

國立交通大學
電機與控制工程研究所

碩士論文

使用光流影像感測的全方向運動平台
之空間軌跡追蹤系統設計



Spatial Path Tracking System Design for an
Omni-directional Platform Using
Optical Flow Sensors

研究生： 鄭 榮 煌
指導教授： 胡 竹 生 博士

中華民國九十五年七月

使用光流影像感測的全方向運動平台
之空間軌跡追蹤系統設計

Spatial Path Tracking System Design for an
Omni-directional Platform Using
Optical Flow Sensors

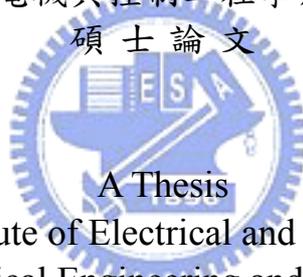
研究生：鄭 榮 煌

Student : Jung-Hung Cheng

指導教授：胡 竹 生 博士

Advisor : Jwu-Sheng Hu

國立交通大學
電機與控制工程學系
碩士論文



Submitted to Institute of Electrical and Control Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of Master
in

Electrical and Control Engineering

July 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

使用光流影像感測的全方向運動平台 之空間軌跡追蹤系統設計

研究生：鄭 榮 煌

指導教授：胡 竹 生 博士

國立交通大學電機與控制工程研究所碩士班

摘 要



本論文針對室內環境的輪式機器人提供一套高整合且低成本的定位及控制系統；它可以在一般的室內環境中利用光流影像定位出機器人在空間中的相對方向和位置；同時利用可即時速度控制之空間性軌跡追蹤法來控制該輪式機器人，使得該機器人擁有在多種不同且未知情況的地面上進行速度控制、軌跡追蹤、碰撞偵測以及動態軌跡規劃的能力。本論文亦提出以全向輪為基礎的輪式機器人平台，它於一般的室內環境中擁有較高的運動靈活度；同時我們也整合了嵌入式網路技術，以達成分散式運算及遠端操控機器人之目的。

目前，已經可以透過無線網路控制該平台在多種地形，包含：木板、地毯、磨石地磚以及傾斜的木板之上進行速度控制、軌跡追蹤、碰撞偵測以及動態路徑規劃的任務。

本論文主要分成三大部分，第一部分介紹極值搜尋演算法、軌跡追蹤演算法、光流影像定位演算法以及全方向運動平台演算法。第二部分介紹實驗平台的軟硬體架構。第三部分則是演算法在實驗平台上的實作，結果與分析。

Spatial Path Tracking System Design for an Omni-directional Platform Using Optical Flow Sensors

Student : Jung-Hung Cheng

Advisor : Prof. Jwu-Sheng Hu

Institute of Electrical and Control Engineering

Abstract



This thesis proposes an integrated and low-cost system of localization and control for an indoor omni-directional wheeled mobile robot using optical flow sensors. The system is capable of commanding the robot to perform velocity control, path tracking, collision detection and dynamic path planning tasks in various unknown landforms. Moreover, the omni-directional wheels offer a higher mobility in an indoor environment, and the embedded Ethernet technology is used to achieve the objectives of distributed computation and remote control.

Presently, we can command this platform via wireless network to perform velocity control, path tracking, collision detection and dynamical path planning tasks in all following landforms: plank, carpet, terrazzo floor and inclined plank.

The thesis is divided into three parts: The first part is the introduction to the extremum seeking algorithm, the path tracking method, the optical flow localization method and the omni-directional motion algorithm. The second part introduces the overall hardware and software architecture of the platform. The third part is the implementation of overall system on the platform and the discussions and analyses on the experimental results.

誌 謝

對於本論文的完成，首先感謝指導老師胡竹生教授兩年多來的辛勤指導。在這段日子裡，老師不僅教導我學術上的知識，更教導我做學問應有的態度和方法；另外，亦在耳濡目染當中從他身上學習到待人處事的方法，對於老師兩年多來的教導，誠摯的致上最真誠的謝意。

而在這些日子，必須感謝實驗室眾多學長姐，同學和學弟們的陪伴與協助。感謝機器人組的立偉學長，在理論及實務上都提供我很多的建議和幫忙；感謝鄭博士對我的加油鼓勵；感謝劉大人帶我們去重訓，讓大家都擁有健康的體魄；感謝宗敏學長架的網頁，讓大家可以獲得新知以及排解無聊時光；感謝開新車超講義氣的帥興學長，讓大家知道怎麼當一個很 Man 的型男；感謝終於當上實驗室一姐的鏗元學姐，讓大家可以吃到好吃的自製餅干；感謝聽到冷笑話會很捧場的鳥蕙，讓實驗室時時充滿歡樂的笑聲；感謝超會唱歌的朱杰倫，讓實驗室充滿活力；感謝超會拉桿上籃的恒嘉，讓大家可以常常運動強健體魄；感謝超有愛心長得又甜美的佩靜，讓實驗室充滿溫暖的感覺；感謝機器人組的好夥伴耀賢和永融，有你們的支持和幫忙，讓我可以充滿動力做好我的研究；還有，講義氣的弘齡，有責任感的楷祥，以及實力超強的 Alphar。感謝你們大家在我研究所的生涯中給我許多的歡樂和幫忙，讓我在這兩年間留下深刻美好的回憶。

最後，我要感謝的是我的家人，父親鄭棟連先生、母親莊梨珠女士、大姐雅綺、二姐美玲、大哥榮泰和三姐美玉，感謝你們給我一個幸福溫暖的家庭，讓我可以無後顧之憂地專心於學業研究上；還有，感謝眾多的好朋友們；尤其是小勻，感謝你一路上的陪伴和鼓勵；謝謝你們。

目 錄

摘 要	i
Abstract	ii
誌 謝	iii
目 錄	iv
表 列	vi
圖 列	vii
第一章 緒論	1
1.1 研究動機	1
1.2 研究目標	2
1.3 序言	3
第二章 即時速度控制之空間性軌跡追蹤法	4
2.1 離散型極值三角搜尋法	4
2.1.1 極值搜尋法(Extremum Seeking Algorithm, ESA)	4
2.1.2 離散型極值三角搜尋法(Discrete-time Triangular Search Extremum Seeking Algorithm, DTS-ESA)[9]	8
2.1.3 使用 DTS-ESA 進行運動平台之速度控制	11
2.2 空間性軌跡追蹤法	19
2.2.1 時間性軌跡追蹤法(Temporal Path Tracking Method, TPTM)	19
2.2.2 空間性軌跡追蹤法(Spatial Path Tracking Method, SPTM)[10]	20
2.2.3 即時速度控制之空間性軌跡追蹤法(Real-time Velocity Control - SPTM, RVC-SPTM)	23
2.2.4 使用 RVC-SPTM 進行運動平台之軌跡控制	24
2.2.5 利用 RVC-SPTM 進行碰撞偵測及動態路徑規劃	27
第三章 光流影像偵測定位法	33
3.1 光流(Optical Flow)的定義	33
3.2 光流影像偵測定位法之原理[11]	34
第四章 全方位運動平台	37
4.1 簡介	37

4.2	運動學分析	38
4.3	運動模式	41
第五章	軟硬體設計與實現	43
5.1	系統架構	43
5.2	全方向運動平台-海星號(Starfish)之架構	44
5.2.1	電路功能介紹	45
5.3	主機端架構	51
第六章	實驗結果與討論	54
6.1	光流影像定位實驗結果	54
6.2	速度控制及軌跡追蹤實驗結果	57
6.3	碰撞偵測及動態路徑規劃實驗結果	68
第七章	結論	73
7.1	研究成果	73
7.2	未來展望	73
Reference:	75



表 列

表 1 光流定位誤差表-實驗 1.....	54
表 2 光流定位誤差表-實驗 2.....	55
表 3 光流定位誤差表-實驗 3.....	55
表 4 光流定位誤差表-實驗 4.....	56



圖 列

圖 1 某系統之成本函數.....	4
圖 2 極值搜尋法之工作迴圈.....	5
圖 3 某系統輸入與輸出之間的對應函式.....	10
圖 4 利用 p_k 和 q_k 來決定下一次的控制輸入值 u_{k+1}	10
圖 5 利用 p_{k+1} 和 q_{k+1} 來決定下一次的控制輸入值 u_{k+2}	10
圖 6 利用 p_{k+2} 和 q_{k+2} 來決定下一次的控制輸入值 u_{k+3}	11
圖 7 找到最佳的控制輸入值(u^*)，.....	11
圖 8 平台速度方向定義圖.....	12
圖 9 木板上 vi_x 與 vo_x 的關係圖.....	12
圖 10 木板上 vi_y 與 vo_y 的關係圖.....	13
圖 11 地毯上 vi_x 與 vo_x 的關係圖.....	13
圖 12 地毯上 vi_y 與 vo_y 的關係圖.....	13
圖 13 磨石地磚上 vi_x 與 vo_x 的關係圖.....	14
圖 14 磨石地磚上 vi_y 與 vo_y 的關係圖.....	14
圖 15 有斜度的木板(方向為 X 軸)上 vi_x 與 vo_x 的關係圖.....	14
圖 16 有斜度的木板(方向為 X 軸)上 vi_y 與 vo_y 的關係圖.....	15
圖 17 有斜度的木板(方向為 Y 軸)上 vi_x 與 vo_x 的關係圖.....	15
圖 18 有斜度的木板(方向為 Y 軸)上 vi_y 與 vo_y 的關係圖.....	15
圖 19 vi_x 與 vo_x 之間呈線性遞增關係.....	17
圖 20 vi_x 與 vo_x 之間呈非線性遞增關係.....	17
圖 21 vi_x 與 vo_x 之間呈非線性遞增關係.....	17
圖 22 雙輸入變數的平台系統輸出函數.....	18
圖 23 切角(CUT CORNERS).....	19
圖 24 軌跡及軌跡相關量值.....	20
圖 25 空間性軌跡追蹤法的控制架構圖.....	21
圖 26 ε -Controller 的相關量值.....	22
圖 27 RVC-SPTM 的控制架構圖.....	24
圖 28 RVC-SPTM 的控制流程圖.....	26
圖 29 碰撞偵測_1.....	28
圖 30 碰撞偵測_2.....	28
圖 31 碰撞偵測_3.....	28
圖 32 動態路徑規劃_1.....	30
圖 33 動態路徑規劃_2.....	31
圖 34 含碰撞偵測及動態路徑規劃之 RVC-SPTM 控制流程圖.....	32
圖 35 光流向量場.....	33
圖 36 光流感測器與平台的幾何關係圖.....	34

