

$$R_{i,p}(u) = \frac{N_{i,p}(u)\omega_i}{\sum_{j=0}^n N_{j,p}(u)\omega_j} \quad (5.5)$$

$$N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (5.6)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (5.7)$$

5-3 NURBS 參數式曲線之樹狀結構運算

在上一小節中介紹了 NURBS 參數式曲線的數學模型，在實際的應用上則可分為數個步驟如圖 5-1 所示：

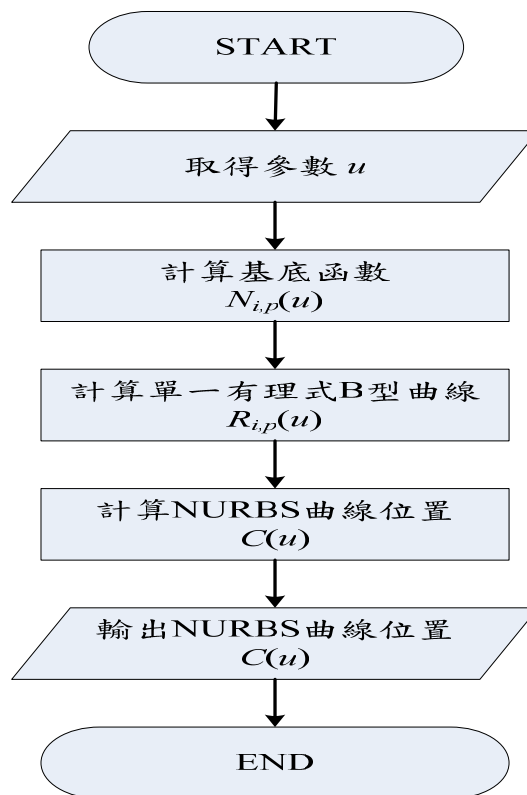


圖 5-1 NURBS 參數式曲線計算流程圖

在流程中首先要計算的是 $p+1$ 階的基底函數 $N_{i,p}(u)$ ，其使用當前的參數 u 代入公式，並在計算基底函數時使用樹狀結構的運算方法[13]，這種運算方法十分適合用於程式語言的撰寫，可使用簡單的迴圈完成其運算，圖 5-2 表示 $p+1$ 階的基底函數 $N_{i,p}(u)$ 的樹狀結構。

基底函數計算完後利用公式(5.2)，由基底函數中最高階的值與權重值計算單一有理式 B 型曲線 $R_{i,p}(u)$ ，最後利用單一有理式 B 型曲線與控制點算出當前參數 u 應對於 NURBS 參數式曲線的位置 $C(u)$ 。

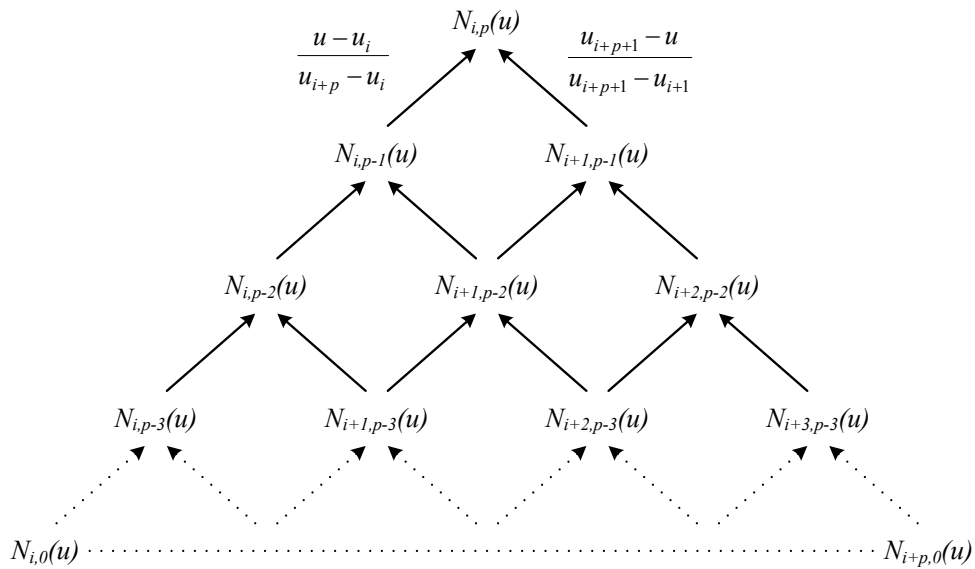


圖 5-2 $p+1$ 階的基底函數 $N_{i,p}(u)$ 的樹狀結構圖

5-4 NURBS 參數式曲線運算化簡

在 NURBS 參數式曲線計算的過程中，雖然使用樹狀結構的運算方法十分方便，但大多數的運算量也都集中在樹狀結構計算，由樹狀結構的運算方法可以看出基底函數是基於第零階基底函數疊代而成，也就是

說第零階基底函數會向上影響，如圖 5-3 所示。

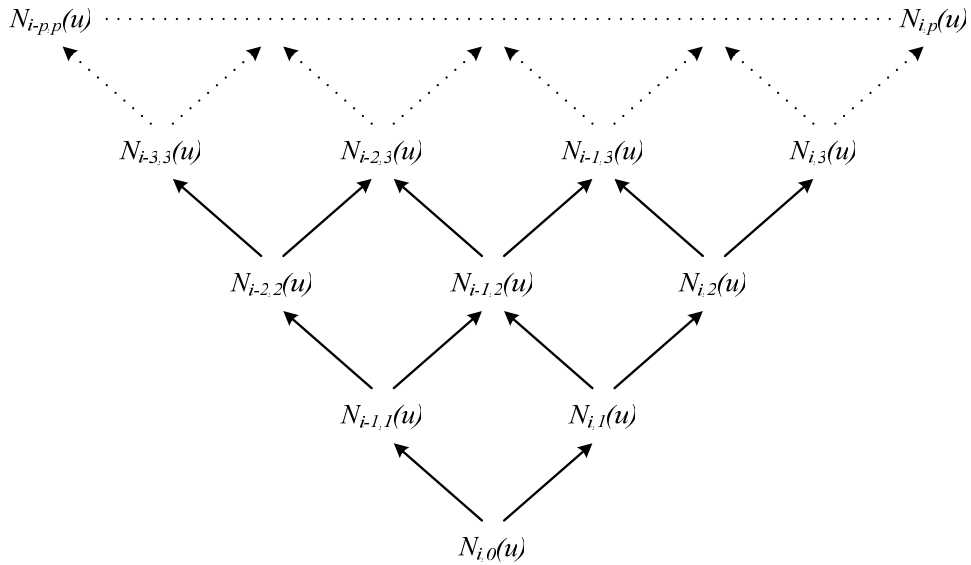


圖 5-3 第零階基底函數對非零階基底函數影響圖

由圖 5-1 所示之流程可知每次僅輸入一個參數 u ，所以重新檢視公式 (5.6)，可以發現在每次僅輸入一個參數 u 的情況下，第零階基底函數僅有一個值為一，其他皆為零，而如圖 5-3 所示第零階基底函數會向上影響，如此一來發現在樹狀結構之中大多數都在做零的運算。假設一 NURBS 曲線其中 $p=2$ 、 $n=6$ ，並且 $u_3 \leq u < u_4$ ，而其樹狀結構如圖 5-4 所示，僅餘一倒三角型區域有非零的值，故在運算基底函數時僅須計算此一倒三角型區域。

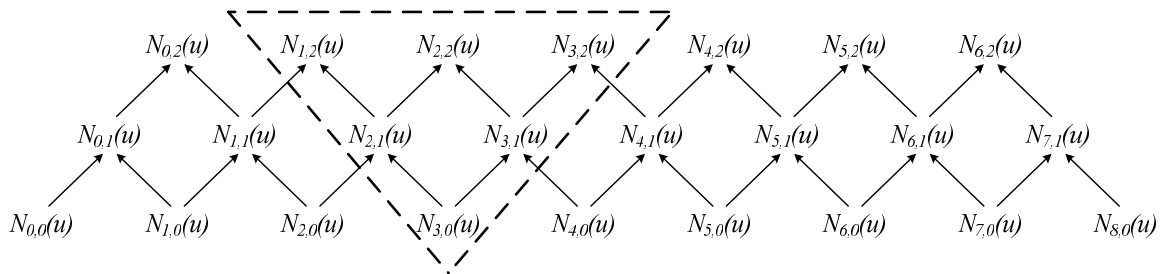


圖 5-4 樹狀結構中非零區域圖

計算完基底函數後由圖 5-1 所示，接下來利用基底函數中最高階的值($N_{i,2}(u)$)與權重值(ω_i)來計算單一有理式 B 型曲線($R_{i,2}(u)$)。由公式(5.5)得知，僅餘與 $N_{i,2}(u)$ (其中 $1 \leq i \leq 3$)有關項為非零值，故單一有理式 B 型曲線僅需計算與非零區域有關的部份。最後利用單一有理式 B 型曲線($R_{i,2}(u)$)與控制點(P_i)算出當前參數 u 對應於 NURBS 參數式曲線的位置 $C(u)$ 。由公式(5.4)得知，僅餘與 $R_{i,2}(u)$ (其中 $1 \leq i \leq 3$)有關項為非零值，故僅需計算非零區域。由上述的例子可得知化簡後的運算流程如圖 5-5 所示。

