

國立交通大學

電機與控制工程學系

碩士論文

發展 QEP-CORDIC 之伺服馬達

19-bit 弦波解碼器及電路實現

Design and Realization of the 19-bit Sinewave Decoder

via the QEP-CORDIC Algorithm for Servo Motors

研究生：李建龍

指導教授：徐保羅 博士

中華民國九十七年七月

發展 QEP-CORDIC 之伺服馬達
19-bit 弦波解碼器及電路實現

Design and Realization of the 19-bit Sinewave Decoder
via the QEP-CORDIC Algorithm for Servo Motors

研究生：李建龍

Student: Chien-Lung Lee

指導教授：徐保羅 博士

Advisor: Prof. Pau-Lo Hsu



A Thesis

Submitted to Department of Electrical and Control Engineering
College of Electrical and Computer Engineering
National Chiao-Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master
in
Electrical and Control Engineering
July 2007
Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

發展 QEP-CORDIC 之伺服馬達

19-bit 弦波解碼器及電路實現

研究生：李建龍

指導教授：徐保羅 博士

國立交通大學電機與控制工程學系

摘要

伺服馬達控制系統中包含伺服馬達、感測器及控制器三部份，想要完成一個高效能的伺服馬達控制系統三者缺一不可，本論文著重於弦波編碼器的解碼演算法發展實現，及低速控制適用的估測法及控制法。在弦波編碼器解碼部分，文中首先使用線性化法進行解碼實驗，之後再提出一個 QEP-CORDIC 演算法，與一般 CORDIC 演算法不同處在於 QEP-CORDIC 演算法特別適用於弦波編碼器的解碼，除了有自動線上及時的 A/D 轉換器之校正外，另外加入了輸入向量位置的對應、以及狀態轉換的 QEP 計數方式等等，並以模擬及實驗找出最合適的遞迴次數，最後完成 DSP-Based 線性化法 19bit 解碼及 FPGA-Based QEP-CORDIC 19bit 弦波編碼器之解碼電路實現，使用 17521 個邏輯閘、150ns 解碼計算效率。

本研究進一步將所研製的 19-bit 弦波編碼器應用在伺服馬達低速控制，在速度方面首先比較三種不同的估測法，(1)差分法加低通濾波器、(2)最小平方近似法、及(3)卡曼濾波器，經過模擬及實驗結果證實卡曼濾波器在低速估測上擁有最佳的效果，能有效降低回授速度變異量；在低速控制法則方面首先使用傳統 PI 控制器，並且調整一組適合低速控制的參數與另一組適合中高速的參數做比較，之後再以 PI 控制器加上狀態空間控制法實現，經過實驗證實傳統 PI 控制器在低速控制上有最低的速度變異量，效果最佳。最後將所有適用於低速控制的 PI 控制器及卡曼低速估測法成功的應用於 CNC 上。

Design and Realization of the 19-bit Sinewave Decoder via the QEP-CORDIC Algorithm for Servo Motors

Student : Chien-Lung Lee

Advisor : Dr. Pau-Lo Hsu

ABSTRACT

This study focuses on the development and realization of the decoder algorithm of sinewave encoders for precision motion control, especially for low-speed operations. The linearization method for sinewave decoding has been implemented on a DSP microcontroller with high-level-programming realization based on the C language. Furthermore, The proposed QEP-CORDIC algorithm is realized on the FPGA-based 19-bit decoding. The present QEP-CORIDC algorithm which employs the CORDIC algorithm with the proposed techniques as : (1) auto-calibration of ADC converter, (2) vector mapping of QEP, and (3) the state counter of the QEP. Also, the best recursive procedure is also determined by both simulation and experimental results, with 17521 logic gates and 150ns decoding time.

The developed 19-bit sinewave encoder is applied to the serve motor with a low-speed control. There are three methods are compared to estimate the velocity from the encoder output: (1) the differential approach with a lowpass filter, (2) the east-squares fit, and (3) the Kalman filter. The Kalman filter has been proven to be the best performance with the minimum velocity variance. Both the PI controller and the Kalman filter has been applied to the CNC to render satisfactory control performance in extremely low-speed motion control.

誌謝

首先要感謝指導教授徐保羅 博士在我兩年研究中細心指導，在研究遇到問題與困難時，能與我一起尋找突破的方法，使我能順利完成碩士學位。同時也要感謝口試委員董蘭榮老師、陳鴻祺老師與葉賜旭老師對於本論文的指正和給予寶貴的建議，使得本論文更加完善。另外也要感謝一同參與計畫的張隆國老師在平時開會時所給予的指導，讓我能從不同的角度去思考問題。

感謝實驗室的學長、同學與學弟們：鎮洲學長、琮政學長、煒生學長、建良學長、孝麟學長、瑞原學長、興漢學長、元銘學長、宗勝學長、兩坤學長、林億學長、煒智、孟哲、正忠、子強、任斌以及與我同屆的哲維、旭志、醇偉及也強，在我的學業與研究中，互相勉勵，遇到困境時給予我鼓勵打氣，並且時常提供意見給我。生活上偶爾在實驗室中間聊談心，讓實驗室不僅是研究的場所，更帶有歡笑、活力與和諧的氣氛，得以在如此美好的環境下順利完成學業。

最後要感謝我的家人及朋友們，特別是我的外公、外婆及母親，能夠在艱困的環境下支持我完成學業，讓我無後顧之憂的專心於研究，因為過去有你們的支持，今天才得以完成我的碩士學位，在此將本論文獻給各位。

目錄

中文摘要.....	i
英文摘要.....	ii
誌謝.....	iii
目錄.....	I
表目錄.....	III
圖目錄.....	IV
第一章 序論.....	1
1.1 研究動機.....	1
1.2 問題界定.....	1
1.3 研究方法及步驟.....	3
1.4 論文架構.....	4
第二章 弦波編碼器解碼演算法.....	5
2.1 弦波編碼器.....	5
2.1.1 弦波編碼器之介紹.....	5
2.1.2 弦波編碼器輸出分析.....	5
2.1.3 弦波編碼器解碼概要.....	6
2.2 粗略位置解碼.....	8
2.3 精細位置解碼.....	9
2.3.1 線性化法解碼 [3].....	9
2.3.2 CORDIC 演算法解碼 [6].....	15
2.3.3 QEP-CORDIC 演算法.....	20
第三章 弦波編碼器解碼實現.....	22
3.1 DSP-Based 線性化法解碼實驗.....	22

3.1.1 硬體介紹	22
3.1.2 即時線上 ADC 校準	27
3.1.3 實驗結果	33
3.2 FPGA-Based QEP-CORDIC 演算法實驗	35
3.2.1 硬體介紹	35
3.2.2 狀態轉換計數	39
3.2.3 實驗結果	41
3.3 線性化法與 QEP-CORDIC 之實現討論	45
第四章 串列式編碼器在低速控制應用	50
4.1 低速估測	50
4.1.1 差分法加低通濾波器	50
4.1.2 最小平方近似法 [14]	52
4.1.3 卡曼濾波器 [16]	54
4.1.4 各估測法模擬比較	57
4.2 各估測法實驗比較	58
4.3 低速控制	65
4.3.1 PI 控制器	65
4.3.2 PI 控制器加狀態控制器	69
4.3.3 應用於 CNC 機台上之低速控制	74
4.4 傳統編碼器與弦波編碼器之控制比較	77
4.5 以 19-bit 串列式編碼器作為低速控制比較基準	81
第五章 結論與未來發展	86
5.1 結論	86
5.2 未來發展	87
參考文獻	88

表目錄

表 2-1 線性化解碼法有無補償比較表.....	15
表 2-2 不同遞迴次數下 CORDIC 的誤差比較表.....	19
表 2-3 象限對應對照表.....	20
表 3-1 弦波編碼器各線功能表.....	23
表 3-2 QEP-CORDIC 不同遞迴次數實現比較表.....	38
表 3-3 傳統編碼器與弦波編碼器變異量比較表.....	44
表 3-4 傳統編碼器與弦波編碼器平均值比較表.....	45
表 3-5 線性化法與 QEP-CORDIC 演算法實驗比較.....	48
表 4-1 各估測法變異量與平均值比較.....	58
表 4-2 不同速度估測法下的變異量.....	64
表 4-3 不同速度估測法下的平均值.....	64
表 4-4 PI 控制器參數調整前後變異量比較表.....	69
表 4-5 狀態回授控制後實驗變異量比較表.....	72
表 4-6 空載與 CNC 機台上實驗變異量比較表.....	76
表 4-7 不同邊碼器解析度對控制效能所造成變異量的影響.....	80
表 4-8 不同編碼器低速控制效能之變異量比較表(19bit).....	84
表 4-9 不同編碼器低速控制效能之平均值比較表(19bit).....	84

圖目錄

圖 2-1 弦波編碼器實驗輸出分析圖	6
圖 2-2 粗略位置與精細位置相對關係圖	7
圖 2-3 弦波編碼器解碼資訊概念圖	7
圖 2-4 弦波編碼器解碼方塊圖	8
圖 2-5 編碼器轉換成 QEP 信號圖	9
圖 2-6 兩相輸入及線性輸出關係圖	10
圖 2-7 類線性化輸出誤差圖	11
圖 2-8 不同參數 A 下的補償函數最大誤差圖	12
圖 2-9 在最佳參數 $A=0.2233$ 下補償函數誤差圖	13
圖 2-10 不同參數 B 下的補償函數最大誤差圖	14
圖 2-11 在最佳參數 $B=0.2798$ 下補償函數誤差圖	14
圖 2-12 線性化輸出修改變化圖	15
圖 2-13 CORDIC 角度旋轉示意圖 [7]	16
圖 2-14 CORDIC 旋轉模式示意圖	17
圖 2-15 不同遞迴次數下原始 CORDIC 輸出圖	18
圖 2-16 遞迴次數與累計誤差關係圖	19
圖 2-17 遞迴次數與最大誤差關係圖	19
圖 2-18 象限對應圖	20
圖 2-19 CORDIC 與 QEP-CORDIC 之比較圖	21
圖 2-20 理想精細位置與粗略位置對照波形	21
圖 3-1 弦波編碼器實體圖	23
圖 3-2 TMS320F2812 功能方塊圖	25
圖 3-3 弦波編碼器介面電路圖	26
圖 3-4 DSP-Based 弦波編碼器解碼硬體圖	27

圖 3-5 ADC 校準示意圖 [9].....	28
圖 3-6 不同時間下 ADC 取樣 1.352V 輸出	30
圖 3-7 ADC 校準前後比較圖.....	32
圖 3-8 弦波編碼器解碼實驗硬體圖.....	33
圖 3-9 不同轉速下的 DSP-Based 解碼實驗	35
圖 3-10 FPGA-Based 弦波編碼器解碼實驗版	37
圖 3-11 Unrolled 架構圖 [6].....	38
圖 3-12 QEP-CORDIC 電路實現遞迴次數與閘數關係圖.....	39
圖 3-13 錯位時的位置輸出結果.....	39
圖 3-14 錯位原因分析圖.....	39
圖 3-15 狀態轉換機制示意圖.....	40
圖 3-16 錯位補償機制圖.....	40
圖 3-17 狀態轉換機制補償前.....	41
圖 3-18 狀態轉換機制補償後.....	41
圖 3-19 FPGA-Based 解碼對照比較	44
圖 3-20 線性化法與不同遞迴次數 CORDIC 模擬誤差.....	46
圖 3-21 兩種解碼法同時在 DSP 實現的誤差比較圖	47
圖 4-1 低解析度位置資訊與高解析度位置資訊比較圖.....	51
圖 4-2 16-bit 差分加低通濾波估測速度	52
圖 4-3 19-bit 高解析度估測速度	52
圖 4-4 不同參數 M 下的累積誤差圖	53
圖 4-5 不同參數 N 下的累積誤差圖.....	53
圖 4-6 16-bit 最小平方近似法估測速度	54
圖 4-7 19-bit 高解析度估測速度	54
圖 4-8 16-bit 卡曼濾波器估測速度	56
圖 4-9 19-bit 高解析估測速度	56

圖 4-10 不同估測法結果與實際差分速度圖(約 11rpm).....	57
圖 4-11 差分加低通濾波控制結果.....	60
圖 4-12 最小平方近似法控制結果.....	62
圖 4-13 卡曼濾波器控制結果.....	64
圖 4-14 伺服馬達控制系統實體圖.....	66
圖 4-15 PI 控制器不同轉速實驗結果($K_p = 2$ 、 $K_i = 0.0103$).....	67
圖 4-16 調整後 PI 控制器不同轉速實驗結果($K_p = 6$ 、 $K_i = 0.0082$).....	68
圖 4-17 狀態回授控制方塊圖.....	69
圖 4-18 系統鑑別用測試輸入.....	70
圖 4-19 系統鑑別輸出.....	70
圖 4-20 原有系統波德圖.....	70
圖 4-21 加入狀態回授控制後系統波德圖.....	70
圖 4-22 狀態回授控制後不同轉速實驗.....	71
圖 4-23 不同控制法下變異量與速度關係圖.....	72
圖 4-24 300rpm 時兩種控制器實驗結果.....	73
圖 4-25 CNC 機台照片.....	74
圖 4-26 負載之比較.....	76
圖 4-27 空載與 CNC 變異量與速度命令關係圖.....	77
圖 4-28 編碼器比較實驗架構圖.....	77
圖 4-29 不同編碼器解析度的控制效果.....	79
圖 4-30 不同編碼器控制效能變異量趨勢.....	80
圖 4-31 控制效能比較實驗架構圖.....	81
圖 4-32 19-bit 編碼器比較兩組伺服馬達控制結果.....	84
圖 4-33 不同編碼器控制變異量趨勢(19bit 對接).....	85

第一章 序論

1.1 研究動機

隨著人們生活習慣的改變和高度的工業自動化，各式的馬達應用也越多，其中可精準定位的伺服馬達更廣泛地應用在各式各樣的自動化工業場合中，使得物件的加工更快速、更精準且更安全，自動化生產的良率也提升。伺服馬達的應用場合相當廣泛，例如電腦數值工具機(CNC)、高速高精密的自動生產線、紡織機、機械手臂、機械人、電動車等等。

在一般伺服馬達控制系統中，包含有控制器(controller)、伺服馬達(servomotor)及感測器(sensor)三大部分，其中感測器作為回授信號的產生，提供了位置及速度資訊給與控制器使用。傳統所使用的編碼器(encoder)及解角器(resolver)因為其高強健性及可靠性，已普遍被工業界所使用數十年，但隨著科技的進步及各項軟硬體的發展，對於伺服馬達控制要求越來越精密，對於量測回授的需求也逐漸提高，傳統的感測裝置已不足應付當前的需求，所以便衍生出融合編碼器及解角器兩者特點的弦波編碼器。由於弦波編碼器是一種強健且具有高可信度的位置感測器，不只能應用在一般的伺服馬達控制中，更特別適用在需要高精度位置資訊的嚴苛環境中，如：機器人、飛行航空、雷達及衛星天線等。

弦波編碼器因為融合兩者間的特性，所以解讀技巧也與傳統所用的技巧有所差異，本文主要目的在於尋找一套適合弦波編碼器解碼之演算法，且實現於不同硬體，發展出高速、精準及通用的解碼器，並就其低速控制應用方面多加延伸使用。

1.2 問題界定

弦波編碼器

1. 基本弦波解碼演算法：

因為弦波編碼器具有傳統解角器特性，會因為不同演算法而產生不同的誤差值，執行時間及運算複雜度也大不相同，如何有效的利用兩相輸出解讀出目前位置，找到一個誤差值小、運算速度快的演算法，乃是本文重點之一。

2. 適用 QEP 及弦波編碼器之演算法：

弦波編碼器除具有傳統解角器的特性之外，同時具有傳統編碼器的特性，如何同時結合這兩個特性，取其長處補其短處，發展一個適合弦波編碼器的修改型演算法，也是其問題之一。

3. 信號結合後的錯位問題：

弦波編碼器的解碼過程中，習慣將信號分成兩部分：粗略位置及精細位置兩部份，獨立計算，然而這兩個信號在結合的時候容易發生錯位的問題，如何避免錯位的發生亦是弦波編碼器解碼的一大重點。

低速控制應用

1. 低速估測：

高解析度編碼器的最大貢獻在於低速伺服馬達控制之應用，除了回授信號的改善之外，低速控制時的位置及速度估測也是影響低速控制效能的一大重點，嘗試找出一個最適合的低速估測法也是一大問題。

2. 低速控制：

除了可靠的感測器及回授估測之外，控制法則也是影響控制效能的另一部份，不同的控制法有其不同的適用場合，因此找出一套適合低速時使用的控制法將是一個問題。

1.3 研究方法及步驟

弦波編碼器

1. 基本弦波解碼器演算法：

演算法首先以舊有解角器線性化解碼法為基礎，再加上誤差補償函數以降低線性化誤差；另外使用 CORDIC 演算法，利用座標旋轉的概念，藉此計算出目前的角度。最後以 DPS 及 FPGA 分別實現兩種不同演的算法，就理論及實驗兩方面分析誤差及演算時間等能力。

2. 適用結合 QEP 及弦波編碼器之演算法：

為配合弦波編碼器的特性，本文中提出一個 QEP-CORDIC 演算法，藉由輸入向量的對應關係，將原本不適合的用於信號結合的 CORDIC (coordinate rotation digital computer)演算法輸出轉換為適合要求之輸出，以作為 FPGA 實現編碼器解碼演算法使用。

3. 信號結合後的錯位問題：

為了避免錯位問題，文中提出狀態轉換機制代替原有的計數方式，狀態轉換方式有別以往直接以 QEP (quadrature encoder pulse)信號做為計數來源，改以線性化運算或是 QEP-CORDIC 演算的輸入向量做為計數來源，可完全避免錯位發生。

低速控制應用

1. 低速估測：

研究中將使用現有的差分法加低通濾波器，加上常見的最小平方近似法及卡曼濾波器共三種不同速度估測法，分析、比較、模擬各速度估測法之差異，並且實際應用於伺服馬達控制系統中，驗證其效能。

2. 低速控制：

文中分別比較 PI 控制器及 PI 控制器外加狀態回授控制器兩種不同控制架構，將兩者加上速度估測器後實現於伺服馬達控制中，並且得到最適合低速所用的控制架構，實際於 CNC 機台上測試其低速效能。

1.4 論文架構

本論文共分為五章，第一章旨在說明研究動機、目的，並且界定出研究的問題、研究方法及步驟。第二章首先介紹弦波編碼器的基本解碼原理及輸出分析，再個別介紹粗略解碼位置及精細解碼位置的各種解碼方法原理。第三章分別以數位信號處理器(DSP)及現場可程式化閘陣列(FPGA)為基礎，實現第二章所介紹之解碼技巧，並且針對 ADC 取樣校準問題、弦波編碼器錯位問題提出適合的解決方法，最後分別以理論及實驗兩方面比較不同解碼技巧之間的差異。第四章為串列式編碼器的低速應用，首先為低速回授的估測法比較，試圖尋找最適合低速範圍的估測法，再比較不同控制器其低速控制之效能差異，並將比較後所得最佳方案用於 CNC 機台上進行實際測試，比較傳統編碼器及高解析度編碼器在控制效能差異，再以 19-bit 編碼器為基準比較兩者的控制效果。最後第五章將對本文進行結論，並提出未來可供發展、應用及改善之方向。

第二章 弦波編碼器解碼演算法

2.1 弦波編碼器

本節介紹弦波編碼器的基本背景、分析弦波編碼器的輸出，並且說明弦波編碼器的解碼概要，解碼概念是將弦波編碼器的資訊分為兩個部份分別計算，最後再合成完整的位置資訊。

2.1.1 弦波編碼器之介紹

近年來在伺服馬達控制系統中對於位置回授的解析度要求越來越高，有鑑於此，傳統的編碼器(encoder)及解角器(resolver)逐漸被擁有更高解析度的弦波編碼器(sinusoidal encoder)所取代。弦波編碼器有別於傳統編碼器的工作原理，利用一對相差 90° 、隨著轉動角度而改變的正弦(sine)及餘弦(cosine)信號，可以提供更高解析度的位置資訊。

$$\begin{aligned} V_s &= \sin(P \times \theta) \\ V_c &= \cos(P \times \theta) \end{aligned} \quad (2.1)$$

P 是每 360° 弦波編碼器所產生的弦波週期數。由於其特殊的輸出信號特性，需要不同於以往的解讀方法，學者紛紛發展各式的解讀技巧，一般常見的有反正切函數運算(direct arctangent computation) [1]、或是根據舊有的解角/數位轉換器(resolver-to-digital converter)所衍生出的追跡及濾波迴路(tracking and filtering loop) [1,2]、線性化技巧(linearization) [3,4,5]或是內插法、查表法等等。在實現的方式上也可分為硬體實現及軟體實現兩種方式，本章將分別介紹兩種適合不同實現方式的解碼技巧。

2.1.2 弦波編碼器輸出分析

弦波編碼器的輸出如上所述是一對相差 90° 的正弦/餘弦信號，將這兩信號分別置於

x 軸與 y 軸座標上，令 $(x, y) = (V_c, V_s)$ 可以得到一圓，仔細分析圓形可以檢視弦波編碼器經過 ADC 取樣之後信號是否受雜訊干擾。理論上應得到一圓心在原點的正圓形，如果圓心不在原點上，表示輸入信號有 offset 的情形，如果不是正圓而是橢圓的話表示兩相出入有 gain 值的差異。下圖 2-1 是弦波編碼器的實際輸出，使用的是 HPB 公司所生產的 2048 PPR (period per revolution，即一圈為 2048 個週期)弦波編碼器，由圖得知弦波編碼器經過取樣後的輸出圓心接近原點表示 offset 非常小，兩相 gain 值也是差距不大。

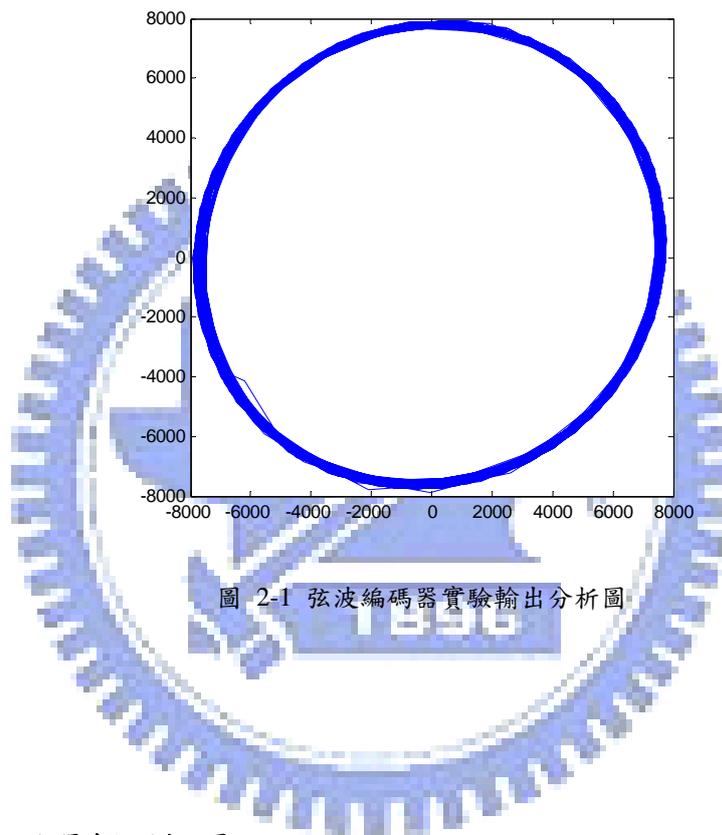


圖 2-1 弦波編碼器實驗輸出分析圖

2.1.3 弦波編碼器解碼概要

弦波編碼器解碼結果可分成兩個部份，一為粗略解析度資訊(coarse resolution data)，另一部分為精細解析度資訊(fine resolution data) [1]。粗糙解析度資訊類似原有光學編碼器(optical encoder)的 QEP 信號，負責計算弦波經過的個數；精細解析度資訊則是利用解角器特有的弦波輸出，在原本的信號中內插細分，使解析度提升，兩者相對關係如下

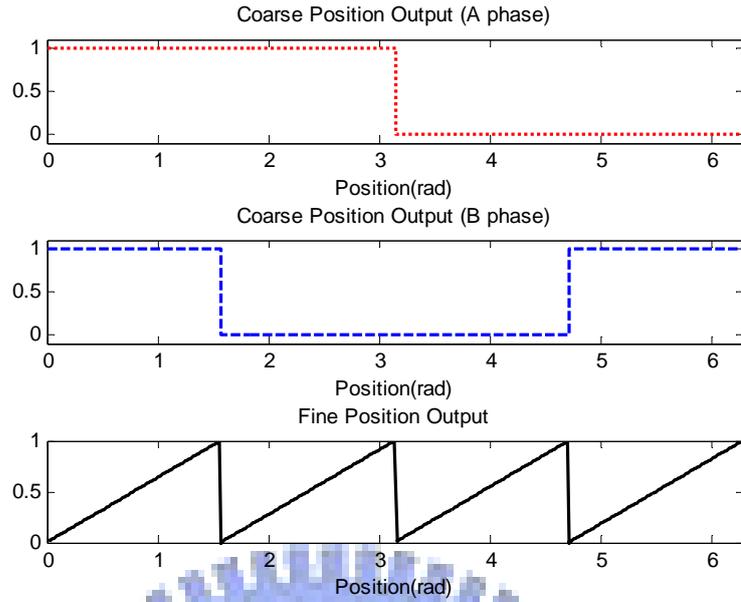


圖 2-2 粗略位置與精細位置相對關係圖

如前一節所述，將輸出分別置於 x 座標及 y 座標得 $(x, y) = (\cos\theta, \sin\theta)$ ，可做出一半徑為 1 且圓心位於原點的圓，構想是以粗糙解析度資訊計算所經過的象限數目，再加上精細解析度資訊求出每個象限內的角度便可求出高解析度位置資訊。

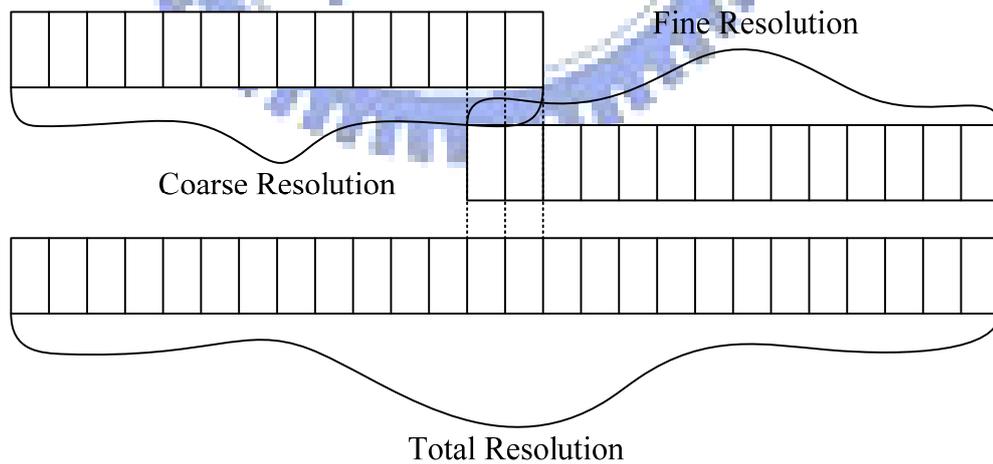


圖 2-3 弦波編碼器解碼資訊概念圖

上圖 2-3 是弦波編碼器的粗糙資訊與精細資訊組合圖，顯示在粗糙解析度與精細解

析度間有 2 位元重疊，重疊的原因是因為粗糙解析度計算的象限數目正巧與精細解析度的前兩位元所表示的位置相同。弦波編碼器的解析度是由粗糙解析度加上精細解析度所得，粗糙解析度限制主要是在於弦波編碼器硬體所限制，例如 2048 PPR 的弦波編碼器就可提供 $2048 \times 4 = 8192$ 的粗糙解析度；精細解析度則是受到硬體 ADC 解析度及不同的精細位置解碼技巧所共同決定，例如以 12 位元的 ADC 取樣最多可得 12 位元的解析度再加上解碼技巧的不同有所減少。以一個 2048 PPR 的弦波編碼器搭配 12 位元的 ADC 最高可得 $2048 \times 4 \times 2^{12} \div 4 = 2^{23}$ 的解析度，其中除 4 就是扣除掉精細解析度與粗糙解析度重疊的部份，但實際上受到硬體電路雜訊、ADC 取樣精確度及解碼技巧不同的影響，解析度可能低於理論值。

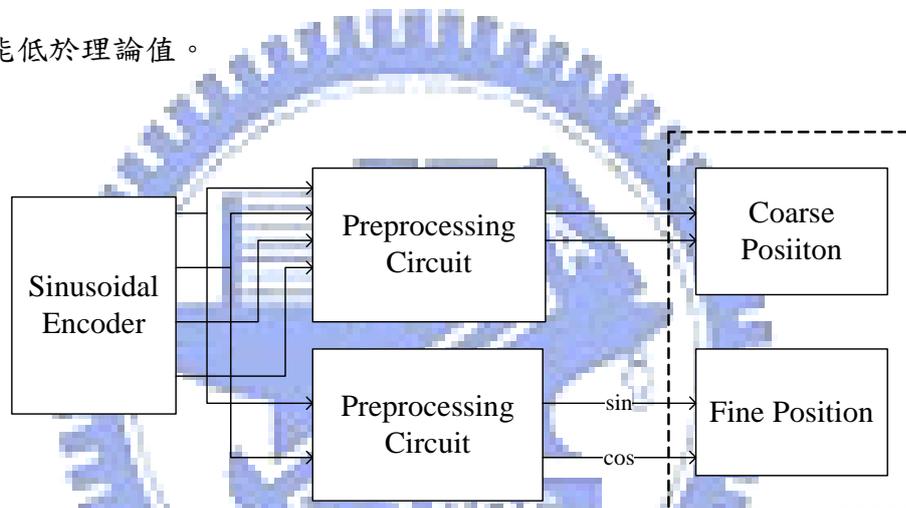


圖 2-4 弦波編碼器解碼方塊圖

上圖 2-4 是弦波編碼器的解碼方塊圖，粗略解碼位置與精細解碼位置都會先經過一些前處理電路，再進入解碼的步驟，而前處理電路的好壞會嚴重造成電路雜訊問題，直接影響到弦波編碼器的解析度。

2.2 粗略位置解碼

粗略解碼位置的基本概念是用原本的 QEP 計數方法來計算目前位置，所以必須將原本的弦波波形轉成數位方波，最簡單的方法是將原本的方波經過一個比較器之後轉換成數位方波，如下圖 2-5 所示

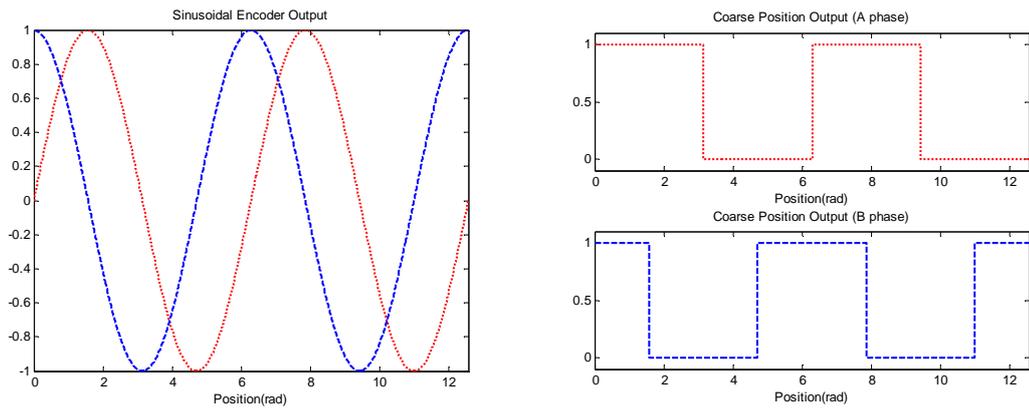


圖 2-5 編碼器轉換成 QEP 信號圖

在比較器的兩端分別放入兩相弦波中的一個信號，另一端則是放零電位作為基準，所以這個基準零電位非常重要，若是零電位漂移，會造成粗略解碼位置與精細解碼位置的資料無法重疊，產生錯位的問題。以一個 2048 PPR 的弦波編碼器來可有 $2048 \times 4 = 8192 = 2^{13}$ 的解析度，意即 13-bit 的解析度。

2.3 精細位置解碼

在這節裡介紹了兩種不同的精細位置解碼方法，一種是依傳統解角/數位轉換器所使用的線性化解碼法，另一種則是利用座標旋轉計算反切函數的 CODRIC 演算法，並且提出一個適用於弦波編碼器的 CORDIC 輸入向量對應方法，稱之為 QEP-CORDIC。

2.3.1 線性化法解碼 [3]