圖 37 光流軌跡視窗	36
圖 38 三種不同型態的全向輪	37
圖 39 全向輪平台簡圖	38
圖 40 全向輪簡圖	38
圖 41 運動模式示意圖	42
圖 42 完整系統架構	43
圖 43 (左)上蓋, (右)上蓋與底盤組合透視圖	44
圖 44 嵌入式網路板	46
圖 45 馬達控制板	46
圖 46 感測器控制板	46
圖 47 馬達裝置	47
圖 48 AP(包含 SWITCH HUB)	47
圖 49 降壓模組與電池組	47
圖 50 光流感測器	48
圖 51 QUADRATURE MODE OUTPUT STATE	48
圖 52 QUADRATURE MODE OUTPUT WAVEFORM	49
圖 53 內部電路圖	49
圖 54 底部照片	50
圖 55 VISUAL BASIC 所建立之使用者圖形介面	52
圖 56 利用 DDE 將資料傳至 MATLAB 進行記錄	52
圖 57 利用 MATLAB 所繪製的速度及指向圖	53
圖 58 利用 MATLAB 所繪製的誤差及軌跡圖	53
圖 59 光流定位軌跡圖(左)及平台移動示意圖(右)-實驗 1	54
圖 60 光流定位軌跡圖(左)及平台移動示意圖(右)-實驗 2	55
圖 61 光流定位軌跡圖(左)及平台移動示意圖(右)-實驗 3	55
圖 62 光流定位軌跡圖(左)及平台移動示意圖(右)-實驗 4	56
圖 63 對照組 1:於木板上進行直線軌跡追蹤任務之速度及指向圖	58
圖 64 對照組 1:於木板上進行直線軌跡追蹤任務之軌跡圖	58
圖 65 對照組 1:於木板上進行直線軌跡追蹤任務之移動誤差圖	58
圖 66 對照組 2:於木板上進行弧線軌跡追蹤任務之速度及指向圖	59
圖 67 對照組 2:於木板上進行弧線軌跡追蹤任務之軌跡圖	59
圖 68 對照組 2:於木板上進行弧線軌跡追蹤任務之移動誤差圖	59
圖 69 實驗組 1-1:於木板上進行直線軌跡追蹤任務之速度及指向圖	60
圖 70 實驗組 1-1:於木板上進行直線軌跡追蹤任務之軌跡圖	60
圖 71 實驗組 1-1:於木板上進行直線軌跡追蹤任務之移動誤差圖	60
圖 72 實驗組 1-2:於木板上進行弧線軌跡追蹤任務之速度及指向圖	61
圖 73 實驗組 1-2:於木板上進行弧線軌跡追蹤任務之軌跡圖	61
圖 74 實驗組 1-2:於木板上進行弧線軌跡追蹤任務之移動誤差圖	61
圖 75 實驗組 2-1:於地毯上進行直線軌跡追蹤任務之速度及指向圖	62
圖 76 實驗組 2-1:於地毯上進行直線軌跡追蹤任務之軌跡圖	62
圖 77 實驗組 2-1:於地毯上進行直線軌跡追蹤任務之移動誤差圖	62

圖 78	實驗組 2-2:於地毯上進行弧線軌跡追蹤任務之速度及指向圖	63
圖 79	實驗組 2-2:於地毯上進行弧線軌跡追蹤任務之軌跡圖	63
圖 80	實驗組 2-2:於地毯上進行弧線軌跡追蹤任務之移動誤差圖	63
圖 81	實驗組 3-1:於磨石地磚上進行直線軌跡追蹤任務之速度及指向圖	64
圖 82	實驗組 3-1:於磨石地磚上進行直線軌跡追蹤任務之軌跡圖	64
圖 83	實驗組 3-1:於磨石地磚上進行直線軌跡追蹤任務之移動誤差圖	64
圖 84	實驗組 3-2:於磨石地磚上進行弧線軌跡追蹤任務之速度及指向圖	65
圖 85	實驗組 3-2:於磨石地磚上進行弧線軌跡追蹤任務之軌跡圖	65
圖 86	實驗組 3-2:於磨石地磚上進行弧線軌跡追蹤任務之移動誤差圖	65
圖 87	實驗組 4-1:於傾斜木板上進行直線軌跡追蹤任務之速度及指向圖	66
圖 88	實驗組 4-1:於傾斜木板上進行直線軌跡追蹤任務之軌跡圖	66
圖 89	實驗組 4-1:於傾斜木板上進行直線軌跡追蹤任務之移動誤差圖	66
圖 90	實驗組 4-2:於傾斜木板上進行弧線軌跡追蹤任務之速度及指向圖	67
圖 91	實驗組 4-2:於傾斜木板上進行弧線軌跡追蹤任務之軌跡圖	67
圖 92	實驗組 4-2:於傾斜木板上進行弧線軌跡追蹤任務之移動誤差圖	67
圖 93	實驗 1:碰撞偵測及動態軌跡規劃之速度及指向圖	69
圖 94	實驗 1:碰撞偵測及動態軌跡規劃之軌跡圖	69
圖 95	實驗 2:碰撞偵測及動態軌跡規劃之速度及指向圖	70
圖 96	實驗 2:碰撞偵測及動態軌跡規劃之軌跡圖	70
圖 97	實驗 3:碰撞偵測及動態軌跡規劃之速度及指向圖	71
圖 98	實驗 3:碰撞偵測及動態軌跡規劃之軌跡圖	71



第一章 緒論

1.1 研究動機

自從機器人走入家庭環境以來，機器人在室內環境中的定位以及控制一直是重要的議題。

對於機器人的定位問題，現今已有多種不同的解決方法，分別使用各式不同的工具來量測並且推算出機器人的方位，例如：輪式編碼器、慣性導航系統、CCD攝影機、紅外線收發器、RFID、超音波測距儀以及雷射掃描器等[1-4]；然而這些儀器在實際使用上仍然有相當的限制，例如：輪式編碼器會因為輪子打滑所產生累積誤差、慣性導航系統(陀螺儀、加速度儀)容易因為精確度不足而產生誤差，而視覺定位導航系統(CCD攝影機)則是因為資料量過大、演算法複雜而且容易受到環境光線變化影響，所以其準確性和實用性並不高。因此在本文中，我們將使用精確度較高的光流影像感測器(雷射滑鼠)來偵測機器人的運動速度和移動軌跡，因為它是直接量測地面的相對移動資訊，因此不會受到輪子打滑和環境光線變化的影響，所以量測誤差較小；並且在光學滑鼠的大量普及使用之下，現今已成為一低價位之產品，因而大幅降低機器人定位系統所需之成本。

接下來，談論到機器人的控制問題，目前常見的室內機器人多為輪式移動型態；但是，輪式移動型態的機器人是一個容易受打滑現象影響的系統。對於輪式機器人的移動軌跡控制，一般經常使用的方法多為時間性(Temporal or Time-based)軌跡控制法[5]，但此類的方法容易受到時間延遲而產生控制誤差。在本文中，我們則是使用空間性(Spatial or Space-based)軌跡控制法，此種方法將機器人的移動路徑定義為幾何空間的函數而非時間的函數，因此不會受到時間延遲而造成控制誤差，也比較不容易受到系統不穩定特性和外部雜訊的影響。另外，我們也使用極值三角搜尋法(Triangular Search Extremum Seeking Algorithm)來進行機器人的運動速度控制，此速度控制方法為一適應性方法，故可以在多種不同環境下進行即時的速度控制，以改善空間性軌跡追蹤法於實際運用時的控制成效。

接著，談論到機器人的碰撞偵測及動態軌跡規劃問題，一般室內機器人通常使用紅外線感測器、雷射測距儀或是觸碰開關等來偵測機器人是否遇到障礙物而無法前進[6]，但是這些儀器除了價格昂貴之外也很容易損壞。本文則是利用光流影像感測器可以量測輪式機器人即時運動速度的特性，來偵測該輪式機器人是否與路徑上的障礙物發生碰撞。當碰撞發生時，我們所使用的空間性軌跡追蹤法可以即時地規劃一條新的路徑，以使該輪式機器人可以避開障礙物，並回到原先要求的路徑上，繼續往目的地前進。

常見的輪式機器人通常有3到4個輪子，其中有1到2個輪子為轉向輪(steering wheel)；此種輪式機器人進行運動時，若要執行一邊移動一邊轉動的運動行為，所需的迴轉空間相當的大；另外，此類機器人也不能直接往側方向移動，操作起來有很多不方便的地方。在本文中，我們將使用全向輪(Omnidirectional Wheel)的技術以取代傳統的輪式移動模式，使機器人具有較高的運動自由度，可以在家庭環境中靈活的移動。