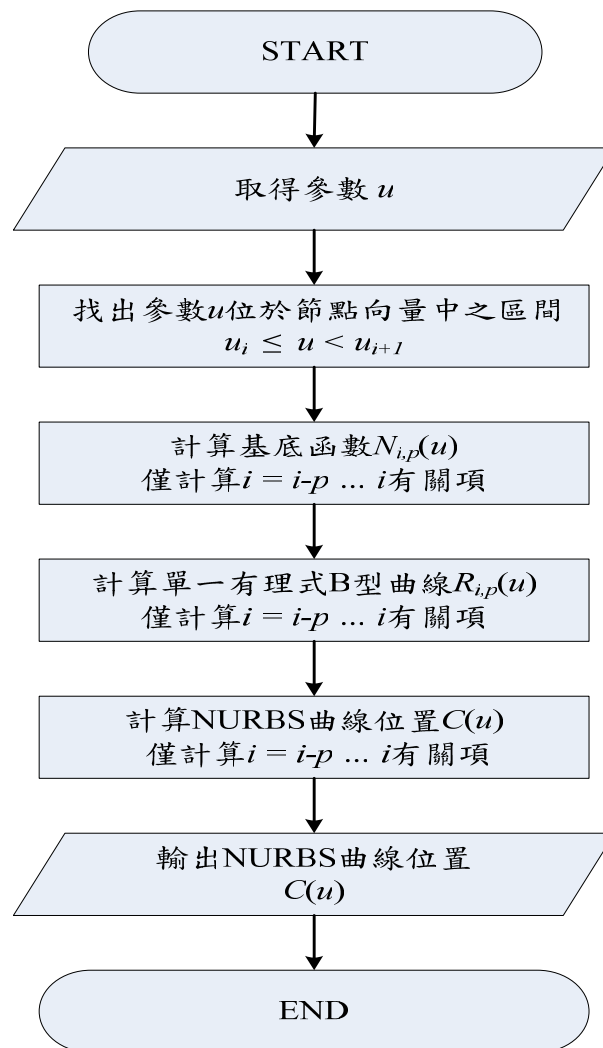


圖 5-5 化簡後之 NURBS 參數式曲線計算流程圖

5-5 樹狀結構與運算化簡之計算方法比較

本節主要在比較 NURBS 參數式曲線樹狀結構的計算方法與去零化簡的計算方法有何不同，而比較的項目有二：一為以 NURBS 參數式曲線軟體設計所注重的計算次數來看，一為以 NURBS 參數式曲線硬體設計所注重的記憶體空間來看。

5-5-1 運算次數比較

在軟體撰寫時，因為硬體架構相同，所以減少運算次數也就是提升效能。在運算次數比較上共分三大部分：

- ◆ 計算基底函數時的運算次數：
包含的運算方式為加法、減法、乘法與除法。
- ◆ 計算單一有理式 B 型曲線時的運算次數：
包含的運算方式為加法、乘法與除法。
- ◆ 計算 NURBS 參數式曲線位置時的運算次數：
包含的運算方式為加法與乘法。

1.) 計算基底函數時的運算次數

使用 NURBS 參數式曲線樹狀結構的計算方法時，其所包含的運算方式之計算次數公式如下所示：

$$\text{加法次數} = \sum_{i=1}^p (n+i) \quad (5.8)$$

$$\text{減法次數} = 4 \sum_{i=1}^p (n+i) \quad (5.9)$$

$$\text{乘法次數} = 2 \sum_{i=1}^p (n+i) \quad (5.10)$$

$$\text{除法次數} = 2 \sum_{i=1}^p (n+i) \quad (5.11)$$

使用 NURBS 參數式曲線去零化簡的計算方法時，其所包含的運算方式之計算次數公式如下所示：

$$\text{加法次數} = \sum_{i=1}^p i \quad (5.12)$$

$$\text{減法次數} = 4 \sum_{i=1}^p i \quad (5.13)$$

$$\text{乘法次數} = 2 \sum_{i=1}^p i \quad (5.14)$$

$$\text{除法次數} = 2 \sum_{i=1}^p i \quad (5.15)$$

2.) 計算單一有理式 B 型曲線時的運算次數

使用 NURBS 參數式曲線樹狀結構的計算方法時，其所包含的運算方式之計算次數公式如下所示：

$$\text{加法次數} = n \quad (5.16)$$

$$\text{乘法次數} = n + 1 \quad (5.17)$$

$$\text{除法次數} = n + 1 \quad (5.18)$$

使用 NURBS 參數式曲線去零化簡的計算方法時，其所包含的運算方式之計算次數公式如下所示：

$$\text{加法次數} = p \quad (5.19)$$

$$\text{乘法次數} = p + 1 \quad (5.20)$$

$$\text{除法次數} = p + 1 \quad (5.21)$$

3.) 計算 NURBS 參數式曲線位置時的運算次數

使用 NURBS 參數式曲線樹狀結構的計算方法時，其所包含的運算方式之計算次數公式如下所示：

$$\text{加法次數} = 2n \quad (5.22)$$

$$\text{乘法次數} = 2(n + 1) \quad (5.23)$$

使用 NURBS 參數式曲線去零化簡的計算方法時，其所包含的運算方式之計算次數公式如下所示：

$$\text{加法次數} = 2p \quad (5.24)$$

$$\text{乘法次數} = 2(p + 1) \quad (5.25)$$

藉由上述公式可計算出兩種方法對於整個 NURBS 參數式曲線的

所有運算次數，表 5-1 顯示兩種方法的所有運算次數。

表 5-1 不同方法之運算次數表(次)

樹 狀 架 構 計 算 方 法	$n \backslash p$	2	3	4	5	6
	5	156	228	309	399	498
	6	181	262	352	451	559
	7	206	296	395	503	620
	8	231	330	438	555	681
	9	256	364	481	607	742
去 零 化 簡 計 算 方 法	$n \backslash p$	2	3	4	5	6
	5	45	79	122	174	235
	6	45	79	122	174	235
	7	45	79	122	174	235
	8	45	79	122	174	235
	9	45	79	122	174	235

由表 5-1 中可看出本論文使用的去零化簡計算方法，相較於樹狀結構計算方法在計算次數上有大幅的減少，並且在相同級數 p 的情況下，其計算次數也不會隨總和上標 n 的大小而改變，故於軟體實現時能提供較佳的效率。

5-5-2 記憶體空間比較

本論文所探討之記憶體空間比較，主要目的為計算在 NURBS 參數式曲線過程中，需要多大的記憶體空間來儲存資料。由於使用軟體撰寫時其資料記憶體屬於硬體架構(通常屬於外接記憶體範圍)，即早在運動控制卡設計架構時便固定了其容量，最多在軟體撰寫時做位址的映射與配置，但在使用硬體設計時，由於必須考慮邏輯單元的多寡，而記憶體空間也是由邏輯單元所構成，所以在相同的目標下減少邏輯單元的使用是硬體設計的要點，故在相同的運算單元與控制單元的情況下，減少記憶體空間就是硬體設計的重點需求。對於 NURBS 參數式曲線描述的硬體設計而言，總共有多少資料需要儲存，則可分成下列兩部分比較：

- ◆ 儲存輸入參數時所需之記憶體空間：

包含需儲存的資料共有參數 u 、控制點矩陣、權重值矩陣與節點向量矩陣。

- ◆ 計算 NURBS 參數式曲線過的程中所需之記憶體空間：

包含需儲存的資料共有基底函數、單一有理式 B 型曲線與 NURBS 參數式曲線位置。

1.) 儲存輸入參數時所需之記憶體空間

使用樹狀結構的計算方法與去零化簡的計算方法，在儲存輸入參數時所需之記憶體空間相同，僅需考慮資料形式的不同，在本論文中資料使用 48 Bits 的形式，並做 Q32 的定點假浮點運算，所以儲存輸入參數的記憶體空間如下所示：

$$\text{參數 } u \text{ 的記憶體空間} = 48\text{Bits} \quad (5.26)$$

$$\text{控制點矩陣的記憶體空間} = 2(n+1) \times 48\text{Bits} \quad (5.27)$$

$$\text{權重值矩陣的記憶體空間} = (n+1) \times 48\text{Bits} \quad (5.28)$$

$$\text{節點向量矩陣的記憶體空間} = (p+n+2) \times 48\text{Bits} \quad (5.29)$$

2.) 計算 NURBS 參數式曲線過的程中所需之記憶體空間

使用樹狀結構的計算方法時，其所包含的記憶體空間如下所示：

基底函數的記憶體空間=

$$\frac{[(n+1) + (n+p+1)] \times (p+1)}{2} \times 48\text{Bits} \quad (5.30)$$

單一有理式 B 型曲線的記憶體空間=

$$(n+1) \times 48\text{Bits} \quad (5.31)$$

NURBS 參數式曲線位置的記憶體空間=

$$2 \times 48\text{Bits} \quad (5.32)$$

使用去零化簡的計算方法時，其所包含的記憶體空間如下所示：

基底函數的記憶體空間=

$$\frac{[1 + (p+1)] \times (p+1)}{2} \times 48\text{Bits} \quad (5.30)$$

單一有理式 B 型曲線的記憶體空間=

$$(p+1) \times 48\text{Bits} \quad (5.31)$$

NURBS 參數式曲線位置的記憶體空間=

$$2 \times 48\text{Bits} \quad (5.32)$$

藉由上述公式可以推算出兩種方法於計算 NURBS 參數式曲線過的程中所需之記憶體空間，表 5-2 顯示兩種方法的所需之記憶體空間。

表 5-2 不同方法所需之記憶體空間(Bits)

樹 狀 架 構 計 算 方 法	$n \backslash p$	2	3	4	5	6
	5	2736	3216	3744	4320	4944
	6	3120	3648	4224	4848	5520
	7	3504	4080	4704	5376	6096
	8	3888	4512	5184	5904	6672
	9	4272	4944	5664	6432	7248
去 零 化 簡 計 算 方 法	$n \backslash p$	2	3	4	5	6
	5	1872	2160	2496	2880	3312
	6	2064	2352	2688	3072	3504
	7	2256	2544	2880	3264	3696
	8	2448	2736	3072	3456	3888
	9	2640	2928	3264	3648	4080

由表 5-2 中可看出本論文所使用的去零化簡計算方法，相較於樹狀結構計算方法在所需之記憶體空間上有大幅的減少。由於輸入參數時所需之記憶體空間是由 CAD/CAM 指令所提供，故無法掌控，但在計算過程中的所需之記憶體空間則可看出有大幅節省的情形。