線性化解碼法的基本概念是把兩相的弦波經過適當的數學運算之後轉換成一個隨著轉動角度而增加的電壓信號，並且透過特別的補償方法讓輸出更接近完美的線性波形。這個方法的特點在於不需要額外的記憶體來建表查詢，但是計算比較複雜，且精確度會隨著補償函數的不同而誤差也會不同。

首先定義一個類似線性化的三角波形 $V_o(\theta)$ ，將兩相信號取絕對值後相減可以得到

$$V_o(\theta) = |\sin \theta| - |\cos \theta| \quad (2.2)$$

下圖是兩相輸出與類線性化輸出 $V_o(\theta)$ 的相對關係圖

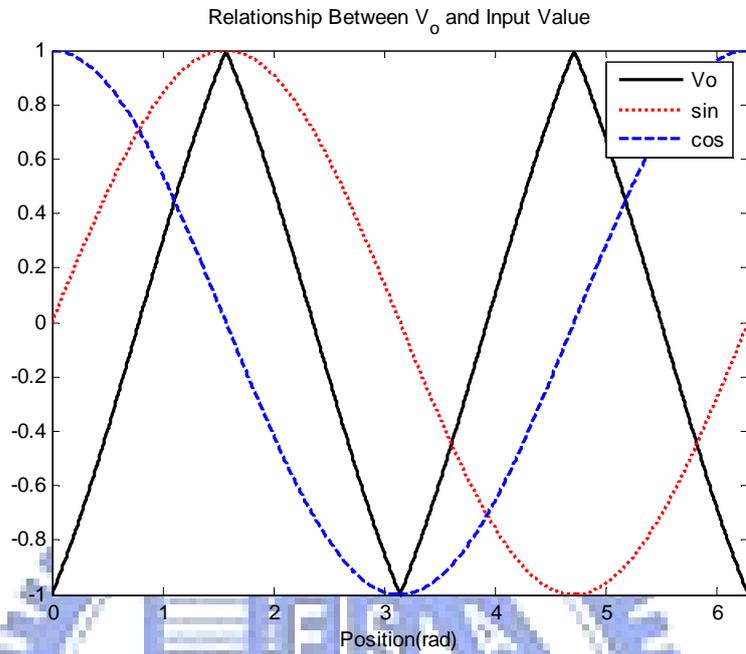


圖 2-6 兩相輸入及線性輸出關係圖

接下來我們定義完美三角波 $PT(\theta)$ 的方程式

$$PT(\theta) = -1 + \frac{4}{\pi} \sin^{-1} |\sin \theta| \quad (2.3)$$

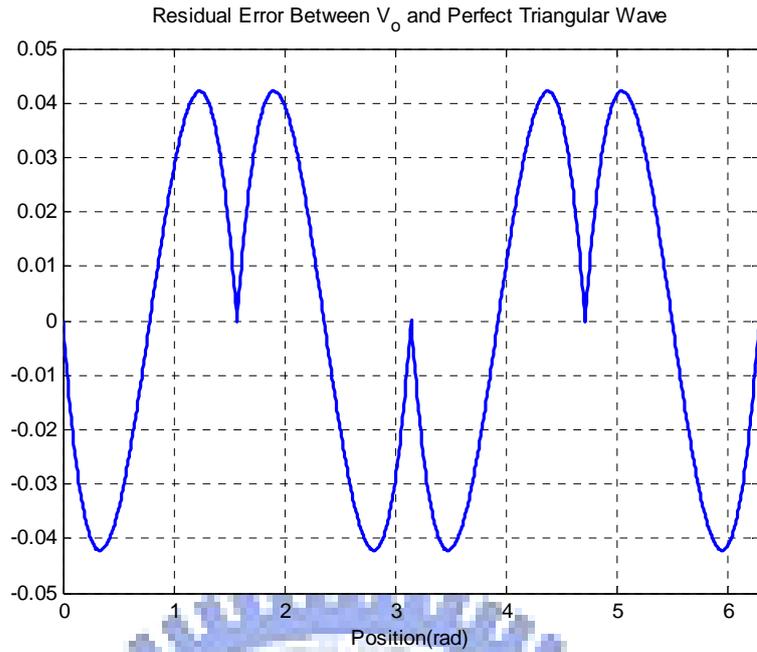


圖 2-7 類線性化輸出誤差圖

$$E(\theta) = V_o(\theta) - PT(\theta) \quad (2.4)$$

上圖 2-7 是 $V_o(\theta)$ 與完美三角波 $PT(\theta)$ 的誤差圖 $E(\theta)$ ，可以看出最大誤差約是 $\pm 4.2\%$ 的弦波振幅，以 2048 PPR 的弦波編碼器換算，誤差約是在 $360/2048/4 \times \pm 0.042 = \pm (1.8457 \times 10^{-3})^\circ$ 左右，換算精度約是 17-bit 左右，但這是在不考慮 ADC 解析度及雜訊干擾下所做的理想推論值。

接下來我們補償類線性化與完美三角波之間的誤差，首先定義補償過後的類線性化輸出電壓信號 $V_{oL}(\theta)$ ，補償函數 $C(\theta)$ ，可得下補償後的輸出式

$$V_{oL}(\theta) = V_o(\theta) + C(\theta) \approx PT(\theta) \quad (2.5)$$

我們的目的在於找到一個適當的函式 $C(\theta)$ 使其盡可能的接近誤差 $E(\theta)$ ，假設第一種補償函數 $C_1(\theta)$ 為

$$C_1(\theta) = A \times |\sin \theta| \times |\cos \theta| \times (|\sin \theta| - |\cos \theta|) \quad (2.6)$$

則補償過後的線性化輸出誤差函數為 $e(\theta)$ ，可得

$$e(\theta) = V_{oL}(\theta) - PT(\theta) = V_o(\theta) + C(\theta) - PT(\theta) \quad (2.7)$$

將 $V_o(\theta)$ 、 $PT(\theta)$ 與 $C_1(\theta)$ 代入 $e(\theta)$ 中可求出誤差與位置角度的關係，但首先必須先

調整補償函數 $C_1(\theta)$ 中的參數 A 。下圖 2-8 是補償函數 $C_1(\theta)$ 在不同參數 A 的情況下與補償誤差函數 $e(\theta)$ 的最大誤差圖。在此我們是使補償誤差函數的最大誤差值最小化，而不是使累積誤差最小化，此目的在於每個位置都可以達到最小的誤差平衡。圖中顯示在 $A=0.2233$ 的情況下可以使最大誤差最小化，此時的最大誤差值為 $\pm 0.2714\%$ ，換算為 2048 PPR 的弦波編碼器誤差約是 $360/2048/4 \times \pm 0.002714 = \pm (1.1927 \times 10^{-4})^\circ$ 左右，換算精度約是 21-bit 左右，這一樣是不考慮到 ADC 解析度下所做的理想推論值。

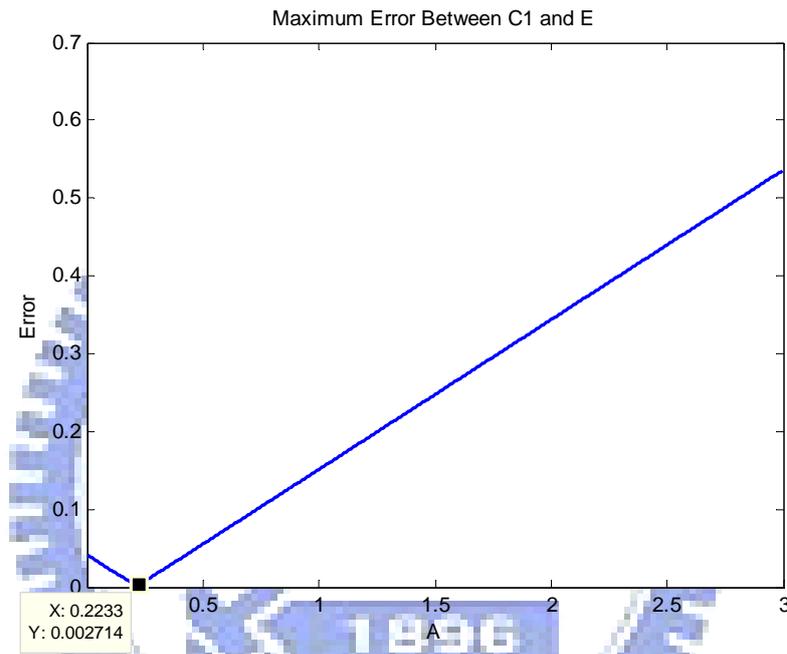


圖 2-8 不同參數 A 下的補償函數最大誤差圖

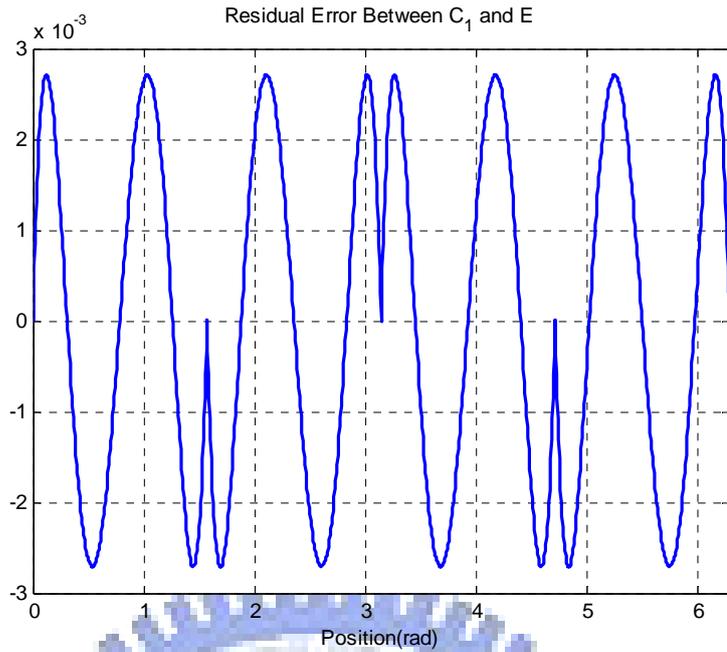


圖 2-9 在最佳參數 $A=0.2233$ 下補償函數誤差圖

圖 2-9 是在 $A=0.2233$ 下所繪出的誤差與位置關係圖，在一圈週期中最大誤差一共發生 8 次，分別發生在 $m\pi \pm 0.1257rad$ 與 $\left(m + \frac{1}{2}\right)\pi \pm 0.1257rad$ 上 (m 為整數)。整合 $V_o(\theta)$ 、 $V_{ol}(\theta)$ 、 $C_1(\theta)$ 與參數 A 後得到第一種補償後線性化輸出式

$$V_{ol1}(\theta) = (1 - 0.2233 \times |\sin \theta| \times |\cos \theta|) \times (|\sin \theta| - |\cos \theta|) \quad (2.8)$$

接下來考慮另一種比較複雜的補償函數 $C_2(\theta)$ [4]，如下

$$C_2(\theta) = \frac{B \times |\sin \theta| \times |\cos \theta| \times (|\sin \theta| - |\cos \theta|)}{|\sin \theta| + |\cos \theta|} \quad (2.9)$$

其中的參數 B 也是用來調整補償函數的誤差值，意義與補償函數 $C_1(\theta)$ 中的 A 相同，為了求出最佳的 B 值我們模擬不同參數 B 下對應的補償後誤差函數 $e(\theta)$ 的最大值。由下圖 2-10 中得到在 $B=0.2798$ 時有最小的最大誤差值，此時誤差值約為 $\pm 0.02414\%$ ，換算成 2048 PPR 得弦波編碼器之後實際誤差最大值約是 $360/2048/4 \times \pm 0.0002414 = \pm (1.0608 \times 10^{-5})^\circ$ ，相當於 24-bit 的解析度，當然這一樣是不考慮到 ADC 取樣解析度及雜訊干擾下的理論值。

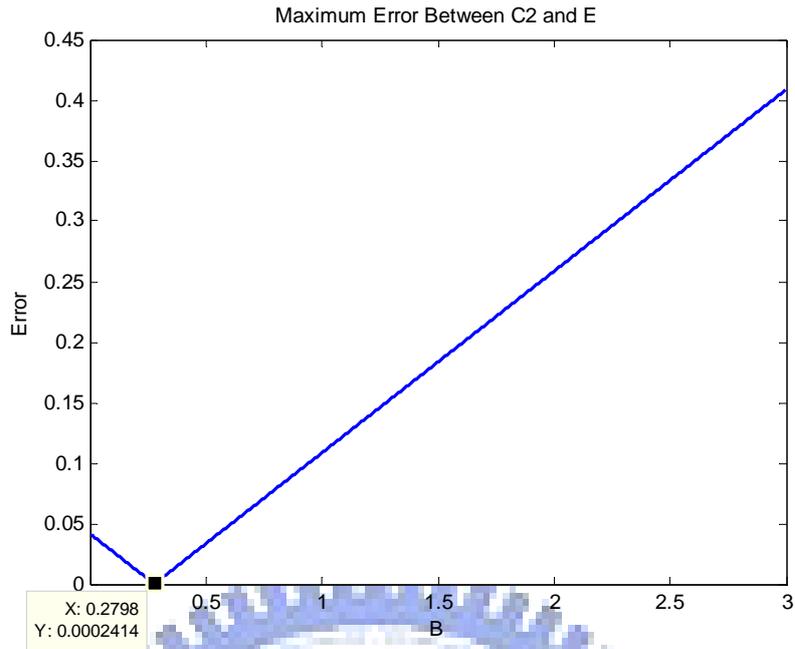


圖 2-10 不同參數 B 下的補償函數最大誤差圖

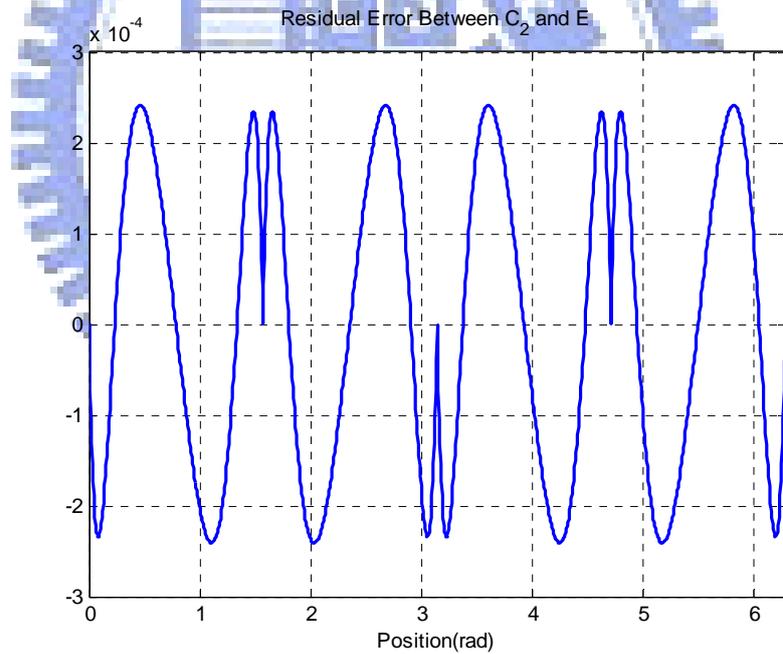


圖 2-11 在最佳參數 $B=0.2798$ 下補償函數誤差圖

上圖顯示一樣在一圈中有 8 個位置會出現最大誤差，分別是在 $m\pi \pm 0.465rad$ 與 $\left(m + \frac{1}{2}\right)\pi \pm 0.465rad$ 的地方 (m 為整數)。整合 $V_o(\theta)$ 、 $V_{oL}(\theta)$ 、 $C_2(\theta)$ 與參數 B 後得到第二種補償後線性化輸出式

$$V_{oL2}(\theta) = \left(1 - \frac{0.2798 \times |\sin \theta| \times |\cos \theta|}{|\sin \theta| + |\cos \theta|} \right) (|\sin \theta| - |\cos \theta|) \quad (2.10)$$

我們在此得到兩種補償函數 $C(\theta)$ ，明顯的補償函數 $C_2(\theta)$ 比 $C_1(\theta)$ 更接近誤差函數 $E(\theta)$ ，具有更高的補償度。但無論是 $V_{oL2}(\theta)$ 或 $V_{oL1}(\theta)$ 都不適合直接作為弦波編碼器的精細解碼位置，直接與粗略解碼位置做結合，原因在於 $V_{oL1}(\theta)$ 與 $V_{oL2}(\theta)$ 的週期都是 π 與粗略解碼位置的變化週期 $\frac{\pi}{2}$ 並不符合，因此需要經過小修改將 $\frac{\pi}{2} \sim \pi$ 及 $\frac{3\pi}{2} \sim 2\pi$ 這兩個區間內的輸出做反向轉換，使 $V_{oL1}(\theta)$ 與 $V_{oL2}(\theta)$ 成為週期為 $\frac{\pi}{2}$ 的信號，變化關係如

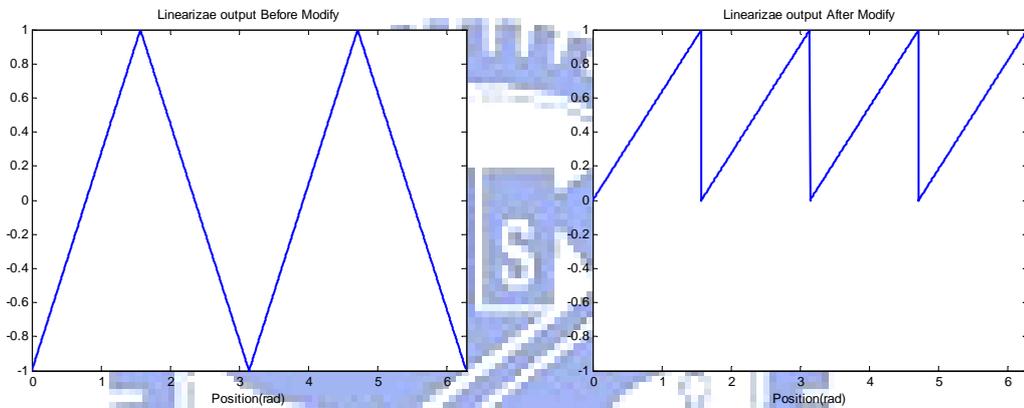


圖 2-12 線性化輸出修改變化圖

表 2-1 線性化解碼法有無補償比較表

	無補償	補償函數 C_1	補償函數 C_2
最大誤差角度 $^{\circ}$	1.8457×10^{-3}	1.1927×10^{-4}	1.0608×10^{-5}
解析度	17bit	21bit	24bit

2.3.2 CORDIC 演算法解碼 [6]

接下來討論另一種有別於前一節解碼技巧，CORDIC 全名為 coordinate rotation digital computer，中文稱為座標旋轉數位演算法，顧名思義這是一個特別適用於數位系統上實現的座標旋轉演算法。Volder 在 1959 年提出 CORDIC 演算法的概念，雖然這個

想法不是個創舉，但是在計算一般除法、三角函數及指數函數上都有著相當大的好處，甚至對於以往較難處理的反三角函數、雙曲線函數等等都可以經由這套演算法得到解答。此法的最大優點在於只需使用到加法器及位移暫存器加上不斷的遞迴運算便可完成計算，完全不需使用到乘法器，特別適用於硬體電路的實現。

下列公式是我們所熟知的座標旋轉公式，假設起始位置為 (x, y) ，旋轉角度為 θ ，終結位置是 (x', y')

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \cos \theta \begin{bmatrix} 1 & -\tan \theta \\ \tan \theta & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.11)$$

接下來我們建立兩個很重要的假設：

1. 我們所要旋轉的角度 θ 是由 n 個已經預設好的角度所組成，也就是

$\theta \cong \pm\theta_0 \pm\theta_1 \pm\theta_2 \pm\cdots \pm\theta_{n-1}$ ，如下圖。



圖 2-13 CORDIC 角度旋轉示意圖 [7]

2. 每個預設角度 θ_i 是 2 的負幕次方的反切函數，也就是 $\theta_i = \tan^{-1} 2^{-i}$ ，並且將 θ_i 值存在 ROM 中以供使用。

將上述兩個假設代入公式中，可以得到近似式

$$\begin{aligned} \begin{bmatrix} x' \\ y' \end{bmatrix} &= \cos \theta \begin{bmatrix} 1 & -\tan \theta \\ \tan \theta & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \cong \prod_{i=0}^{n-1} \cos \theta_i \prod_{i=0}^{n-1} \begin{bmatrix} 1 & -\tan \theta_i \\ \tan \theta_i & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= K \prod_{i=0}^{n-1} \begin{bmatrix} 1 & -\tan \theta_i \\ \tan \theta_i & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = K \prod_{i=0}^{n-1} \begin{bmatrix} 1 & -d_i \cdot 2^{-i} \\ d_i \cdot 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned} \quad (2.12)$$

當中的 $K = \prod_{i=0}^{n-1} \cos \theta_i$ ，只是個常數增益，僅影響向量的長度變化， $d_i = \pm 1$ 代表 θ_i 的

旋轉方向，在不考慮常數增益的影響後，可以改寫成下列遞迴方程式

$$\begin{aligned} x_{i+1} &= x_i - d_i \cdot 2^{-i} \cdot y_i \\ y_{i+1} &= y_i + d_i \cdot 2^{-i} \cdot x_i \end{aligned} \quad (2.13)$$

此外再加上轉角累加方程式

$$\theta - \sum_{i=0}^{n-1} \theta_i \cong 0 \Rightarrow z_{i+1} = z_i - \theta_i = z_i - d_i \cdot \tan^{-1} 2^{-i} \quad (2.14)$$

上兩式合起來便是傳統 CORDIC 的遞迴運算公式，若是在遞迴運算式中稍加改變，加入參數 m 後，其數學運算更可擴大分為圓周(circular)函數、線性(linear)函數與雙曲線(hyperbolic)函數三種

$$\begin{aligned} x_{i+1} &= x_i - m \cdot d_i \cdot 2^{-i} \cdot y_i \\ y_{i+1} &= y_i + d_i \cdot 2^{-i} \cdot x_i \\ z_{i+1} &= z_i - d_i \cdot \theta_i \end{aligned} \quad \begin{cases} m=1 & \text{circular} \\ m=0 & \text{linear} \\ m=-1 & \text{hyperbolic} \end{cases} \quad \theta_i = \begin{cases} \tan^{-1} 2^{-i} & m=1 \\ 2^{-i} & m=0 \\ \tanh^{-1} 2^{-i} & m=-1 \end{cases} \quad (2.15)[8]$$

除此之外，還可依向量旋轉的方式分為旋轉模式(rotation mode)及向量模式(vectoring mode)兩種。旋轉模式是將初始向量旋轉一個我們所欲旋轉的角度 θ 後，去求得新的向量位置；而向量模式則是將初始向量旋轉到 x 軸上，可以計算出初始向量與 x 軸間的夾角 α 。在弦波編碼器的解碼上，我們使用的是圓周函數下的向量模式。

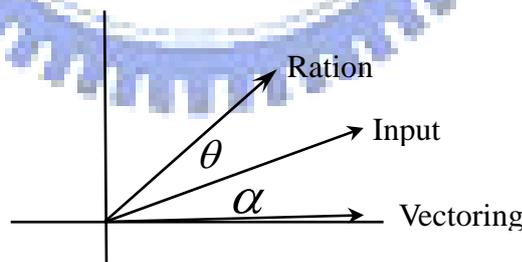


圖 2-14 CORDIC 旋轉模式示意圖

在旋轉模式下， z_0 放入我們所想要旋轉的角度 θ ，每次的遞迴運算的結果 z_i 就是剩餘未旋轉的角度，此模式下旋轉方向 d_i 是由剩餘未旋轉角度 z_i 所決定， $d_i = \text{sign}(z_i)$ ，最後的 z_n 表示未轉動的剩餘角度，代表期望旋轉角度 θ 與實際旋轉角度

$\pm\theta_0 \pm\theta_1 \pm\cdots \pm\theta_{n-1}$ 之間的誤差；在向量模式中我們試圖將初始向量旋轉到 x 軸上，也就是使 $y'=0$ ，將 z_0 設為 0， d_i 則是由 x_i 與 y_i 所決定， $d_i = -\text{sign}(x_i \cdot y_i)$ ，每次向量旋轉的角度都累計於 z_i 中，當最後向量轉至 x 軸後， z_n 便是所有旋轉過後的角度總和，也就是初始向量與 x 軸的近似夾角 α 。下圖 2-15 是 CORDIC 在不同遞迴次數的模擬輸出圖，因為 CORDIC 向量模式下所求得的是反切函數 \tan^{-1} 的值，所以在 $\frac{\pi}{2} \sim \frac{3\pi}{2}$ 之間的輸出會與 $-\frac{\pi}{2} \sim \frac{\pi}{2}$ 將相同。

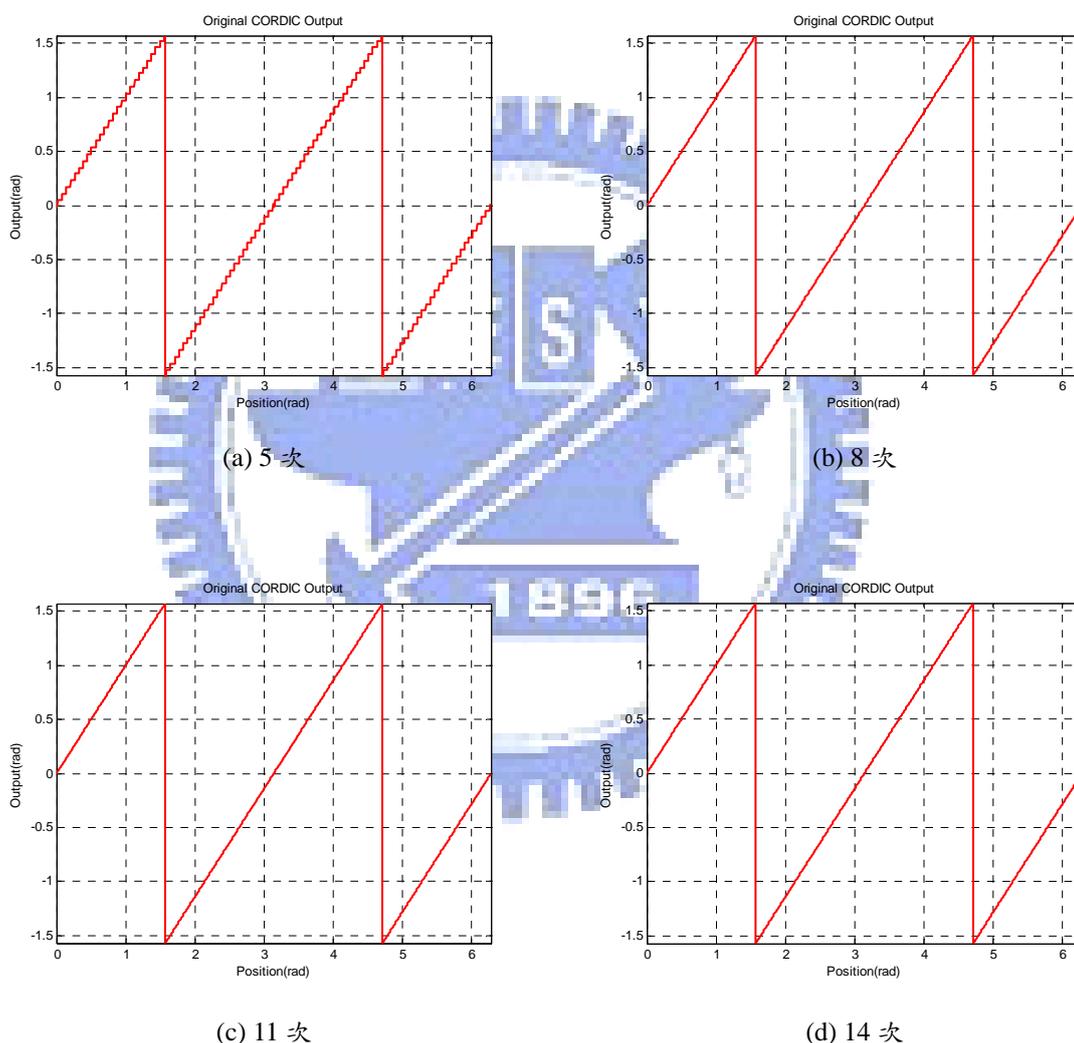


圖 2-15 不同遞迴次數下原始 CORDIC 輸出圖

利用 CORDIC 演算法解碼的最大解析度會依據遞迴運算的次數有所不同，而遞迴運算次數則會受到原始輸入信號位元長度而限制。每多執行一次遞迴運算時就會受到 2^{-i}

的影響將遞迴輸入多向右位移一位元，也就是說，當遞迴次數超過原始輸入的位元長度時，右移結果將會為 0，輸出已達收斂。

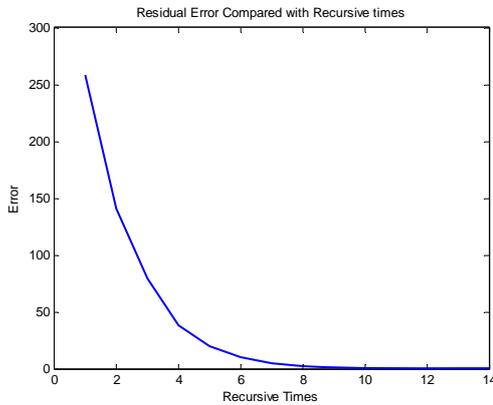


圖 2-16 遞迴次數與累計誤差關係圖

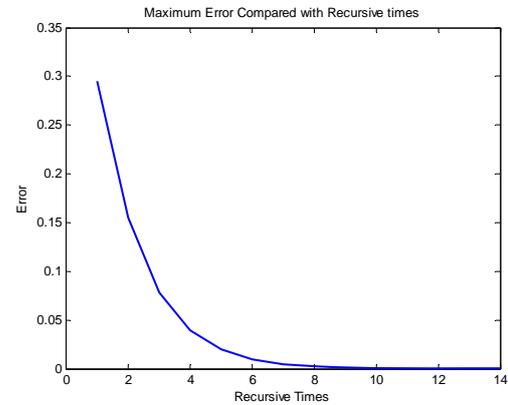


圖 2-17 遞迴次數與最大誤差關係圖

上兩圖是模擬 CORDIC 在不同遞迴運算次數下的弦波編碼器解碼後一周期(360°，分 2000 份，每份 0.18°)累計誤差及最大誤差，圖中顯示隨著遞迴次數的增加兩種誤差量都會越來越少，這是因為我們使用越多的內建已知角度去近似所求得的角度關係。同樣以 2048 PPR 的弦波編碼器作估算，在執行 8 次遞迴運算後的最大誤差約是±0.24853%，換算為角度則是 $360/2048/4 \times \pm 0.0024853 = (1.0921 \times 10^{-4})^\circ$ ，此時的解析度約是 21-bit 左右；若遞迴次數增加到 12 次時最大誤差約剩下±0.01552%，換算成角度是 $360/2048/4 \times \pm 0.0001552 = (6.8203 \times 10^{-6})^\circ$ ，擁有約 25-bit 的解析度；若是再增加遞迴次數到 14 次時誤差已剩下±3.8674×10⁻³%，約是 $360/2048/4 \times \pm 3.8674 \times 10^{-5} = (1.7000 \times 10^{-6})^\circ$ ，此時有 27-bit 的解析度。

表 2-2 不同遞迴次數下 CORDIC 的誤差比較表

	8 次	12 次	14 次
最大誤差角度°	1.0921×10^{-4}	6.8203×10^{-6}	1.7000×10^{-6}
解析度	21bit	25bit	27bit

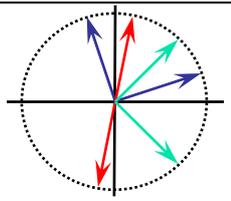
從上表中可以看出一個規律，範例中所使用的 2048 PPR 弦波編碼器本身已由粗略位置資訊提供 13-bit 的解析度，在遞迴運算 8 次得到 21-bit 解析度，額外增加 21-13=8 bit 的解析度，12 次則增加 25-13=12 bit 的解析度，14 次則額外獲得 27-13=14 bit 的解析度。也就是說遞迴次數會等於額外所增加的解析度，原因在於我們的遞迴次數受到輸入

向量的位元長度限制，8 位元長度的輸入向量最多只能進行 8 次遞迴運算，而輸入向量的位元長度則是受到 ADC 的解析度所決定，所以一個 8-bit 的 ADC 最多只能提供 8 位元的輸入向量位元長度，最多就只能提供 CORDIC 做 8 次的遞迴運算，故可以增加的解析度大約就是 8-bit，並且不可能超過 8-bit。

2.3.3 QEP-CORDIC 演算法

從圖 2-15 中得知原始 CORDIC 的輸出週期與線性化輸出一樣是 π ，而且範圍在 $-\frac{\pi}{2} \sim \frac{\pi}{2}$ 之間，不適合與粗略位置資訊做結合，因此在這提出一個專用於弦波編碼器解碼的修正法則。首先釐清目前的兩個問題及想要達到的目標，一是要把原本週期是 π 的輸出轉為週期為 $\frac{\pi}{2}$ 輸出；二是把原本輸出的範圍由 $-\frac{\pi}{2} \sim \frac{\pi}{2}$ 轉為 $0 \sim 1$ 之間。