1.2 研究目標

本論文的研究目標如下：

1. 使用全向輪來建構一個擁有高運動自由度的全方向運動平台。
2. 使用光流感測器來量測該全方向運動平台的即時運動速度和移動軌跡。
3. 發展一套控制系統，使得該全方向運動平台於家庭環境中，可以在多種不同且未知的地面上進行速度控制和軌跡追蹤，同時，擁有碰撞偵測和動態軌跡規劃的功能。
4. 最後，配合嵌入式網路技術，使得該全方向運動平台能透過無線網路的機制來達到遠端操控和分散式運算的目標。

1.3 序言

本論文主要包含了三個部分，分別是演算法理論的介紹、實驗平台軟硬體架構的介紹與演算法的驗證。底下大致描述這三個部分的內容：

第一部分：

第二章：介紹可即時速度控制之空間性軌跡追蹤方法。

第三章：介紹光流影像偵測定位法。

第四章：介紹全方向運動平台。

第二部分：

第五章：介紹實驗平台完整架構。

第三部分：

第六章：演算法在實驗平台上的驗證。

第七章：結論與未來展望。



第二章 即時速度控制之空間性軌跡追蹤法

2.1 離散型極值三角搜尋法

在這一章裡，我們將會介紹一種空間性軌跡追蹤法，它可以將二維的軌跡追蹤問題轉換成一個純量調整問題，因而降低了機器人平台進行軌跡追蹤的難度。但是，該方法中的速度控制部分必須以輪子不發生打滑及馬達特性不失真為前題才能成立；因此，在實際狀況下該方法是會有缺失的。

所以，我們將會介紹一種離散型極值三角搜尋法，它可以適應性地自動調整馬達控制輸入值，使得機器人平台可以在多種不同而且未知的地面上以一預設之速度移動前進，進而改善前述空間性軌跡追蹤法之缺點。

2.1.1 極值搜尋法(Extremum Seeking Algorithm, ESA)

對於線性與非線性系統的適應性控制議題，主流的方法只能針對已知的參考資訊點進行分析處理。在一些應用中，參考資訊與系統輸出的對應關係存在著極值(極大值或是極小值)，而控制的目標就是要找到最佳的參考資訊點使得系統的輸出保持在極值。對於無法確定參考資訊與系統輸出之間的對應關係時，則需要使用適應性的搜尋方法來找到適當的參考資訊點使得系統輸出可以被極值化；這一類的問題被稱為 extremum seeking control 或是 self-optimizing control，在 1950 及 1960 年代時非常地盛行[7]，其中，利用弦波擾動(sinusoidal perturbation)則是最受歡迎的一種方法。

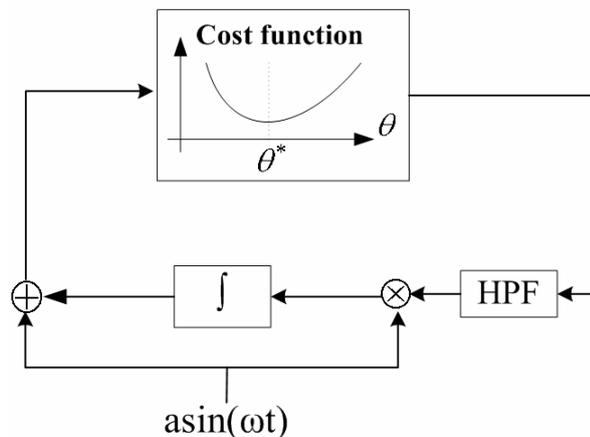


圖 1 某系統之成本函數

一開始我們先以文字簡單地敘述一下 Extremum Seeking Algorithm 的工作原理。圖 1 中的曲線是某個系統的成本函數(cost function)，我們的控制目標是找到這個成本函數的極小值(f^*)。 $a \sin \omega t$ 是加入的弦波擾動訊號，我們將此擾動訊號與參考資訊(θ)相加，然後輸入到此系統中，然後可以得到此系統中參考資訊(θ)與成本函數之間的相關資訊，再經由高通濾波器(HPF: $\frac{s}{s+h}$)將不需要的訊息濾除，接著乘上 $a \sin \omega t$ 進行解調變(demodulate)的動作，便可以得到正確的傾向(ξ)，最後再經由一個積分器(低通濾波)的作用來決定新的參考資訊輸入值，如此週而復始地不斷搜尋，最後便可以找到最佳的參考資訊輸入值(θ^*)，同時可以得到該成本函數的極值(f^*)。這種方法的優點在於可以做到即時的適應性控制處理，在目標的極值大小以及成本函數皆會變動的情況下，依然可以找到極值，但缺點在於只能找出局部的極值而無法找出全域的極值。

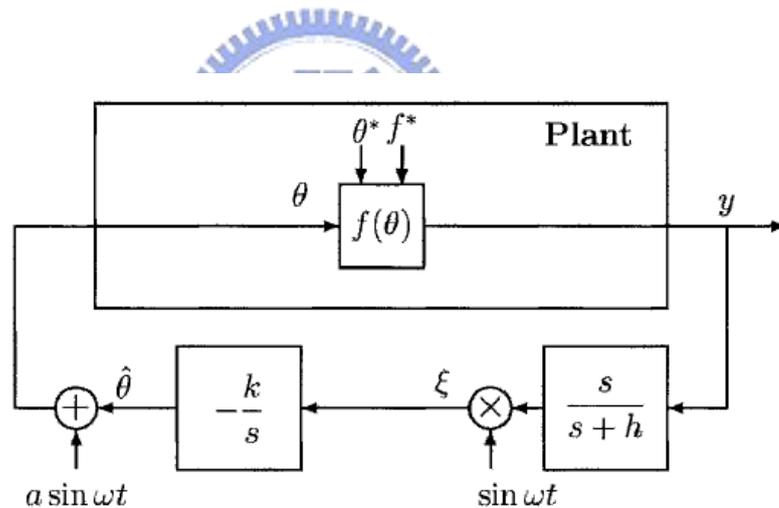


圖 2 極值搜尋法之工作迴圈

接下來，我們將以數學運算式來驗證極值搜尋法的穩定收斂性質[7]。圖 2 為一個 extremum seeking problem 的基本工作迴圈；任何可以二次微分的函式 $f(\theta)$ 均可區域性地表示成下列的形式：

$$f(\theta) = f^* + \frac{f''}{2}(\theta - \theta^*)^2 \quad (1)$$

其中為了不失一般性，我們假設 $f'' > 0$ ；若是在真實情況下 $f'' < 0$ ，我們只需將圖 2 中的 $k (k > 0)$ 替換為 $-k$ 即可。極值搜尋法的目標是讓 $\theta - \theta^*$ 盡可能的縮小，如此一來，系統的輸出 $f(\theta)$ 便會趨向於它的極值 f^* 了。一開始，一個 $a \sin \omega t$ 形式的弦波擾動訊號必須被輸入至系統當中，來幫助我們量測 $f(\theta)$ 的梯度 (gradient) 資訊。

圖 2 中的 $\hat{\theta}$ 代表的是對於未知的最佳輸入值 θ^* 的預測值，我們將預測誤差設定為：

$$\tilde{\theta} = \theta^* - \hat{\theta} \quad (2)$$

然後可以得到：

$$\theta - \theta^* = a \sin \omega t - \tilde{\theta} \quad (3)$$

將(3)式代入(1)式之後，可以得到：

$$y = f^* + \frac{f''}{2} (\tilde{\theta} - a \sin \omega t)^2 \quad (4)$$

利用基本三角關係式 $2 \sin^2 \omega t = 1 - \cos 2\omega t$ ，可以將(4)式擴展為：

$$\begin{aligned} y &= f^* + \frac{f''}{2} \tilde{\theta}^2 - af'' \tilde{\theta} \sin \omega t + \frac{a^2 f''}{2} \sin^2 \omega t \\ &= f^* + \frac{a^2 f''}{4} + \frac{f''}{2} \tilde{\theta}^2 - af'' \tilde{\theta} \sin \omega t + \frac{a^2 f''}{4} \cos 2\omega t \end{aligned} \quad (5)$$

接著，將輸出 y 值經過一個高通濾波器：

$$\frac{s}{s+h} \quad (6)$$

可以發現輸出 y 值中的 $f^* + \frac{a^2 f''}{4}$ (DC value) 將會被過濾掉：

$$\frac{s}{s+h} [y] \approx \frac{f''}{2} \tilde{\theta}^2 - af'' \tilde{\theta} \sin \omega t + \frac{a^2 f''}{4} \cos 2\omega t \quad (7)$$

接下來，將這個訊號乘上 $\sin \omega t$ 以進行解調變的動作：

$$\xi \approx \frac{f''}{2} \tilde{\theta}^2 \sin \omega t - af'' \tilde{\theta} \sin^2 \omega t + \frac{a^2 f''}{4} \cos 2\omega t \sin \omega t \quad (8)$$

利用三角關係式 $2 \cos 2\omega t \sin \omega t = \sin 3\omega t - \sin \omega t$ ，可以將(8)式擴展成：

$$\xi \approx -\frac{af''}{2} \tilde{\theta} + \frac{af''}{2} \tilde{\theta} \cos 2\omega t + \frac{a^2 f''}{8} (\sin \omega t - \sin 3\omega t) + \frac{f''}{2} \tilde{\theta}^2 \sin \omega t \quad (9)$$

因為 θ^* 是一個常數值，所以從(2)式可以得到：

$$\dot{\hat{\theta}} = -\dot{\hat{\theta}} \quad (10)$$

所以，我們可以得到：

$$\begin{aligned} \tilde{\theta} \approx \frac{k}{s} \left[-\frac{af''}{2} \tilde{\theta} + \frac{af''}{2} \tilde{\theta} \cos 2\omega t \right. \\ \left. + \frac{a^2 f''}{8} (\sin \omega t - \sin 3\omega t) + \frac{f''}{2} \tilde{\theta}^2 \sin \omega t \right] \end{aligned} \quad (11)$$