5-6 架構設計

本節介紹 NURBS 曲線產生 IP 之硬體架構設計，其架構如圖 5-6 所示，其中共分為六個主要區塊，其功能如下。Input Block：栓鎖輸入參數 u 並存有級數、控制點、節點向量與權重值。FSM：為有限狀態機負責控制協調各區塊間的運作。Debounce Block：由於 FSM 在狀態改變時輸出會有彈跳現象產生導致輸出錯誤故使用此區塊改善。Arith Block：資料運算之區塊。RAM：儲存運算暫存資料。Address Block：記憶體位置計數與判斷。

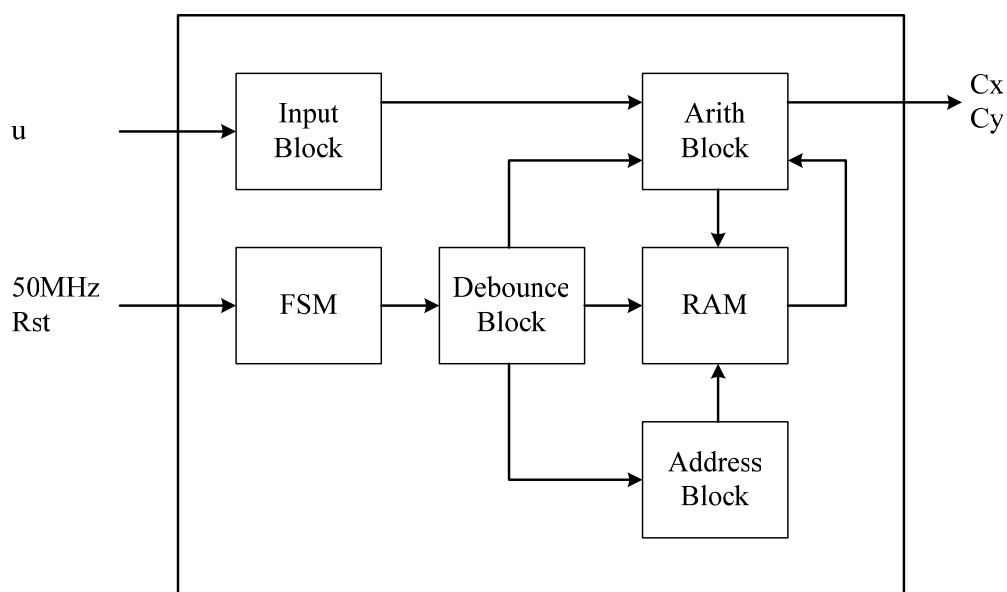


圖 5-6 NURBS IP 硬體架構圖

5-7 小結

本章節主要目的為發展 NURBS 參數式曲線產生 IP，由於以硬體方

式實現 NURBS 參數式曲線時將需用到相當多的邏輯運算單元，因此，本章節中則進一步探討本論文所規劃之運算化簡方法於硬體實現之成效。其結果顯示，該方法以硬體架構方式實現於 FPGA 時，確能有效降低所需之邏輯運算元並減少所需記憶體空間大小量。且由下一章節的實驗中，亦可以發現本論文所發展之方法的確可用於不同之 NURBS 曲線產生。

第六章 實驗設備介紹與實驗結果

本章節中，將對本論文所開發之 IP 完成軟體模擬與硬體實驗，以驗證其功能達到所需之目標。

6-1 軟硬體設備與硬體實驗架構

由於本論文所開發之 IP 其目的為達成各自所規劃之功能，在驗證架構上可分為軟體模擬與硬體實驗兩部分，本論文所建構之驗證設備及應用軟體說明如下：

軟體設備：

1. Quartus II v5.0：

為 Altera 公司所開發之發展軟體，支援其所有 FPGA 與 CPLD 的設計軟體，具有電路輸入 (design entry)、電路編譯 (compilation)、時序分析 (timing analysis)、電路模擬 (simulation)、晶片規劃 (programming)... 等功能，本論文中 IP 的開發與軟體模擬皆使用此工具。

2. MATLAB v6.5：

為 MathWorks 公司所開發之計算核心軟體，並具有許多不同功能之 Toolbox，本論文使用其作為資料處理與數據整理繪圖的工具。

3. Code Composer：

為 Texas Instruments 公司所開發之 DSP 開發軟體，其具有程式編輯、編譯、除錯...等功能，本論文使用該發展軟體以配合 DSP 作為資料接取的工具。

硬體設備：

1. Nios Development Board：

本論文所採用的開發板之主要核心為 Stratix EP1S40F780C5 之 FPGA 晶片，如圖 6-1 所示，該板共具有 41,250 個邏輯單元，其支援一般 FPGA 與嵌入式軟核處理器 IP Nios 的開發，且具有高度的硬軟體整合性。本論文於發展上為一般 FPGA 的開發方式，並將所開發之 IP 下載至 FPGA 上完成硬體實驗。



圖 6-1 Altera Stratix EP1S40F780C5 FPGA 發展板

2. STC-VC33 DSP 控制卡：

開發板之主要核心為 TMS320VC33 DSP 晶片，如圖 6-2，本

論文使用其計時中斷與外部觸發中斷等功能，用以擷取 FPGA 所產生之脈波訊號。



圖 6-2 史賓納 STC-VC33 DSP 控制卡

3. MITSUBISHI AC 伺服馬達及驅動器：

本論文於加減速規劃及脈波輸出驗證上使用三菱公司所出產型號為 HC-KFS43AC 之伺服馬達，如圖 6-3，其額定轉速為 3000 rpm，額定輸出功率為 400 Watt，而 AC 伺服驅動器型號為 MR-JS2-40A，本論文於馬達無負載狀況下作硬體實驗。

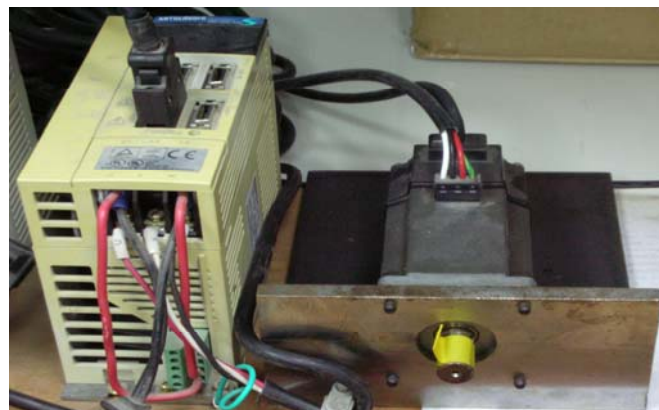


圖 6-3 MITSUBISHI AC 伺服馬達及驅動器

6-2 實驗結果

本論文所開發之運動命令規劃 IP，實驗將分為兩部份來探討，第一部份為整合加減速規劃 IP 與脈波命令輸出 IP 的實現部份，第二部份為以 FPGA 晶片完成 NURBS 曲線產生 IP。

1.) 整合加減速規劃 IP 與脈波命令輸出 IP

在本論文中，為了驗證加減速規劃及輸出脈波的正确與否，將以整合加減速規劃 IP 與脈波命令輸出 IP 的形式，驗證所發展的兩種 IP 架構並於 FPGA 上實現點對點運動控制。

圖 6-4 至圖 6-7 為以 Quartus II 模擬各種不同條件下之加減速曲線及補償輸出型態圖，其中圖 6-4 與圖 6-5 為實現梯型加減速曲線分別補償於速度曲線末端與速度曲線的加速及減速曲間，圖 6-6 與圖 6-7 為實現 S 型加減速曲線分別補償於速度曲線末端與速度曲線的加速及減速曲間。於圖上所標示圓圈為表示脈波補償區間所在。例如：圖 6-4 為要求總長為 450 個輸出脈波，最大速度為 40 個輸出脈波(於一個 Δt 週期內)，設定梯型加減速 N_1 為 7 之梯型加減速曲線脈波輸出，其餘數補償機制為於梯型加速度曲線之末端插入 6 個補償脈波輸出。圖 6-5 為要求總長為 650 個輸出脈波，最大速度為 30 個輸出脈波(一個 Δt 週期內)，梯型加減速 N_1 設定為 15 之梯型加減速曲線脈波輸出，其餘數補償機制為於梯型加速度曲線之加速度區及減速度區各插入 10 個補償脈波輸出。

為了進一步了解本論文所規劃之加減速及輸出脈波是否符合所探討的加減速形態，本論文將上述個別之脈波儲存輸出狀態值後，以 Matlab 繪出個別加減速曲線圖，如圖 6-8 所示。其中縱軸為最大輸出脈波數(一個 Δt 週期內)，橫軸為 Δt 個數。圖 6-8(a)(b) 為梯型加減速規劃，圖 6-8(c)(d) 為 S 型加減速規劃。

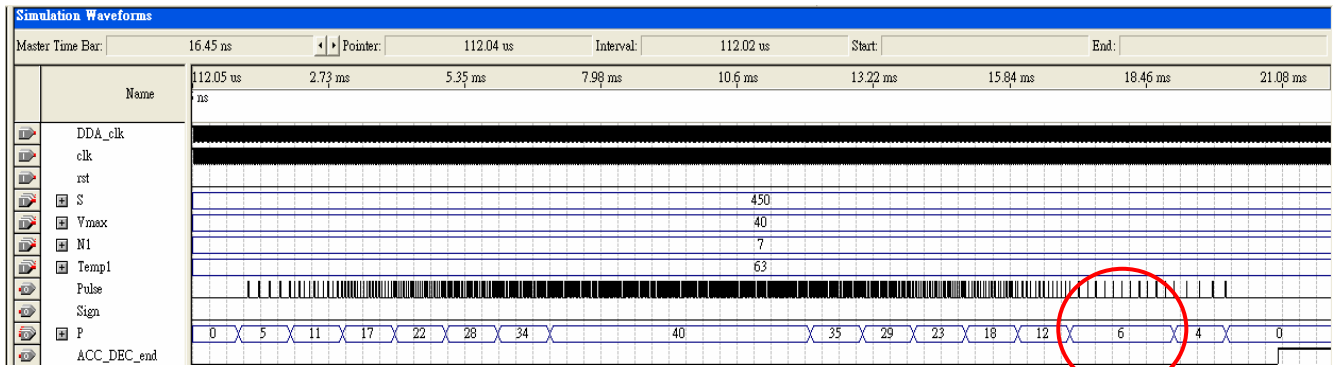


圖 6-4 梯型加減速曲線補償末端脈波輸出圖(S:450,V:40,N₁:7)