就第一個問題而言，本文提出一個輸入信號象限對應轉換的方法，將所有的輸入信號轉換到相對應的第一象限位置上，如此一來便可使輸出周期變為 $\frac{\pi}{2}$ ，而輸出範圍則變為 $0 \sim \frac{\pi}{2}$ 之間。

表 2-3 象限對應對照表			
原初始位置	修正後初始位置	修正後起始軸	
第一象限 (x_0, y_0)	(x_0, y_0)	+x	
第二象限 (x_0, y_0)	$(y_0, -x_0)$	+y	
第三象限 (x_0, y_0)	$(-x_0, -y_0)$	-x	
第四象限 (x_0, y_0)	$(-y_0, x_0)$	-y	

利用表 2-3 很快的便可知道新舊輸入的對應關係，如下圖 2-19 顯示週期確實降為 $\frac{\pi}{2}$ ，就如同把 $0 \sim \frac{\pi}{2}$ 這段輸出複製到其他象限一樣的效果。並且與圖 2-20 相比看出經過轉換後的輸出與理想波形都齊集大小都相當符合。

針對第二個問題最直接的方法則是由 ROM 中的 θ_i 去做改變，如 2.3.2 中所做的第

二個假設，我們在 ROM 中需要存著 $\theta_i = \tan^{-1} 2^{-i}$ 的數值以供轉角累加方程式做累加用，所以我們只需要把 θ_i 都同除 $\frac{\pi}{2}$ 後再存入 ROM 中，就能夠將輸出縮小在 0~1 之間了。下

圖 2-19 是 CORDIC 輸出與 QEP-CORDIC 輸出的比較，對照圖 2-20 則可證實 QEP-CORDIC 的輸出才是適合與 QEP 粗略位置信號作合成使用。

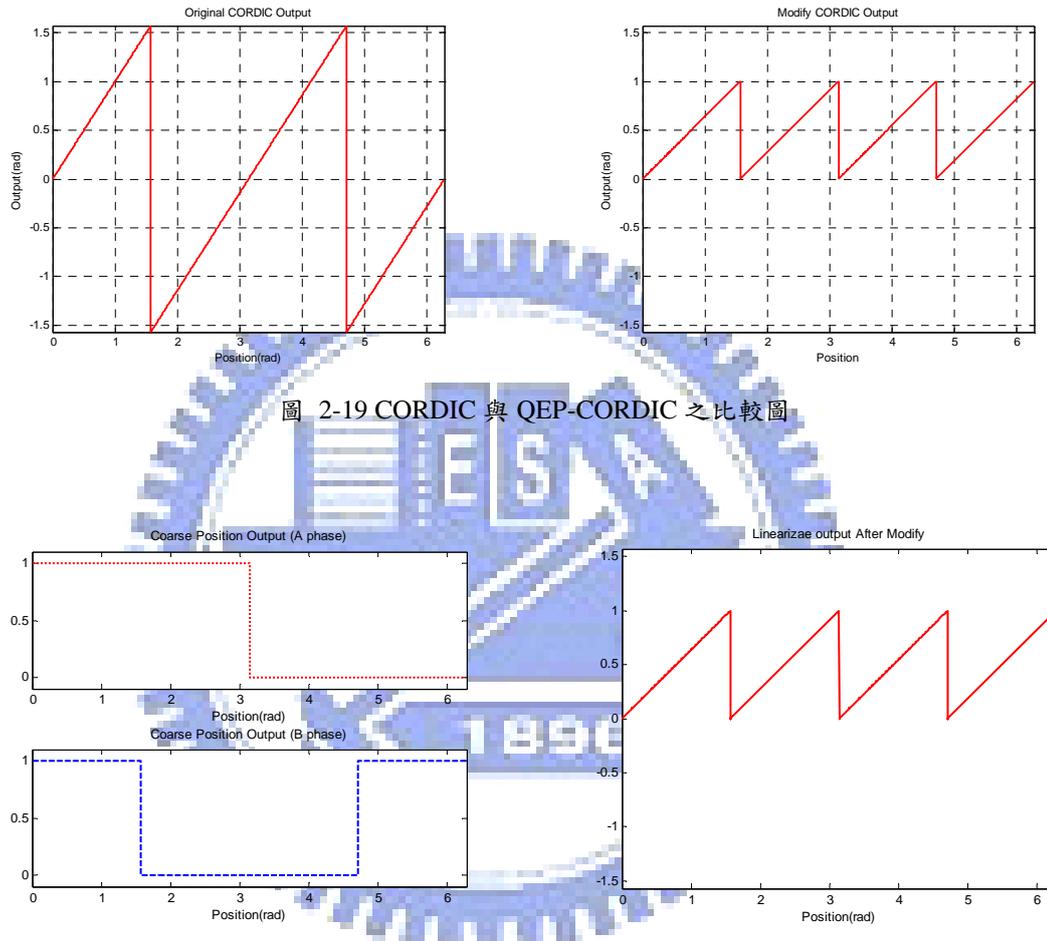


圖 2-19 CORDIC 與 QEP-CORDIC 之比較圖

圖 2-20 理想精細位置與粗略位置對照波形

第三章 弦波編碼器解碼實現

3.1 DSP-Based 線性化法解碼實驗

本節首先介紹 DSP-Based 線性化解碼法的各項硬體，包含了實驗用的弦波編碼器及測試用的伺服馬達連接裝置及 DSP 實驗版，在 3.1.2 中介紹了 ADC 校準法以提供更精確的 ADC 取樣信號，增加解碼後的精度，最後實際測試 DSP-Based 線性化解碼後的速度回授結果。

3.1.1 硬體介紹

弦波編碼器

本論文所用的弦波編碼器是由 HPB 公司所生產的弦波編碼器，可以提供 2048 PPR 的正弦/餘弦信號，同時也提供 1 PPR 的正弦/餘弦信號作為絕對位置的判斷，除此之外傳統編碼器所提供的 Z 相信號也以弦波的方式輸出。實體如下圖





圖 3-1 弦波編碼器實體圖

編碼器需要供給 5V 電源，信號輸出的方式是以差動輸出以減少雜訊影響，所有的弦波電壓範圍都是在 2~3V 之間，並且額外提供 2.5V 直流電壓以供判斷使用，下是弦波編碼器各信號線功能說明

表 3-1 弦波編碼器各線功能表

線色	功能	輸入/出電壓
藍	A^+	$2.5 + \sin(2048 \times \theta)$
藍黑	A^-	$2.5 - \sin(2048 \times \theta)$
綠	B^+	$2.5 + \cos(2048 \times \theta)$
綠黑	B^-	$2.5 - \cos(2048 \times \theta)$
黃	Z^+	Z index
黃黑	Z^-	Z index
褐	A'^+	$2.5 + \sin(\theta)$
褐黑	A'^-	$2.5 - \sin(\theta)$
白	B'^+	$2.5 + \cos(\theta)$
白黑	B'^-	$2.5 - \cos(\theta)$
灰	Med	2.5
灰黑	Gnd	0
紅	V_{cc}	5

黑	Gnd	0
---	-----	---

DSP 解碼核心

近年來由於超大型積體電路(VLSI)的技術迅速發展,使得 DSP 數位訊號微控器的運算速度及周邊處理功能有大幅的提升,取代了傳統的微處理器(microprocessor),在許多應用領域如行動通訊、資訊家電與馬達驅動等都被廣泛的使用。因此本研究採用由德州儀器(Texas Instrument, TI)公司研發的 TMS320F2812 這顆 DSP 作為控制核心。

利用 DSP 強大的數學運算能力,可以最少的硬體周邊電路實現馬達控制所需要的功能。而在德州儀器公司所生產的 DSP 中,'C2000 系列的硬體發展卡是專為馬達控制而設計的,其中本論文所使用的是此系列中的 F2812 晶片控制器,其擁有 150 MHz (6.67 ns cycle time)的高速運算處理能力,比 TI 早期出產的 DSP 'C240 快 7 倍,比傳統 8051 單晶片快約 150 倍。為了提升運算效能,核心部份的算數邏輯單元(ALU)、累積器(ACC)均採用 32 位元定點運算,並以硬體方式實現乘法器、位移器等,如此僅需一個指令週期(instruction cycle)即可完成乘加運算。在整數計算方面,為減少數值計算所衍生的問題如溢位(overflow)等,採用硬體式的位移器來提高軟體執行的精確度。

F2812 主要擁有的記憶體有 28K × 16 Flash EEPROM、兩組 4K × 16 Single-Access RAM (SARAM)與一組 8K × 16 SARAM,並採用哈佛匯流排(Harvard bus)架構。舊型的 DSP 大部分是採用分離式的程式記憶體、資料記憶體與 I/O 記憶體,而新型的 DSP 如 F2812 則是使用包含上述三種記憶體的單獨記憶體空間,使用起來更為有彈性,也可以減少不同記憶體之間搬動資料的指令集。

Spectrum Digital 公司利用 F2812 晶片結合自行開發的周邊模組,發展出 eZdspTM F2812 初學板(DSP Started Kit, DSK),其週邊包含 16 個通道的 12 位元類比數位轉換器(ADC),串列傳輸支援四種常用的型式:SPI (serial peripheral interface)、SCI (serial communications interface)、eCAN (enhanced controller area network)與 McBSP (multi-channel buffered serial port),另外最高可支援 56 個 GPIO (general-purpose I/O)。為 TMS320F2812 功能方塊圖。

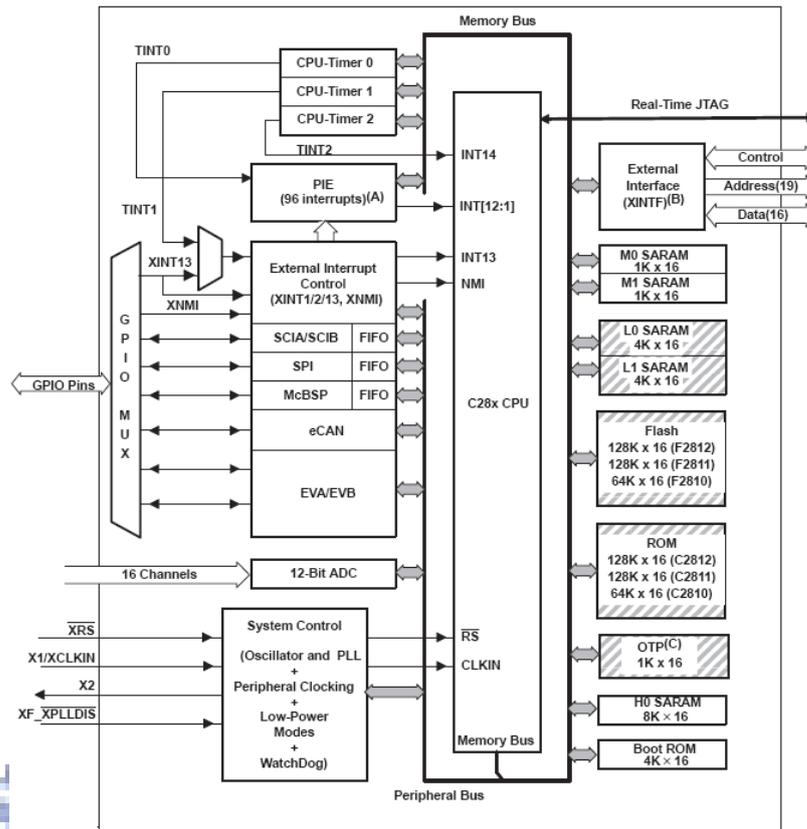


圖 3-2 TMS320F2812 功能方塊圖

由於弦波編碼器的解碼中類比數位轉換器占有相當重要的地位，所以特別加以說明內建之 12 位元的類比數位轉換器的各項功能如下：

- 內建兩組取樣及維持(Sample-and-hold)模組的 12 位元 ADC
- 兩組取樣及維持模組共有 16 的輸入通道
- 取樣及維持模組可分為同步取樣模式或是依序取樣模式
- 類比輸入範圍 0~3V
- 快速的轉換能力，最大取樣率為 12.5 Msps
- 自動取樣轉換功能，可在一次中斷中執行 16 次的 ADC 轉換，每次的轉換可任意指定 16 個通道中的任一個
- 16 個轉換結果暫存器，最高可儲存 16 次的轉換結果
- 擁有四種不同的方法啟動 ADC 轉換
- 可設定連續轉換模式

介面電路

由上述基本硬體說明中得知 DSP 的內建 ADC 類比輸入範圍是在 0~3V 之間，而弦波編碼器的輸出則是在 2~3V 之間，經過差動放大之後範圍會在 -1~1V 間，所以必須設計一個電位提升電路將電壓範圍提升到 0.5~2.5V 之間，至於為什麼沒有提升到 0~3V 的 ADC 全部範圍的原因在下節將會說明。

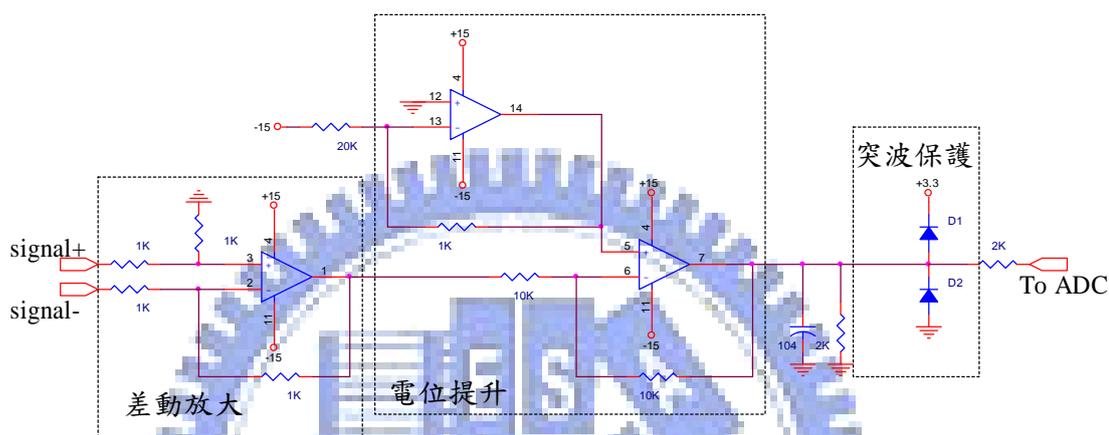


圖 3-3 弦波編碼器介面電路圖

首先將信號經過差動放大減少雜訊後，再進行電位提升，要將電壓由原本的 0V 提升到 1.5V 需由另一個 OP 放大器提供 0.75V 的提升參考電壓，最後設有一個突波保護電路防止過大電壓輸入 ADC 造成損壞。



圖 3-4 DSP-Based 弦波編碼器解碼硬體圖

上圖 3-4 是由 eZdsp 實驗發展版與介面電路所組合而成的 DSP-Based 弦波編碼器解碼硬體圖，上層是介面電路板，除了電位的提升功能之外，還包含下節所使用到的 ADC 校準電路及伺服馬達控制所需的其他功能，在此便不多做說明，而下層則是 eZdsp 實驗發展版。為了將雜訊影響降到最低，兩者採上下兩層重疊方式組合而成，減少線路連接所造成的雜訊干擾。

3.1.2 即時線上 ADC 校準

弦波編碼器的解碼過程中，ADC 取樣弦波信號用來解碼精細位置資訊，所以 ADC 的取樣信號可信度是影響弦波解碼後精度的最大因素之一。DSP-Based 的弦波編碼器解碼是以線性化解碼法為主，使用的 ADC 是 TI 公司 TMS320F2812 數位信號處理器(DSP) 所內建的 12-bit ADC，由於所使用的實驗發展版 eZdsp 上並沒有設置校準 ADC 的周邊硬體電路，而弦波取樣需要極為精確可信的取樣結果，所以我們使用一個 ADC 校準方

法，利用外部電路提供兩個穩定已知的電壓信號作為參考電壓，強制將這兩個取樣結果指定到我們所要的輸出位置上，搭配上 DSP 本身所具有的運算功能，便可算出實際的增益值及偏移量，在每次取樣時都能進行電壓的校準及補償。

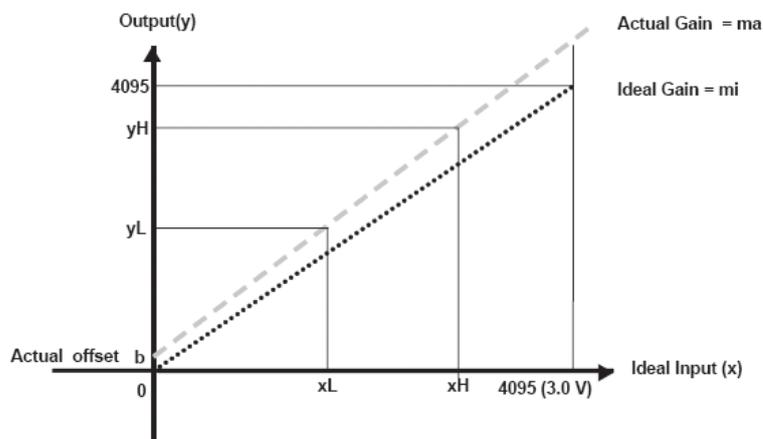


圖 3-5 ADC 校準示意圖 [9]

上圖 3-5 是 ADC 的校準示意圖，我們假設理想的 ADC 方程式為

$$y = x \times mi \quad (3.1)$$

其中輸入為 x ，輸出為 y ， mi 是理想的增益值，以實驗用的 12-bit 對應到 0~3V 的 ADC 來說，增益值應該為 $\frac{2^{12}-1}{3-0} = 1365$ 。而實際的 ADC 輸出方程式則是

$$y = x \times ma + b \quad (3.2)$$

ma 是實際的增益， b 則是實際偏移量(offset)，這兩個數值在每次進行校準運算的時候都會從新運算。

校準的流程是在每次 ADC 取樣後，我們都會得到兩組輸入與三組輸出數據，分別是已知輸入的高電壓輸出 y_H 、已知輸入的低電壓輸出 y_L 及未知輸入的電壓輸出 y ，在加上我們原本做為基準的輸入 x_H 與 x_L 共五個參數。

首先求出實際增益量 ma

$$ma = \frac{y_H - y_L}{x_H - x_L} \quad (3.3)$$

接著求出實際偏移量 b

$$b = y_L - x_L \times ma \quad (3.4)$$

估測出未知輸出 y 所應對應到的輸入 \hat{x}

$$\hat{x} = \frac{y-b}{ma} \quad (3.5)$$

最後將估測出的輸入 \hat{x} 經過理想的 ADC 輸出方程式得到理想輸出值

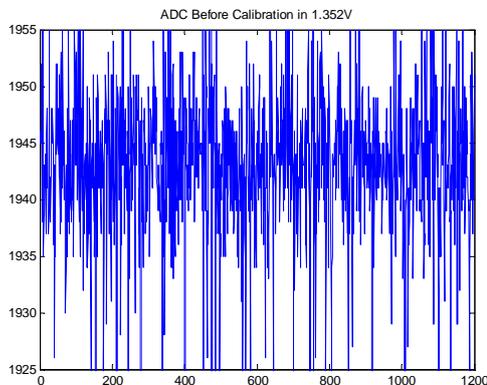
$$\hat{y} = \hat{x} \times mi \quad (3.6)$$

即時線上 ADC 校準流程如下：

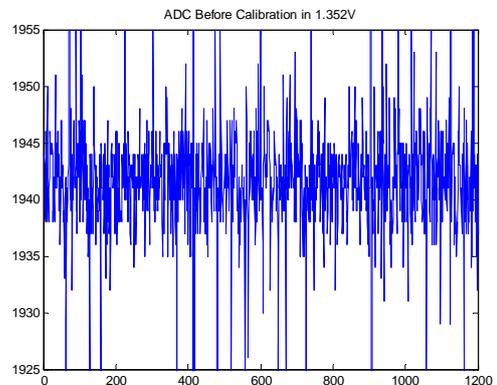
1. 先以方程式(3.3)算出實際增益量 ma
2. 再以方程式(3.4)求出實際偏移量 b
3. 利用 ADC 取樣結果 y 與方程式(3.5)估測出目前輸入電壓 \hat{x}
4. 最後以方程式(3.6)將估測輸出電壓對應到理想輸出 \hat{y}

經過上述四個步驟便可將原本未經校準補償的輸出 y 補償成理想輸出 \hat{y} ，而且在每次 ADC 取樣時都執行這樣校準，以得到最佳的 ADC 取樣效果。實驗中所使用提供穩定電壓的 IC 分別為 LD1117 與 LM317 兩種，LD1117 提供 2.5V 左右的穩定電壓，實際測量後實驗所用的 LD1117 應為 2.485V，而 LM317 則是一顆可調整輸出的穩壓 IC，其最低輸出電壓約為 1.25V，在實際測量後實驗用的 LM317 輸出為 1.260V。

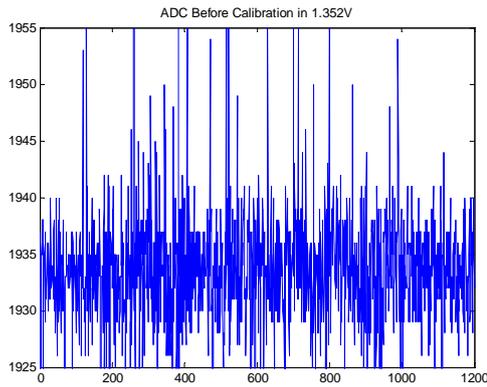
以下實驗是在不同時間裡用同樣的電壓測試 ADC 輸出，經過實驗證實內建 ADC 在不同時間下取樣可能會有時變問題，所以每次取樣都應該進行即時線上校準，不適合以離線方式先行算出再補償。



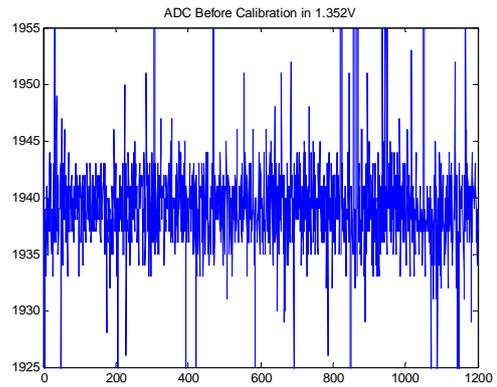
(a) 第一次取樣



(b) 第二次取樣



(c) 第三次取樣



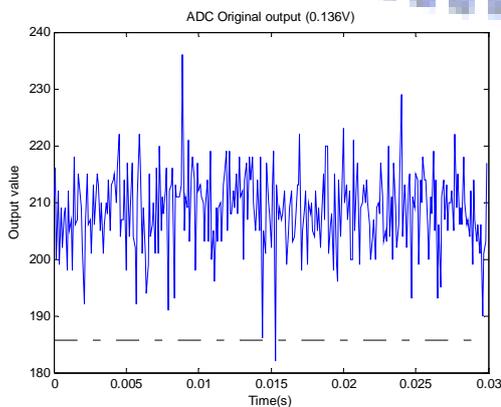
(d) 第四次取樣

圖 3-6 不同時間下 ADC 取樣 1.352V 輸出

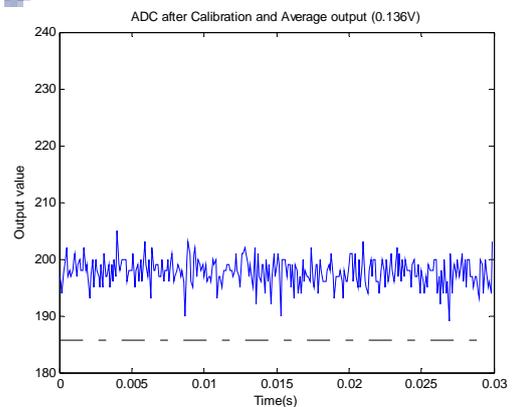
當然這樣的校準都是在假設 ADC 的輸出是線性為主，實際上 ADC 的輸出並不是線性關係，若是要做非線性估測可能會使用過多系統效能，所以在此指假設 ADC 為線性輸出，並且在實驗中也將弦波編碼器的範圍調整在線性區間內。

除了 ADC 校準可以提升精度之外，因為 TMS320F2812 的 ADC 設有多通道的取樣及維持(sample-and-hold)功能，利用此功能同時對取樣弦波編碼器的兩相輸出各進行取樣 7 次(最後剩下的兩次取樣留給校準用的參考電壓轉換用)，再取其平均值為輸出，可以減少高頻雜訊的影響。

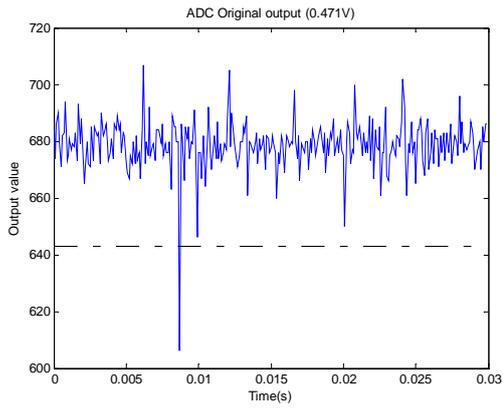
下列各圖是在不同電壓下經過即時線上校準及平均之後的輸出比較



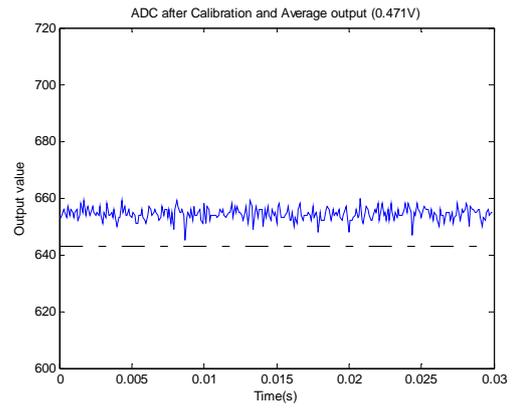
(a) 0.136V 校準前



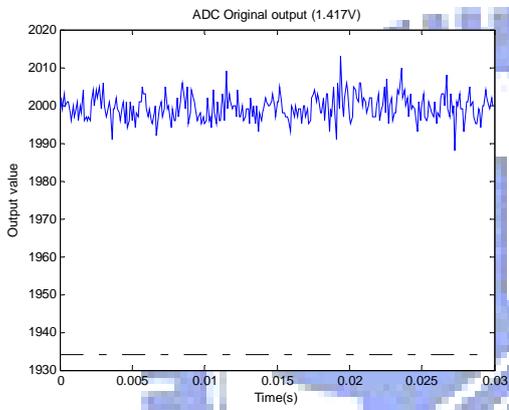
(b) 0.136V 校準後



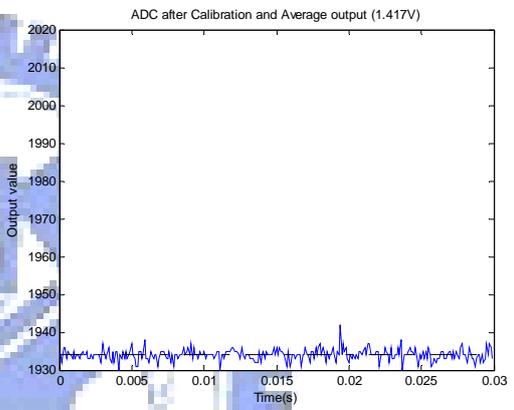
(c) 0.471V 校準前



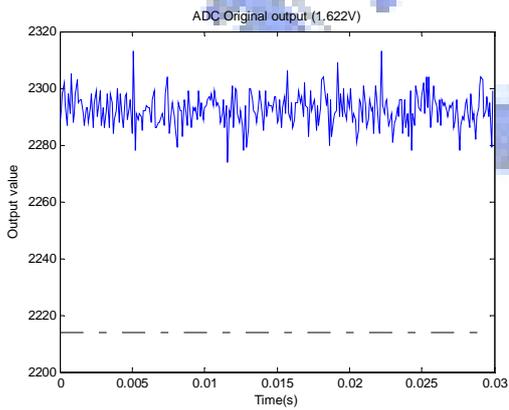
(d) 0.471V 校準後



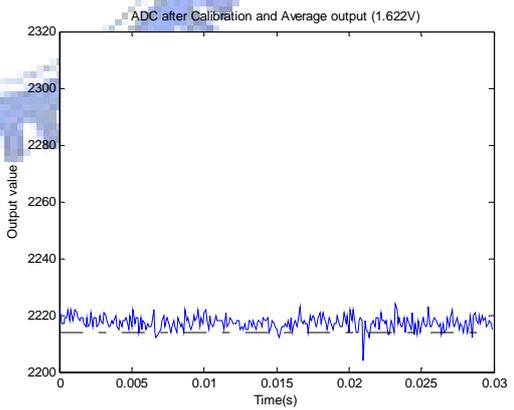
(e) 1.417V 校準前



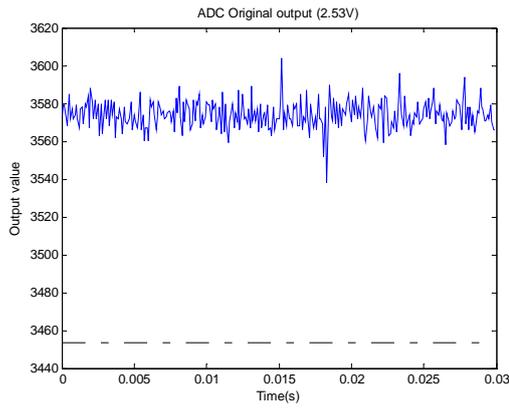
(f) 1.417V 校準後



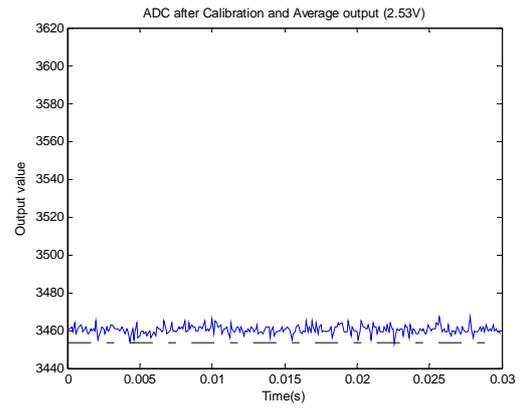
(g) 1.622V 校準前



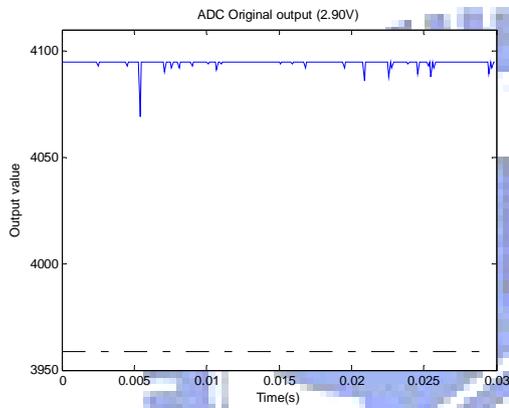
(h) 1.622V 校準後



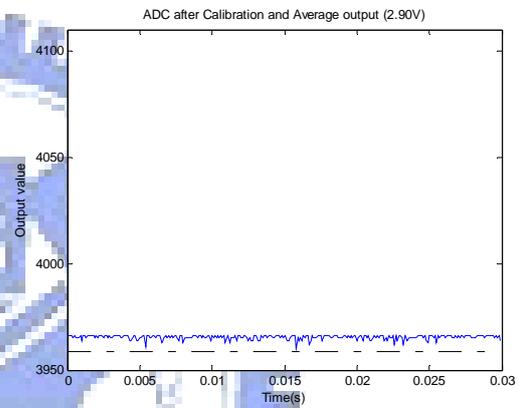
(i) 2.53V 校準前



(j) 2.53V 校準後



(k) 2.90V 校準前



(l) 2.90V 校準後

圖 3-7 ADC 校準前後比較圖

由上面各張在不同區間的比較圖看來，輸入電壓在偏低或偏高的區間校準的效果似乎比較少，這是因為在這些區域內本來就屬非線性區間，已與假設有所出入，再加上這個區間並不在我們提供的兩個基準電壓間，所以屬於外插估測，準確度確實會大大減少。另外在 2.90V 的時候 ADC 的取樣輸出已經達到最大值 4095，這個時候再怎麼補償也無法達到效果，因此在弦波編碼器解碼時我們僅使用 0.5~2.5V 這個線性區間。