接下來，我們將(11)式中的最後一項省略掉，這是因為它是 $\tilde{\theta}$ 的二次方項，而我們只對 $\tilde{\theta}$ 的區域分析有興趣：

$$\begin{aligned} \tilde{\theta} \approx \frac{k}{s} \left[-\frac{af''}{2} \tilde{\theta} + \frac{af''}{2} \tilde{\theta} \cos 2\omega t \right. \\ \left. + \frac{a^2 f''}{8} (\sin \omega t - \sin 3\omega t) \right] \end{aligned} \quad (12)$$

從(12)式中，可以發現後面兩項是屬於高頻的訊號，當我們將此訊號經過積分器(低通濾波)作用之後，後面這兩項會被濾掉，因此可以得到：

$$\tilde{\theta} \approx \frac{k}{s} \left[-\frac{af''}{2} \tilde{\theta} \right] \quad (13)$$

或是

$$\dot{\hat{\theta}} \approx -\frac{kaf''}{2} \tilde{\theta} \quad (14)$$

因為 $kaf'' > 0$ ，所以這是一個穩定的系統；因此，我們可以推論 $\tilde{\theta} \rightarrow 0$ ，也就是 $\hat{\theta}$ 會趨近於 θ^* ，因此系統的輸出 $f(\theta)$ 便會趨向於它的極值 f^* 了。

在這裡我們只針對靜態系統(Static System)的穩定性提出驗證，而對於動態系統(Dynamic System)的穩定性討論則是在 Ref. [7]中有詳細的介紹。

2.1.2 離散型極值三角搜尋法(Discrete-time Triangular Search Extremum Seeking Algorithm, DTS-ESA)[9]

在上一小節中，我們已經介紹了極值搜尋法(ESA)的工作原理及穩定性質。極值搜尋問題的最佳解通常可以透過兩種方法來求得，一種是調整迴授控制器的參數值，另外一種則是直接調整控制輸入值；而在 Ref. [8]中，使用連續性擾動方法來調整迴授控制器參數值的問題被詳細的分析與討論，並且被應用在航空器引擎壓縮機的控制問題上。

接下來，我們將要介紹一種可以直接調整控制輸入值的離散型極值三角搜尋法(DTS-ESA)，此種方法係利用前幾次的效能量測(performance measures)來形成一個三角標示點(triangular locator)以決定下一次的控制輸入值；因此，它被稱為一種三角搜尋演算法(triangular search algorithm)。

因為三角搜尋演算法是一種自行驅動切換(self-driving switching)的系統，因此資料的平滑度以及去雜訊化是非常重要的；如此一來，效能指標才能對於演算法目前正在接近或是遠離極值點，提供有意義的資訊。

為了過濾雜訊和平滑資料，我們必須使用時間平均化(time-averaging)以及降低取樣頻率(down-sampling)這兩種方法來幫助我們達成目標；這一部分的討論在 Ref. [9]中有很詳細的介紹，因此在這裡便不再詳述。

假設有一個經過時間平均化及降低取樣頻率之後的離散時間系統，我們可以將它表示成下列的形態：

$$y_k = f_k(u_k) + v_k \quad (15)$$

其中， y_k 是系統在取樣時間為 k 時的輸出值， u_k 是控制輸入值， v_k 是量測誤差值，而非線性函式 f_k 則是代表此系統輸入與輸出之間的對應函式。

假設當控制輸入值 u_k 等於最佳解 u^* 時，輸入與輸出之間的對應函式 $f_k(u)$ 有唯一的最小值 $f_k^* = f_k(u^*)$ ；此時， $f_k'(u^*) = 0$ ，並且 $\exists \delta > 0$ 使得 $f_k'(u^* + \delta_u) > 0$ ， $f_k'(u^* - \delta_u) < 0$ ， $\forall \delta_u \in (0, \delta)$ 。

底下，我們列出三角搜尋法的遞迴公式：

$$(p_k, q_k) = \begin{cases} (u_k, p_{k-1}) & \text{if } y_k \leq y_{p_{k-1}} \\ (p_{k-1}, u_k) & \text{if } y_k > y_{p_{k-1}} \end{cases} \quad (16)$$

其中， $y_{p_{k-1}}$ 代表控制輸入 $u_k = p_{k-1}$ 時系統的輸出。而控制輸入值的更新公式則可以表示成：

$$u_{k+1} = p_k + \mu_{k+1} \text{sign}(p_k - q_k) \quad (17)$$

其中， $\mu_{k+1} > 0$ 是 step size，而 p_k, q_k 是用來記憶到目前為止最佳的控制輸入值；因為更新公式總是在最佳的控制輸入值附近變動，所以新的控制輸入值並不會離最佳值太遠。當 $y_k \leq y_{p_{k-1}}$ 時，代表找到一個新的最佳控制輸入值，因此， p_k 要被更新為 u_k ；而當 $y_k > y_{p_{k-1}}$ 時，代表舊的最佳控制輸入值仍然是目前的最佳值，所以 p_k 被更新為 p_{k-1} 。因此，下一次的 control 輸入值 u_{k+1} 將會以目前的最佳 control 輸入值 p_k 為起點，然後往可以讓輸出值 y 下降的方向(由 q_k 指向 p_k 的方向)移動 μ_{k+1} 的距離；在經過數次的尋找之後，此演算法將可以找到該系統的最佳 control 輸入值 (u^*)，同時，系統的輸出將會達到最小值 $f_k^* = f_k(u^*)$ 。

接下來，我們將會利用圖形來解釋上述的各項情況：

- (1) 圖 3 代表某系統輸入與輸出之間的對應函式 f_k 。
- (2) 如圖 4 所示，假設在取樣時間為 k 時，此演算法 DTS-ESA 中已經記憶一組 (p_k, q_k) 值，所以可以使用更新公式及 (p_k, q_k) 值來決定下一次的 control 輸入值 u_{k+1} 。
- (3) 如圖 5 所示，此時取樣時間為 $k+1$ ，因為 control 輸入值 u_{k+1} 所得到的系統輸出 $y_{k+1} \leq y_{p_k}$ ，代表找到新的最佳 control 輸入值，所以 p_{k+1} 要被更新為 u_{k+1} ， q_{k+1} 要被更新為 p_k ，接著可以使用更新公式及 (p_{k+1}, q_{k+1}) 值來決定下一次的 control 輸入值 u_{k+2} 。
- (4) 如圖 6 所示，此時取樣時間為 $k+2$ ，因為 control 輸入值 u_{k+2} 所得到的系統輸出 $y_{k+2} > y_{p_{k+1}}$ ，代表舊的最佳 control 輸入值仍然是目前的最佳值，所以 p_{k+2} 被更新為 p_{k+1} ， q_{k+2} 要被更新為 u_{k+2} ，接著可以使用更新公式及 (p_{k+2}, q_{k+2}) 值來決定下一次的 control 輸入值 u_{k+3} 。
- (5) 如圖 7 所示，經過數次的尋找之後，演算法 DTS-ESA 將可以找到該系統的最佳 control 輸入值 (u^*)，而且系統的輸出將會達到最小值 $f_k^* = f_k(u^*)$ 。