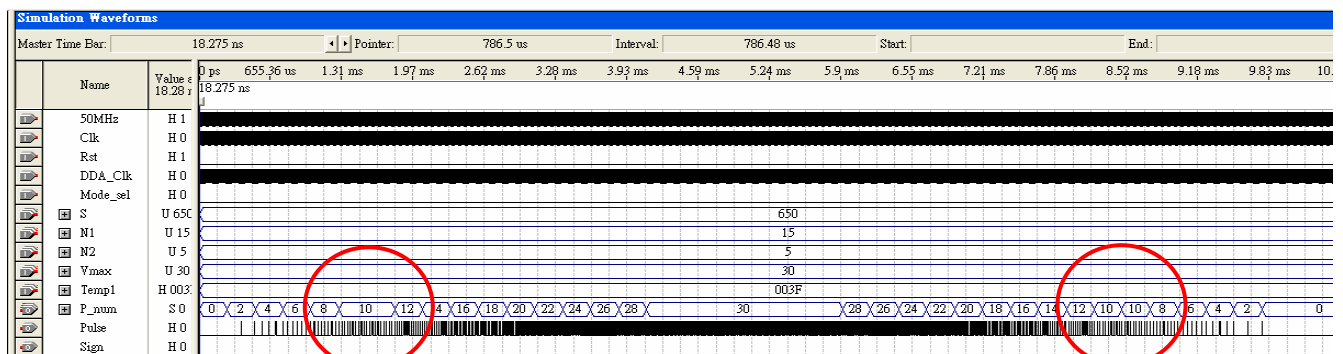


圖 6-5 梯型加減速曲線補償加速及減速區脈波輸出圖(S:650,V:30,N₁:15)

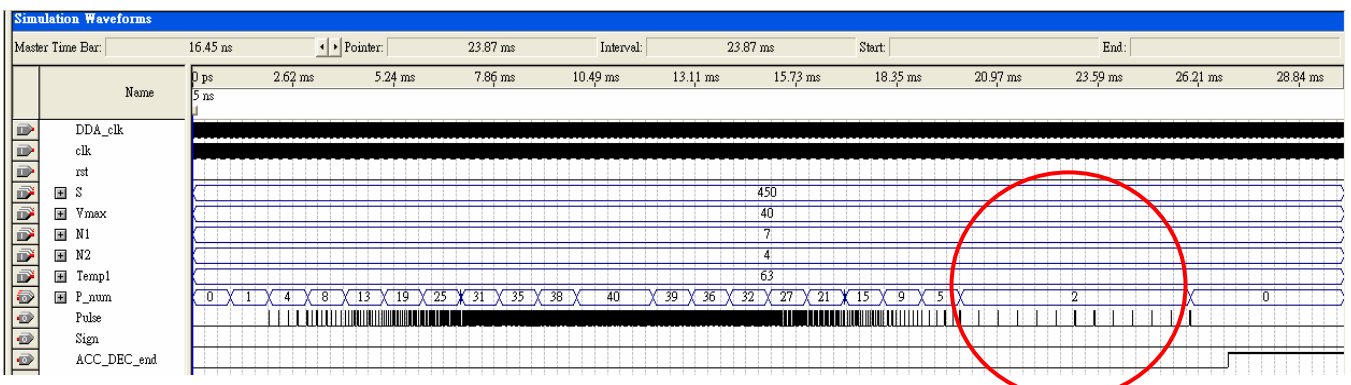


圖 6-6 S Curve 加減速曲線補償末端脈波輸出圖(S:450,V:40,N₁:7, N₂:4)

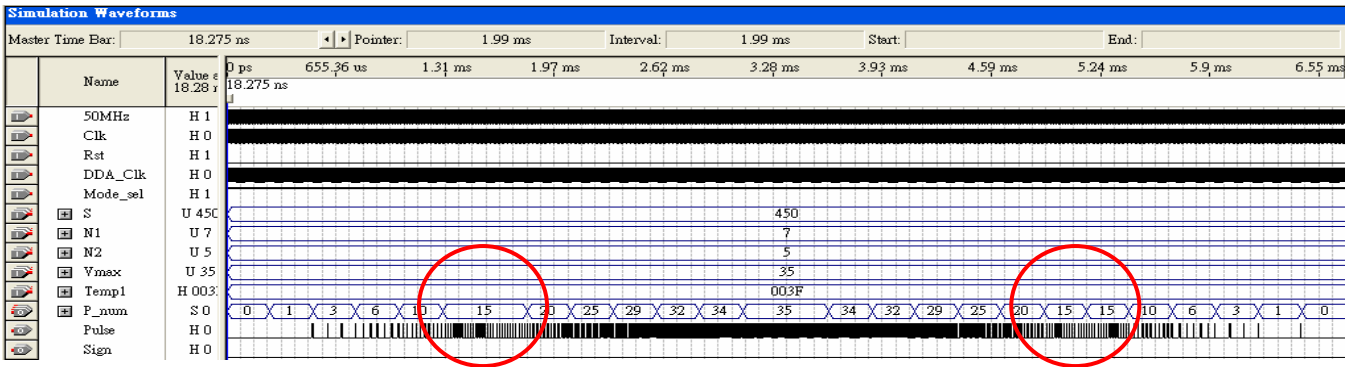


圖 6-7 S Curve 加減速曲線補償加速及減速區脈波輸出圖(S:450, V:40, N₁:7, N₂:5)

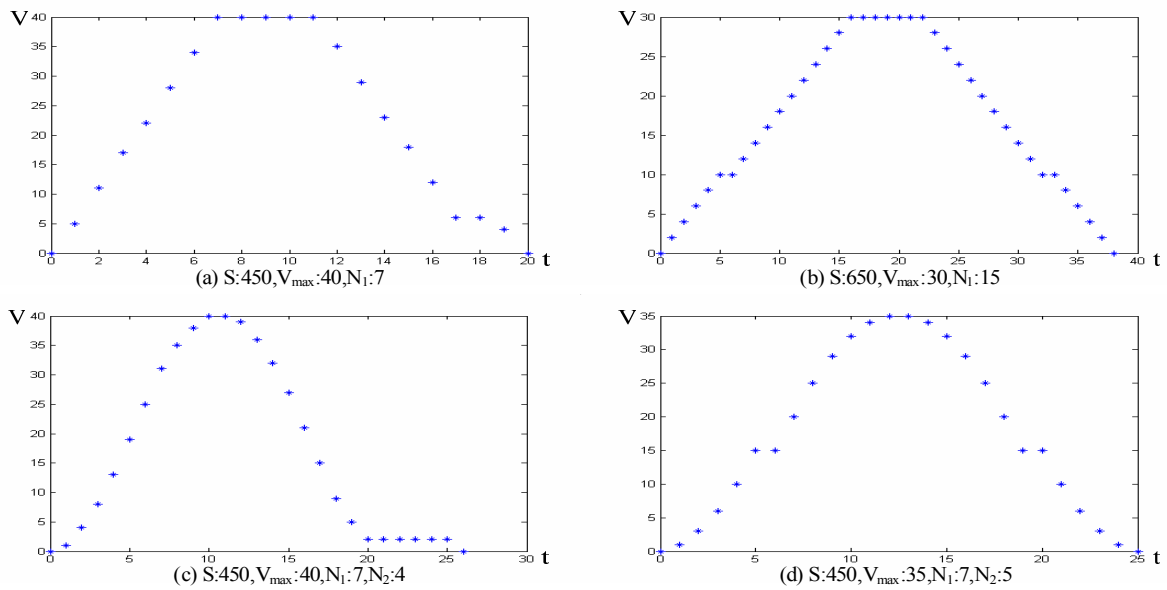


圖 6-8 不同條件下之梯型及 S Curve 加減速曲線圖

於探討過所規劃之加減速曲線及補償機制後，本論文將透過 FPGA 晶片將實際脈波輸出以實現點對點運動之硬體實驗，其整體實驗發展架構如圖 6-9 所示。本實驗之主要目的為驗證本文所發展之運動命令產生 IP 於點對點運動命令脈波輸出時之精確度，所考量的依據為輸出是否具有誤差失步的情況產生。

本論文實驗架構為以使用單軸 AC 伺服馬達來驗證，方式為以透過 FPGA 產生具加減速曲線脈波輸出至馬達驅動器驅動伺服馬達運轉，再由

伺服馬達上所配置之編碼器回傳至 FPGA 內之解碼器 IP，以計數馬達行走的位置量。於本實驗中，脈波輸出命令(位置)與解碼器解出之位置間為存在一倍數關係，其原因為實驗所設定的伺服驅動器之電子齒輪比參數、編碼器參數與 FPGA 內解碼器具有倍頻之關係，其換算如公式(6.1)所示。其中 P 為輸入脈波數，CMX 為電子齒輪分子，CDV 為電子齒輪分母，R 為解碼器 IP 除頻數，131,072 為馬達解析度，4,096 為編碼器解析度。本文中設定電子齒輪比為 8:1，R 為 8 倍頻。解碼器輸出脈波計算公式如下：

$$\text{解碼器輸出脈波} = \frac{P \times \frac{CMX}{CDV}}{131072} \times \frac{4096}{R} \quad (6.1)$$