3.1.3 實驗結果

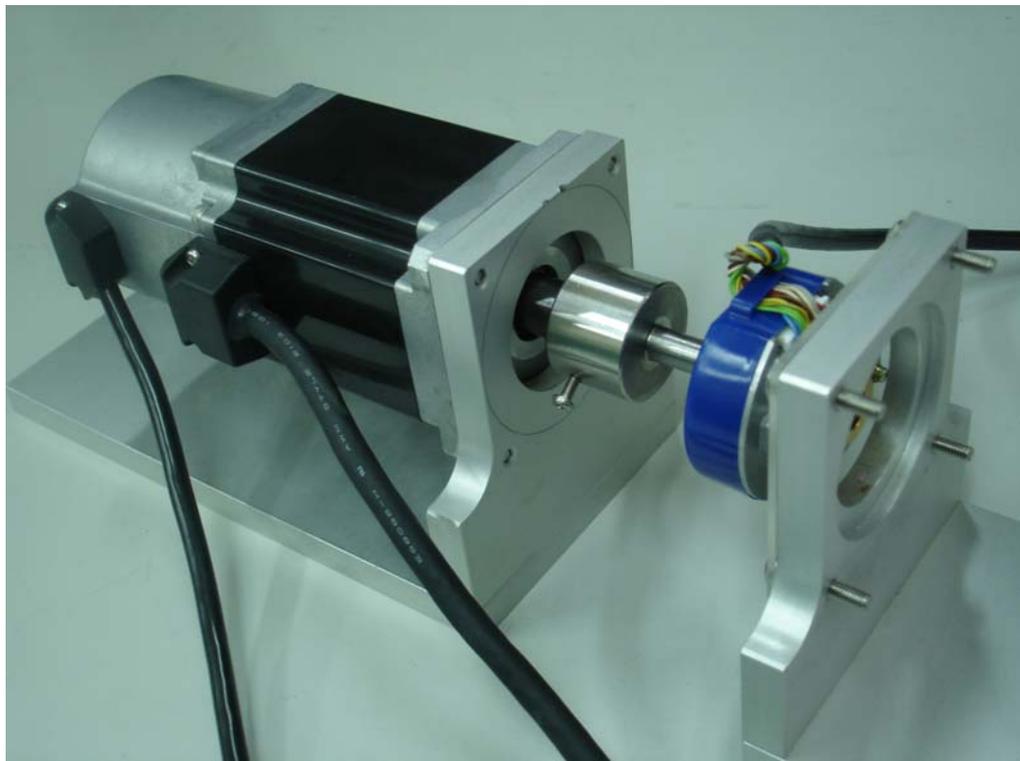
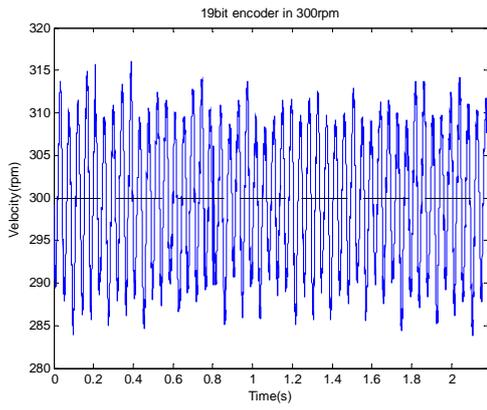
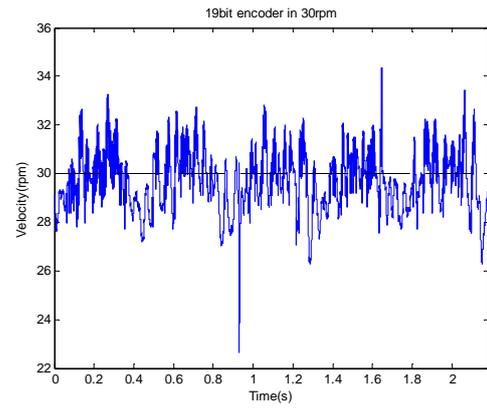


圖 3-8 弦波編碼器解碼實驗硬體圖

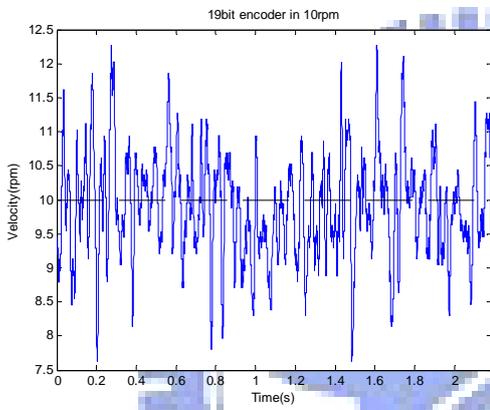
實驗中以附有 10000 PPR(此指 pulse per revolution)的傳統編碼器的 400 W 伺服馬達以套管帶動弦波編碼器，如上圖 3-8。用弦波編碼器位置回授後的速度估測來看解碼的效果，取樣頻率為 7500 Hz。因為 DSP-Based 的解碼方法花費的時間比較多，所以無法以一個 DSP 同時完成 DSP-Based 的解碼以及馬達控制兩個功能。實驗中分別以一個 DSP 為馬達控制器，另一個 DSP 為弦波編碼器解碼用，所以僅紀錄 DSP-Based 弦波解碼的速度輸出。



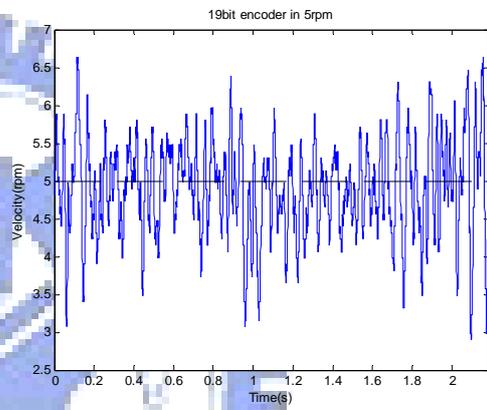
(a) 300rpm



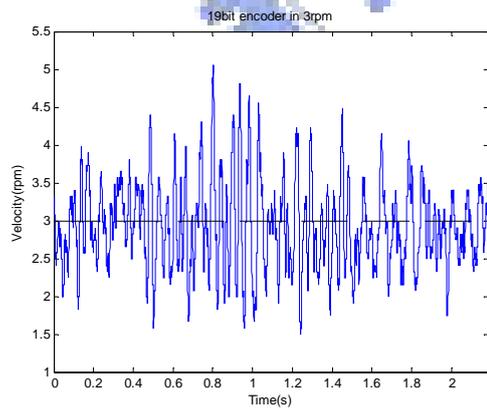
(b) 30rpm



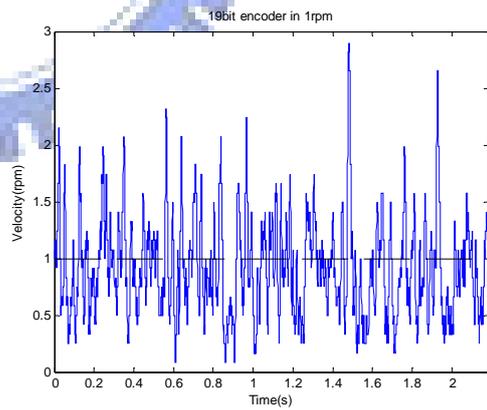
(c) 10rpm



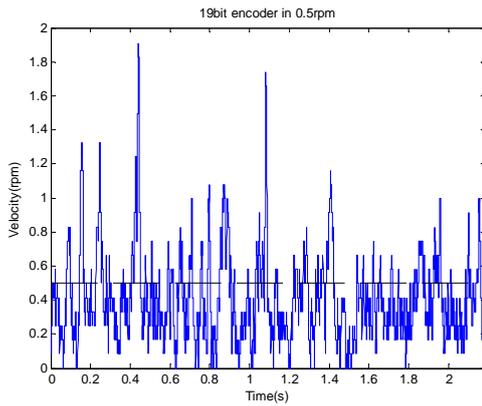
(d) 5rpm



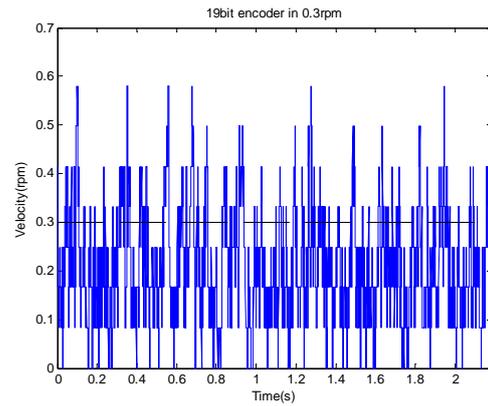
(e) 3rpm



(f) 1rpm



(g) 0.5rpm



(h) 0.3rpm

圖 3-9 不同轉速下的 DSP-Based 解碼實驗

由上面八種不同轉速下的實驗結果顯示，在 $1rpm$ 以上時 DSP-Based 的解碼都能夠有良好的效果，並且平均值也與速度命令將近，但在 0.5 及 $0.3rpm$ 時平均值似乎低於速度命令。

由實驗得知 DSP-Based 弦波編碼器解碼效果確實可靠，能夠提供馬達控制器所需的位置資訊，但由於線性化法所需的計算較為複雜，所花費的計算時間較多，下一節將改用 FPGA 為實驗硬體，搭配上不同的解碼技巧以大大縮短運算時間。

3.2 FPGA-Based QEP-CORDIC 演算法實驗

本節實際將 QEP-CORDIC 演算法實現於 FPGA 上，利用此演算法計算弦波編碼器的精細位置，除了基本的硬體介紹之外，在 3.2.2 中進一步討論了錯位問題的發生，並且提出另一種新的粗略位置解碼方法來改善這個問題，最後實際測試 FPGA-Based QEP-CORDIC 演算法解碼後的速度回授。

3.2.1 硬體介紹

FPGA-Based 的實驗硬體與前節所用的弦波編碼器是相同的，介面電路也是相似，僅提升的範圍有所不同，唯一的差異在於解碼本體的 FPGA 及外部 ADC 的使用兩部份

現場可程式化邏輯閘(Field programmable gate array, FPGA) [10]

實驗所使用的 FPGA 是 Xilinx 公司所發展的 Spartan-3AN 系列，是專為空間有限的環境下所設計，為業界第一片內建 Flash RAM 且具有多重開機模式的非失真性 FPGA。實驗所使用的詳細型號為 XC3S700AN(FGG484)，此款 FPGA 提供了 700K 個邏輯閘(system gate)、1472 個可配置邏輯方塊(configurable logic block, CLB)、92K 的分散式記憶體(distributed RAM)、360K 的區塊型記憶體(block RAM)、372 隻外部 I/O 腳，並設有 20 個 32bit 的專用乘法器，其內建 In system Flash RAM 更是高達 8Mbit。

實驗所使用發展版是以 Spartan-3AN 為核心的 Spartan-3AN Starter Kit Board 初學者使用套件[11]，實驗版上設有 133.33MHz 及 50MHz 兩個震盪器可依需求選擇使用；FPGA 的開機配置記憶體則設置了 4Mbit 的可配置化 Flash PROM、32Mbit 的並列 NOR Flash 及兩個 16Mbit 的串列 SPI Flash 三種來源，加上內建的 In system Flash RAM 共有四種不同的開機配置記憶體來源可供使用；一個具有雙輸入通道附有可程式化放大器的 14bit SPI-Based ADC；一個具有四輸出通道的 12bit SPI-Based DAC；其他一般 I/O 裝置則有 PS/2 port、10/100 乙太網路孔、12bit VGA port、9 pin RS232 port (DTE and DCE)、兩行 16 字元的 LCD 顯示器及基本的按鍵與 LED 顯示燈等等，另外設有 USB 下載裝置。

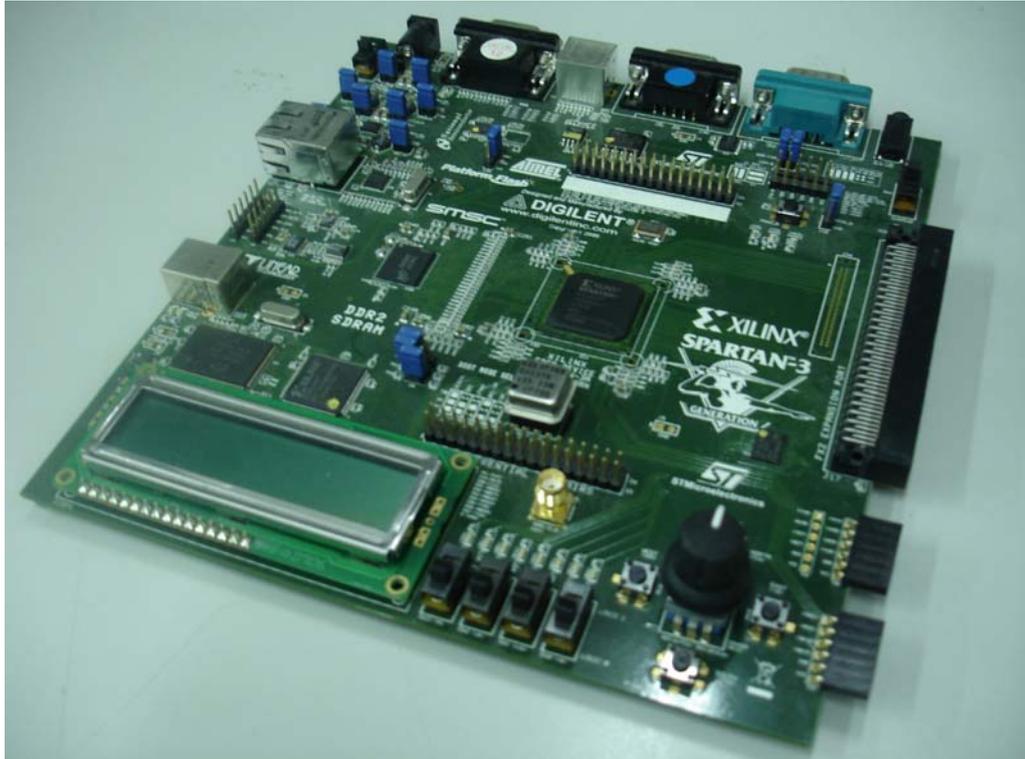


圖 3-10 FPGA-Based 弦波編碼器解碼實驗版

上圖 3-10 是 Spartan-3AN Starter Kit Board 的實體圖。接下來對於實驗版上所配置的 14bit SPI-Based ADC 功能進行簡單說明。

Linear Technology LTC1407A-1 基本功能：

- 具有雙通道同步差動輸入的 ADC
- 每通道最大取樣速度為 1.5Msps
- 類比電壓輸入範圍為 $\pm 1.25V$
- 設定為 SPI slave mode
- 轉換結果須使用 34 個 SPI 周期進行傳輸
- SPI clock 不可大於 51MHz

FPGA-Based 的解碼介面電路與 DSP-Based 解碼介面電路大致相同，在此不再說明。

FPGA 實現架構

要以 FPGA 實現 CORDIC 演算法有 Bit-Serial、Bit-Parallel、High-Radix 及 Unrolled

[12]等等的架構，也有更多學者致力研究更新更快速的架構，但這並不是本文討論的重點。在此我們選擇 unrolled 的架構來實現，其特點有：

- 每個位移暫存器的位移數是固定的
- 角度累計值 θ_i 分散於系統各處
- 可以非常快的系統時脈執行
- 相對於其他架構需要較多的空間及資源

以下為 unrolled 的 CORDIC 架構圖

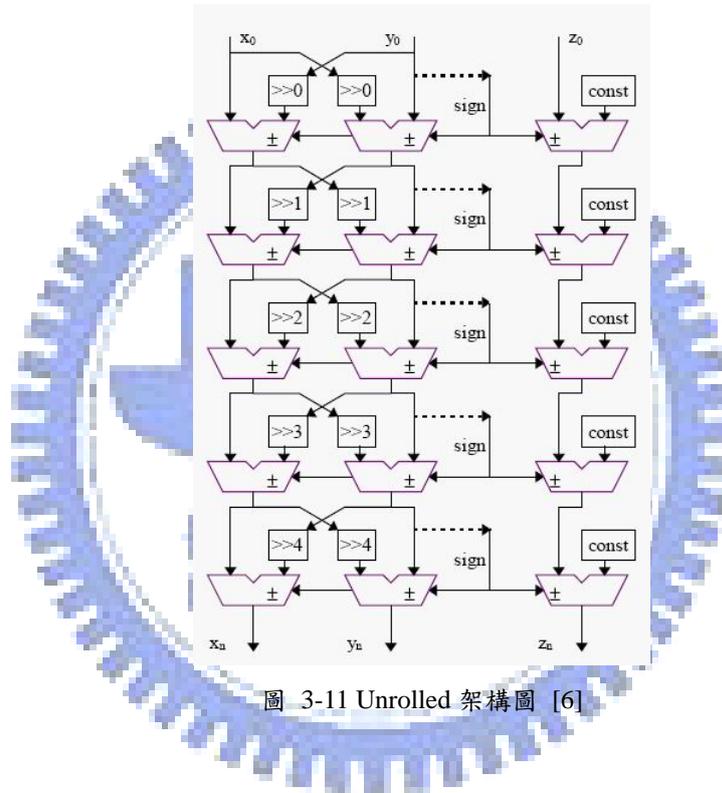


圖 3-11 Unrolled 架構圖 [6]

以 unrolled 架構實現 QEP-CORDIC 演算法，再搭配上節所提出的狀態轉換計數機制，不同遞迴次數實現後所使用的邏輯閘數及運算週期比較如下表

表 3-2 QEP-CORDIC 不同遞迴次數實現比較表

	2 次	5 次	8 次	11 次	14 次
使用閘數	313	4231	8782	13177	17521
運算週期	8	11	14	17	20

由上表得知減少遞迴次數後運算週期隨之減少，但基本上是 6 個基本週期，再加上遞迴運算的次數就是全部運算所需花費的週期。邏輯閘使用數方面則是每減少三次遞迴運算約減少 4300 個邏輯閘數，但是隨著遞迴次數的增加，使用的邏輯閘數會更多，不

單是增加了額外遞迴所需要的邏輯閘，還必須將所有內部資料位元長度增加。

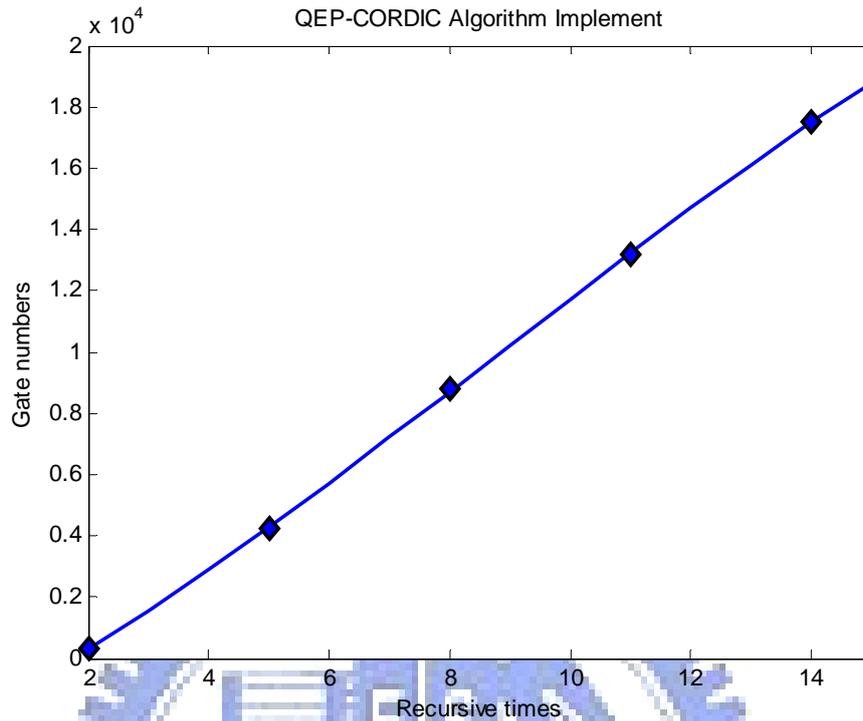


圖 3-12 QEP-CORDIC 電路實現遞迴次數與閘數關係圖

3.2.2 狀態轉換計數

因為 QEP 信號與 QEP-CORDIC 計算來源是屬於兩條不同的信號線，在取樣時難免會有兩個信號零電位漂移的問題，此時會造成兩個信號的輸出在做結合時產生偏移錯位，如下圖 3-13

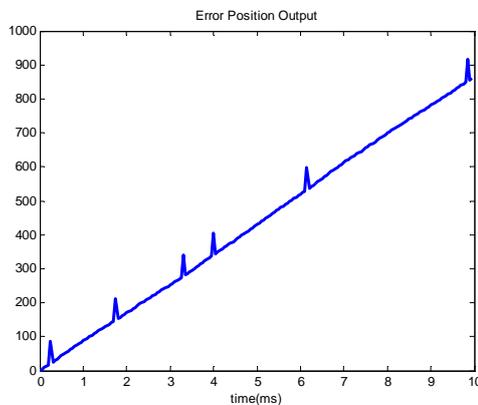


圖 3-13 錯位時的位置輸出結果

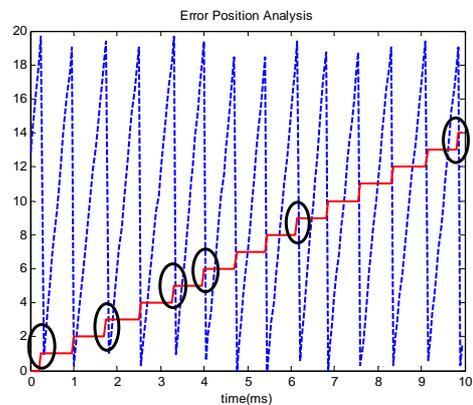


圖 3-14 錯位原因分析圖

CORDIC 輸出信號在做變化的同時，理論上 QEP 信號也應該同時產生變化，但實際上因為兩者的取樣基準浮動容易造成無法同時變化。這樣的情況在 DSP-Based 的解碼下可以簡單的利用軟體判斷的方式避開，但是 FPGA-Based 則不容易實現，因此為了避免這樣的情形發生，我們設計一個狀態機機制來根據 CORDIC 輸入向量的狀態做為計數依據，示意圖如下

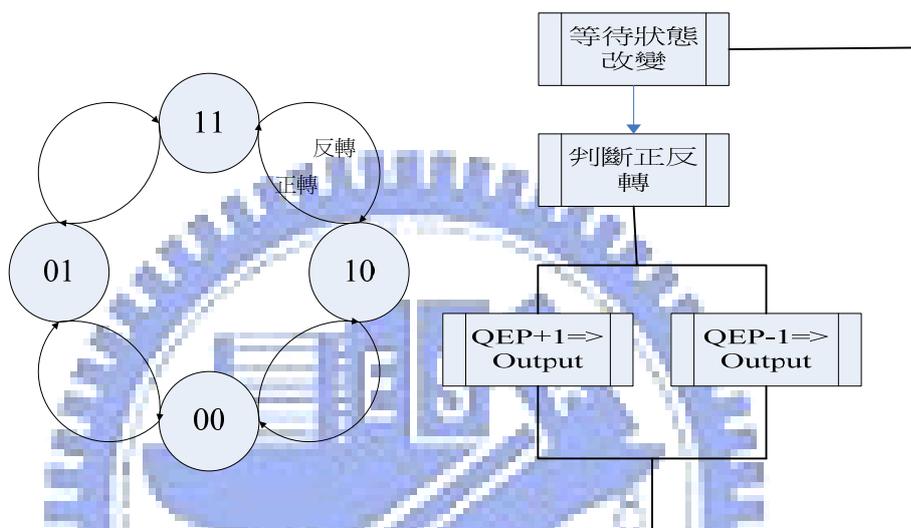


圖 3-15 狀態轉換機制示意圖

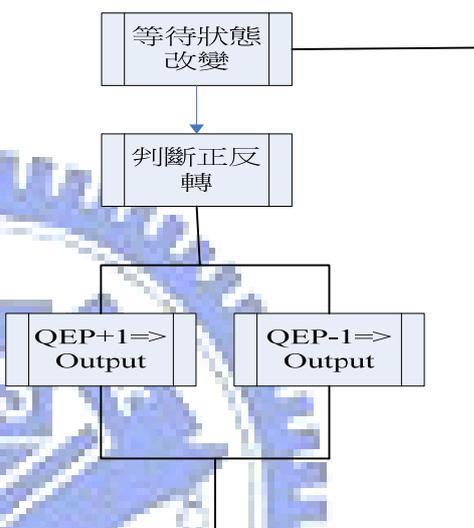


圖 3-16 錯位補償機制圖

在圖 3-15 中的狀態是以弦波編碼器兩相輸出經過 ADC 取樣後的正負號為狀態

$$state = sign(\sin) \& sign(\cos) \quad (3.7)$$

修正式 CORDIC 演算法是以兩輸入信號的正負號作為象限的判斷，同時象限轉換時也代表著 QEP 信號應該同時轉換，我們依此為基準來控制 QEP 信號的加減，而外部的 QEP 信號可作為內部 QEP 計數器的修正用。

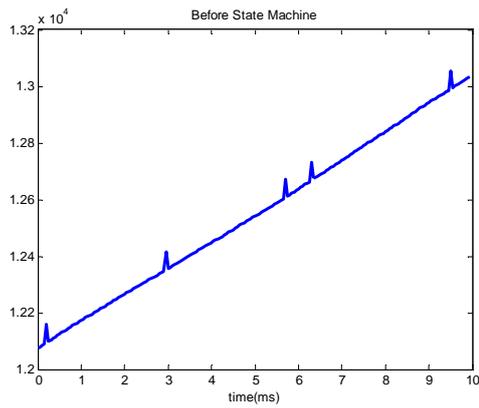


圖 3-17 狀態轉換機制補償前

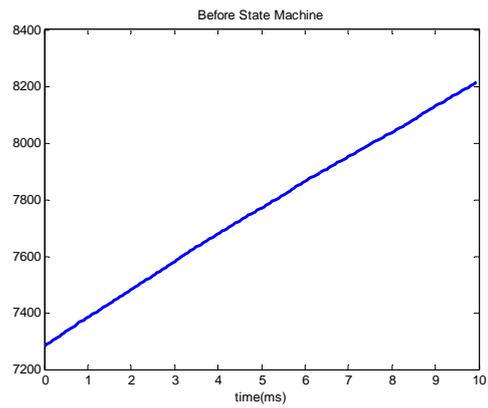
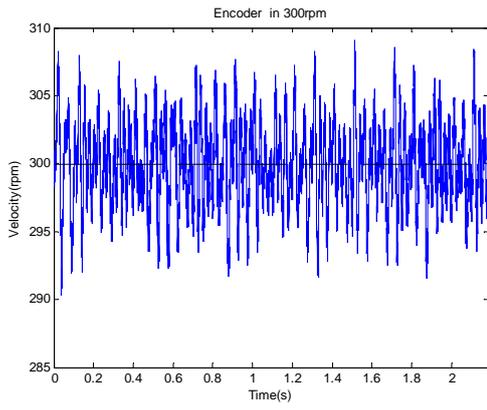


圖 3-18 狀態轉換機制補償後

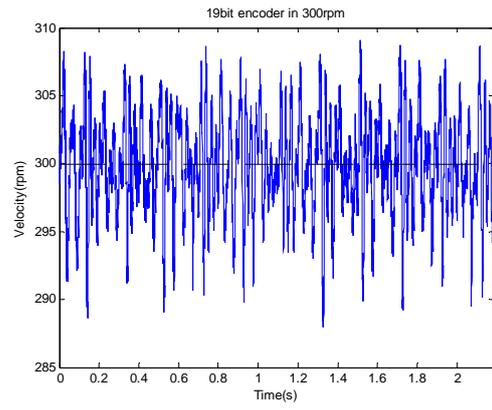
比較兩圖在經過狀態改善過後已經不再有錯位的問題，可以正確的解出位置。但在轉速超過 $1800rpm$ 的情況下會造成 QEP 轉態機制過慢而來不及跟上實際的 QEP 脈波數，意即每次 ADC 取樣間距會有兩次以上的象限變化，造成 pulse lose 發生，此時則轉換為高速狀態，只以實際 QEP 脈波數當作位置資訊，解析度僅有 13-bit，當轉速低於 $1800rpm$ 時則可使用高解析度 19-bit 輸出。

3.2.3 實驗結果

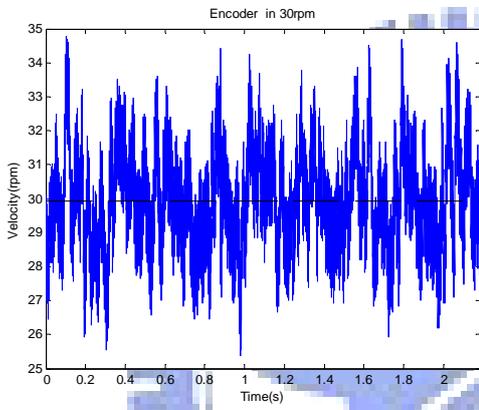
實驗的硬體基本上跟前一節的 DSP-Based 相同，唯一的差別是因為 FPGA-Based 的解碼速度較快，並且是獨立於控制器外的 FPGA 硬體實現，所以馬達控制器在控制過程中可以同時抓取 FPGA 的位置資訊，可以同時紀錄 19-bit 的解碼器與傳統 10000 PPR 編碼器之結果。