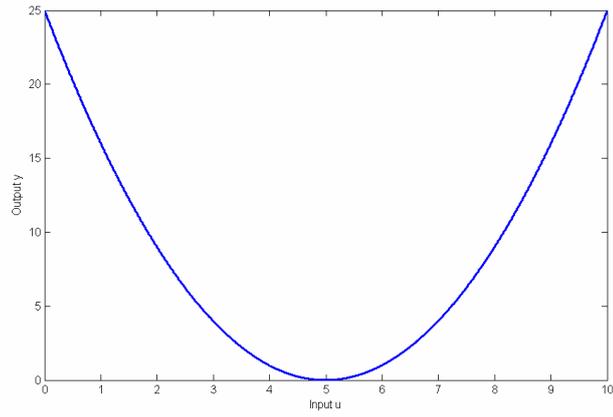


圖 3 某系統輸入與輸出之間的對應函式

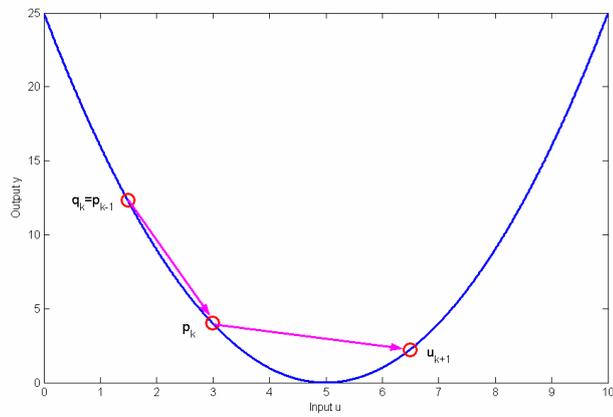


圖 4 利用 p_k 和 q_k 來決定下一次的控制輸入值 u_{k+1}

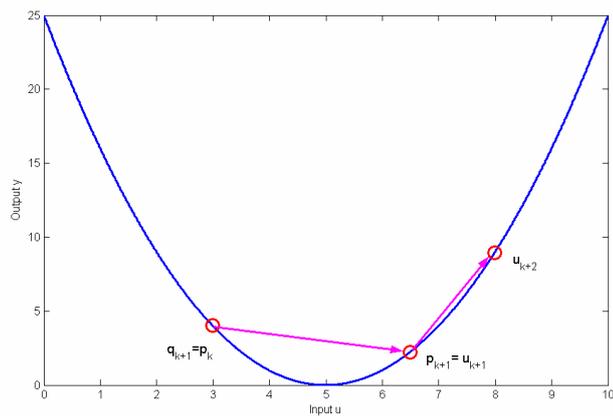


圖 5 利用 p_{k+1} 和 q_{k+1} 來決定下一次的控制輸入值 u_{k+2}

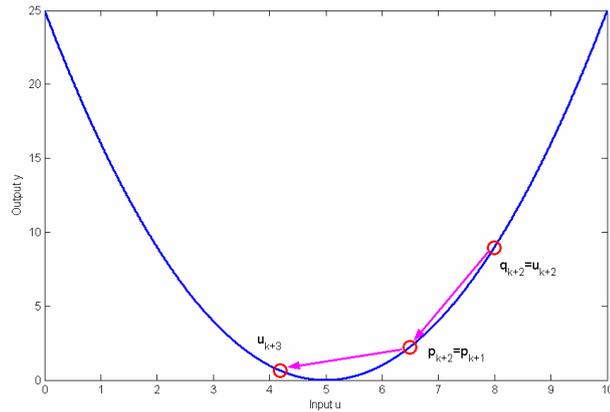


圖 6 利用 p_{k+2} 和 q_{k+2} 來決定下一次的控制輸入值 u_{k+3}

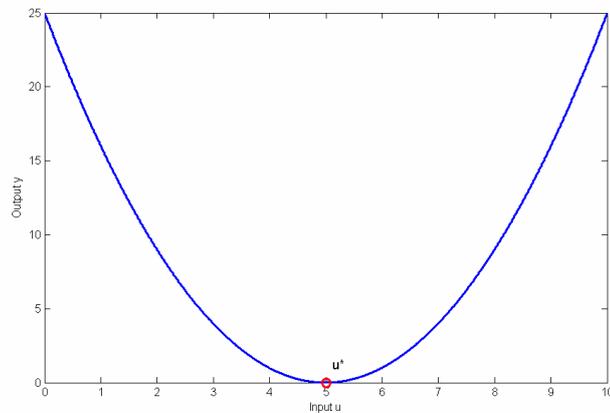


圖 7 找到最佳的控制輸入值(u^*)，

同時，系統的輸出將會達到最小值 $f_k^* = f_k(u^*)$

2.1.3 使用 DTS-ESA 進行運動平台之速度控制

在上一小節中，我們已經詳細地介紹演算法 DTS-ESA 是如何運作，以得到一個系統的最佳的控制輸入值以及最小的系統輸出值；接下來，我們將把 DTS-ESA 的使用空間維度擴展成二維，並實際應用到全方向運動平台的即時速度控制上面。

我們使用全向輪的機構以當作全方向運動平台的移動載具，全向輪可以提供該全方向運動平台較高的移動自由度(此部分將會在第四章中詳細地介紹)；但是，不可避免的，全向輪與地面之間還是會發生打滑的現象，因而造成控制該全方向運動平台的難度提高。此外，家庭環境中的地面材質通常很多樣化，而每種

材質的地面對全向輪所產生的動摩擦力及滾動摩擦力都不盡相同，因此也會影響到控制該全方向運動平台的效能。

舉例來說，理論上當我們輸入速度控制指令 $(v_{i_x}, v_{i_y})=(v_{t_x}, v_{t_y})$ 給全方向運動平台時，平台應該會以 (v_{t_x}, v_{t_y}) 這個速度在地面上行進；但是，實際上平台並不是以 (v_{t_x}, v_{t_y}) ，而是以 (v_{o_x}, v_{o_y}) 這個速度在地面上行進，如圖 8 所示。底下，我們列出在幾種地面上以及斜坡上， (v_{i_x}, v_{i_y}) 與 (v_{o_x}, v_{o_y}) 之間的關係圖形，以說明上述的情況。

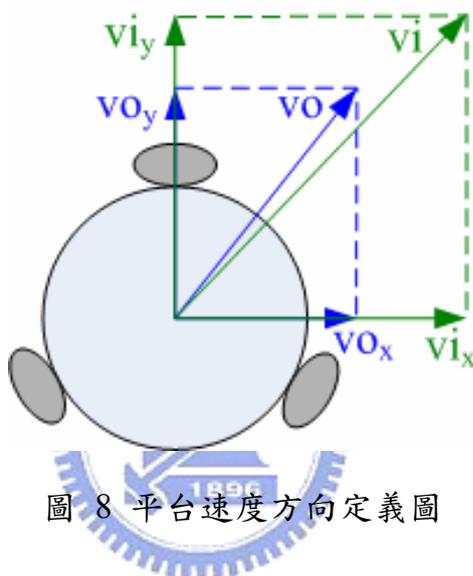


圖 8 平台速度方向定義圖

(1) 木板:

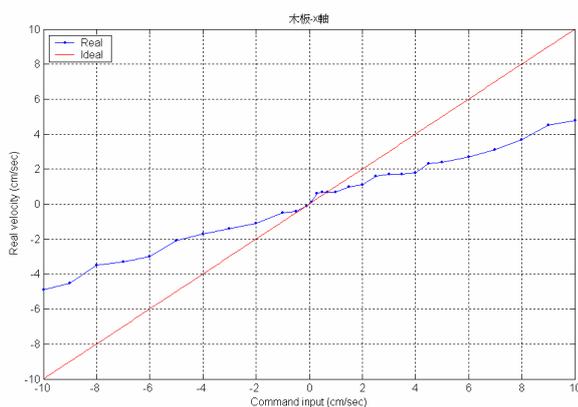


圖 9 木板上 v_{i_x} 與 v_{o_x} 的關係圖

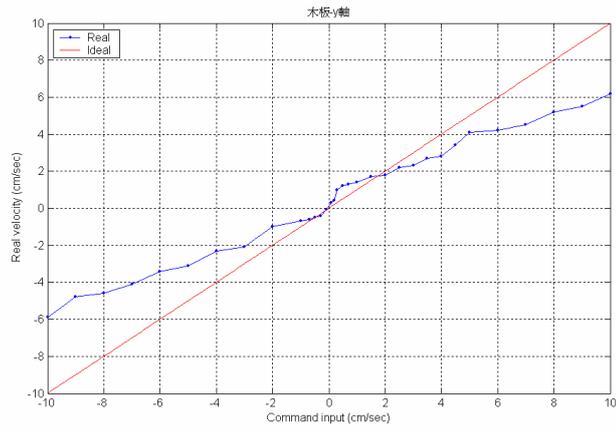


圖 10 木板上 v_{iy} 與 v_{oy} 的關係圖

(2) 地毯:

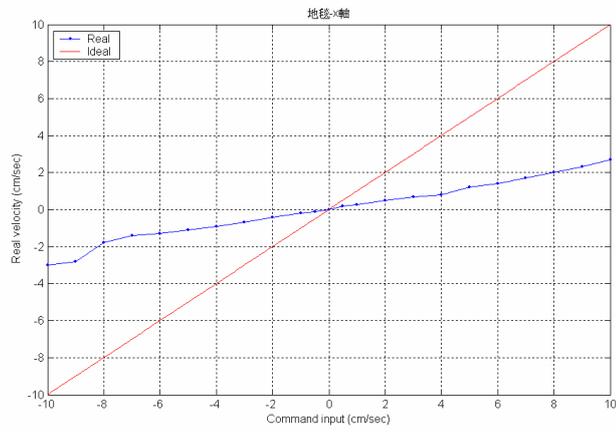


圖 11 地毯上 v_{ix} 與 v_{ox} 的關係圖

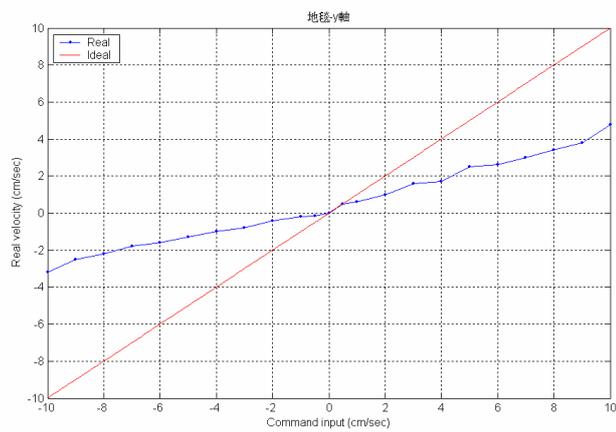


圖 12 地毯上 v_{iy} 與 v_{oy} 的關係圖

(3) 磨石地磚:

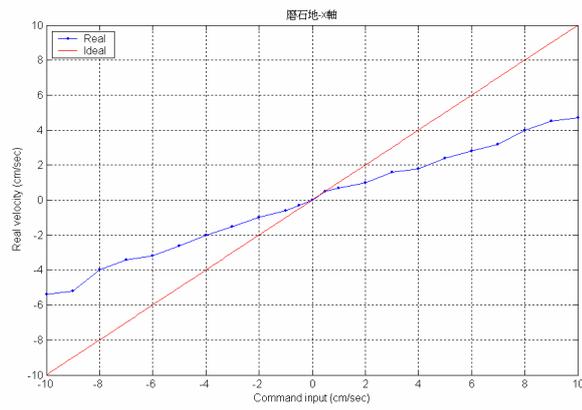


圖 13 磨石地磚上 vi_x 與 vo_x 的關係圖

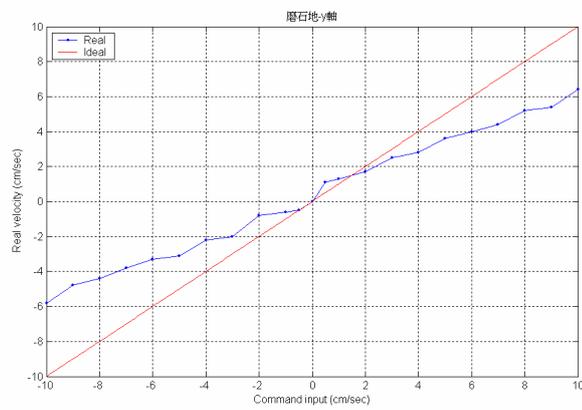


圖 14 磨石地磚上 vi_y 與 vo_y 的關係圖

(4) 有斜度的木板(坡度為 18%, 傾斜方向為 x 軸):