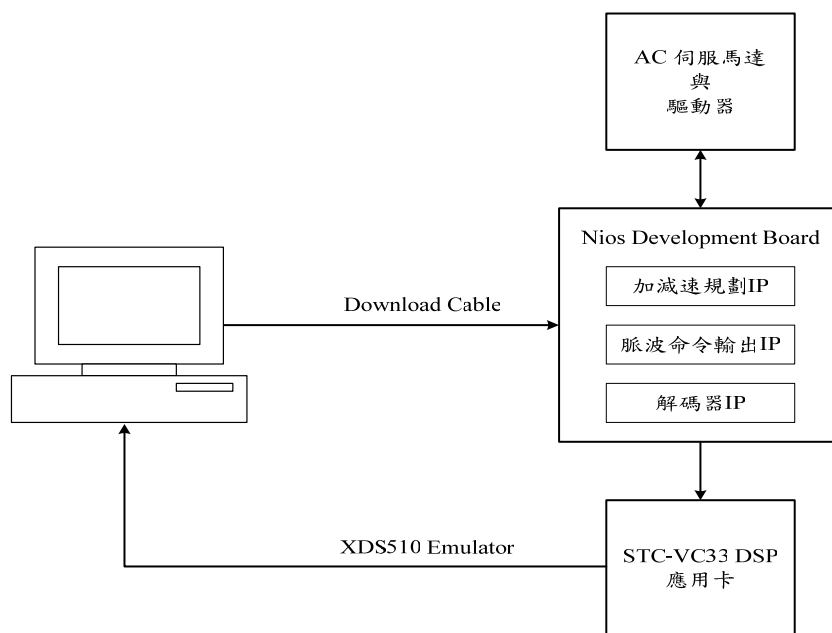


圖 6-9 點對點運動實驗驗證架構圖

圖 6-10 至 6-17 為點對點脈波輸出運動於不同加減速曲線及補償機制的情況下，輸出命令位置與馬達轉動之位置關係圖，實驗所共同設定的

參數如下：

1. 輸入脈波步數 32,000 步
2. DDA 週期 $\Delta t = 5.12ms$
3. $N_1 = 30, N_2 = 15$

解碼器輸出脈波計算公式如(6.1) 所示，當輸入脈波為 32000 步時，輸出轉換之解出馬達位置應為 1000 步。實際上由圖 6-10 至圖 6-17 可知，本論文所發展之加減速脈波輸出機制的確能達到零失步的實驗結果。再者，由圖 6-11、圖 6-13、圖 6-15 及圖 6-17 等馬達位置檢出圖之局部放大圖中，亦可以觀察到本論文所發展之加減速規劃於驅動伺服馬達起始及終止時，明顯的有和緩的加減速效果產生。