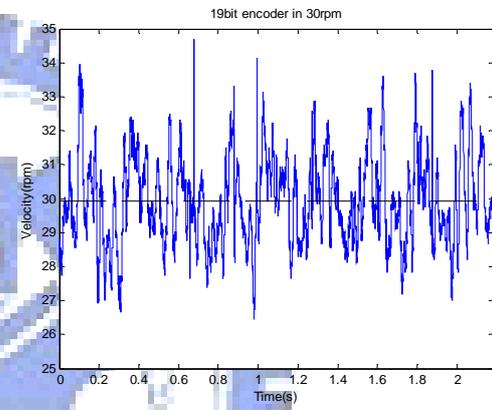
(a) 300rpm 時傳統編碼器



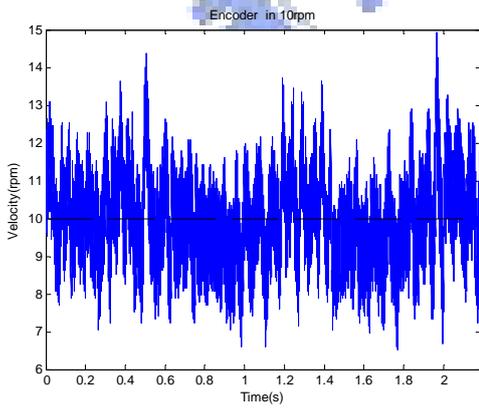
(b) 300rpm 時 FPGA-Based 19-bit 解碼



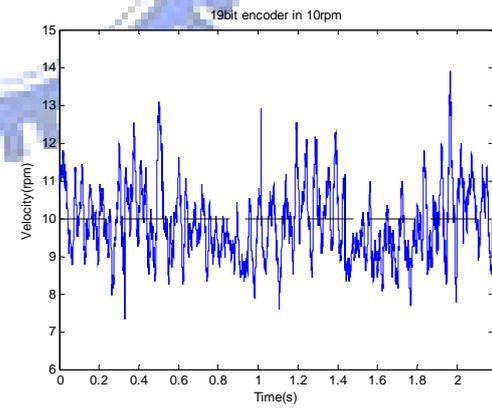
(c) 30rpm 時傳統編碼器



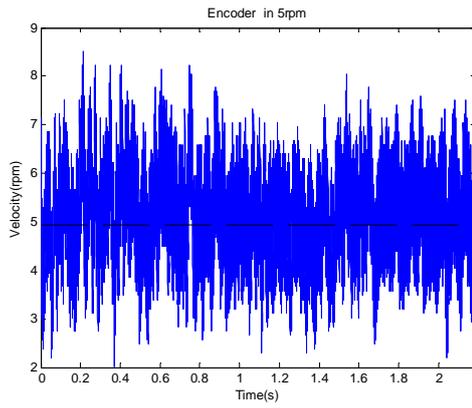
(d) 30rpm 時 FPGA-Based 19-bit 解碼



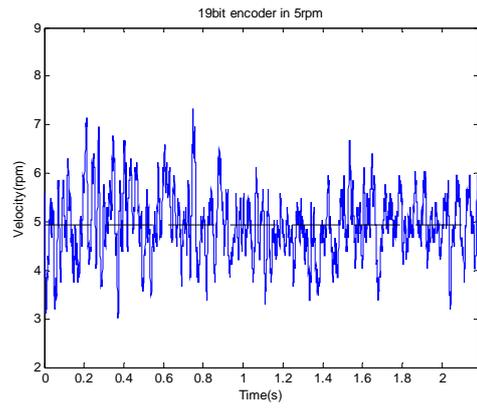
(e) 10rpm 時傳統編碼器



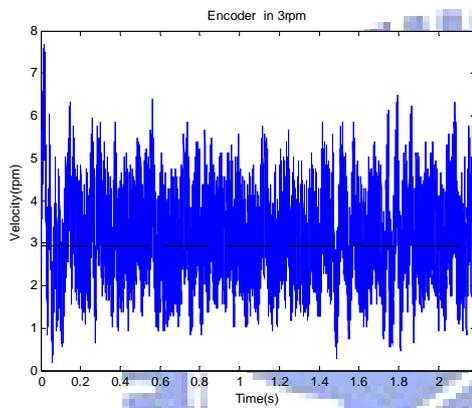
(f) 10rpm 時 FPGA-Based 19-bit 解碼



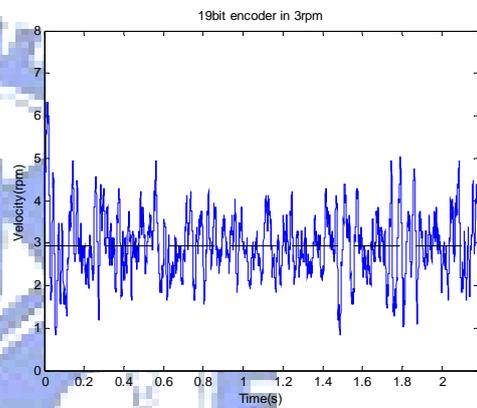
(g) 5rpm 時傳統編碼器



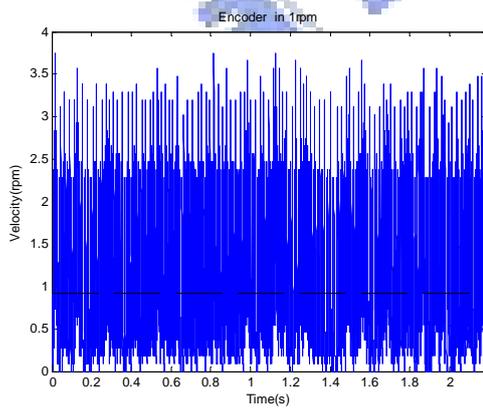
(h) 5rpm 時 FPGA-Based 19-bit 解碼



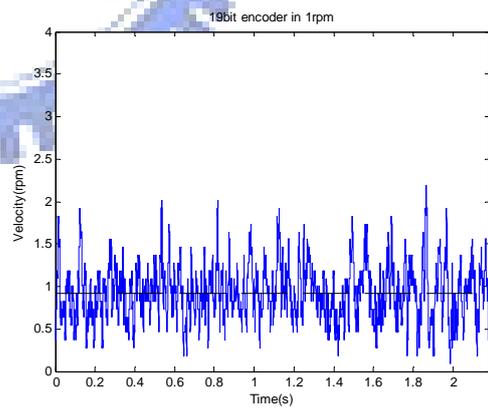
(i) 3rpm 時傳統編碼器



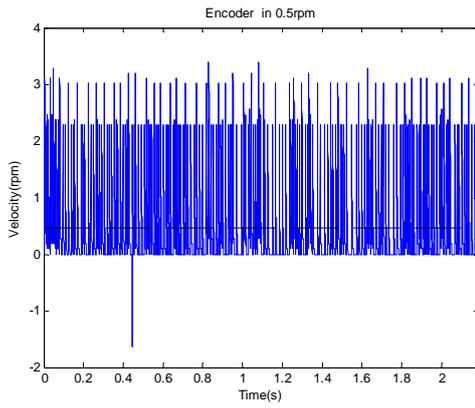
(j) 3rpm 時 FPGA-Based 19-bit 解碼



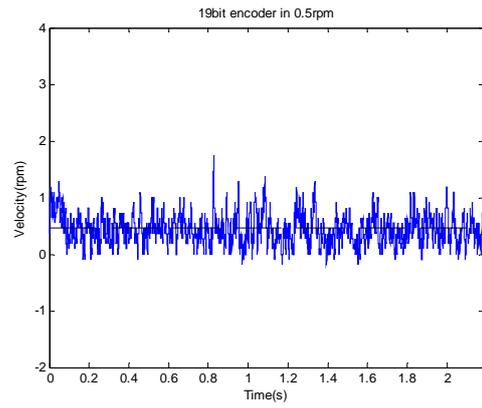
(k) 1rpm 時傳統編碼器



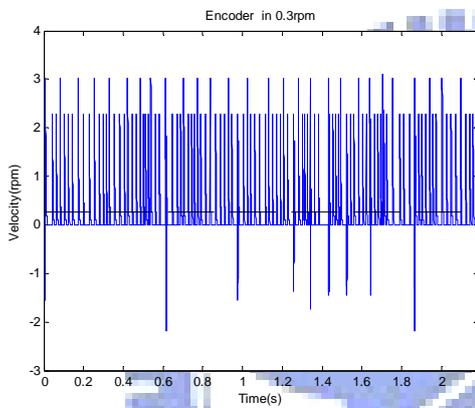
(l) 1rpm 時 FPGA-Based 19-bit 解碼



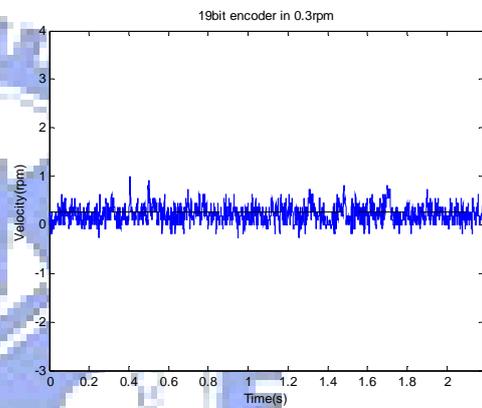
(m) 0.5rpm 時傳統編碼器



(n) 0.5rpm 時 FPGA-Based 19-bit 解碼



(o) 0.3rpm 時傳統編碼器



(p) 0.3rpm 時 FPGA-Based 19-bit 解碼

圖 3-19 FPGA-Based 解碼對照比較

在 300rpm 的情況下因為傳統編碼器的解析度已經足夠，所以不管是傳統編碼器或是弦波編碼器兩者無太大差異，但是在 45rpm 以下的範圍因為傳統編碼器解析度已經不足，已經使用到速度估測內插，而弦波編碼器還有足夠的解析度，所以明顯看出兩者間的差異，尤其在極低速的範圍內更是明顯見到弦波編碼器的回授信號明顯比傳統編碼器穩定。下表是上列各實驗結果的變異量及平均值比較

表 3-3 傳統編碼器與弦波編碼器變異量比較表

	300rpm	30rpm	10rpm	5rpm
Encoder	9.2786	2.1571	1.4148	1.0191
19bit encoder	13.3040	1.6668	0.8659	0.4574

傳統編碼器與弦波編碼器變異量比較表(續)

	3rpm	1rpm	0.5rpm	0.3rpm
Encoder	1.0984	0.6046	0.4393	0.3056
19bit encoder	0.5576	0.1024	0.0700	0.0308

表 3-4 傳統編碼器與弦波編碼器平均值比較表

	300rpm	30rpm	10rpm	5rpm
Encoder	299.9673	29.9401	9.8925	4.9600
19bit encoder	299.9408	29.9143	9.8666	4.9320

(b) 傳統編碼器與弦波編碼器平均值比較表(續)

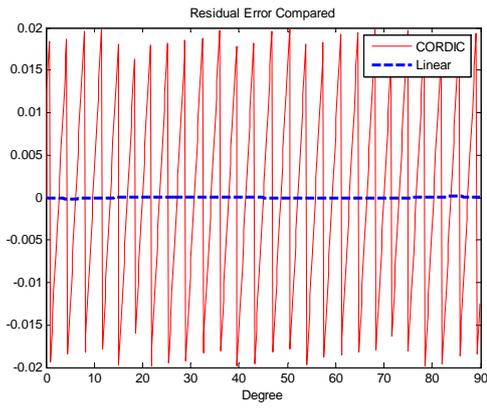
	3rpm	1rpm	0.5rpm	0.3rpm
Encoder	2.9757	0.9575	0.4715	0.2551
19bit encoder	2.9471	0.9315	0.4446	0.2193

總和上面四個表，在平均值方面傳統編碼器大約都比弦波編碼器略高 0.02rpm 左右，較接近速度命令，但是差異很小影響不大，但是在變異量方面弦波編碼器在低速範圍內則是遠遠勝過傳統編碼器，在 0.3rpm 下變異量甚至小了傳統編碼器 10 倍之多。

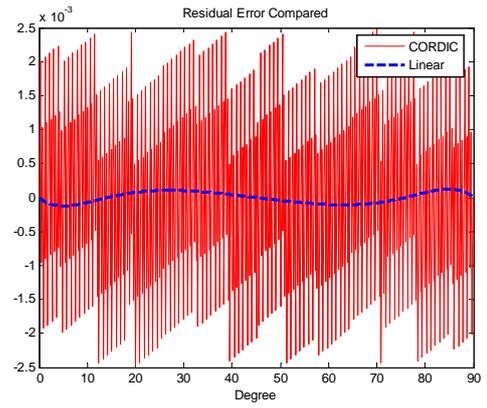
由實驗證實 FPGA-Based 的弦波編碼器解碼，確實能快速準確的解讀出目前位置，能夠明顯的增加位置回授的解析度，提供更穩定可靠的位置資訊。

3.3 線性化法與 QEP-CORDIC 之實現討論

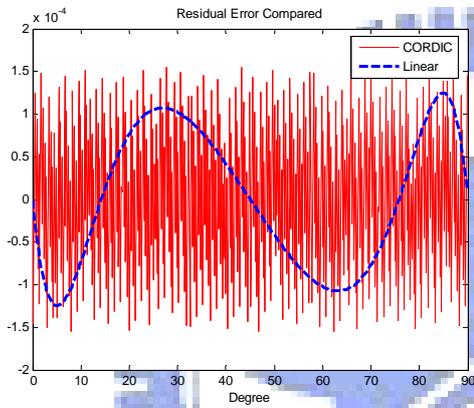
此節將對新提出使用的 QEP-CORDIC 演算法與原本所使用的線性化法做比較，並就模擬及實做兩方面進行分析。首先進行模擬，下圖 3-20 是我們分別進行兩種不同的算法，在不同的遞迴次數下模擬後所得到的誤差圖。



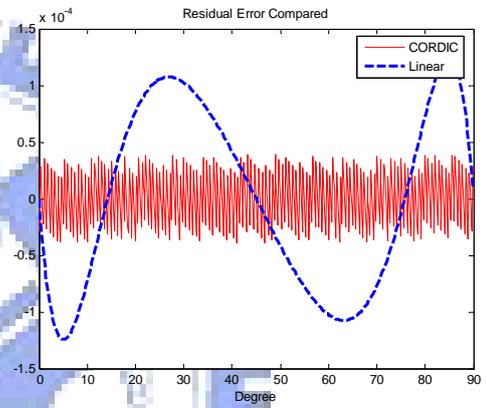
(a) 5 次



(b) 8 次



(c) 12 次

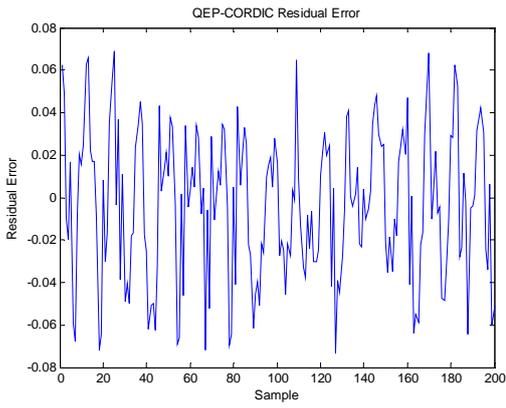


(d) 14 次

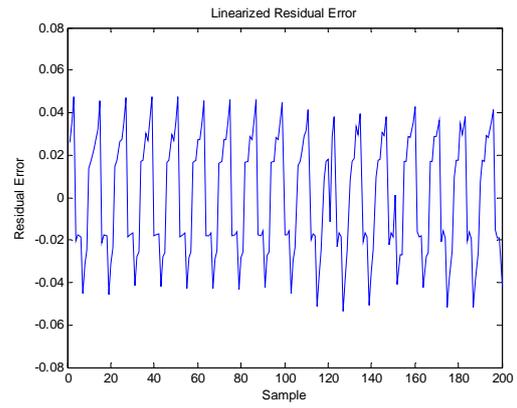
圖 3-20 線性化法與不同遞迴次數 CORDIC 模擬誤差

模擬顯示線性化法誤差大小並不會改變，而 QEP-CORDIC 演算法在遞迴次數 5 次及 8 次時所產生的誤差皆遠大於線性化法的誤差，即使遞迴次數增加到 12 次時誤差值仍大於線性化法，但是在遞迴 14 次時誤差僅剩線性化法的三分之一左右。

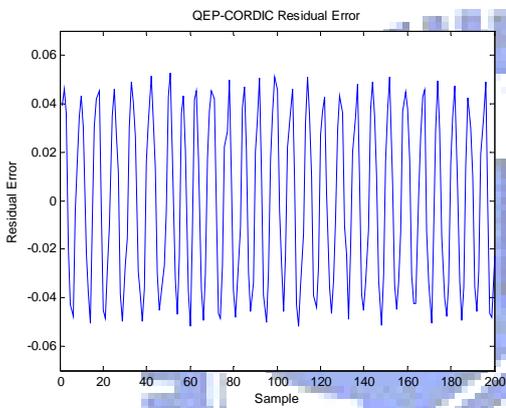
接下來以實驗比較兩者誤差，為求公平 QEP-CORDIC 演算法也是以 eZdsp 實驗版實現。因為 DSP F2812 的內建 ADC 只有 12-bit，所以最多只能做到 12 次遞迴運算，因此我們只實驗 5、8 及 12 次遞迴 QEP-CORDIC 運算，分別與線性化法做比較。



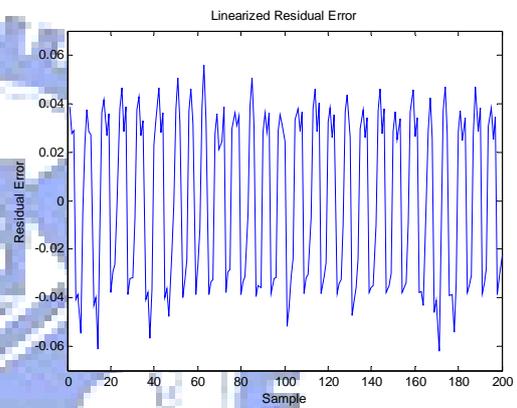
(a) CORDIC 遞迴次數 5 次



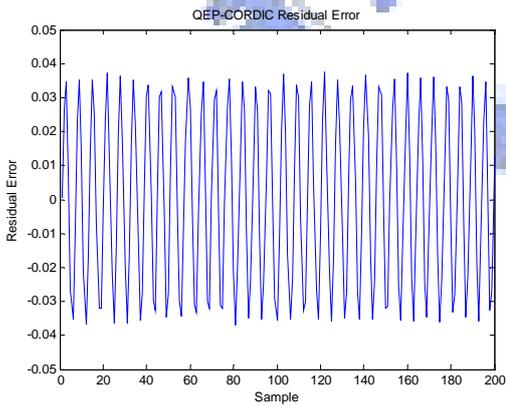
(b) 線性化法誤差



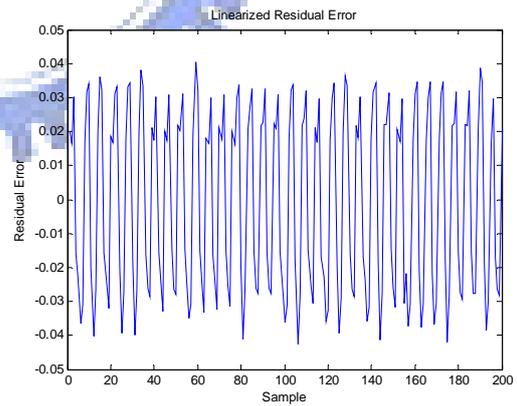
(c) CORDIC 遞迴次數 8 次



(d) 線性化法誤差



(e) CORDIC 遞迴次數 12 次



(f) 線性化法誤差

圖 3-21 兩種解碼法同時在 DSP 實現的誤差比較圖

結果顯示，在 5 次遞迴運算時 QEP-CORDIC 演算法誤差略大於線性化法，而在 8

次及 11 次遞迴運算時 QEP-CORDIC 演算法則與線性化法的誤差相當接近，下則表 3-5 是兩者間的比較

表 3-5 線性化法與 QEP-CORDIC 演算法實驗比較

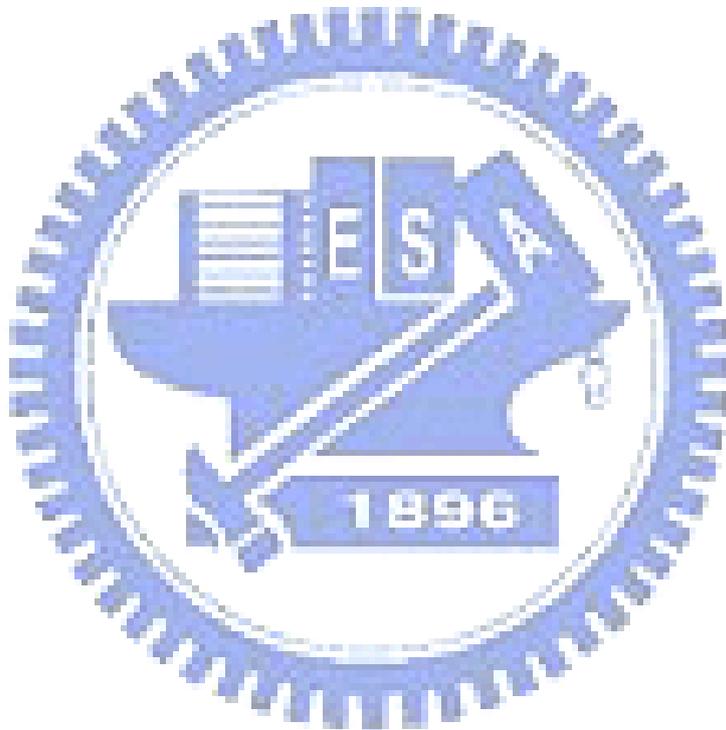
	最大誤差量	誤差變異量
線性化法	5.2687×10^{-2}	7.4188×10^{-4}
QEP-CORDIC(5)	7.3423×10^{-2}	1.1879×10^{-3}
QEP-CORDIC(8)	5.2771×10^{-2}	1.1235×10^{-3}
QEP-CORDIC(12)	3.7509×10^{-2}	6.4868×10^{-4}

由上面的模擬及實驗結果發現了相當奇怪的現象，原本模擬的時候在 12 次遞迴以下線性化法的誤差都小於 QEP-CORDIC 法，尤其在遞迴次數越少的時候越明顯，但實驗結果卻是 QEP-CORDIC 演算法在 5 次遞迴時比線性化法誤差略大，在 8 次兩者誤差已相當接近，在 12 次遞迴運算時反而 QEP-CORDIC 誤差已經小於線性化法了，其原因分析如下：

1. ADC 取樣精度未加以考慮：在模擬中都沒有考慮到 ADC 取樣後的精度，而實驗中受限於 ADC 取樣 12-bit 的精度，大大降低了線性化法的精度，但相對於 QEP-CORDIC 演算法來說，本身在模擬時就已經設定為 12 次的遞迴運算，間接考慮到 ADC 取樣的精度問題，所以 ADC 取樣精度對 QEP-CORDIC 法影響較小，而對線性化法則影響較大。
2. DSP 運算時有限位元之影響：線性化法中需要用到複雜的運算元，如：除法、乘法或是取絕對值，DSP 執行這些複雜的運算後都會因為有限位元的限制而忽略掉末端的位元資訊，而 QEP-CORDIC 演算法本身就是為了數位系統所發展的，並無使用到複雜的運算元，僅僅以位移及加法器就可以完成，所以在實做運算時誤差與模擬相差無不多。

除此之外執行速度也大有不同，以 DSP-Based 及 FPGA-Based 實際實現兩種不同的解碼技巧，DSP-Based 的線性化解碼因為需要用到許多乘法及除法，所以需要的時間比較多，大約需要 $75.88 \mu s$ ，所以最高處理速度約是 13.1 KHz 左右；相對的 FPGA-Based 的 QEP-CORDIC 解碼因為是專為數位硬體所設計的演算法，只需使用到位移及加減法即可，所以處理速度極快，以目前實現的 unrolled 架構來說，14 次的遞迴運算來說只需

要 20 個系統週期，以 133.33 MHz 的系統週期來說只需 150ns，約為 6.666 MHz，但實際實現上因為受限於外部 LTC1407A 串列傳輸 ADC 的傳輸限制，最快更新速度只能執行到 245 KHz 左右。



第四章 串列式編碼器在低速控制應用

4.1 低速估測

速度估測在低速控制中是影響控制效能最大的因素之一，本節所提出的速度估測法目的在於未來搭配高解析度串列式編碼器時，可以得到更精確的速度資訊。精確的速度回授除了經由硬體編碼器解析度的提升外，利用信號處理的方法提升回授信號的可信度、降低雜訊也同樣可以增加速度回授的解析度。本節提供三種不同的速度估測法，最後實際比較各種估測法在低速控制上所得到的結果。

4.1.1 差分法加低通濾波器

差分法是最常應用於伺服馬達控制的速度估測法，在一般轉速編碼器解析度足夠的情況下確實是個簡單而且有效的方法。其作法是取兩點之間的差分後再經過一個簡單的低通濾波器濾除雜訊。

定義目前位置 $\theta(k)$ 與前次位置 $\theta(k-1)$ 及本次速度差分值 $\omega(k)$ 關係為

$$\omega(k) = \theta(k) - \theta(k-1) \quad (4.1)$$

接著使用一個連續時間的一階低通濾波器來降低編碼器的雜訊影響，估測速度 $\hat{\omega}(k)$ 為

$$\frac{d\hat{\omega}}{dt} = \frac{1}{\tau_c}(\omega - \hat{\omega}) \quad (4.2)[13]$$

其中 $\tau_c = \frac{1}{2\pi \times f_c}$ 是低通濾波器的截止頻率常數， f_c 為截止頻率，然後將上式用反

向近似法推導，取樣週期為 T

$$\frac{\hat{\omega}(k) - \hat{\omega}(k-1)}{T} = \frac{\omega(k) - \hat{\omega}(k)}{\tau_c} \quad (4.3)$$

$$\Rightarrow \hat{\omega}(k) = \frac{T}{\tau_c + T} \omega(k) + \frac{\tau_c}{\tau_c + T} \hat{\omega}(k-1)$$

以下實驗資訊我們以記錄到的 19-bit 編碼器輸出為實際速度，然後故意將解析度調降為 16-bit，也就是將後三位元無條件捨去，以低解析度 16-bit 的位置資訊做差分法運算，再與 19-bit 直接差分不經過低通濾波器做比較，用來做速度估測的低解析度位置資訊與高解析度位置資訊比較圖

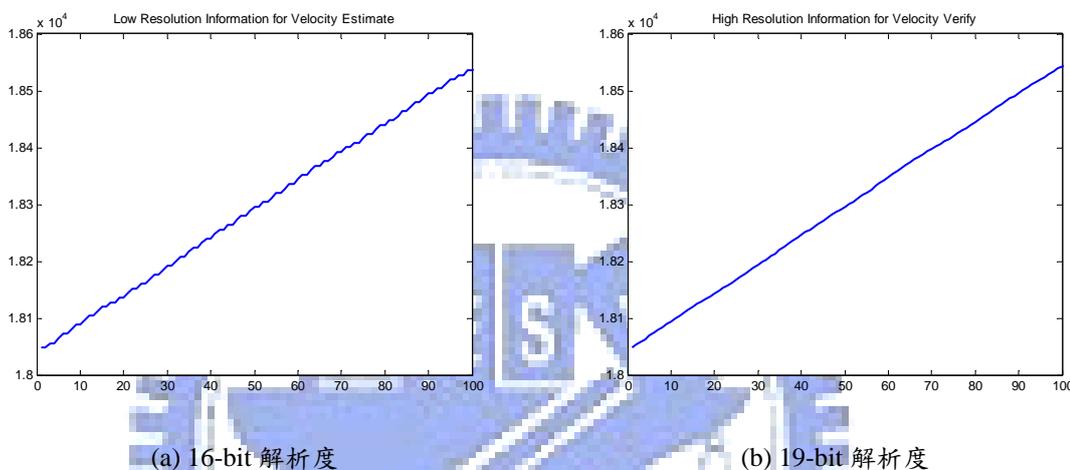


圖 4-1 低解析度位置資訊與高解析度位置資訊比較圖

由低解析度位置資訊中看出在此轉速下 16-bit 已經不足以應付，所以會有些取樣時間位置是沒有變化的，但實際上由右邊高解析度 19-bit 的圖中清楚看出編碼器是有持續且穩定的運轉，用這樣的數據做模擬比較是最容易可以看出效果的。

在差分法的實驗中我們將截止頻率設在 900 Hz，經過軟體進算後比較下兩圖

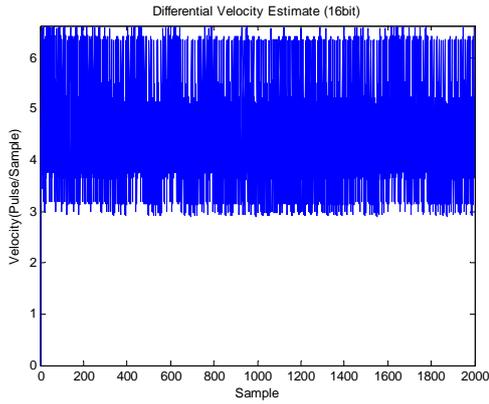


圖 4-2 16-bit 差分加低通濾波估測速度

$$\sigma = 1.5625 \quad \mu = 4.6855$$

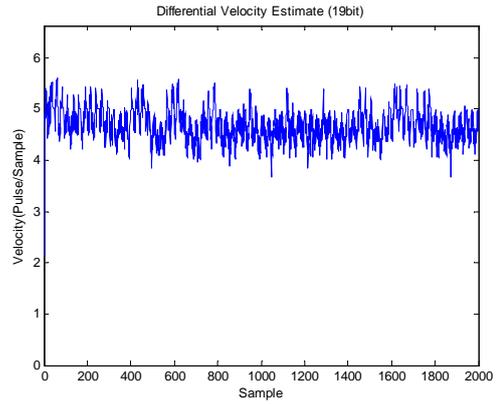


圖 4-3 19-bit 高解析度估測速度

$$\sigma = 0.1157 \quad \mu = 4.6870$$

由上兩圖比較差分法所得到的結果接近實際的速度，實際速度大約是在 4~5 間，而估測結果則是在 3~6 間，雖然震盪範圍略大，變異量約是 1.5625，但是平均值為 4.6855 與高解析度資訊平均值 4.6870 相當接近。

4.1.2 最小平方近似法 [14]

最常使用到的差分法之外，最小平方近似法也是常被用來作為速度估測的方法。最小平方法是假設位置為時間的多項展開式，在時刻 t_k 時所估測的位置 \hat{x}_k 如下：

$$\hat{x}_k = c_0 + c_1 t_k + c_2 t_k^2 + \dots + c_N t_k^N \quad (4.4)$$

N 為我們所設定的階數，這個階數大大影響到速度估測的準確性，階數的選擇可依照命令的形式來設定，或是使用主軸成分分析(principle component analysis) [15]來決定階數。再以前 M 點的資料作為最小平方法的參數估測來源，可得下列式：

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} = \begin{bmatrix} 1 & t_1 & \dots & t_1^N \\ 1 & t_2 & \dots & t_2^N \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_M & \dots & t_M^N \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} \Rightarrow X = AC \quad (4.5)$$

由於我們的取樣時間都是固定間距，故可建立 $t_2 = 2t_1$ 、 $t_3 = 3t_1$... 等等的推論，同時

假設時刻 $t_1 = 1$ ， A 矩陣便可簡化為

$$A \Rightarrow \begin{bmatrix} 1 & 1 & \cdots & 1^N \\ 1 & 2 & \cdots & 2^N \\ \vdots & \vdots & \ddots & \vdots \\ 1 & M & \cdots & M^N \end{bmatrix} \quad (4.6)$$

如此便可利用最小平方法公式求出係數 C

$$C = (A^T A)^{-1} A^T X \quad (4.7)$$

而在下一時刻的位置便可用 $M+1$ 的時間參數來估測

$$\hat{x}_{M+1} = c_0 + c_1 t_{M+1} + \cdots + c_N t_{M+1}^N \quad (4.8)$$

而速度估測便是利用速度為位置對時間微分的這項特性

$$\begin{aligned} \hat{v}_{M+1} &= \frac{d\hat{x}_{M+1}}{dt_{M+1}} = c_1 + 2c_2 t_{M+1} + 3c_3 t_{M+1}^2 + \cdots + Nc_N t_{M+1}^{N-1} \\ &\Rightarrow \begin{bmatrix} 0 & 1 & 2(M+1) & \cdots & N(M+1)^{N-1} \end{bmatrix} \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_N \end{bmatrix}^T \\ &\Rightarrow q_{M+1} \cdot C = q_{M+1} \cdot (A^T A)^{-1} A^T X = H_{M+1} X \end{aligned} \quad (4.9)$$

因為階數 N 及參數估測來源點數 M 都是已知，故 H_{M+1} 便可先由離線算出，再將每次所取得的前 M 點位置代入便可估測現在時刻的速度，實做上猶如一個濾波器般，僅以差分方程式便可實現。以下是不同的 M 及 N 值模擬所得的誤差，在固定低速的條件下 $M=4$ 、 $N=1$ 有最好的效果。

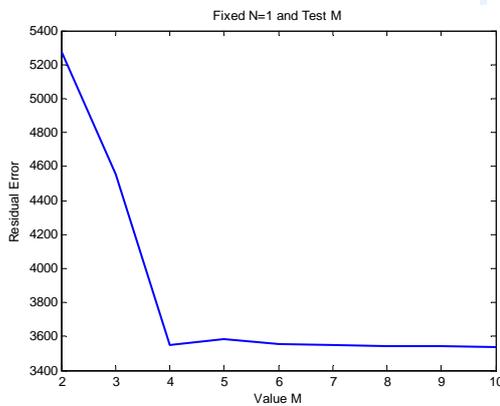


圖 4-4 不同參數 M 下的累積誤差圖

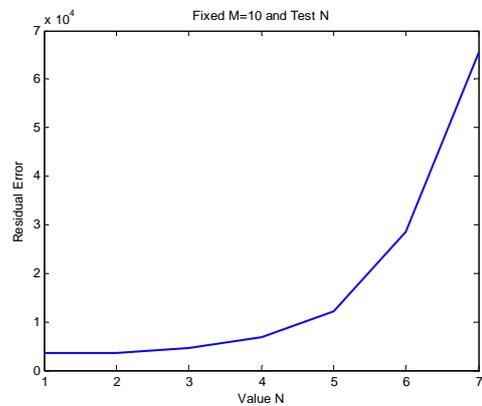


圖 4-5 不同參數 N 下的累積誤差圖

使用的參數為 $M = 4$ 、 $N = 1$ ，將階數 N 設定為 1 時有最小誤差的原因在於實驗是以固定低速的命令為主，所以速度表現應該為單純 1 階的線性方程式，故誤差最小。

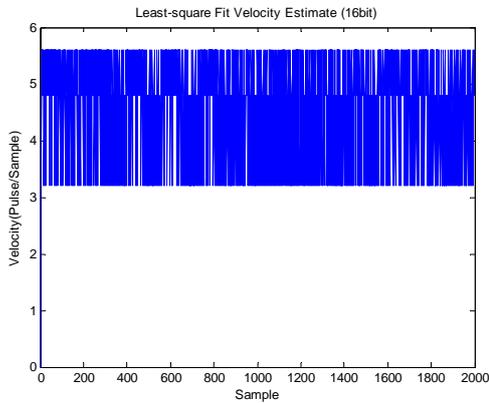


圖 4-6 16-bit 最小平方近似法估測速度

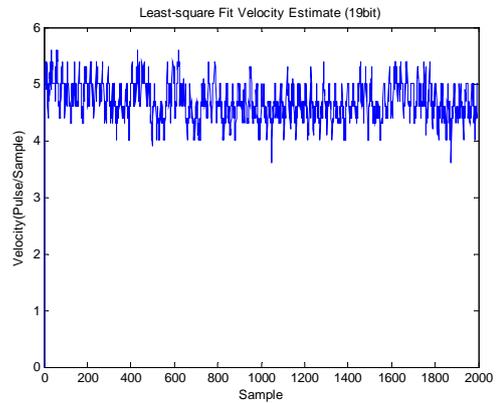


圖 4-7 19-bit 高解析度估測速度

$$\sigma = 0.8578 \quad \mu = 4.6851$$

$$\sigma = 0.1365 \quad \mu = 4.6853$$

利用此組參數進行運算，結果如上圖 4-6，整體輸出範圍約在 3.5~5.5 之間，實際上平均值約為 4.6851 與右邊的高解析度速度資訊所得的 4.6853 是非常相近的，而變異量約為 0.8578，比 900 Hz 時的差分法還要小。

4.1.3 卡曼濾波器 [16]

卡曼濾波器是在 1960 年時由 Rudolf Emil Kalman 所提出的理論，是一種最佳化遞迴資料處理演算法(optimal recursive data processing algorithm)，利用遞迴運算處理離散時間資料的線性濾波問題，由於它的在數位系統有許多的優點，便立刻受到廣泛的研究及應用。