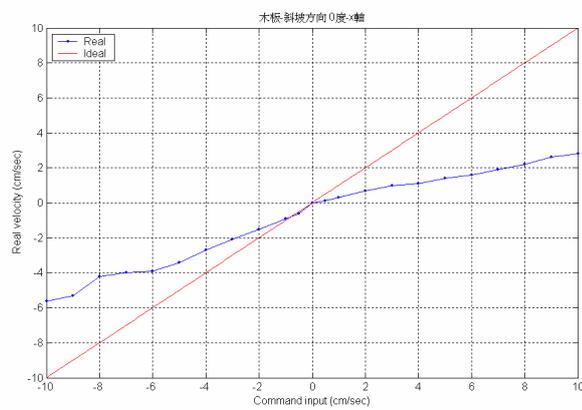


圖 15 有斜度的木板(方向為 x 軸)上 vi_x 與 vo_x 的關係圖

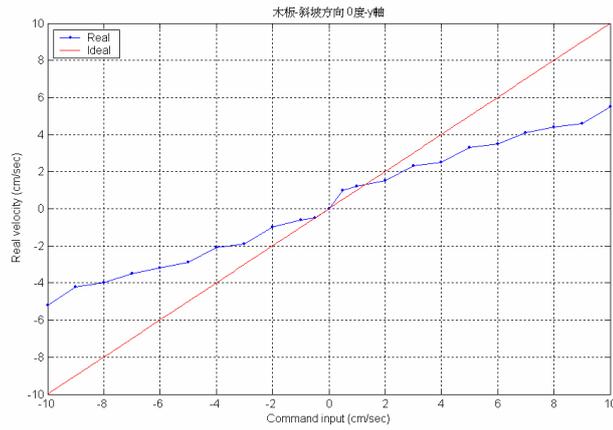


圖 16 有斜度的木板(方向為 x 軸)上 v_{i_y} 與 v_{o_y} 的關係圖

(5) 有斜度的木板(坡度為 18%, 傾斜方向為 y 軸):

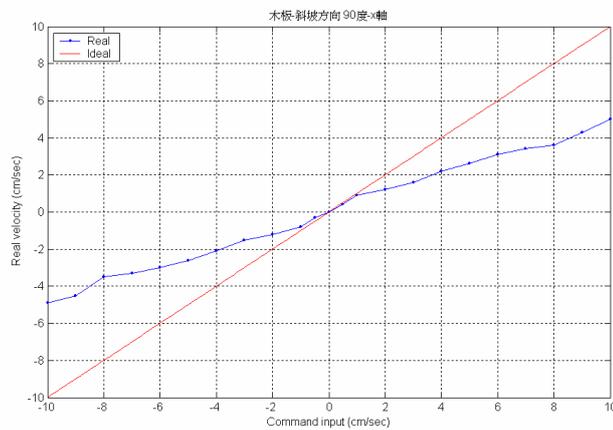


圖 17 有斜度的木板(方向為 y 軸)上 v_{i_x} 與 v_{o_x} 的關係圖

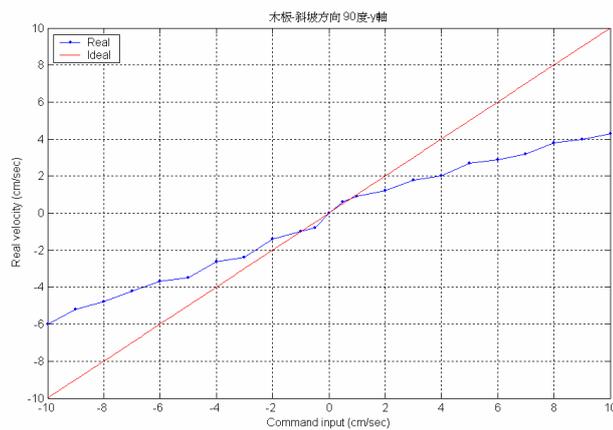


圖 18 有斜度的木板(方向為 y 軸)上 v_{i_y} 與 v_{o_y} 的關係圖

圖 9 到圖 18 中的紅色斜線代表 vi_x 與 vo_x 及 vi_y 與 vo_y 之間的理想關係曲線，而藍色折線則是代表 vi_x 與 vo_x 及 vi_y 與 vo_y 之間的實際關係曲線。經由上述實驗數據，我們發現速度控制指令 (vi_x, vi_y) 與實際運動速度 (vo_x, vo_y) 之間的關係式是會隨著地面材質的不同(圖 9 到圖 14)、及地面傾斜度的不同(圖 15 到圖 18)而改變，而且它們之間的關係式是非線性、時變、且不穩定的；所以我們並無法確定速度控制指令 (vi_x, vi_y) 與實際運動速度 (vo_x, vo_y) 之間精確的關係式，因此我們必須使用適應性的方法來解決這個控制問題。

經由前兩個小節的討論，我們知道 DTS-ESA 為一種離散型極值搜尋法，使用起來很方便，不需要繁雜的運算，而且可以根據系統的輸出表現直接調整控制輸入值；所以接下來我們將會使用 DTS-ESA 來進行全方向運動平台的速度控制。

在這裡先以一維空間來做說明；假設我們希望控制該全方向運動平台以 vt_x (單位為 cm/sec) 的速度在某種地面上行進；透過光流感測器我們可以量測到該全方向運動平台的實際運動速度為 vo_x (單位為 cm/sec)。因此，我們將該全方向運動平台系統的輸出函數(成本函數)設定為：

$$Z_x(vo_x) = (vt_x - vo_x)^2 \quad (18)$$

然後，套用演算法 DTS-ESA 來調整控制輸入值 vi_x ；最後，該演算法將可以找到最佳的控制輸入值 $vi_x = vi_x^*$ ；同時，系統的輸出函數將會達到最小值 $Z_x = Z_x^* = 0$ ，此時代表平台的實際運動速度為 vo_x 將會等於目標速度 vt_x 。

底下我們將會列出 vi_x 與 vo_x 、 vo_x 與 Z_x 以及 vi_x 與 Z_x 之間的關係圖形，用以說明 vi_x 與 vo_x 之間不同的關係式將會影響演算法最後所找到的最佳控制輸入值 vi_x^* 。

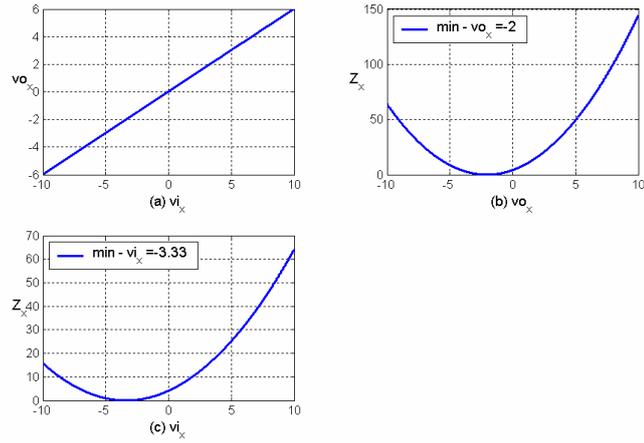


圖 19 v_{i_x} 與 v_{o_x} 之間呈線性遞增關係

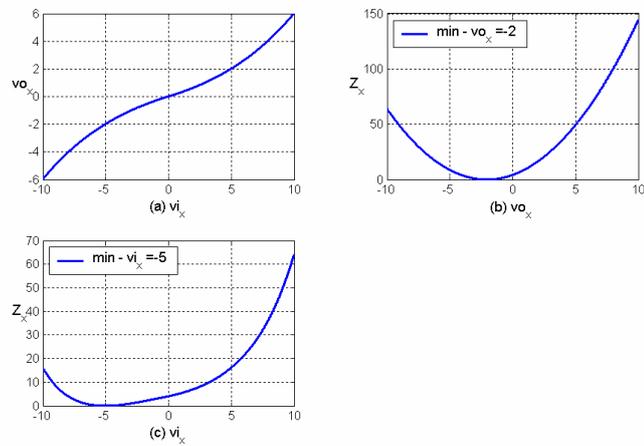


圖 20 v_{i_x} 與 v_{o_x} 之間呈非線性遞增關係

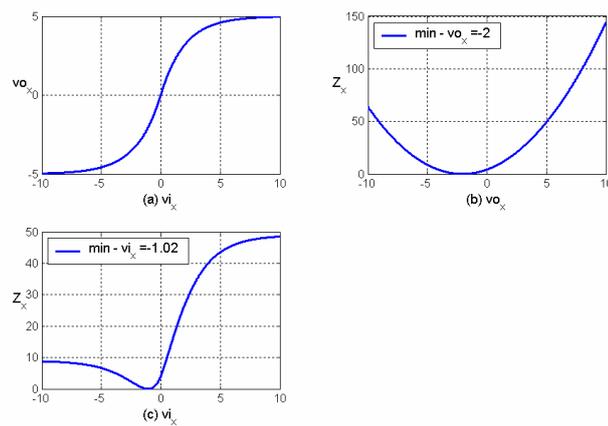


圖 21 v_{i_x} 與 v_{o_x} 之間呈非線性遞增關係

經由上述的實驗數據，我們發現如果 vi_x 與 vo_x 之間呈線性遞增或非線性遞增關係(如圖 19 ~ 圖 21 中(a)部分所示)，我們所定義的輸出函數 $Z_x(vo_x)$ 仍然是一個凸形(convex)函數，存在唯一的最小值，而且 vi_x 與 vo_x 之間的關係式將會影響該輸出函數 $Z_x(vo_x)$ 最小值的發生位置(如圖 19 ~ 圖 21 中(c)部分所示)。