Case 1：補償於末端之梯型加減速脈波輸出定位

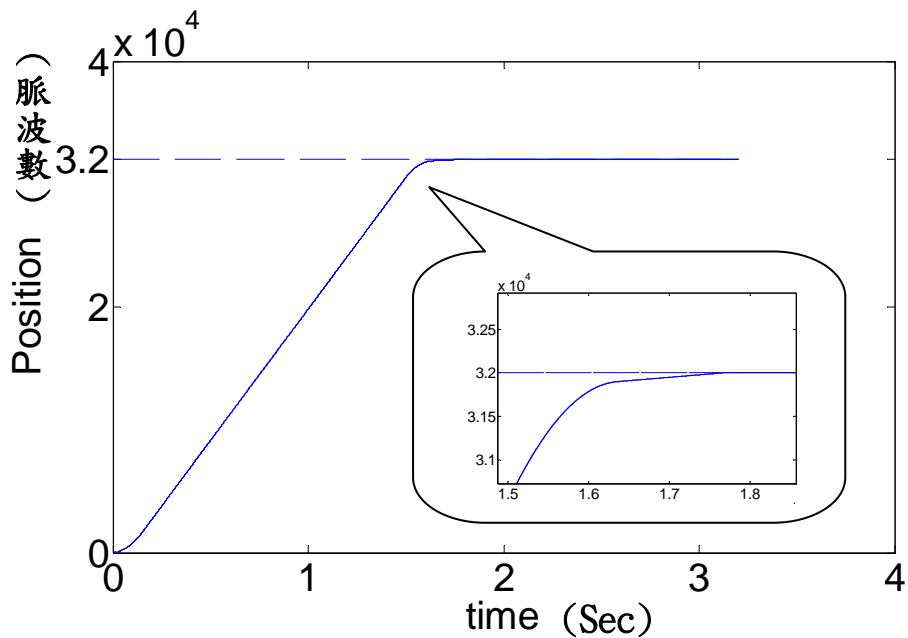


圖 6-10 補償於末端之梯型加減速命令輸出位置

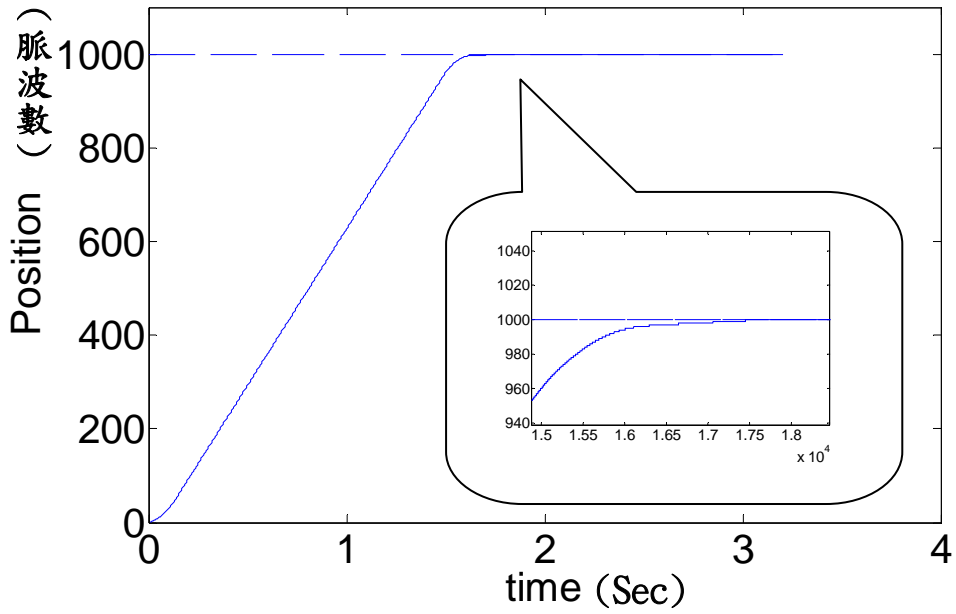


圖 6-11 補償於末端之梯型加減速馬達位置檢出圖

Case 2：補償於加速與減速區之梯型加減速脈波輸出定位

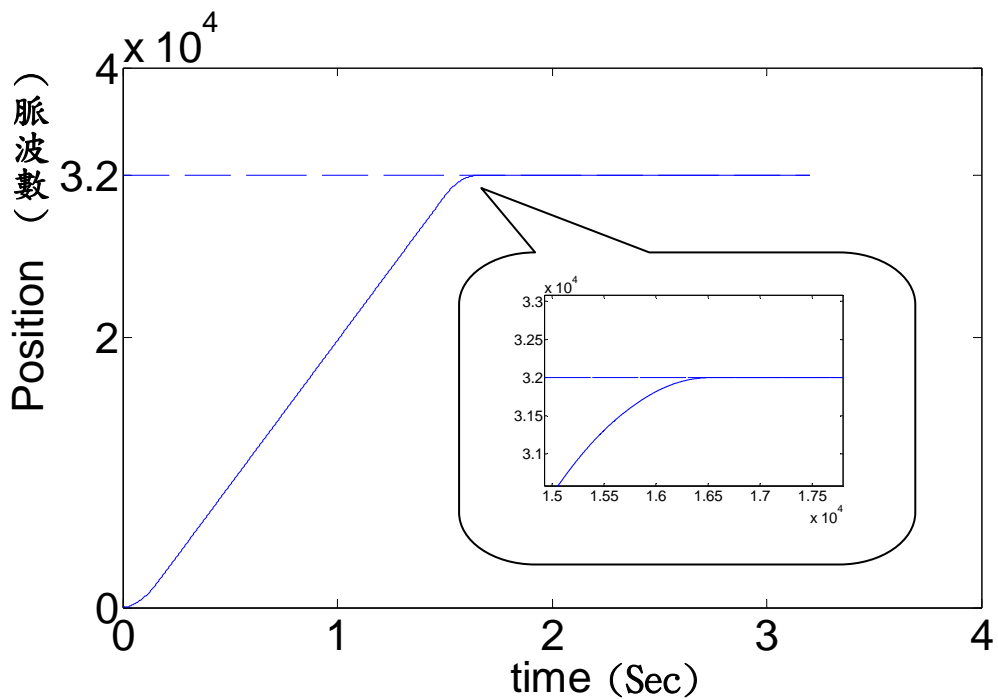


圖 6-12 補償於加速與減速區之梯型加減速命令輸出位置

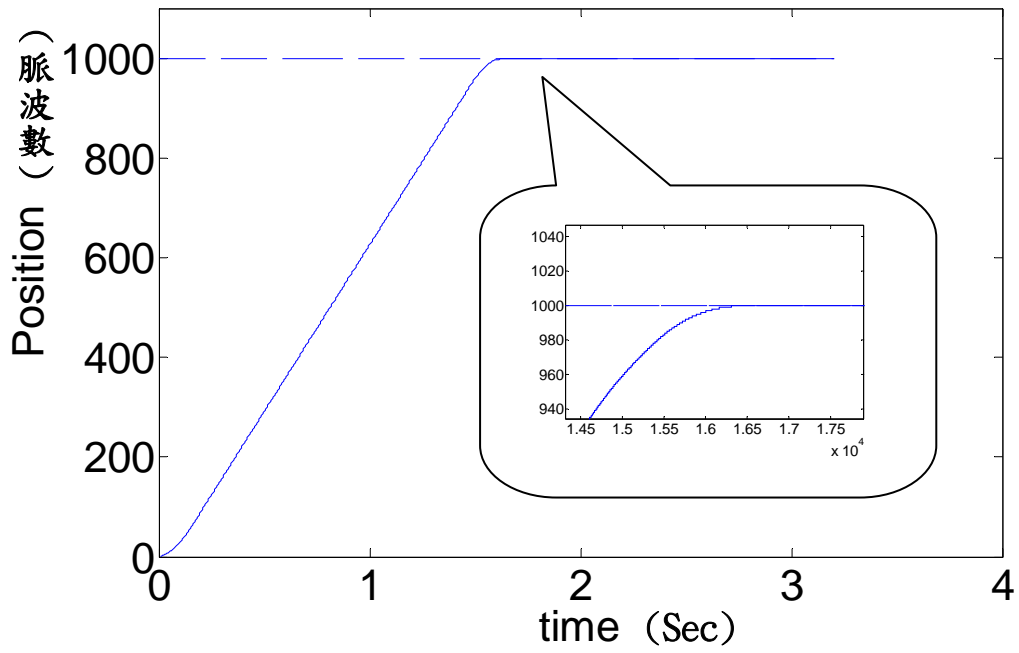


圖 6-13 補償於加速與減速區之梯型加減速馬達位置輸出圖

Case 3：補償於末端之 S Curve 加減速脈波輸出定位

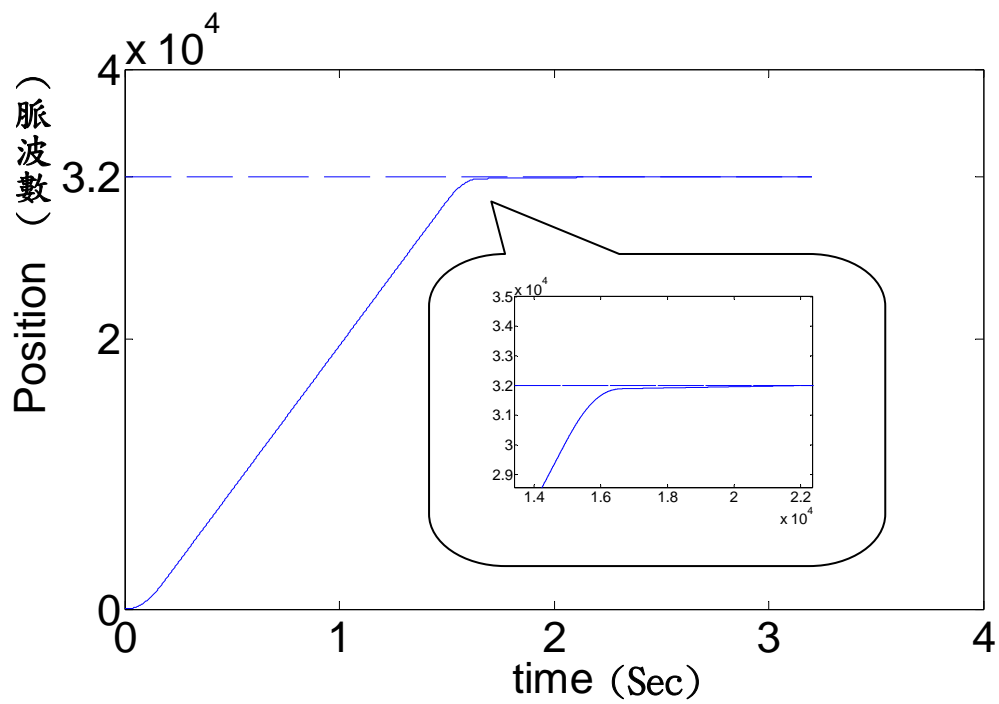


圖 6-14 補償於末端之 S Curve 加減速命令輸出位置

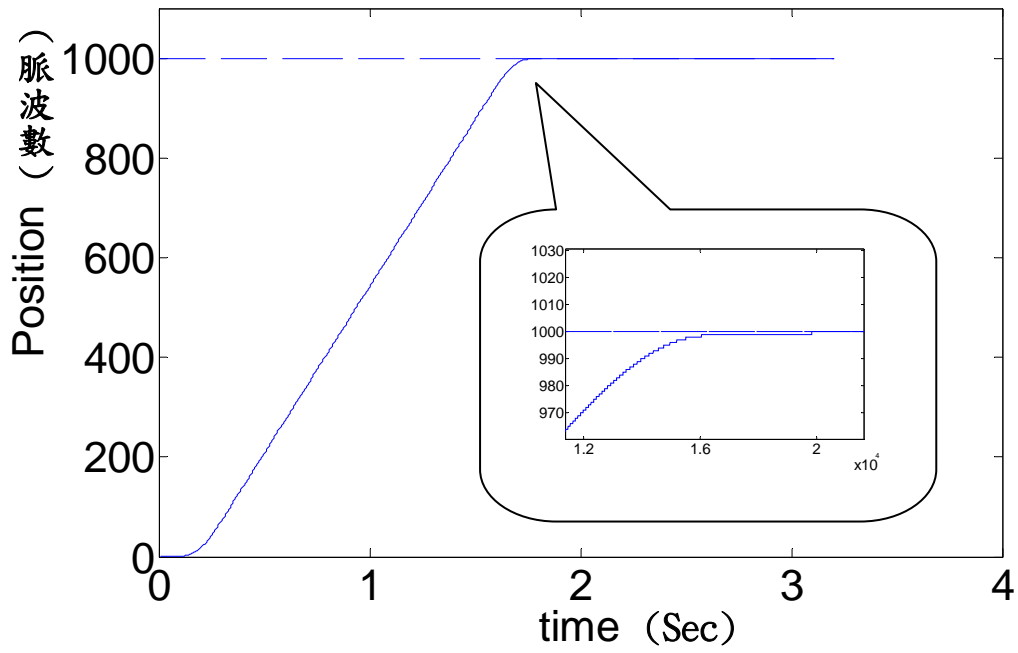


圖 6-15 補償於末端之 S Curve 加減速馬達位置輸出圖

Case 4：補償於加速與減速區之 S Curve 加減速脈波輸出定位

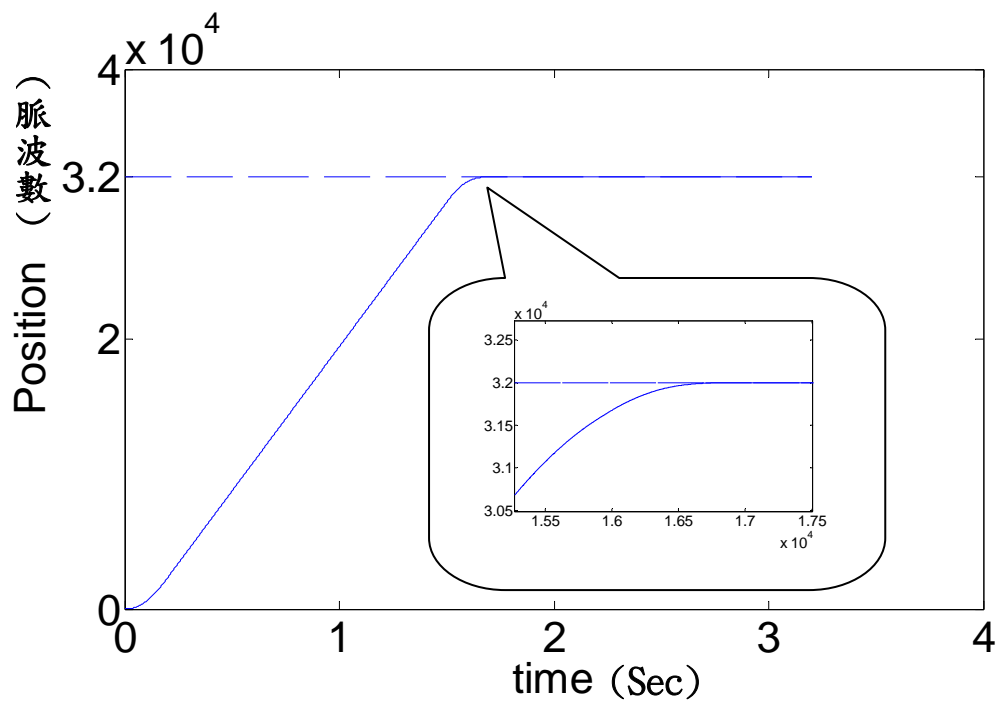


圖 6-16 補償於加速與減速區之 S Curve 加減速命令輸出位置

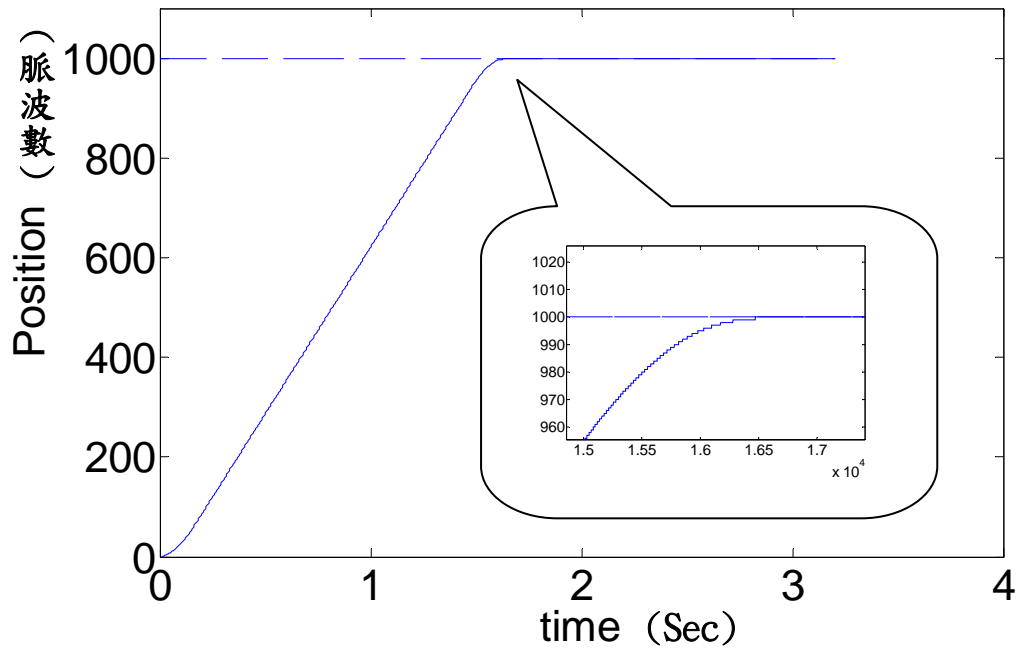


圖 6-17 補償於加速與減速區之 S Curve 加減速馬達位置檢出圖

2.) NURBS 曲線產生 IP

本小節主要為實現第五章中所提到的 NURBS 參數式曲線運算化簡於 FPGA 內，其方式為透過硬體描述語言的撰寫以計算出所產生的曲線位置點，經由記憶體存取後，再經由 Matlab 繪出所規劃之 NURBS 曲線。在本論文中，以四種學者常實現的 NURBS 曲線為實驗曲線，其設定值如 Case 1 至 Case 4 所規劃，其圖形如圖 6-18 至 6-19 所示。