卡曼濾波器利用一組數值化方程式有效遞迴的估測系統狀態，使得累積平方誤差最小化，不但可以估測過去、現在甚至是未來的系統狀態，除此之外在系統模型不夠準確的狀況下更能展現其作用。

假設系統離散狀態方程式為

$$\begin{aligned}x_k &= Ax_{k-1} + Bu_{k-1} + w_{k-1} \\z_k &= Hx_k + v_k\end{aligned}\quad (4.10)$$

其中比較特殊的地方在於引入 w 及 v 兩個參數，分別代表處理雜訊及量測雜訊，假設兩者互相獨立且為常態分布

$$\begin{aligned}p(w) &\sim N(0, Q) \\p(v) &\sim N(0, R)\end{aligned}\quad (4.11)$$

Q 代表處理雜訊的自變量 ($Q = ww^T$)， R 代表量測雜訊的自變量 ($R = vv^T$)，兩者可能隨著時間改變，但在此我們為了方便計算假設他們為常數。通常在實現上我們僅能取得測量雜訊變異量 v ，這個變異量可離線先行分析，但對於處理雜訊變異量 w 則是無從得知，仔細分析 v 與 w 的關係

$$d = \frac{w}{v}\quad (4.12)$$

當 $d \gg 1$ 時，表示處理雜訊 w 相對大於測量雜訊 v ，下次的狀態估測則以測量到的值 z_k 為較大的比重，反之當 $d \ll 1$ 時表示處理雜訊相對於測量雜訊小的多，在做下次狀態估測時將以前次狀態為基準。 d 值影響到卡曼濾波器的收斂速度及暫態響應。

定義時刻 k 的前次測量誤差 e_k^- 及現在測量誤差 e_k 為

$$\begin{aligned}e_k^- &= x_k - \hat{x}_{k-1} \\e_k &= x_k - \hat{x}_k\end{aligned}\quad (4.13)$$

在卡曼濾波器中假設此刻估測狀態 \hat{x}_k 是由前次估測狀態 \hat{x}_{k-1} 及實際測量值 z_k 與預測輸出 $H\hat{x}_{k-1}$ 的相差量的線性組合，其中 $(z_k - H\hat{x}_{k-1})$ 稱為更新項，時刻 k 的估測狀態如下式

$$\hat{x}_k = \hat{x}_{k-1} + K(z_k - H\hat{x}_{k-1})\quad (4.14)$$

K 是一個增益矩陣，利用前次的誤差自變量 $P_k^- = E[e_k^- e_k^{-T}]$ 及量測雜訊自變量 R 使本次誤差自變量 $P_k = E[e_k e_k^T]$ 達到最小化，詳細的推理在此略過，在時刻 k 的時候最佳 K 值如下

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}\quad (4.15)$$

綜合上述的推論推理出下列五條卡曼濾波器的遞迴公式

$$\begin{aligned}
 \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} \\
 P_k^- &= AP_{k-1}A^T + Q \\
 K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\
 \hat{x}_k &= \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \\
 P_k &= (I - K_k H)P_k^-
 \end{aligned} \tag{4.16}$$

其中各維度為 $A \in R^{n \times n}$ 、 $B \in R^{n \times 1}$ 、 $H \in R^{m \times m}$ 、 $x_k \in R^{n \times 1}$ 、 $z_k \in R^{m \times 1}$ 、 $w \in R^{n \times 1}$ 、 $v \in R^{m \times 1}$ 、

$Q \in R^{n \times n}$ 、 $R \in R^{m \times m}$ 、 $P_k \in R^{n \times n}$ 及 $K_k \in R^{n \times m}$ ，需要設定的參數有 x_0 、 P_0 、 w 及 v 。

首先進行一維速度估測的模擬，模擬中是將低解析度位置資訊先經過差分算出速度後，在經過卡曼濾波器的處理，因此作用有點類似一個濾波器。將系統模型假設為

$$\begin{aligned}
 x_k &= x_{k-1} + w \\
 z_k &= x_k + v
 \end{aligned} \tag{4.17}$$

各詳細的參數設定為 $x_0 = 0$ 、 $P_0 = 1$ 、 $w = 0.01$ 及 $v = \sqrt{0.03}$ ，其中 x_0 及 P_0 的決定影響僅前幾個運算週期的結果，隨著運算次數的增加，將逐漸被往後的更新項所主宰決定。

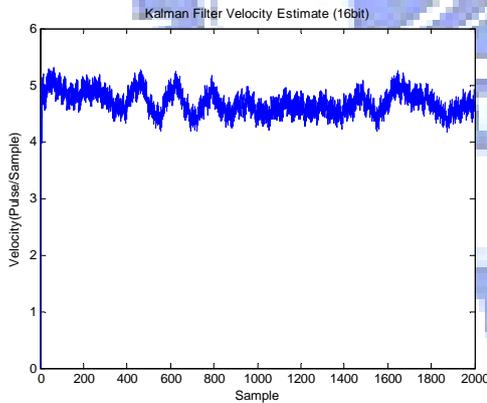


圖 4-8 16-bit 卡曼濾波器估測速度

$$\sigma = 0.0616 \quad \mu = 4.6872$$

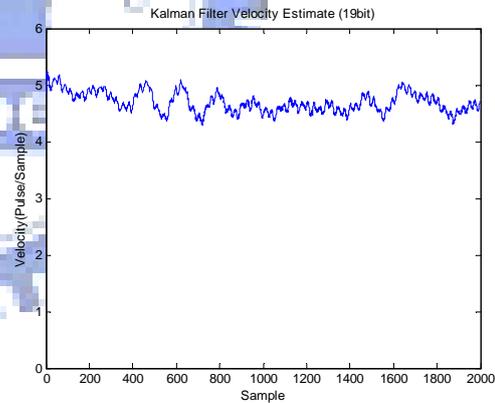


圖 4-9 19-bit 高解析估測速度

$$\sigma = 0.0319 \quad \mu = 4.6956$$

由上兩實驗圖比較得知，卡曼濾波器的估測結果相當優異，雖然我們初始狀態設定在 0，但是很快的在幾個運算週期內就到達穩態的狀態，估測結果的平均值為 4.6872 與實際高解析度的平均值 4.6956 也很接近，但是變異量僅有 0.0616，與前兩種速度估測法

相比減少十倍以上。

4.1.4 各估測法模擬比較

整合上面三種估測法結果做比較，同樣的一筆 16-bit 位置資訊經過三種不同的速度估測法得到不同的結果，將三張圖與 19-bit 差分圖做比較

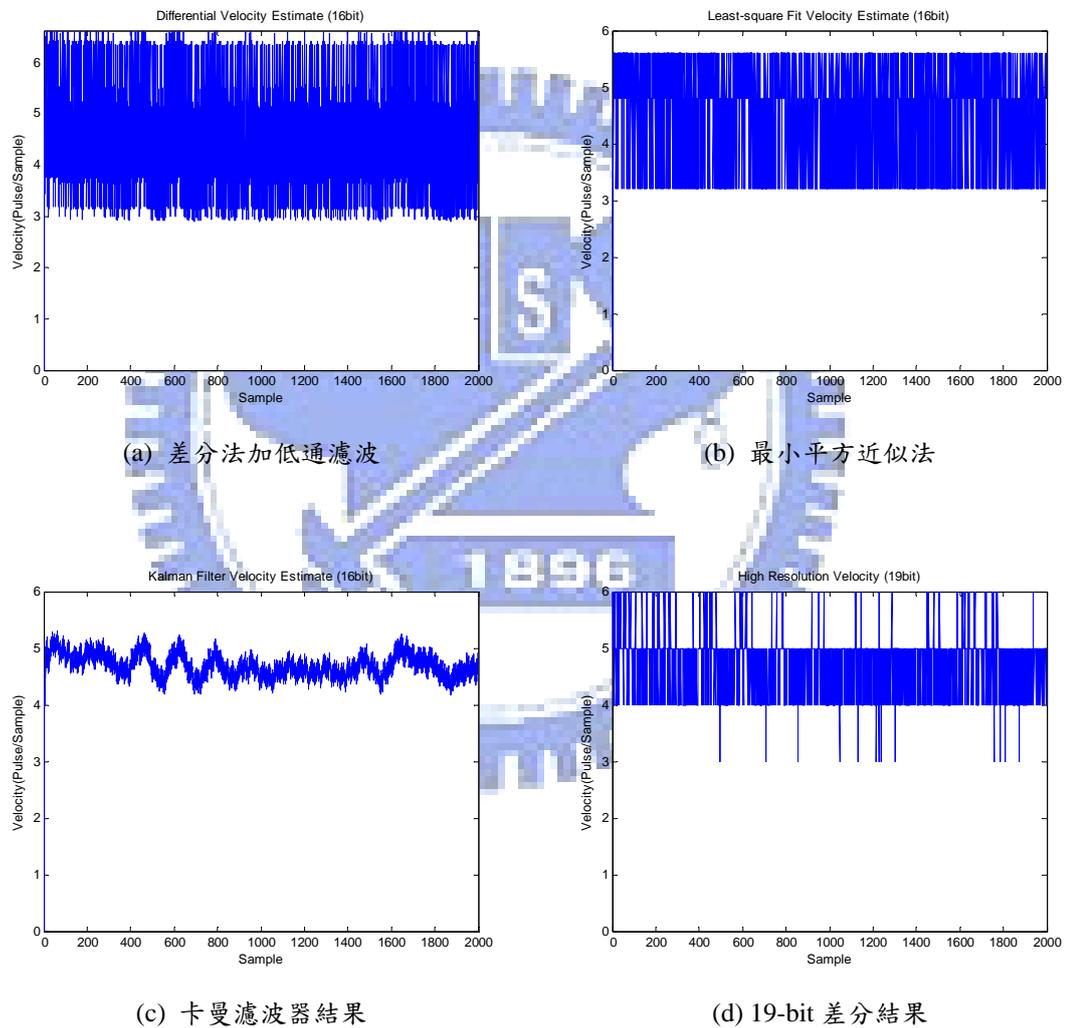


圖 4-10 不同估測法結果與實際差分速度圖(約 11rpm)

表 4-1 各估測法變異量與平均值比較

	差分+低通	最小平方近似	卡曼濾波器	19-bit 差分
變異量	1.5625	0.8578	0.0616	0.3102
平均值	4.6855	4.6851	4.6872	4.6855

由上比較表得知三種估測法的平均值都與高解析度所得結果差不多，但在變異量方面則是卡曼濾波器表現最好，因此推論卡曼濾波器在低速命令下應該是最適合做為速度估測用。

4.2 各估測法實驗比較

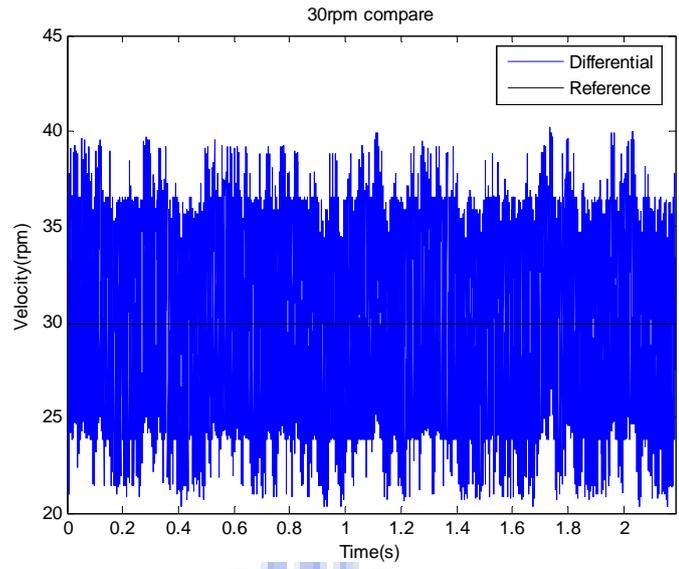
實驗中以傳統的 10000 PPR 光學編碼器為位置感測器，目的在於使用較低解析度的編碼器方便看出各種速度估測法的效果及差異，若使用高解析度弦波編碼器可能因為其優異的位置回授精度，而不易看出速度估測法的效果。

首先判斷 10000 PPR 的傳統編碼器在使用上的極限及速度估測範圍，目前的實驗的取樣頻率為 7.5KHz

$$\frac{7500 \times 60}{10000} = 45rpm \quad (4.18)$$

也就是說在速度低於 45rpm 的情況下每次的取樣週期將可能不再有位置變化，此時速度估測法才有其明顯效果及意義，實驗中將速度命令分別設定在 30、10 及 3rpm 三種不同的速度，實際測試這三種速度估測法對伺服馬達的速度控制效能。

首先是差分法，差分法後所使用的低通濾波器設定在 900 Hz，實際速度回授結果如下



(a) 30rpm

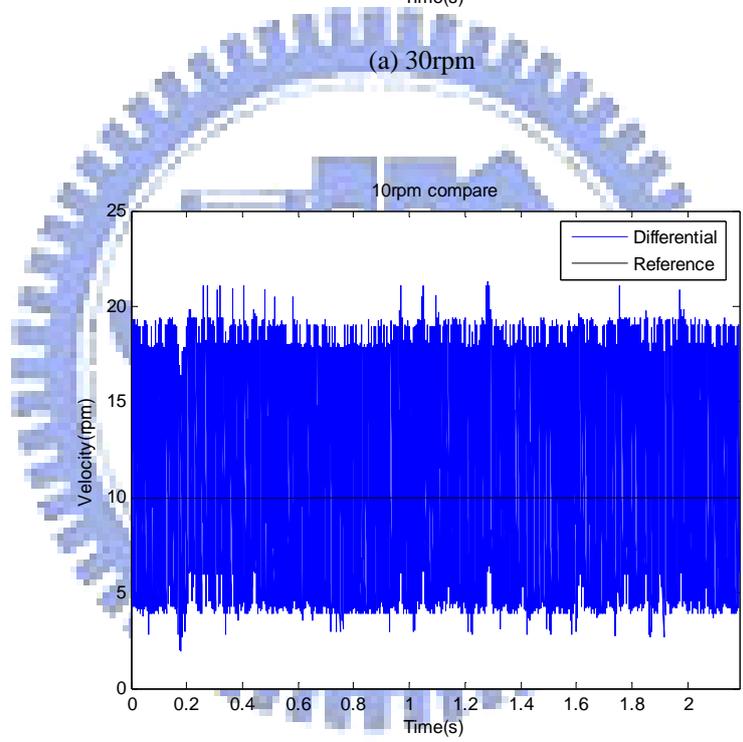
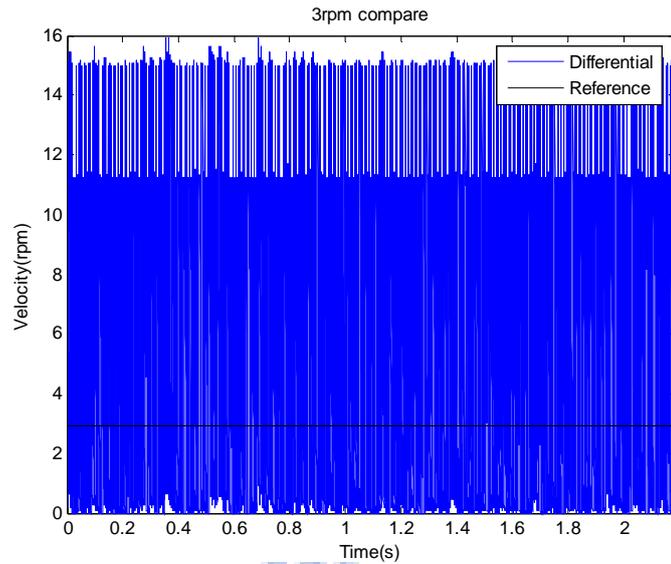


圖 (b) 10rpm

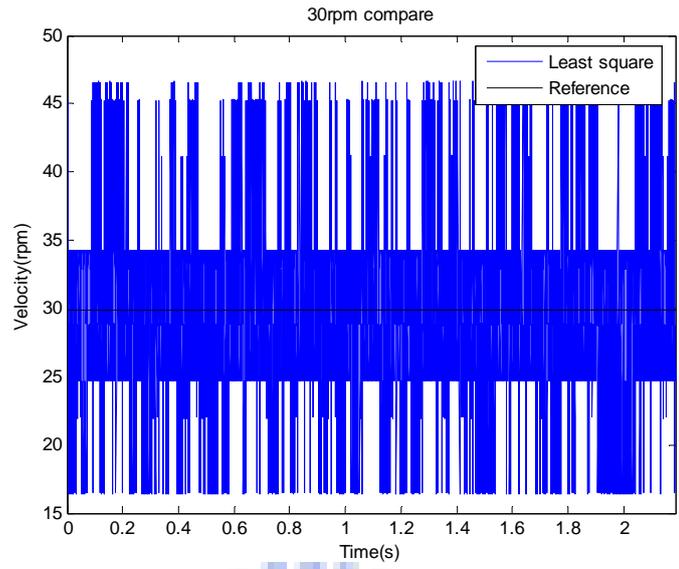


(c) 3rpm

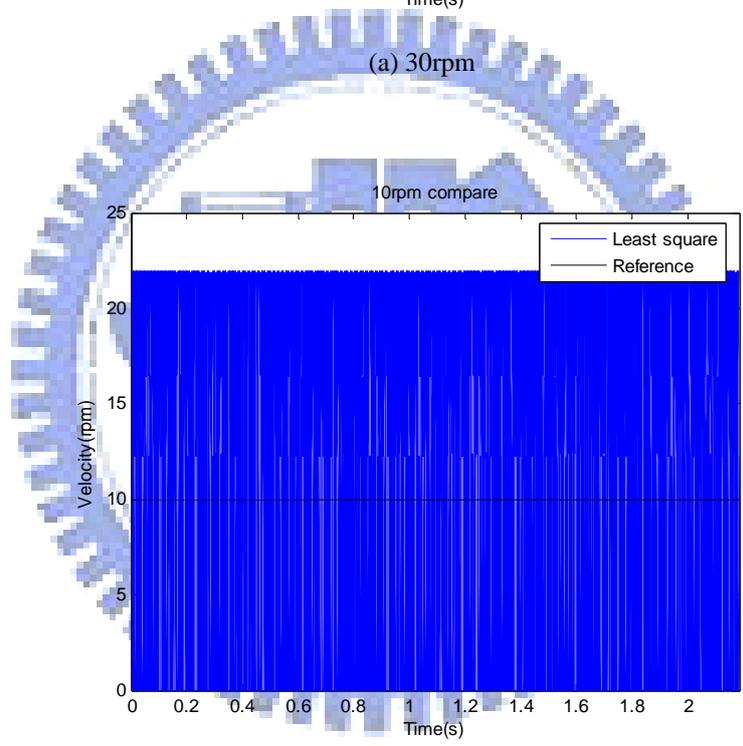
圖 4-11 差分加低通濾波控制結果

由上面三個實際的速度估測及控制結果顯示，在 $30rpm$ 及 $10rpm$ 的速度命令下速度控制的誤差量相對於速度命令都在一定範圍內，速度的平均值也與命令相當接近，值得注意的是在 $3rpm$ 的情況下速度估測有時會發生 0 速度的情形，表示這個估測法似乎已經不適用這樣的低速範圍，但是平均值仍然是與速度命令很接近。

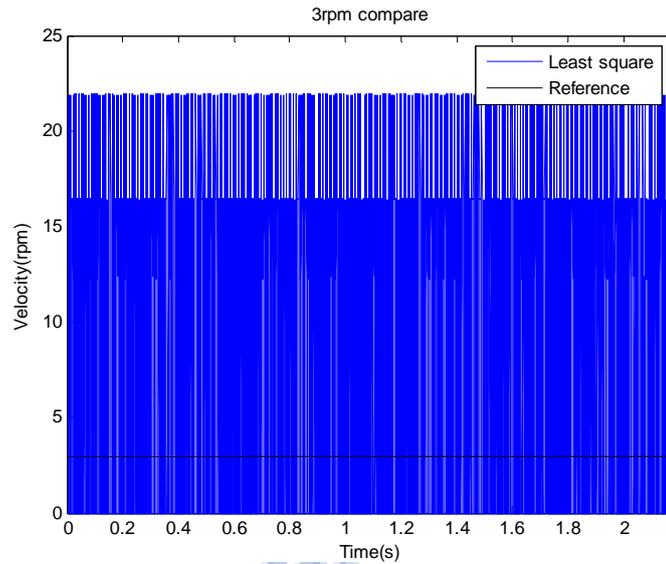
第二種速度估測法，最小平方近似法的實驗使用模擬時所得的最佳參數，階數設定為 1 ，點數設定為 4 ，結果如下



(a) 30rpm



(b) 10rpm

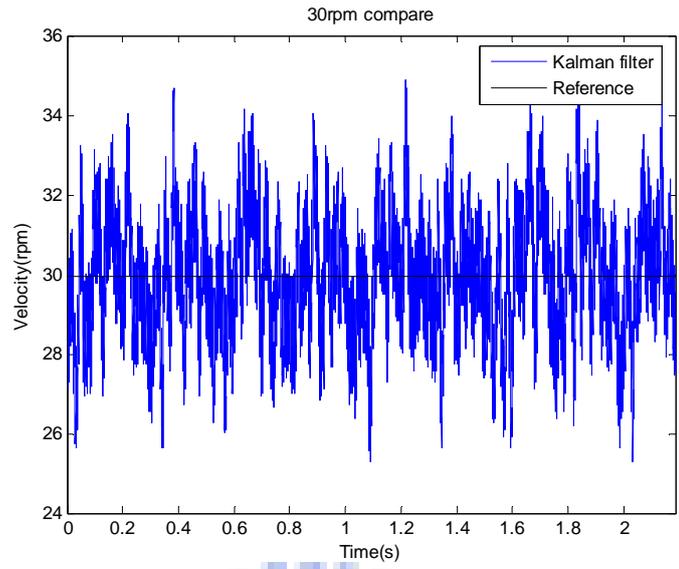


(c) 3rpm

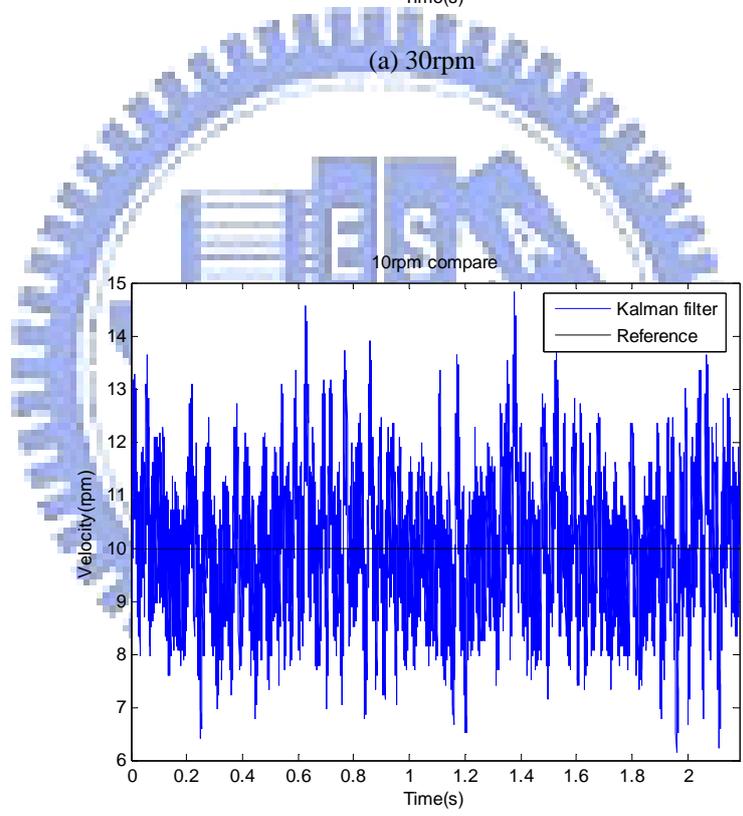
圖 4-12 最小平方近似法控制結果

由上面三種不同的實驗結果來看，最小平方近似估測法在低速區域時並沒有表現的比前述的差分好，在 $10rpm$ 與 $3rpm$ 的情況下速度估測都已經會發生 0 速度的情形，表示已經不適用這樣的低速度，另外速度漣波無論是在 $10rpm$ 或是 $3rpm$ 下都會高達 $20rpm$ 左右，沒有明顯下降的趨勢，說明了在更低速的區域下似乎也無法再得到更好的結果了。而在 $30rpm$ 的情況下漣波還是相當高，但是三種速度下的平均值都與速度命令相近。此法的特點在於少量的計算及簡單的實現，但對於低速領域的速度估測似乎並不是特別適用。

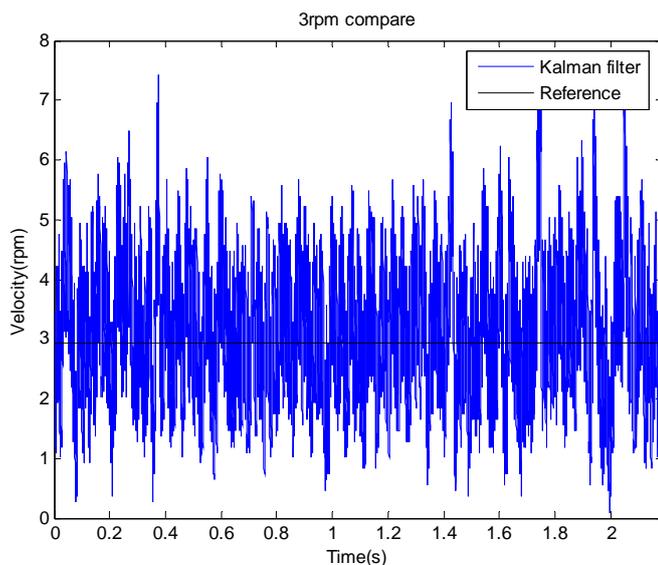
最後實驗的是卡曼濾波器，實驗的參數一樣使用模擬時所用的參數及模型，唯一的改變是將初始狀態 x_0 由原本的 0 改為目前第一筆所給的速度命令，結果如下



(a) 30rpm



(b) 10rpm



(c) 3rpm

圖 4-13 卡曼濾波器控制結果

上面三種不同速度下的估測結果顯示卡曼濾波器有最佳的速度估測能力，不只是在 30rpm 或 10rpm 下能有效明顯的降低漣波，就連 3rpm 的速度下都能明顯的降低速度漣波，而且沒有前兩種估測法所會發生的 0 速度情形，表示至少在 3rpm 的情況下卡曼濾波器仍然有一定的效果。

將上列實驗的結果列表比較

表 4-2 不同速度估測法下的變異量

	30rpm	10rpm	3rpm
差分法+低通濾波	18.8865	17.8261	13.3988
最小平方近似法	21.9879	55.6257	37.5272
卡曼濾波器	2.2705	1.4765	1.2293

表 4-3 不同速度估測法下的平均值

	30rpm	10rpm	3rpm
差分法+低通濾波	29.9065	9.9684	2.9508
最小平方近似法	29.9721	9.9495	2.9926
卡曼濾波器	29.9731	9.9459	2.9667

比較上面兩個表格，三種估測法的速度平均值都相當接近，並沒有特定方法有比較好的效果，但是在變異量方面明顯的卡曼濾波器有最好的效果，也就是說卡曼濾波器的

估測結果最穩定，雜訊較小，速度估測結果較可靠。

4.3 低速控制

控制器是影響低速控制效能的另一個主要因素，本節的目的在於利用不同的控制器比較低速下的控制效能，並且討論在 CNC 機台上加上負載之後的低速控制表現，以後搭配上 19-bit 串列式編碼器後可以更加提升低速控制效能。在前兩小段控制器之比較是使用 400 W 裝載 10000 PPR 傳統編碼器的伺服馬達，4.3.3 的 CNC 機台測試則是以 400 W 裝載 17-bit 高解析度串列式編碼器的伺服馬達進行測試。

4.3.1 PI 控制器

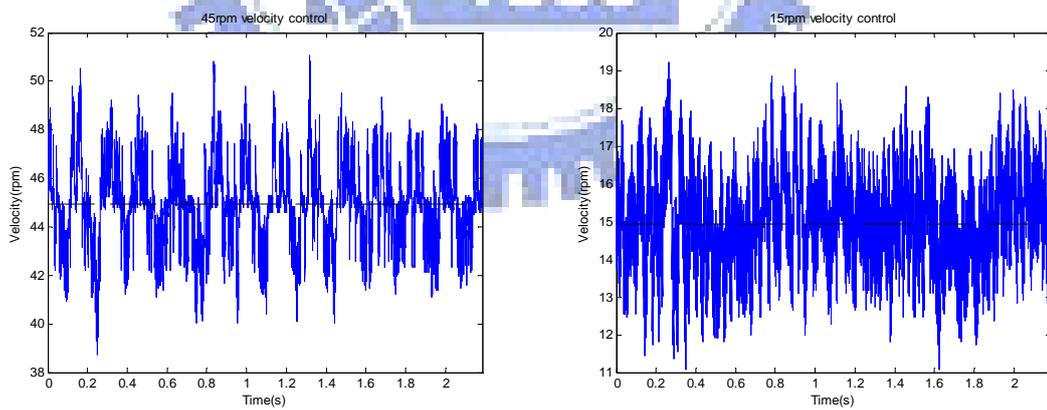
將 4.1.3 節中卡曼濾波器應用於伺服馬達控制之中，速度迴路控制以傳統 PI 控制器做實驗，調整出一組適合低速控制的參數，並比較調整前後的差異。





圖 4-14 伺服馬達控制系統實體圖

實驗中將由速度命令端放入不同的速度命令，偏重於 45rpm 以下之低速範圍，比較穩態時的漣波大小及變異量。以下實驗是在 $K_p = 2$ 、 $K_i = 0.0103$ 下所做的實驗



(a) 45rpm

(b) 15rpm

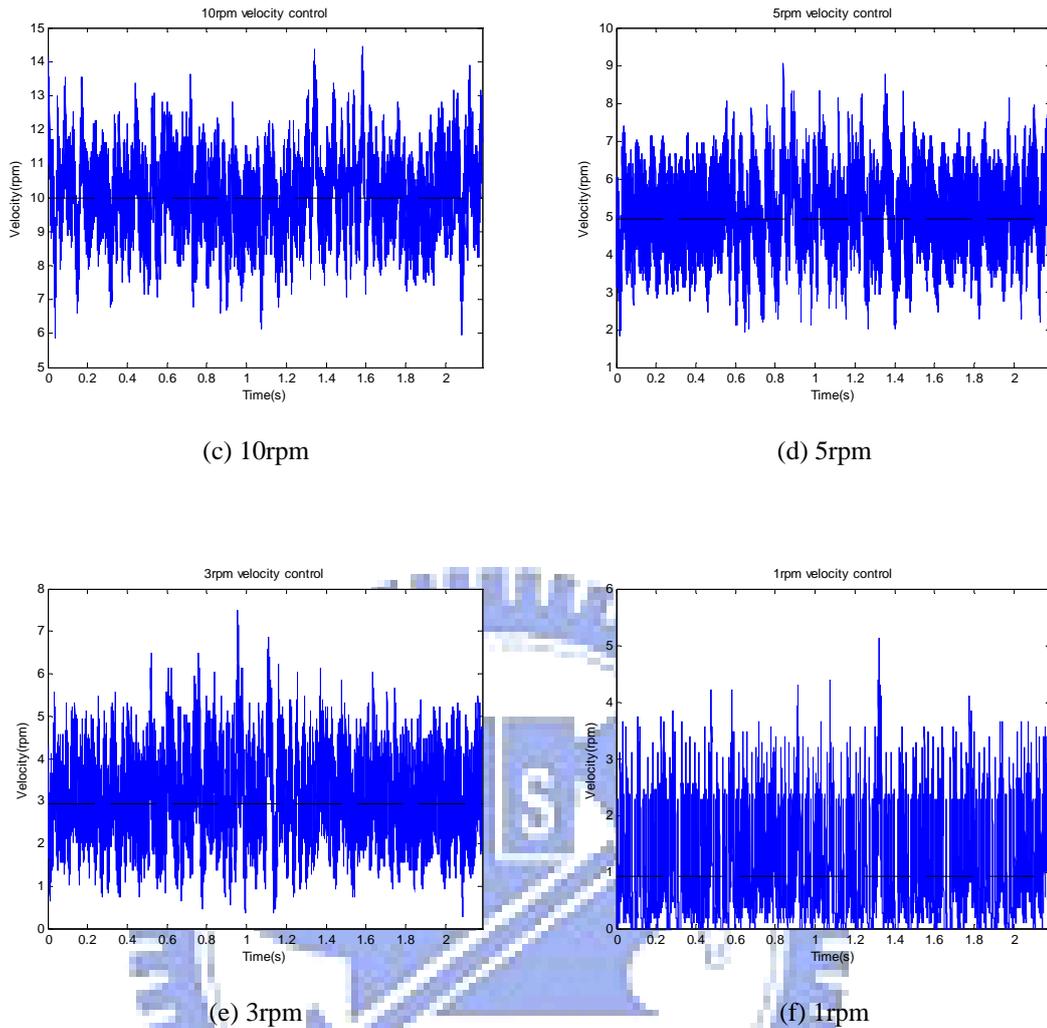


圖 4-15 PI 控制器不同轉速實驗結果($K_p = 2$ 、 $K_i = 0.0103$)