上面的討論僅針對平台 X 軸上的運動控制，若是希望擴展到二度空間(X 軸及 Y 軸)，僅需要將該平台系統的輸出函數(成本函數)設定為：

$$Z(vo_x, vo_y) = (vt_x - vo_x)^2 + (vt_y - vo_y)^2 \quad (19)$$

如圖 22 所示(其中假設 $(vt_x, vt_y) = (5, 5)$)；然後，使用演算法 DTS-ESA 來輪流調整控制輸入值 vi_x 和 vi_y ；最後，該演算法將可以找到最佳的控制輸入值 $(vi_x, vi_y) = (vi_x^*, vi_y^*)$ ；同時，系統的輸出函數將會達到最小值 $Z = Z^* = 0$ ，代表平台的實際運動速度 (vo_x, vo_y) 將會等於目標速度 (vt_x, vt_y) 。

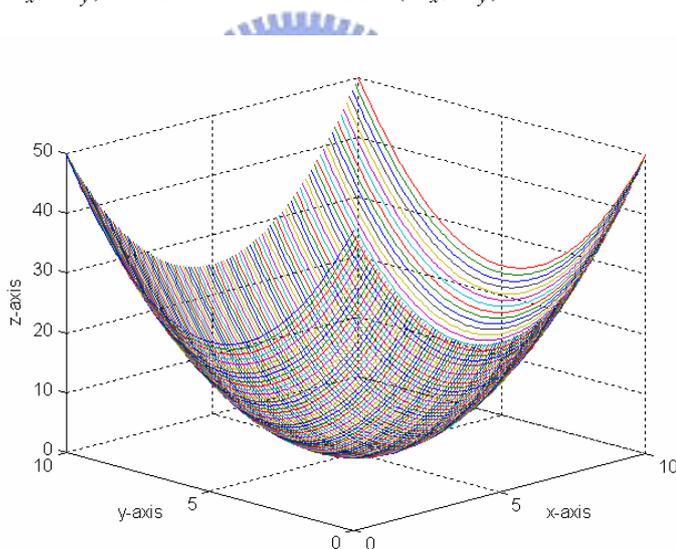


圖 22 雙輸入變數的平台系統輸出函數

從圖 9 到圖 18，我們可以發現在家庭環境的各種地面上，速度控制指令 (vi_x, vi_y) 與實際運動速度 (vo_x, vo_y) 之間的關係式，雖然為非線性、時變、而且會隨著地面性質的不同而改變，但都存在著遞增關係；因此，可以使用 DTS-ESA 演算法來進行速度控制。

2.2 空間性軌跡追蹤法

2.2.1 時間性軌跡追蹤法(Temporal Path Tracking Method, TPTM)

一開始，我們先介紹相對於空間性軌跡追蹤法的時間性軌跡追蹤法；所謂的時間性軌跡追蹤法，是指在進行機器人平台的軌跡控制時，必須將希望行進的軌跡定義成一個時間的函數，這個函數中定義在某個時間機器人必須到達某個地點；理論上，機器人平台就會根據這個函數，在規定的時間到達規定的地點。但是實際上，這是一種非直接的方法，很容易因為平台的動態表現或外界因素影響而產生無法預期的結果；例如，平台的輪子卡到障礙物暫時無法前進、或是將平台操作在一個斜坡上、或者是要求的輸出速度已經超過馬達驅動器可以達到的最高轉速...等現象，都可能會造成時間點延遲而產生“切角(cut corners)”的問題。如圖 23 所示，圖中的弧形為使用者希望機器人平台行進的軌跡，將此軌跡轉換成時間的函數後，代表機器人平台必須在 $t=1\sim 5$ 分別到達圖中 $a\sim e$ 各點。假設在 $t=3$ 且平台到達 c 點的時候，平台的輪子突然卡到障礙物而無法繼續前進；當時間到達 $t=5$ 的時候，平台突然又可以移動；此時，平台會往 \overline{ce} 方向前進而到達 e 點。因此，造成平台移動軌跡在 c 點至 e 點之間出現了一個「切角」。

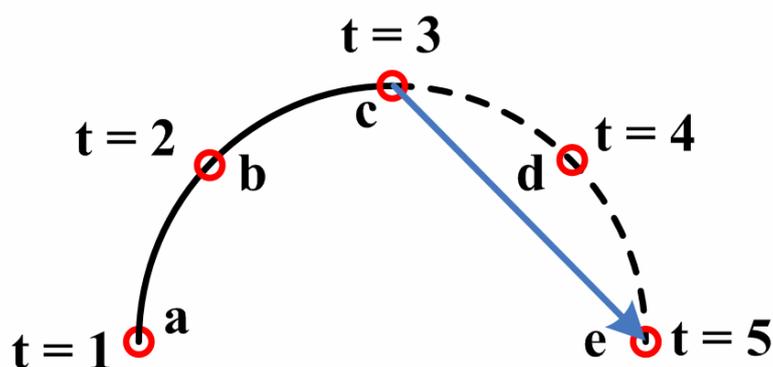


圖 23 切角(cut corners)

另外，此種參數化方法對於沿著運動軌跡進行即時速度改變的情況使用起來並不方便；這是因為每當下達一個新的速度指令時，這個方法必須對剩下的軌跡路徑進行重新計算，以求得適用於新的速度指令的軌跡時間函數。

2.2.2 空間性軌跡追蹤法(Spatial Path Tracking Method, SPTM)[10]

相對於時間性軌跡追蹤法，空間性軌跡追蹤法是指在進行機器人平台的軌跡控制時，必須將希望行進的軌跡定義成一個幾何空間的函數。

在這裡我們將介紹一種非線性空間軌跡追蹤法，這個方法可以將輪式機器人控制在空間中的某個軌跡上，同時，驅動該輪式機器人以一設定之速度往此軌跡的終點前進。

此軌跡追蹤控制器(簡稱為 ε -Controller 或 C_ε)完全建立在下列的空間資訊上，包含運動軌跡的幾何空間描述和輪式機器人所迴授的目前位置資訊，以及預先設定好的移動速度。

首先， C_ε 是建立在一個右手型慣性卡式座標系統(Inertial Cartesian coordinate System, ICS)之上；因此，所有由 C_ε 所產生的向量數值都是相對於該座標系統，如圖 24 中下標為 I 的 x 軸和 y 軸所示。

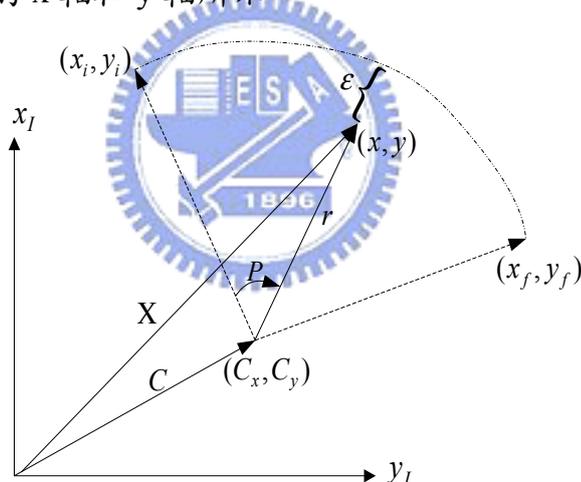


圖 24 軌跡及軌跡相關量值

C_ε 中定義所有要求行進的軌跡都是由弧形線段所組合而成的；而直線是其中的一個特例，可以被近似為一段擁有極大半徑的弧形線段。因此，任何複雜的軌跡都可以由基本的弧形線段所組合而成。

在這樣的定義之下，任何的弧形軌跡都可以由它的起始點 $X_i = [x_i \ y_i]^T$ ，終止點 $X_f = [x_f \ y_f]^T$ 以及帶正負號的半徑 R (根據右手定則，帶正號的 R 代表往順時鐘方向行進) 來進行描述；經由這三個量值就可以定義出該弧形軌跡 C 的中心點位置 (C_x, C_y) 。如圖 24 所示，向量 $X = [x \ y]^T$ 代表機器人平台在 ICS 中的真實位置。

另外，從弧形中心點 (C_x, C_y) 到真實位置 (x, y) 的距離向量 r 在 ICS 中的定義為：

$$r = X - C = \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} \quad (20)$$

而圖 24 中的 P 代表從起始點 X_i 移動到終止點 X_f 的移動進度，它永遠是一個正數值；它是用來同步控制機器人平台的指向，以達成包含指向控制的空間軌跡追蹤任務。

接下來，將要介紹 C_ε 中最主要的一個變數 ε ，它代表機器人平台偏離軌跡的量值，也就是從 $X = [x \ y]^T$ 到弧形軌跡上最近的點之間的距離：

$$\varepsilon = |R| - \|r\| \quad (21)$$

如(21)式所示，當 ε 為正值時，代表機器人平台在弧形軌跡之「內」，而當 ε 為負值時，代表機器人平台在弧形軌跡之「外」，而當 ε 量值為 0 時，代表機器人平台正好在弧形軌跡之上。依據 ε ， C_ε 可以產生慣性速度組合點(inertial velocity setpoints) V_i^* 。