Case 1 :

u : 0~1 每次增加 0.01，共計算 100 次

Degree : 2

Control point : $\begin{bmatrix} 0 & 0 & 50 & 50 & 50 & 0 & 0 \\ 0 & 25 & 25 & 0 & -25 & -25 & 0 \end{bmatrix}$

Knot vector : [0 0 0 0.25 0.5 0.5 0.75 1 1 1]

Weight : [1 0.5 0.5 1 0.5 0.5 1]

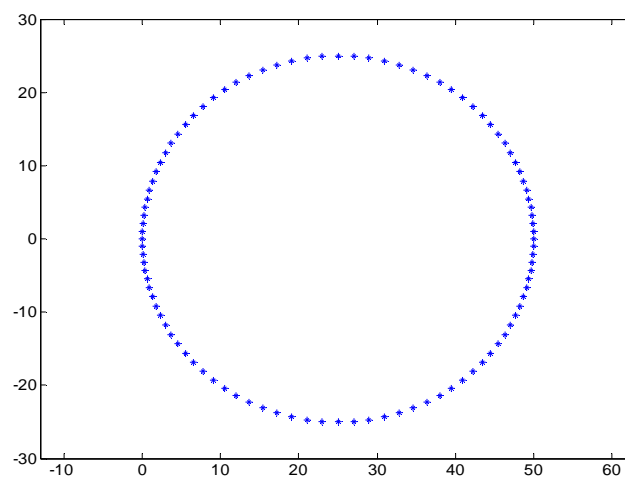


圖 6-18 NURBS 曲線產生實驗結果一

Case 2 :

u : 0~1 每次增加 0.01 , 共計算 100 次

Degree : 2

Control point : $\begin{bmatrix} 10 & 20 & 12 & 10 & 8 & 0 & 10 \\ 0 & 20 & 8 & 20 & 8 & 20 & 0 \end{bmatrix}$

Knot vector : $[0 \ 0 \ 0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1 \ 1 \ 1]$

Weight : $[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$

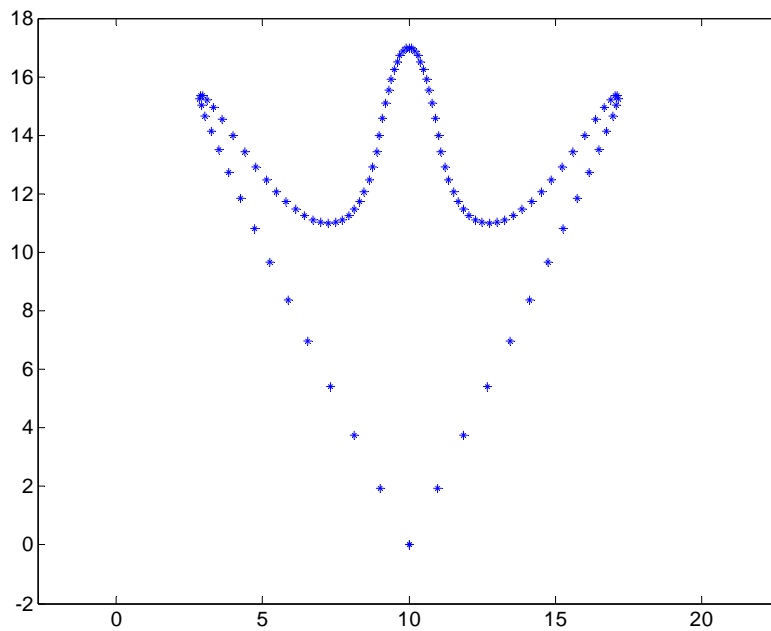


圖 6-19 NURBS 曲線產生實驗結果二

Case 3 :

u : 0~1 每次增加 0.01 , 共計算 100 次

Degree : 2

$$\text{Control point : } \begin{bmatrix} 0 & 10 & 10 & 0 & 0 & 10 \\ 0 & 0 & 10 & 10 & 20 & 20 \end{bmatrix}$$

$$\text{Knot vector : } [0 \ 0 \ 0 \ 0.25 \ 0.5 \ 0.75 \ 1 \ 1 \ 1]$$

$$\text{Weight : } [1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

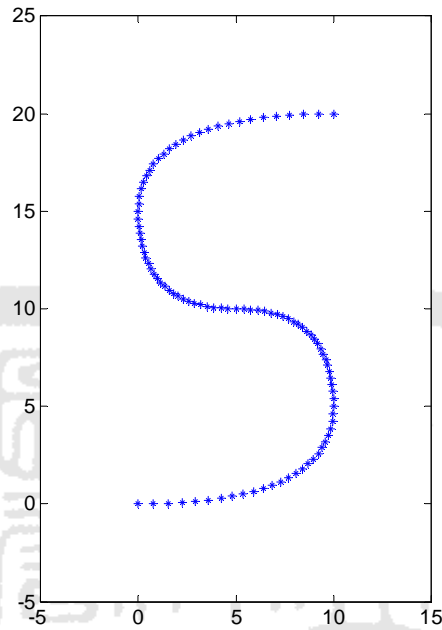


圖 6-20 NURBS 曲線產生實驗結果三

Case 4 :

u : 0~1 每次增加 0.01 , 共計算 100 次

Degree : 2

$$\text{Control point : } \begin{bmatrix} 0 & -50 & -50 & 0 & 50 & 50 & 0 \\ 0 & -50 & 50 & 0 & -50 & 50 & 0 \end{bmatrix}$$

$$\text{Knot vector : } [0 \ 0 \ 0 \ 0.25 \ 0.5 \ 0.5 \ 0.75 \ 1 \ 1 \ 1]$$

$$\text{Weight : } [1 \ 0.85 \ 0.85 \ 1 \ 0.85 \ 0.85 \ 1]$$

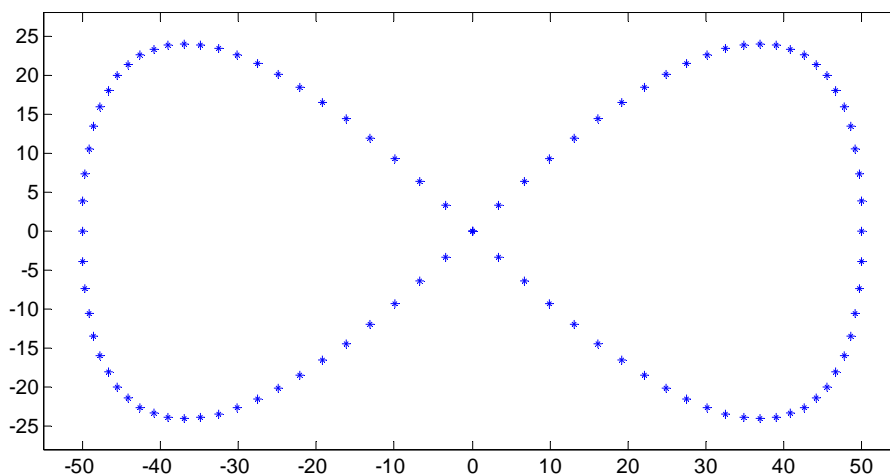


圖 6-21 NURBS 曲線產生實驗結果四

6-3 小結

由本節的各實驗驗證結果得知，驗證了本論文所開發之運動命令規劃 IP 皆可達成預期之目標。於第一個驗證的實驗中，本論文所規劃之整合加減速規劃 IP 與脈波輸出命令 IP 所建構之加減速脈波產生器確能提供零失步誤差脈波輸出，相較於常見的市售脈波產生晶片普遍存在著 0.1% 左右的誤差，本論文所發展之產生器的確能改善此缺點，達到零誤差脈波輸出的優點。於第二個驗證的實驗中，對於 NURBS 曲線產生 IP 部分，本論文探討以運算化簡的方式實現 NURBS 參數式曲線。於本論文之前的章節中，曾經經由記憶體及運算次數的探討了解此種方法的確可以提高整體系統的運算效能。除此之外，於本章節中經由硬體實驗結果證實，此種方法實際上亦適用於多種型態的 NURBS 曲線產生，相較於傳統的 NURBS 曲線產生方式更有利於硬體實現。

第七章 結論與建議

結論:

一般業界對於工具機的應用上，除裝設時會由調機工程師調整控制器之參數，使機台達到所需之響應與精度以維持運作外，之後的操作僅由使用者下達運動命令。也就是說若運動命令規劃良好，則加工工件之精度必能達到所需目標，故運動命令規劃可說是一門重要課題。