由上面六種不同低速範圍下的實驗結果發現原本 PI 控制器在低速範圍內控制效果尚可， $3rpm$ 以上時速度回授都有沿著速度命令做正負擺盪，隨著速度命令的降低擺盪範圍逐漸減少，在 $3rpm$ 時漣波集中在 $\pm 1 \sim 2rpm$ 之間，但是在 $1rpm$ 時漣波範圍似乎比較集中在正方向，但實際平均值有還是約 $1rpm$ 左右。

經過多次實驗之後調得另一組對低速範圍有較佳控制效果的新參數 $K_p = 6$ 、 $K_i = 0.0082$ ，以下是這組新參數的低速控制結果

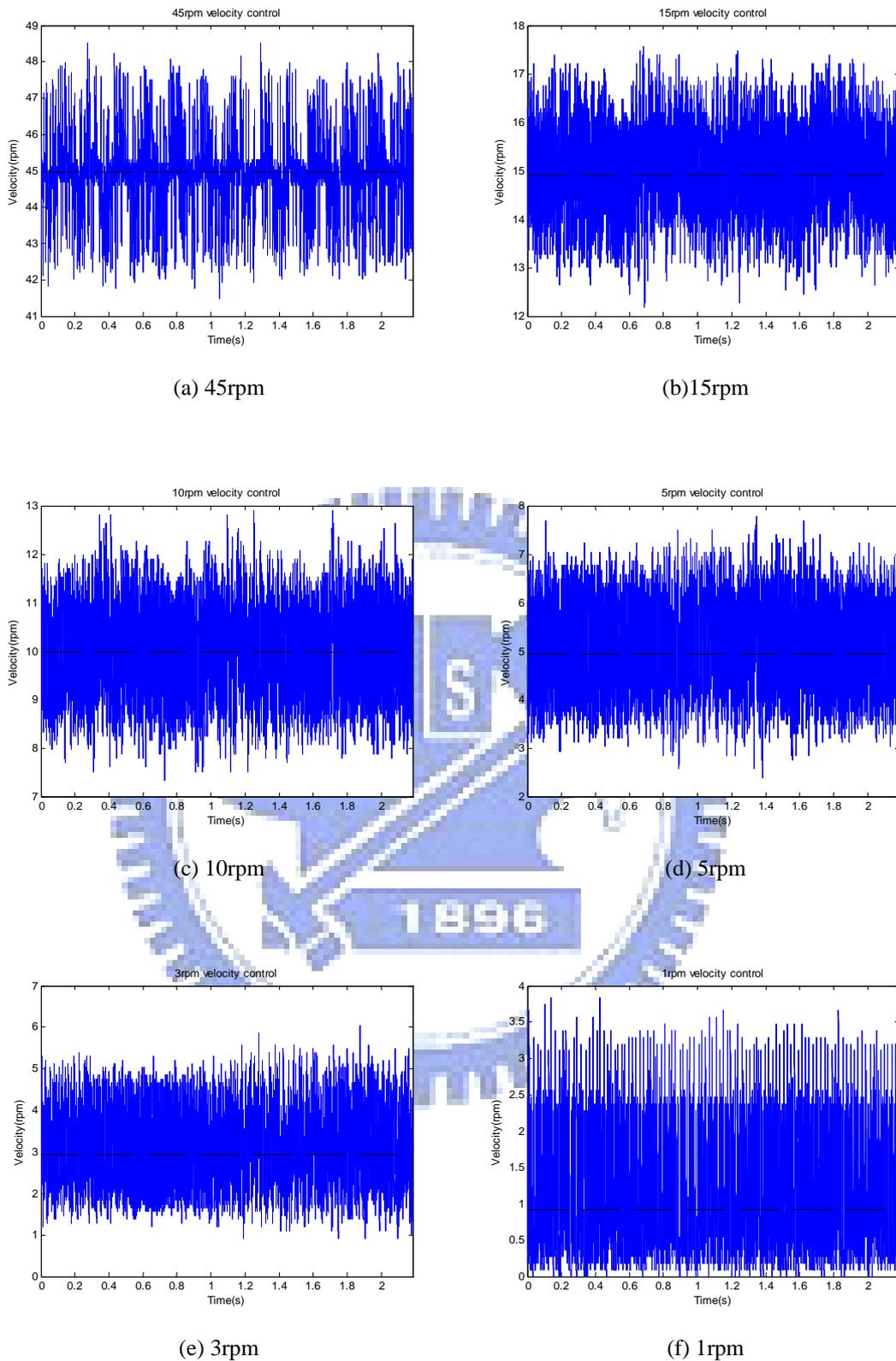


圖 4-16 調整後 PI 控制器不同轉速實驗結果 ($K_p = 6$ 、 $K_i = 0.0082$)

實驗結果證明在實驗的速度範圍內均能有效的減少低速漣波大小，在 45rpm 時減少

了約 2rpm 的漣波大小，在 5rpm 時也能減少約 1rpm 的漣波大小，但在 1rpm 時減少的漣波範圍就不是很明顯了。

表 4-4 PI 控制器參數調整前後變異量比較表

	45rpm	15rpm	10rpm	5rpm	3rpm	1rpm
$K_p = 2$ $K_i = 0.0103$	2.5713	1.5418	1.4589	1.1416	1.0515	0.7308
$K_p = 6$ $K_i = 0.0082$	1.0847	0.7942	0.8048	0.7563	0.7576	0.5819

從上比較表中證實在新調整過後的 PI 控制器參數的確有較好的低速控制效果，在各轉速實驗中都有比原本更好的控制效能。但值得注意的是，在參數調整前，15rpm 以下的實驗誤差會隨著命令的減少而降低，但在新調整過參數後，15rpm 以下的實驗誤差範圍維持都在 $\pm 2rpm$ 左右，沒有明顯減少的趨勢，因此推論此組參數可能接近 PI 控制器的最佳值。

4.3.2 PI 控制器加狀態控制器

本段將討論 PI 控制器再外加一個狀態控制器之後對低速控制的效能之改善，下圖是基本的狀態回授控制方塊圖

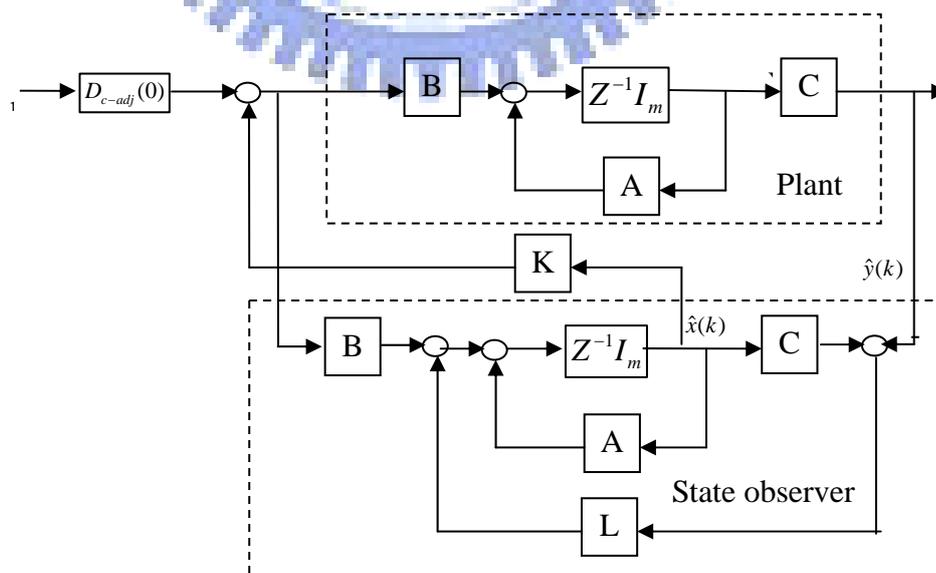


圖 4-17 狀態回授控制方塊圖

要做狀態回授控制首先必須要得到一個準確的系統模型，系統鑑別(system identification)是得到系統模型的唯一方法，利用 Matlab 軟體產生系統鑑別的輸入命令，將命令實際經過系統後的輸出與原命令一起進行系統鑑別。首先利用軟體產生 50~150rpm 的弦波測試信號，如下圖 4-18

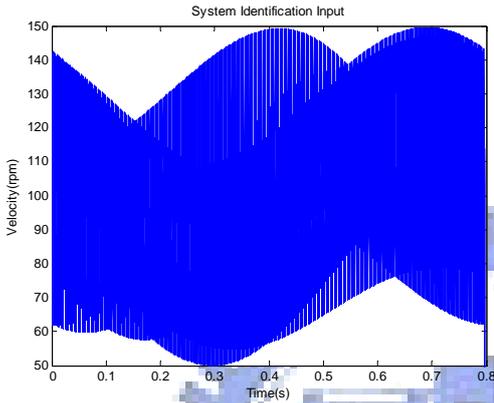


圖 4-18 系統鑑別用測試輸入

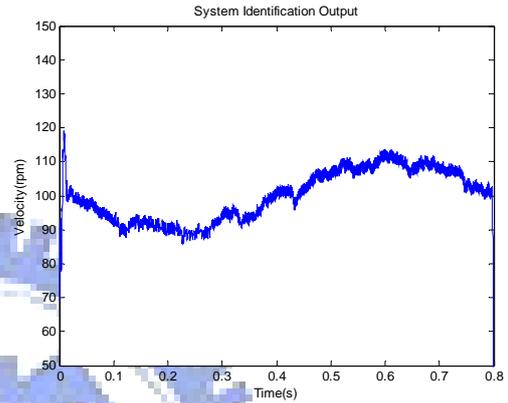


圖 4-19 系統鑑別輸出

此組系統鑑別所得四階系統模型

$$\frac{Y(z)}{X(z)} = \frac{0.0009849z^{-1} + 0.00607z^{-2} + 0.01103z^{-3} + 0.01265z^{-4}}{1 - 0.4848z^{-1} - 0.3227z^{-2} - 0.1916z^{-3} + 0.0298z^{-4}} \quad (4.19)$$

原有系統的 S-domain 極點位置在 $[-15572 \quad -5317 \pm 16908i \quad -143]$ ，將狀態回授後的系統 S-domain 極點設在 $[-10000 \pm 1000i \quad -8000 \pm 1000i]$ ，狀態估測器的極點則是設在兩倍遠的 $[-20000 \pm 2000i \quad -16000 \pm 2000i]$ ，下兩圖是狀態回授控制前後的系統波德圖。

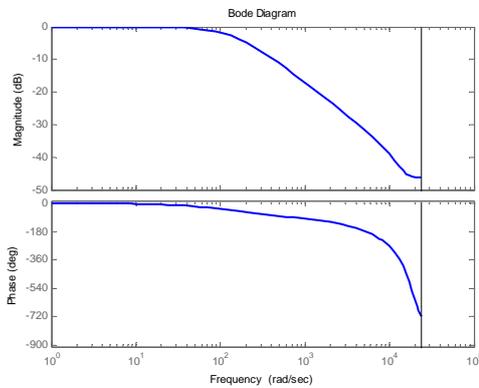


圖 4-20 原有系統波德圖

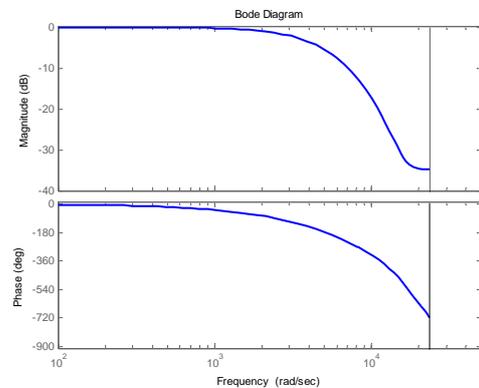


圖 4-21 加入狀態回授控制後系統波德圖

加入狀態回授控制後對系統頻寬有極大的改善，將原本的頻寬由 22.69Hz 一舉提升到 574.72Hz，將此控制架構實現，並與 4.3.1 所調整後的 PI 參數控制效果做比較。

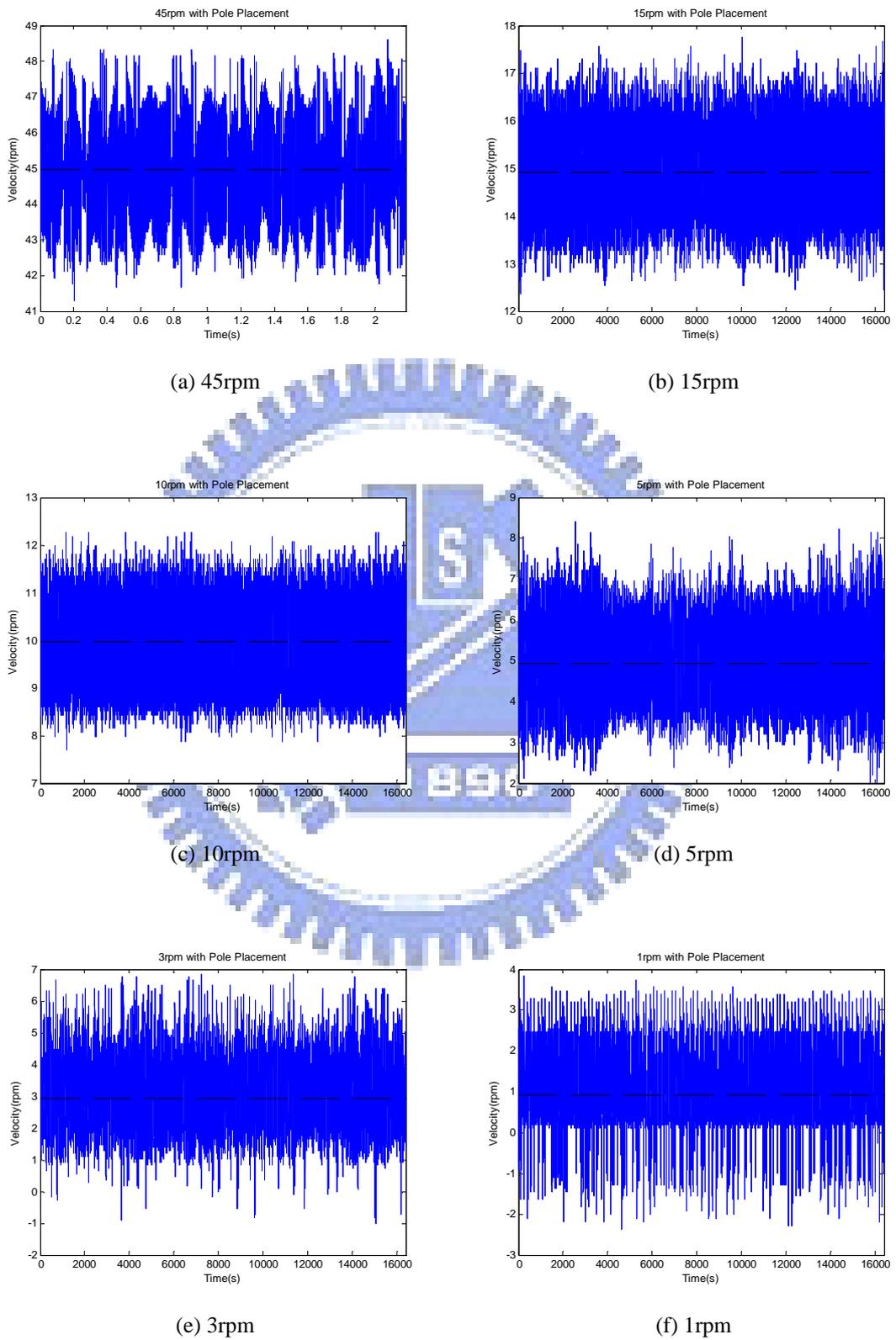


圖 4-22 狀態回授控制後不同轉速實驗

將上面六張實驗結果圖與圖 4-16 做比較，發現在 15rpm 的時候狀態回授控制與 PI($K_p = 6$ 、 $K_i = 0.0082$)控制兩者差距並不大，但是在 15rpm 以內似乎都是 PI 控制器的效果較好，原因在於低速範圍屬於非線性控制區間，要做系統鑑別相當困難的，所以才將系統鑑別的測試輸入設定在 50~150rpm 之間，就是為了取得重現性高的系統模型。

表 4-5 狀態回授控制後實驗變異量比較表

	45rpm	15rpm	10rpm	5rpm	3rpm	1rpm
狀態回授控制	1.4220	0.8084	0.6982	1.0115	1.3443	0.8993
$K_p = 6$ $K_i = 0.0082$	1.0847	0.7942	0.8048	0.7553	0.7576	0.5819

下圖是將三種不同的控制器控制結果變異量利用最小平方法所做的近似曲線，比較不同的控制器在不同速度命令下變異量的變化關係。

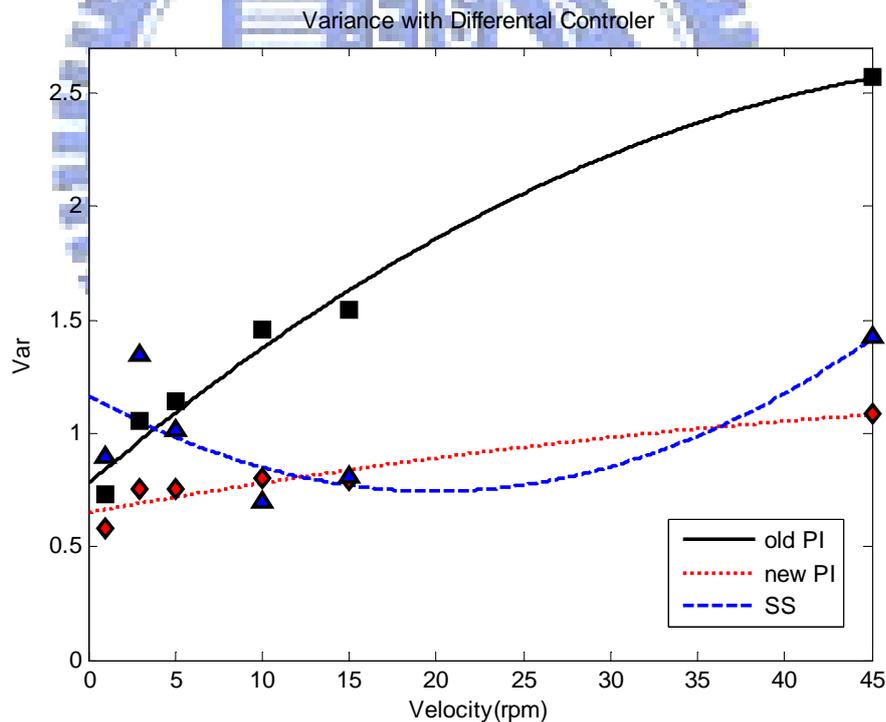


圖 4-23 不同控制法下變異量與速度關係圖

另外比較加入狀態回授控制的實驗結果，發現無論速度命令設定多少，誤差值似乎都控制在 $\pm 2\sim 3$ rpm 之間，變異量與速度命令沒有明顯關係，與原本只有 PI 控制器時誤差

會隨著速度命令降低而減少的情形不同，為證明狀態回授控制是否能將速度誤差控制在一定範圍內，比較 PI 控制器與狀態回授控制器在 300rpm 時的控制結果。

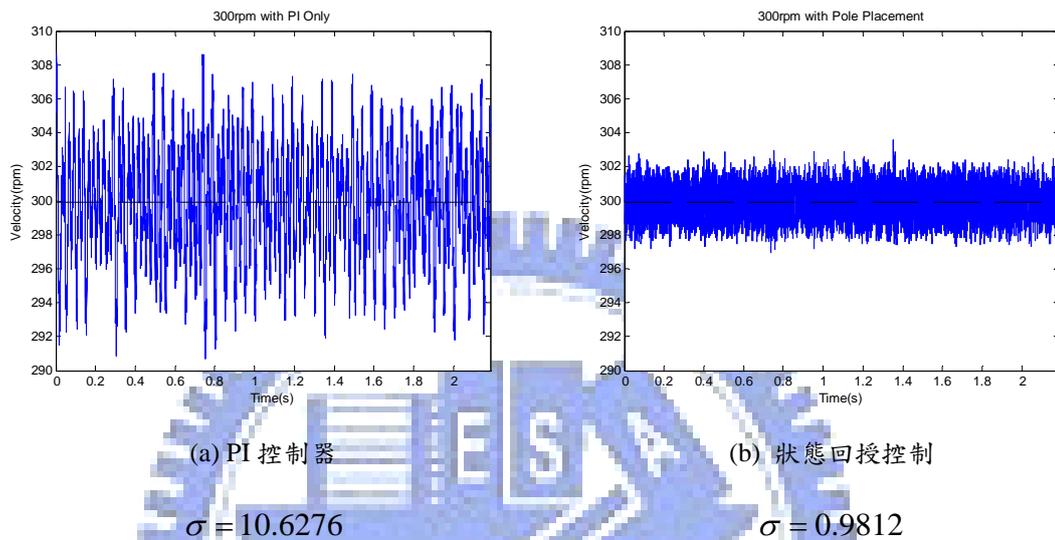


圖 4-24 300rpm 時兩種控制器實驗結果

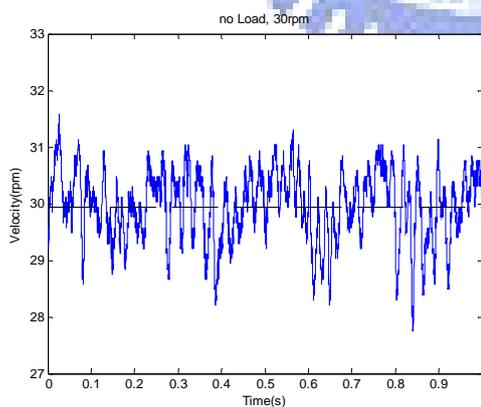
在 300rpm 時 PI 控制器有 10.6276 的變異量，而狀態回授控制只有 0.9812。由上面的實驗證實狀態回授控制器確實能將速度誤差控制在 $\pm 2\sim 3rpm$ 間，相同速度命令下 PI 控制器則有 $\pm 6\sim 8rpm$ 的誤差範圍，證實狀態回授控制器能夠有效的將速度誤差限制在固定的範圍內，而 PI 控制器的誤差範圍則會隨著速度命令的不同而有所改變，以低速範圍來說 PI 控制器有著較優異的控制效能。

4.3.3 應用於 CNC 機台上之低速控制

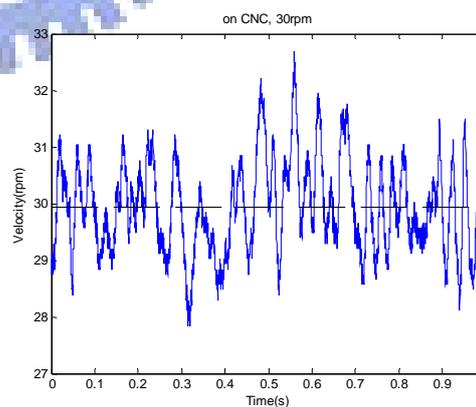


圖 4-25 CNC 機台照片

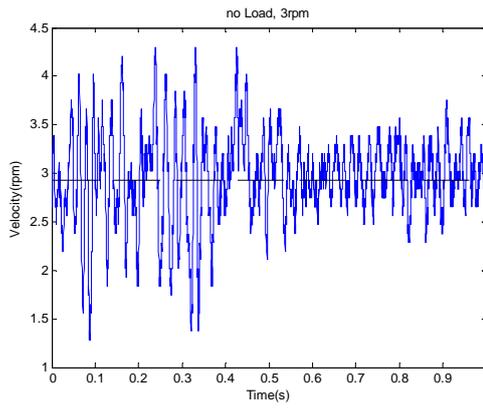
裝於實驗用 CNC 機台上的是 400 W 裝配 17-bit 串列式編碼器的伺服馬達，經過前面的速度估測及控制器比較實驗之後，得知在低速控制的應用中以卡曼濾波器作為速度估測搭配 PI 控制器作速度控制器可得到較佳的低速控制效能，實驗中將以此組合測試空載與裝置上 CNC 機台後之差異($K_p = 2$ 、 $K_i = 0.0103$)。



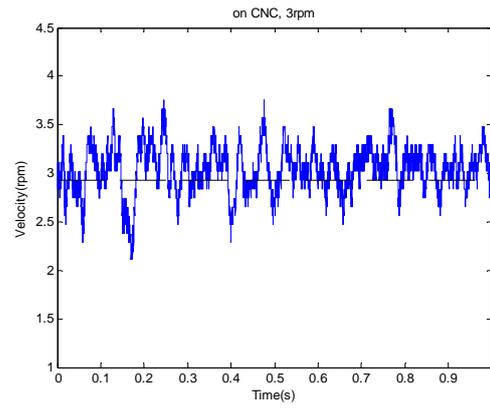
(a) 30rpm 時伺服馬達



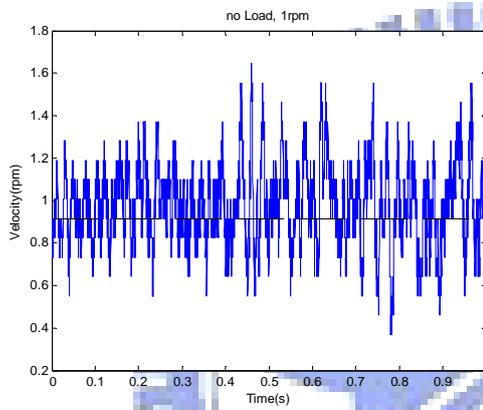
(b) 30rpm 時 CNC



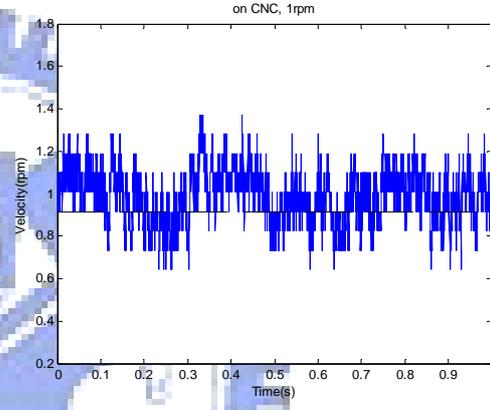
(c) 3rpm 時伺服馬達



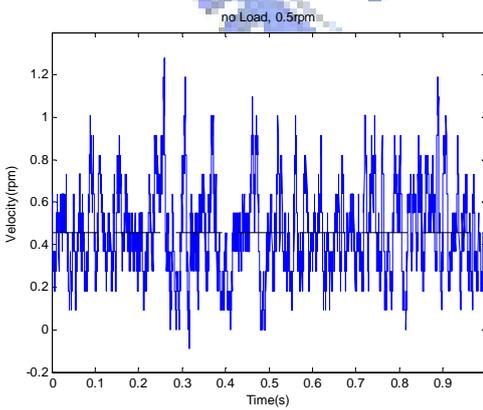
(d) 3rpm 時 CNC



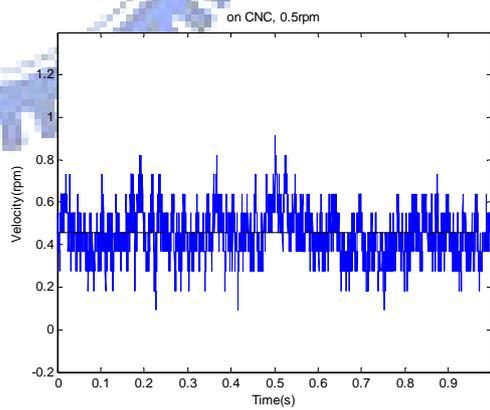
(e) 1rpm 時伺服馬達



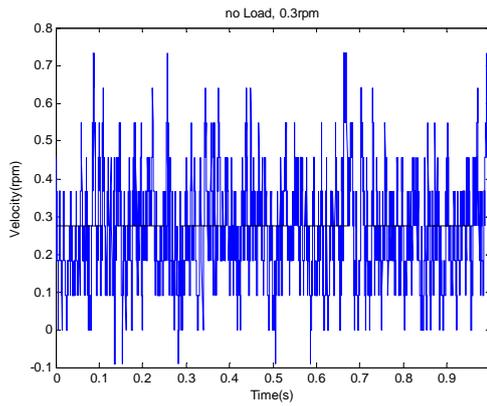
(f) 1rpm 時 CNC



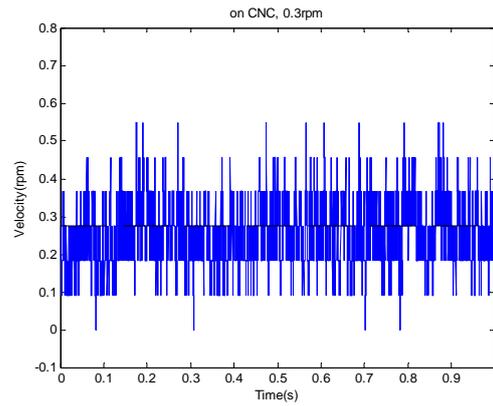
(g) 0.5rpm 時伺服馬達



(h) 0.5rpm 時 CNC



(i) 0.3rpm 時伺服馬達



(j) 0.3rpm 時 CNC

圖 4-26 負載之比較

由上面的實驗結果顯示在 $30rpm$ 的時候無載的控制結果比 CNC 機台上的控制結果好，但在 $3rpm$ 以下的實驗都是 CNC 機台的控制結果較好，推論其原因在於低速控制時受到負載的影響所以控制器輸出 u 較大，對於摩擦力的克服有較大的能力，而沒有負載時輸出 u 較小，受摩擦力影響較大，但在 $30rpm$ 時摩擦力的影響相對於速度命令則算是很小，所以無載時的表現比 CNC 機台上來的好。

表 4-6 空載與 CNC 機台上實驗變異量比較表

	30rpm	3rpm	1rpm	0.5rpm	0.3rpm
空載	0.3554	0.1935	0.0309	0.0484	0.0190
CNC	0.8955	0.0826	0.0167	0.0133	0.0073

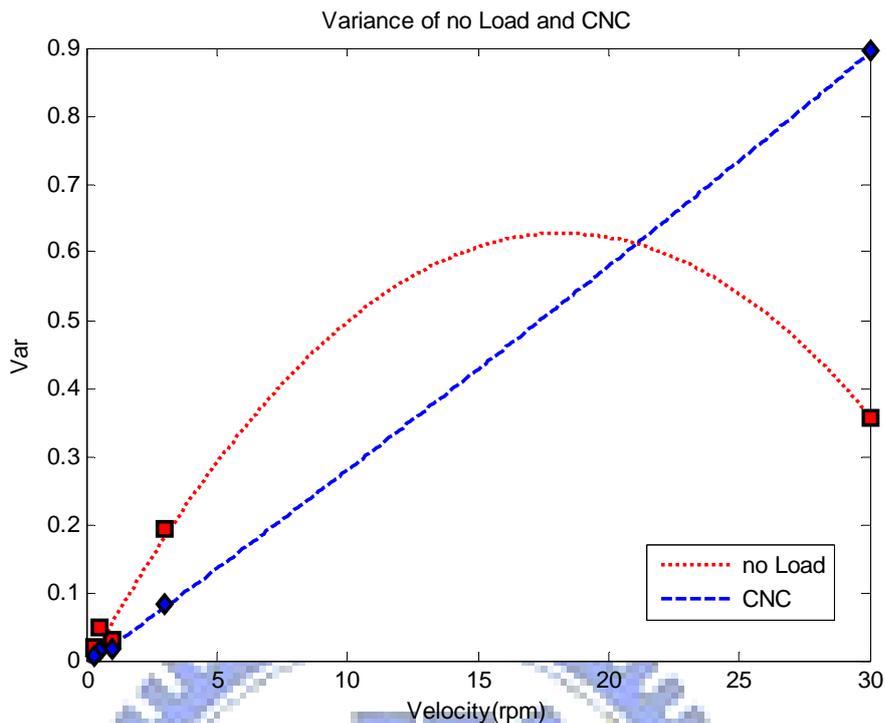


圖 4-27 空載與 CNC 變異量與速度命令關係圖

4.4 傳統編碼器與弦波編碼器之控制比較

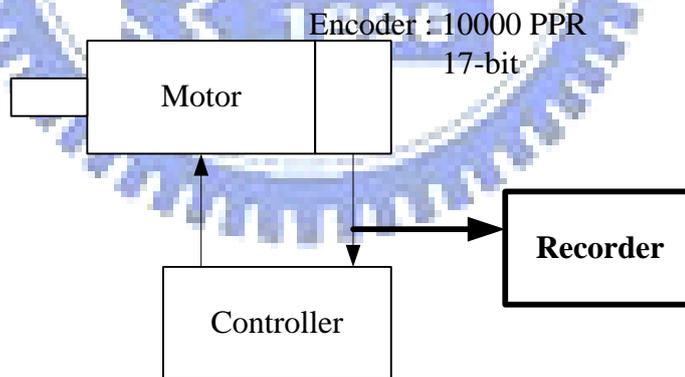


圖 4-28 編碼器比較實驗架構圖

本節首先比較編碼器解析度對伺服馬達低速控制所造成的影響，藉此證明提高編碼器解析度對於控制效能之改善。實驗中同樣以是 400 W 的伺服馬達，在編碼器方面則配備有 10000 PPR 的傳統編碼器與 17-bit 的高解析度串列編碼器兩種，以 7500 Hz 的取樣頻率做速度控制，首先計算兩種編碼器的基本速度回授解析度。