在本論文中，實現了運動命令規劃中的加減速規劃 IP、脈波命令輸出 IP、NURBS 參數式曲線產生 IP 等三種 IP。於加減速規劃 IP 方面，本論文藉由數位迴旋積分的方式提昇精度並減少邏輯單元的使用。而在脈波命令輸出 IP 方面，改良後的架構因週期可調，故能適應更多系統。本論文並透過整合兩種 IP 特性的方式實現具零失步誤差之加減速命令產生器。至於在 NURBS 參數式曲線產生 IP 方面，則藉由演算法的化簡提昇效能與減少記憶體空間。實驗中將本論文所發展的這三種 IP 實際下載至 FPGA 晶片中驗證，結果顯示均能達成所要求之目標。

未來研究方向與建議:

本論文所建構的三種 IP 雖然都可以達到預期規劃之目標，然而運動控制還有其他任務模組有待開發，例如：插值器模組、速度(電壓驅動)模式模組或內建控制器模組。期望未來能整合上述模組開發成為一具整合性之運動控制規劃 ASIC。另外，由於一般市售 FPGA 之內建記憶體有限，因此，若能發展 FPGA 與 PC 間之即時傳輸介面，對於開發上資料的擷取必然能有相當大的助益。

參 考 文 獻

- [1] *Stratix Device Handbook*, Altera Co., San Jose, TX, 2006.
- [2] B.S.Hu, and J.Li, "The Fuzzy PID Gain Conditioner : Algorithm, Architecture and FPGA Implementation," in *Proceedings of 1996 IEEE International Conference on Industrial Technology*, pp.621-624.
- [3] C.S.Chen, and A.C.Lee, "Design of acceleration/deceleration profiles in motion control based on digital FIR filters," *International Journal of Machine Tools & Manufacture*, vol.38, No.7, July 1998, pp.799-825.
- [4] D.Carrica, M.A.Funes, and S.A.González, "Novel Stepper Motor Controller Based on FPGA Hardware Implementation," *IEEE/ASME Transactions on Mechatronics*, vol.8, No.1, Mar. 2003, pp.120-124.
- [5] D.I.Kim, J.W.Jeon, and S.Kim, "Software acceleration/deceleration methods for industrial robots and CNC machining tools," *Mechatronics*, vol.4, No.1, Feb. 1994, pp.37-53.
- [6] D.M.Tsay, and B.J.Lin, "Improving the Geometry design of Cylindrical Cams using Nonparametric Rational B-Splines," *Computer-Aided Design*, vol.28, No.1, Jan. 1996, pp.5-15.
- [7] D.M.Tsay, and Jr.C.O.Huey, "Application of Rational B-Spline to the Synthesis of Cam-Follower Motion Programs," *ASME Journal of Mechanical Design*, vol.115, No.3, 1993, pp.621-626.
- [8] H.T.Yau, M.T.Lin, Y.T.Chan, and K.C.Yuan, "Design and Implementation of Real-time NURBS Interpolator using a FPGA-Based Motion Controller," in *Proceedings of 2005 IEEE International Conference on Mechatronics*, pp.56-61.
- [9] J.W.Jeon, "A Generalized Approach for the Acceleration and Deceleration of CNC Machine Tools," in *Proceedings of 1996 IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, Taipei, Taiwan, pp.1283-1288.
- [10] J.W.Jeon, and Y.Y.Ha, "A Generalized Approach for the Acceleration and Deceleration of Industrial Robots and CNC Machine Tools," *IEEE Transactions on Industrial Electronics*, vol.47, No.1, Feb. 2000, pp.133-139.
- [11] J.W.Jeon, and Y.K.Kim, "FPGA based acceleration and deceleration circuit for industrial robots and CNC machine tools," *International Journal of Machine Tools & Manufacture*, vol.12, No.4, May 2002, pp.635-642.
- [12] K.Yoon, and S.S.Rao, "Cam Motion Synthesis Using Cubic Splines," *ASME Journal of Mechanical Design*, vol.115, No.3, 1993, pp.441-446.
- [13] L.Piegl, "On NURBS : A Survey," *IEEE Transactions on Computer Graphics & Application*, vol.11, No.1, Jan. 1991, pp.55-71.
- [14] L.Piegl, and Tiller.W, *The NURBS Book*. New York: Springer-Verlag, 1997, Ch.4.
- [15] L.Samet, N.Masmoudi, M.W.Kharrat, and L.Kamoum, "A Digital PID Controller for Real Time and Multi-Loop Control : a Comparative Study," in *Proceedings of 1998 IEEE International Conference on Electronics, Circuits and Systems*, Lisboa, Portugal, pp.291-296.
- [16] M.Gopi, and S.Manohar, "A Unified Architecture for the Computation of B-Spline Curves and Surfaces," *IEEE Transactions on Parallel and Distributed Syatem*, vol.8, No.12, Dec. 1997, pp.1275-1287.
- [17] M.G.Egan, J.M.Murphy, E.J.Heffeman, S.V.Lidbolm, and M.L.McGrath, "An

- ASIC-based PWM waveform generator for AC motor control applications,” in *Proceedings of 1988 IEEE International Symposium on Circuits and Systems*, Espoo, Finland, pp.1369-1372.
- [18] *General-Purpose Interface MR-J2-□A Specifications and Installation Guide*, MITSUBISHI Co., Tokyo, 2000.
- [19] M.Y.Cheng, M.C.Tsai, and J.C.Kuo, “Real-time NURBS command generators for CNC servo controllers,” *International Journal of Machine Tools & Manufacture*, vol.42, No.7, May 2002, pp.801-813.
- [20] *MCX314As User's manual*, Ver. 1.5, 2006.Nova electronics Co., Available: <http://www.novaelec.co.jp>
- [21] *Motion processor for microstepping motion control MC2400 series Data sheet*, 1999, Performance Motion Devices Inc., Available: <http://www.pmdcorp.com>
- [22] T.C.Green, M.M.Moud, J.K.Goodfellow, and B.W.Williams, “Field- Programmable Gate-Arrays and Semi-Custom Designs for Sinusoidal and Current-Regulated PWM,” in *Proceedings of 1992 IEE Colloquium on ASIC Technology for Power Electronics Equipment*, pp.4/1-4/4.
- [23] Y.Koren, *Computer Control of Manufacturing System*. New York: McGraw-Hill, 1983, Ch.5.
- [24] 王有正, “FPGA 為基礎之感應馬達強健控制器設計,” 碩士論文, 長庚大學電機工程研究所, 2004。
- [25] 王淳, “以 FPGA 為基礎之精密位置控制 IC,” 碩士論文, 國立中央大學機械工程研究所, 2002。
- [26] 江修, “應用週期及密度可變之 DDA 架構改善脈波輸出性能,” 工研院機械所自動控制技術專輯, 2001, pp.133-146。
- [27] 李文猶, “估測式 T 型/S 型曲線脈波產生器之實現,” 2005 年自動控制研討會論文集, pp.d-four-34 - d-four-39。
- [28] 呂思遠, “數位影碟機主軸馬達伺服控制系統之研究,” 碩士論文, 大葉大學自動化工程研究所, 2003。
- [29] 何建曉, “IP 產業概況,” 交銀通訊, 2002, pp.33-39。
- [30] 林灶生、劉紹漢, *Verilog FPGA 晶片設計*, 全華科技圖書, 2004。
- [31] 林德, “應用 FPGA 與人機介面設計之模糊 PID 控制器,” 碩士論文, 大葉大學電機工程研究所, 2004。
- [32] 官宏霖, “感應馬達速度控制積體電路之研製,” 碩士論文, 國立台灣科技大學電機工程研究所, 2003。
- [33] 邱乾致, “FPGA 為基礎之感應馬達 V/f 控制及短路防止時間補償,” 碩士論文, 長庚大學電機工程研究所, 2003。
- [34] 施慶隆、李文猶, *機電整合與運動控制: 原理與單軸平台實例*, 高立圖書, 1997。
- [35] 袁國欽, “FPGA-Based NURBS 插補器之設計與實現,” 碩士論文, 國立中正大學機電光工程研究所, 2003。
- [36] 徐禮弘, “使用 FPGA 晶片發展一個雕塑曲面設計,” 碩士論文, 國立中央大學電機工程研究所, 2001。
- [37] 唐佩忠, *VHDL 與數位邏輯設計*, 高立圖書, 1999。
- [38] 陳秋帆, “DDA 高次插值器整合最佳化加減速曲線,” 碩士論文, 國立台北科技大學自動化科技研究所, 2002。
- [39] 陳建鈞, “三度空間曲線之 Bicubic DDA 研究及 FPGA 實作,” 碩士論文, 國立交

- 通大學機械工程研究所，2004。
- [40] 許皓鈞，“以 FPGA 為基礎的交流馬達空間向量脈寬調變 IC 之研製，”碩士論文，國立交通大學控制工程研究所，1995。
- [41] 許祿岳，“運動控制器之硬體電路設計，”碩士論文，國立台灣科技大學電機工程研究所，2002。
- [42] 張婷婷，“直流馬達之模糊控制晶片設計與實現，”碩士論文，中華技術學院電子工程研究所，2004。
- [43] 張煌欽，“以 FPGA 為基礎之模糊滑動模式控制器設計應用於感應馬達直接轉矩控制，”碩士論文，大同大學電機工程研究所，1999。
- [44] 曾紹顯，“以場效可規劃邏輯閘陣列為基礎之脈波寬度調變控制器的研製，”碩士論文，國立台北科技大學自動化科技研究所，2001。
- [45] 黃嘉正，“研發具多種速度曲線之雙軸運動控制卡，”碩士論文，國立台北科技大學自動化科技研究所，2001。
- [46] 詹耀德，“FPGA-Based PID 伺服控制器之設計與實現，”碩士論文，國立中正大學機電光工程研究所，2003。
- [47] 鄭光欽、黃孝祖，CPLD 與 VHDL 設計實務，全威圖書，2001。
- [48] 鄭中緯、鄭銘揚、蔡明琪，“即時參數式插值器之設計與實現，”中國機械工程學會第 18 屆全國學術研討會論文集，2001，pp.267-272。
- [49] 劉承愚、賴文智，“SOC 推波助瀾全球 IP 產業掀起合作風，”電子工業週刊，Oct. 2000。