以 10000 PPR 解析度的編碼器來說

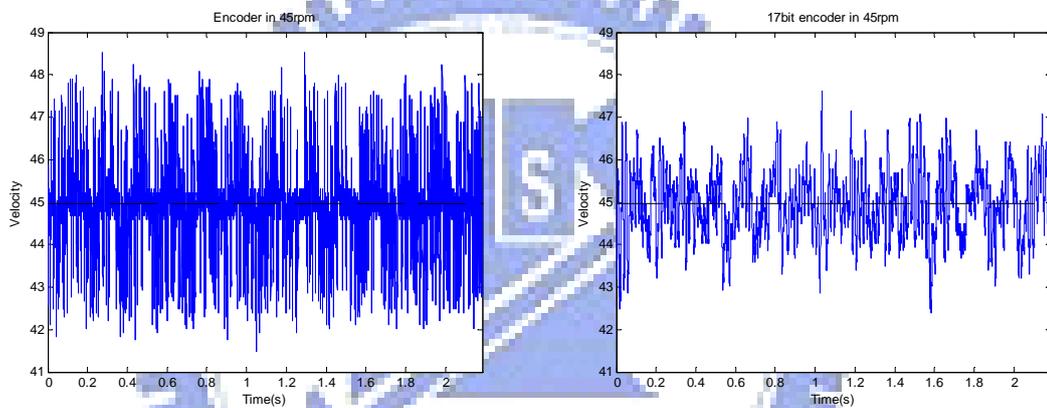
$$\frac{7500 \times 60}{10000} = 45rpm \quad (4.20)$$

也就是說在 45rpm 以下的轉速時傳統編碼器已經無法確保每次取樣頻率下都會有位置改變，所以此時的速度都是以內插估測所得。

另外以 17-bit 解析度的編碼器計算

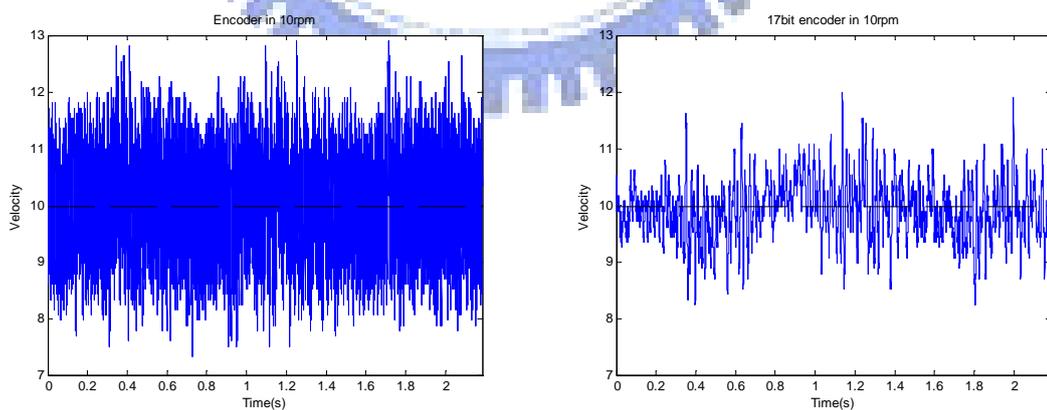
$$\frac{7500 \times 60}{2^{17}} = 3.43rpm \quad (4.21)$$

必須到 3.43rpm 之下才會使用到內插估測速度，所以 45rpm 及 3.43rpm 這兩個速度算是一個分界點。下面是這兩種編碼器在不同的速度命令下所做的速度控制結果



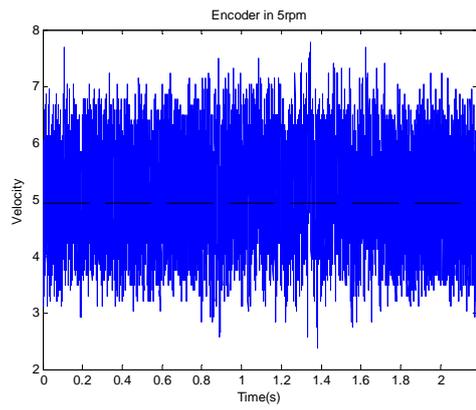
(a) 45rpm 時 10000PPR

(b) 45rpm 時 17bit

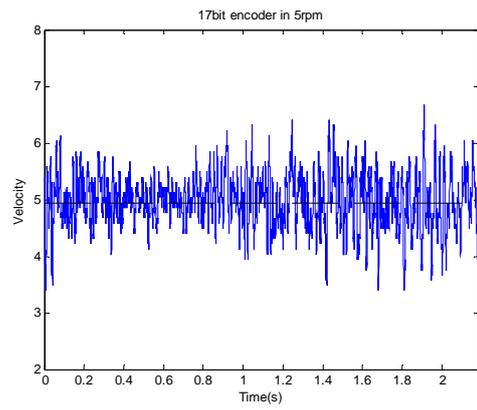


(c) 10rpm 時 10000PPR

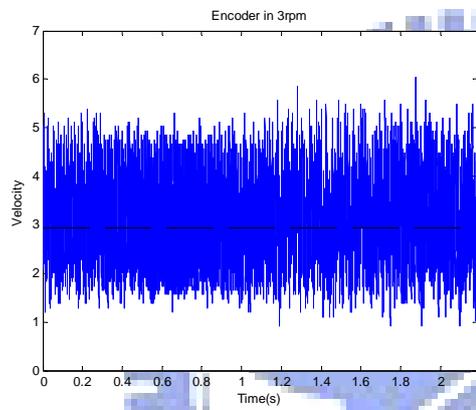
(d) 10rpm 時 17bit



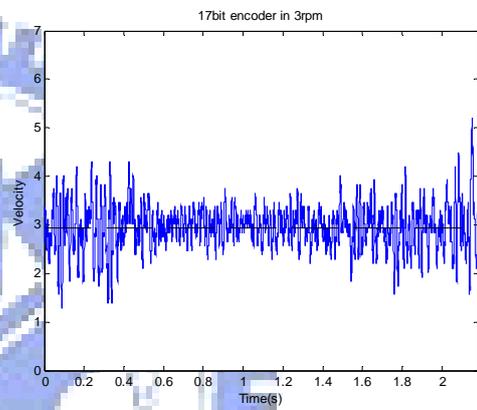
(e) 5rpm 時 10000PPR



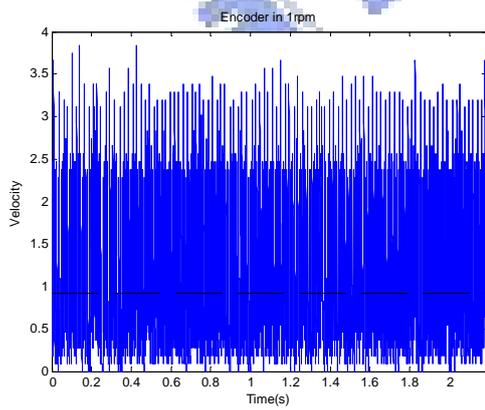
(f) 5rpm 時 17bit



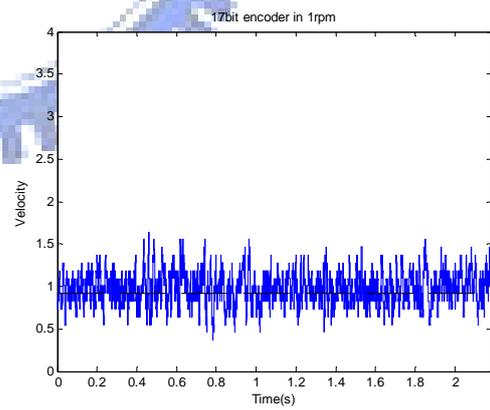
(g) 3rpm 時 10000PPR



(h) 3rpm 時 17bit



(i) 1rpm 時 10000PPR



(j) 1rpm 時 17bit

圖 4-29 不同編碼器解析度的控制效果

實驗結果顯示在各種速度區間內 17-bit encoder 都有較好的控制效能，而且隨著速

度命令的下降在越低速的時候更可明顯看出高解析度所減少的漣波範圍。

表 4-7 不同邊碼器解析度對控制效能所造成變異量的影響

	45rpm	10rpm	5rpm	3rpm	1rpm
10000PPR	1.0847	0.8048	0.7563	0.7576	0.5819
17bit	0.5972	0.2638	0.2148	0.1935	0.0309

由變異量比較結果，17-bit 解析度的控制結果變異量都比 10000 PPR 小，兩者的變異量隨著速度命令的降低而減少，但是 10000 PPR 解析度在 10rpm 以下變異量的減少比例已經不多，相對的 17-bit 解析度降低比例還是相當高。

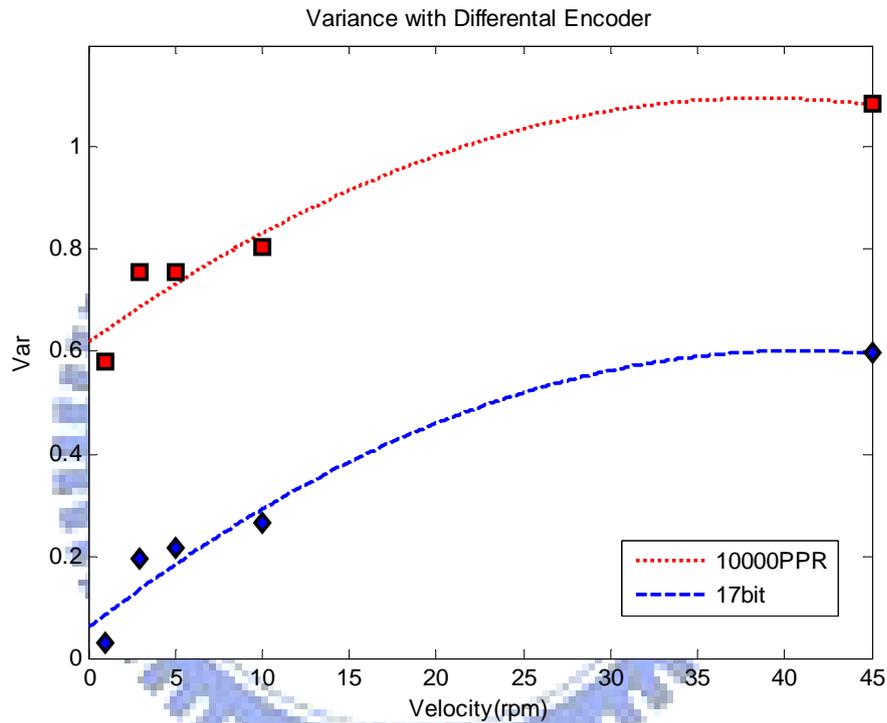


圖 4-30 不同編碼器控制效能變異量趨勢

由實驗證明編碼器解析度的提高有助於低速控制的穩定性，因此若將本文發展的 19-bit 高解析度串列編碼器用於伺服馬達低速控制中，可大大提升低速控制的效能，但對於高速度的區間則不易看出差別。

4.5 以 19-bit 串列式編碼器作為低速控制比較基準

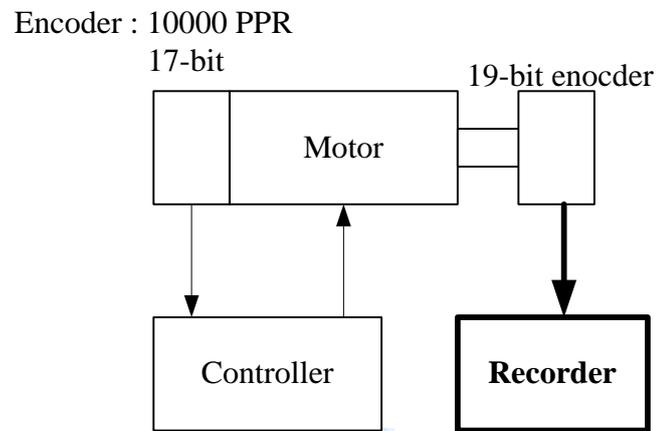


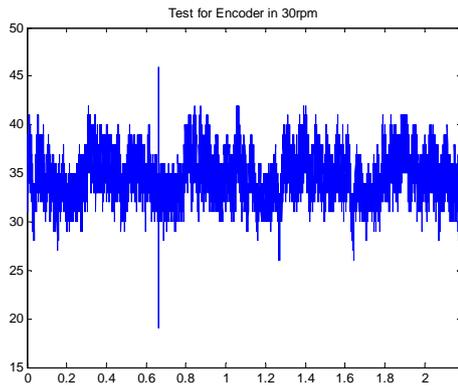
圖 4-31 控制效能比較實驗架構圖

本段將以另一種不同的方法來測量編碼器解析度的提升對低速控制的效能改善，以上述實驗所使用的兩組馬達，分別對接上本文所研發的 19-bit 串列式編碼器，編碼器獨立運作於馬達控制器外，以 7.5 KHz 速度紀錄位置，再使用 Matlab 軟體直接以 PC 差分計算兩次取樣間的位置變化，由此來比較兩種不同編碼器的低速控制效能。

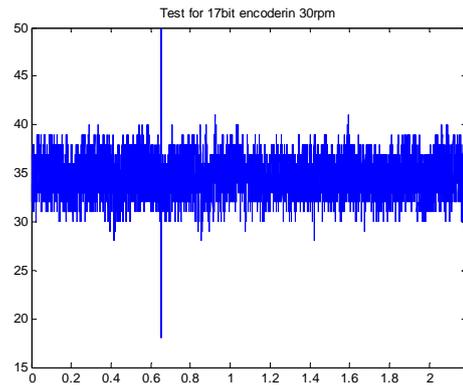
首先討論 19-bit 編碼器的最小位置解析度

$$\frac{7500 \times 60}{2^{17}} = 0.8583 \text{rpm} \quad (4.22)$$

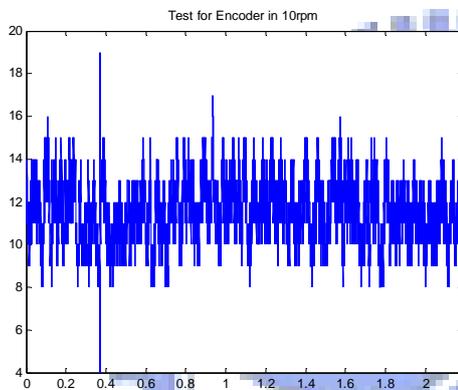
也就是說 19-bit 編碼器最小可直接偵測到的轉速是 0.8583rpm 左右，以下則是以 19-bit 編碼器作為測試基準，於裝有傳統編碼器及 17-bit 編碼器兩種不同編碼器的伺服馬達低速控制結果



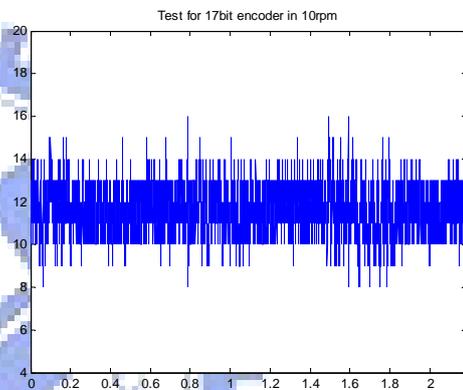
(a) 30rpm 時傳統編碼器



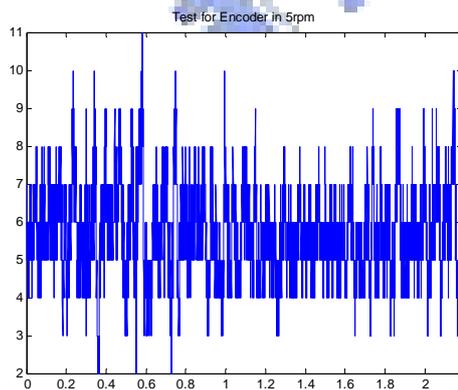
(b) 30rpm 時 17-bit 編碼器



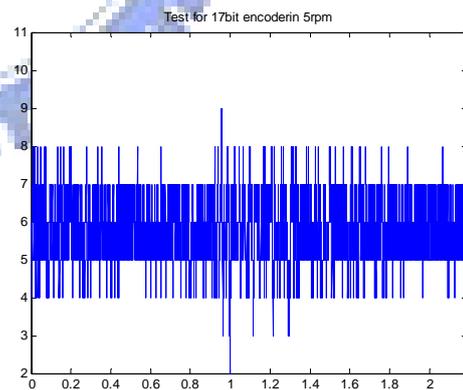
(c) 10rpm 時傳統編碼器



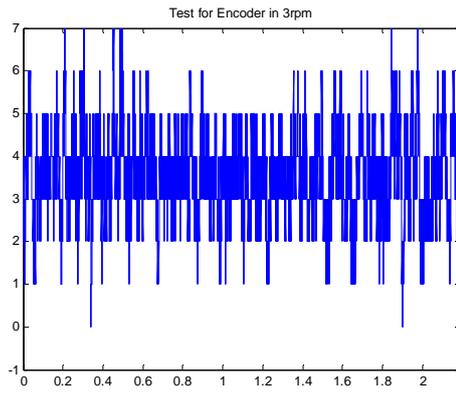
(d) 10rpm 時 17-bit 編碼器



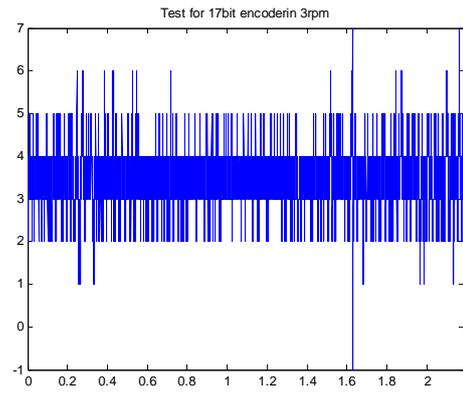
(e) 5rpm 時傳統編碼器



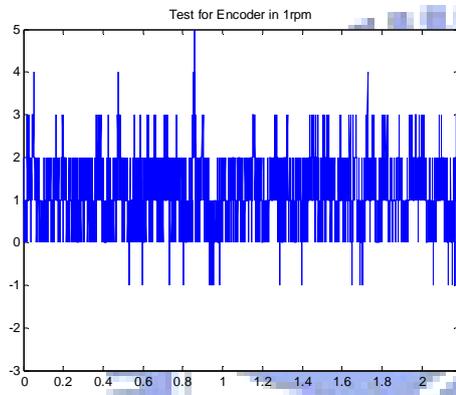
(f) 5rpm 時 17-bit 編碼器



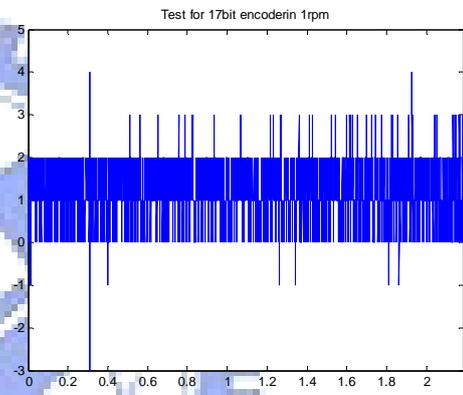
(g) 3rpm 時傳統編碼器



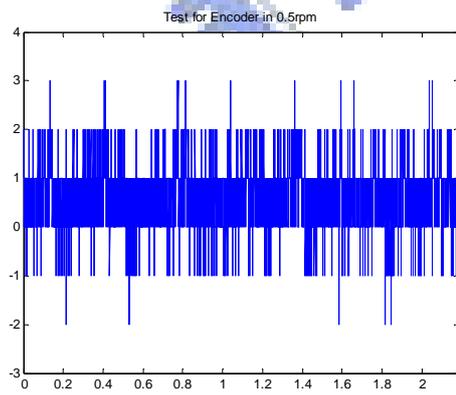
(h) 3rpm 時 17-bit 編碼器



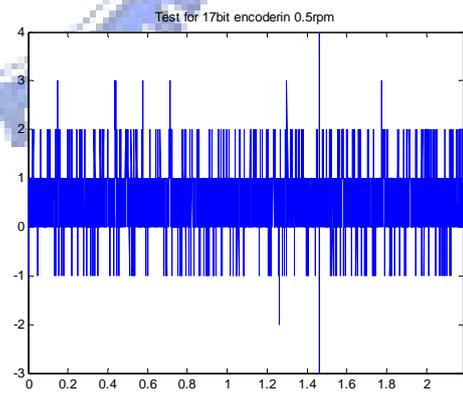
(i) 1rpm 時傳統編碼器



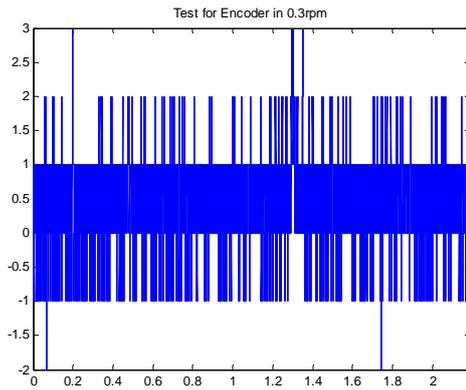
(j) 1rpm 時 17-bit 編碼器



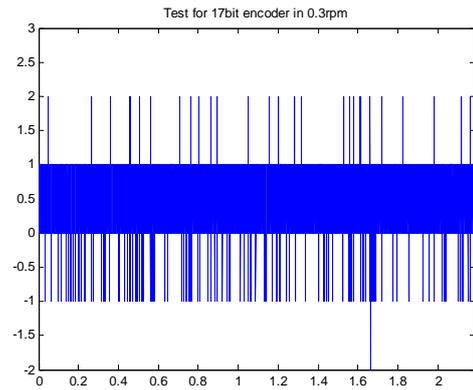
(k) 0.5rpm 時傳統編碼器



(l) 0.5rpm 時 17-bit 編碼器



(m) 0.3rpm 時傳統編碼器



(n) 0.3rpm 時 17-bit 編碼器

圖 4-32 19-bit 編碼器比較兩組伺服馬達控制結果

由上面的實驗結果得知裝有 17-bit 編碼器的伺服馬達低速控制都比裝有傳統編碼器的伺服馬達有更低的速度漣波，以下兩個表格分別是比較兩種伺服馬達在低速控制時的漣波變異量及每次取樣的理想差分量與實際差分量之平均值比較，理想差分量以 10rpm 來說則是

$$\frac{2^{19}}{7500 \times 60} \times 10 = 11.6508 \quad (4.23)$$

表示每次取樣作為比較基準的 19-bit 弦波編碼器應該要有 11.6508pulse 的改變量

表 4-8 不同編碼器低速控制效能之變異量比較表(19bit)

	10rpm	5rpm	3rpm	1rpm	0.5rpm	0.3rpm
10000PPR	1.8608	1.3182	0.9437	0.5310	0.4066	0.3445
17bit	1.0522	0.6121	0.5046	0.3542	0.3557	0.2527

表 4-9 不同編碼器低速控制效能之平均值比較表(19bit)

	10rpm	5rpm	3rpm	1rpm	0.5rpm	0.3rpm
理論值	11.6508	5.8254	3.4953	1.1651	0.5823	0.3495
10000PPR	11.5711	5.7446	3.5418	1.1654	0.5972	0.3769
17bit	11.6454	5.8227	3.4398	1.1667	0.5730	0.3506

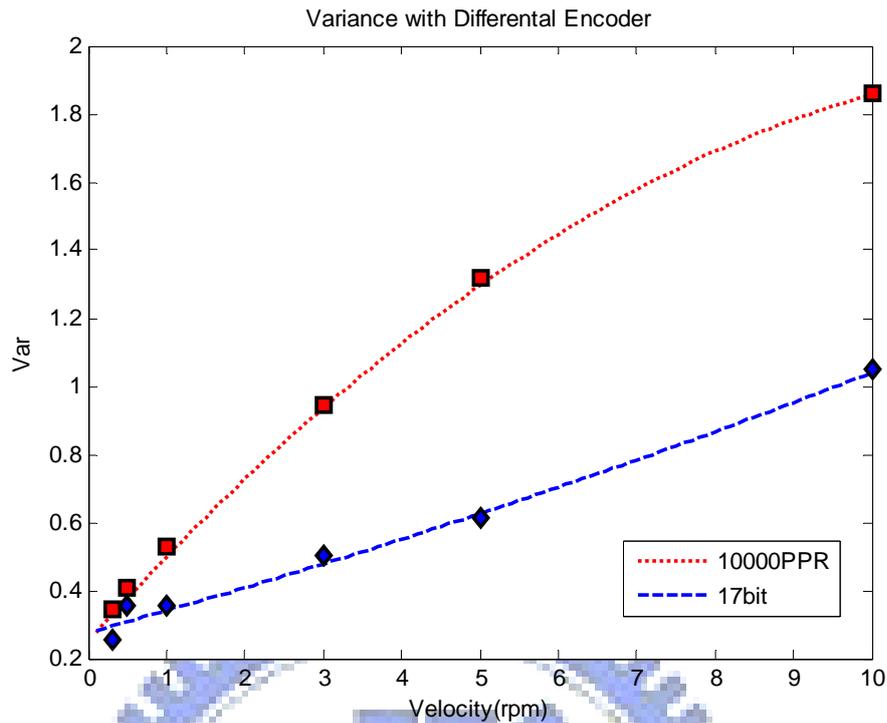


圖 4-33 不同編碼器控制變異量趨勢(19bit 對接)

由上面兩個比較結果看出裝有 17bit 編碼器的伺服馬達的低速控制效能優於傳統編碼器，在穩態漣波的抑制上有良好的效果，但就平均值而言兩者的差異量並不大，再次證明編碼器解析度的提升對於低速控制效能能夠有效的改善。此外也可看出使用傳統編碼器的伺服馬達控制系統事實上控制效果已經相當優良，雖然裝配的傳統編碼器解析度遠低於 17-bit 編碼器十倍以上，但控制效能卻僅略遜於 17-bit 編碼器之伺服馬達控制系統，因為影響低速控制效果的除了編碼器之外，低速估測及控制法則也相當重要。

第五章 結論與未來發展

5.1 結論

本文實現出兩種不同的弦波編碼器的解碼方法，分別以 DSP 及 FPGA 兩種方式實現，並提出一個專為弦波編碼器解碼使用的 QEP-CORDIC 演算法。再延伸至低速控制應用方面，比較編碼器的改善對控制效能的影響、不同的速度估測及控制法則在低速控制的效能。在此歸納出以下幾點結論：

1. 以 DSP-Based 實現線性化解碼：以 F2812 內建 12bitADC 取樣弦波輸入信號之後，先以外部電壓校準法及多通道取樣平均法去除雜訊及 ADC 取樣誤差，再進行線性化解碼計算及補償，最後整合粗略解碼位置及精細解碼位置後可完成 19-bit 弦波編碼器之解碼，全部需花費 $75.88\mu s$ ，最快解碼速度為 13.1 KHz。
2. 狀態轉換計數器：設計一個狀態機制計數弦波編碼器狀態轉換次數，提供一個能夠完全解決粗略解碼位置與精細解碼位置错位的方法，但轉速超過 1800rpm 後此法便不適用。
3. 以 FPGA-Based 實現 QEP-CORDIC 演算法解碼：以 unrolled 架構執行 14 次 QEP-CORDIC 遞迴運算需要 20 個系統週期，完成 19-bit 解碼約需 $150ns$ ，最快可達 6.666 MHz，但因受到外部 ADC 串列通訊之影響，更新速度僅為 245 KHz。設有串列通訊 SPI 介面及並列通訊介面兩種供使用者選用，全部共使用約 21000 個邏輯閘。
4. 串列編碼器應用之低速估測法：比較差分法加低通濾波器、最小平方近似法及卡曼濾波器三種不同的速度估測法，在低速命令時三種估測法平均值都與命令相近，但卡曼濾波器擁有最小的變異量，表示卡曼濾波器最適合低速估測之使用。
5. 串列編碼器應用之低速控制法：比較傳統 PI 控制器與 PI 控制器外加狀態回授控

制兩種控制法，實驗顯示 PI 控制器的速度回授漣波大小會隨著速度命令而有所改變，外加狀態回授控制後漣波大小會固定在 $\pm 2\sim 3rpm$ 之間，不隨速度命令改變，以低速控制而言純 PI 控制器擁有較低的速度漣波，較好的控制效能。

5.2 未來發展

1. 利用信號處理方式及硬體電路方式降低雜訊干擾，提昇 ADC 解析度，可望提升弦波編碼器解碼後的解析度。
2. 以實驗完成的 FPGA-Based QEP-CORDIC 解碼為基礎，發展「整合型通訊介面高解析度弦波編碼器解碼晶片」，能夠減少體積、提升處理速度、降低成本，且可適用於市面上所有的弦波編碼器解碼，並廣泛用於各種需要高解析度轉動位置感測之場合。
3. 雖然實驗中 PI 控制器對低速控制的效能以達相當水準，但若能以非線性控制法或非線性補償等等技術，搭配上本文所研發之高解析度弦波編碼器，必能將伺服馬達低速控制效能再提升。

參考文獻

- [1] J. Burke, J. F. Moynihan, K. Unterkofler, “Extraction of High Resolution Position Information from Sinusoidal Encoder”, *Proc. PCIM-Europe 1999*, pp.217-222.
- [2] S.Balemi, “Automatic Calibration of Sinusoidal Encoder Signals”, *Proceedings of the 16th IFAC World Congress*, Prague, 2005.
- [3] M. Benammar, L. Ben-Brahim, M. A. Alhamadi, “A Novel Resolver-to-360° Linearized Converter”, *IEEE Sensors Journal*, VOL.4, NO. 1, pp.96-101, February 2004.
- [4] M. Benammar, L. Ben-Brahim, M. A. Alhamadi, “A High Precision Resolver-to-DC Converter”, *IEEE Transactions on Instrumentation and Measurement*, VOL.54, NO.6, pp.2289-2296, December 2005.
- [5] M. Benammar, L. Ben-Brahim, M. A. Alhamadi, Mohamed EI-Naimi, “A Novel Converter for Sinusoidal Encoders”, *IEEE Sensors Conference*, pp.1415-1418, October, 2006.
- [6] R. Andraka, “A survey of CORDIC algorithms for FPGA based computers,” *ACM/SIGDA sixth international symposium on Field programmable gate arrays.*, pp.191-200, February 1998.
- [7] Y. H. Hu, “CORDIC-based VLSI architectures for digital signal processing,” *IEEE Signal Processing Mag.*, pp.16-35, July 1992.
- [8] 施忠賢, “一個使用可變因子的新式混合式座標旋轉演算法來計算正弦餘弦函數值,” 逢甲大學資訊工程系, 碩士論文, 1999.
- [9] “F2810, F2811 and F2812 ADC Calibration”, *Texas Instruments Application Report SPRA989A* November 2004.
- [10] “Spartan-3 Generation FPGA User Guide”, Xilinx , UG331, April 2001.
- [11] “Spartan-3A/3AN Starter Kit Board User Guide”, Xilinx , UG334, May 2007.
- [12] S. Vadlamani, W. Mahmoud, “Comparison of CORDIC Algorithm Implementations on FPGA Families”, *Proceedings of the Thirty-Fourth Southeastern Symposium on*,

pp.192-196, 2002.

[13] <http://www.ti.com/>

[14] R. H. Brown, S. C. Schneider, and M. G. Mulligan, "Analysis of Algorithms for Velocity Estimation from Discrete Position Versus Time Data," *IEEE Transactions on Industrial Electronics*, VOL.39, NO.1, pp.11-19, February 1992.

[15] T. Hastie, R. Tibshirani, J. Friedman, "The Elements of Statistical Learning", Springer, 2001.

[16] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Department of Computer Science, University of North Carolina at Chapel Hill.

[17] P. R. Belanger, P. Dobrovolny, A. Helmy, and X. Zhang, "Estimation of Angular Velocity and Acceleration from Shaft-Encoder Measurements," *IEEE International Conference on Robotics and Automation*, May 1992.

[18] 鄭群星, "FPGA/CPLD 數位晶片設計入門-使用 Xilinx ISE 發展系統", 全華科技圖書股份有限公司, 2006.

[19] 黃宗勝, "永磁式伺服馬達之自動切換式 Fuzzy/Lead-Lag 控制器設計", 國立交通大學電機與控制工程系, 碩士論文, 2007.