如圖 25 所示，我們先列出此空間性軌跡追蹤法的整體架構圖，用以說明此空間性軌跡追蹤法的整體運作情形。其中，MakeSetPoint(MSP)演算法可以將慣性速度指令轉換成機器人平台的輪子轉速指令(此部分將會在 4.2 節中詳細介紹)；但值得注意的是，這必須以輪子不會發生打滑現象為前提。藉由 V_i^* 及運動學分析，MSP 可以產生相對應的驅動速度指令給每一顆輪子的馬達控制器；而 C_w 控制器則是會提供馬達控制訊號 E 給機器人平台上的馬達；至於 C_ψ 控制器則是用來控制機器人平台的指向。

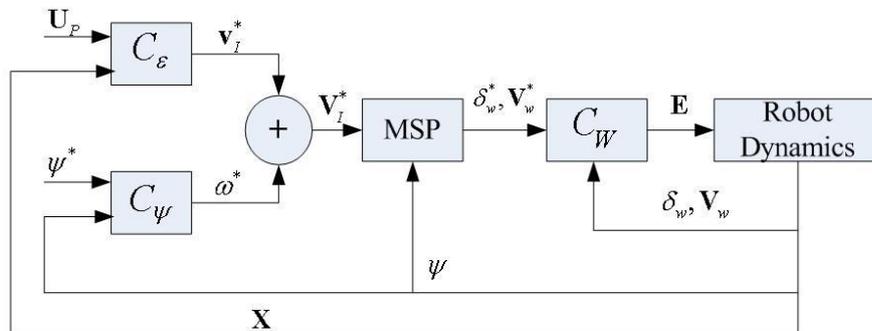


圖 25 空間性軌跡追蹤法的控制架構圖

$$d\mathbf{V}_i^* = \mathbf{V}_n + \mathbf{V}_t \text{sign}(R) \quad (22)$$

接下來，如(22)式所示，我們定義 $d\mathbf{V}_i^*$ 為平行速度向量 \mathbf{V}_n 及垂直速度向量 \mathbf{V}_t 兩者之合；其中， $d\mathbf{V}_i^*$ 代表的是 \mathbf{V}_i^* 的方向，而 sign 項會依據移動路徑的方向而改變 \mathbf{V}_t 的方向。

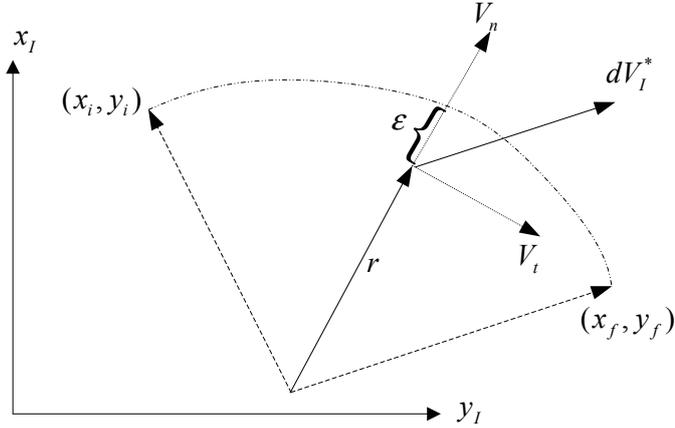


圖 26 ε -Controller 的相關量值

如圖 26 所示，平行速度向量 \mathbf{V}_n 代表的是 C_ε 中的修正動作項，是用來降低偏移誤差量 ε ；在這裡，我們套用 PI 控制器的型式，因此，可以表示成：

$$\mathbf{V}_n = \left(K_p \varepsilon(P) + \frac{1}{2} K_i \int_0^P (R^2 - \|r\|^2) dP \right) \left[\frac{r}{\|r\|} \right] \quad (23)$$

其中， K_p 和 K_i 為控制器的增益值；而 $\left(\frac{r}{\|r\|} \right)$ 則是用來指定 \mathbf{V}_n 的方向。

而垂直速度向量 \mathbf{V}_t 被定義成：

$$\mathbf{V}_t = (V_d - \|\mathbf{V}_n\|) \left(\frac{r_t}{\|r_t\|} \right) \quad (24)$$

其中， $r_t = [-(y - C_y) \quad (x - C_x)]^T$ ， V_d 為設定的平台行進速度，而且 $(V_d - \|\mathbf{V}_n\|)$ 被限制不能為負數。這樣的限制條件使得 $\|\mathbf{V}_t\|$ 成為一個 ε 的函數；當 ε 增加時， $\|\mathbf{V}_t\|$ 將會趨近於 0，而當 ε 減少至 0 時， $\|\mathbf{V}_t\|$ 將會趨近於 V_d 。

因此， \mathbf{V}_i^* 便可以表示成：

$$\mathbf{V}_i^* = V_d \frac{d\mathbf{V}_i^*}{\|d\mathbf{V}_i^*\|} \quad (25)$$

(25)式代表的是一個軌跡追蹤控制器，當機器人平台偏離軌跡時，它可以引導該平台回到軌跡上距離現在位置最近的點；而當機器人平台接近軌跡時，該控制器將會產生一個垂直軌跡的速度向量，以驅使平台往軌跡終止點前進。

最後，值得注意的是，這裡所提出的 ε -Controller 是一種運作在純量 ε 上的非線性控制器；它可以有效地將二維的軌跡追蹤問題轉換成一個相對於 ε 的純量調整問題，因而降低了機器人平台進行軌跡追蹤的難度。

2.2.3 即時速度控制之空間性軌跡追蹤法(Real-time Velocity Control - SPTM, RVC-SPTM)

經由上一小節的討論，我們知道空間性軌跡追蹤法(ε -Controller or C_ε)是一種運作在純量 ε 上的非線性控制器；它可以將二維的軌跡追蹤問題轉換成一個相對於 ε 的純量調整問題，因而有效地降低機器人平台進行軌跡追蹤的難度。

但是，經過實際的運用之後，我們發現此空間性軌跡追蹤法的控制架構中是有可以改善的地方的。圖 25 中的 MSP(MakeSetPoint)演算法負責將 ε -Controller 所產生的慣性速度指令轉換成機器人平台的輪子轉速指令，但這必須以輪子不發生打滑現象而且馬達特性不失真為前提才能成立；但是，實際運作時全向輪與地面之間很容易發生打滑現象，而且每種地面對輪子產生的動摩擦力和滾動摩擦力都不相同；另外，每種不同特性的地面也都會對馬達特性產生不同程度的影響。因此，MSP 對於空間性軌跡追蹤法的整個控制架構而言是關鍵而且重要的部分。

在這裡，我們提出一個改良過的空間性軌跡追蹤法，名稱為「即時速度控制之空間性軌跡追蹤法」(Real-time Velocity Control - SPTM, RVC-SPTM)。

RVC-SPTM 與原本的空間性軌跡追蹤法(SPTM)比較起來，最大的改良之處在於 RVC-SPTM 中使用離散型極值三角搜尋法(DTS-ESA)的概念取代原本 SPTM 中的 MSP 演算法；我們稱之為 Modified-MSP(MMSP)演算法。

如圖 27 所示，MMSP 演算法是使用 DTS-ESA 來進行機器人平台的即時速度控制；利用光流影像所偵測到平台的即時移動速度，然後迴授至 MMSP 演算法中的 DTS-ESA 進行控制，使得該機器人平台的實際移動速度能與 C_ε 所指定的慣性速度指令一致；進而增加空間性軌跡追蹤法的精準度及實用性。

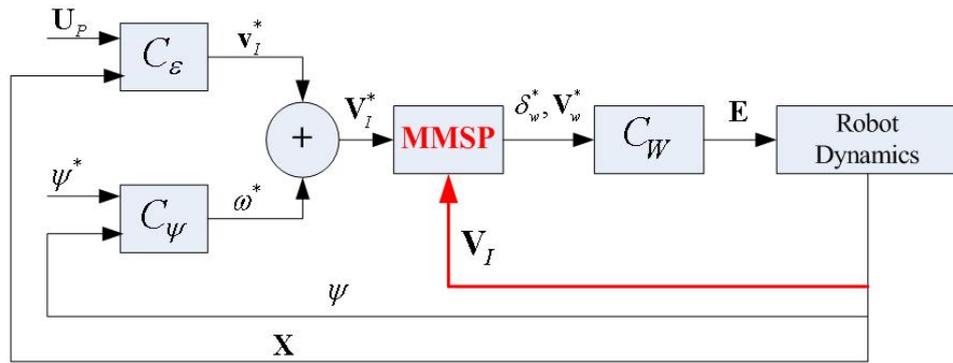


圖 27 RVC-SPTM 的控制架構圖

2.2.4 使用 RVC-SPTM 進行運動平台之軌跡控制

如上一個小節所述，改良過後的 RVC-SPTM 不但擁有空間性軌跡追蹤的功能，更擁有可以適應於多種地面的即時速度控制功能。

因此，學生將 RVC-SPTM 使用在我們所設計的全方向輪式運動平台的軌跡控制上面；讓使用者可以更精確地控制該平台以一個設定的速度進行軌跡追蹤的任務。

圖 28 為 RVC-SPTM 控制演算法的控制流程圖，以下為流程圖說明。

- (1) 在「Start」階段，使用者先設定好平台的移動軌跡資訊，包含軌跡起始點、終止點及半徑，平台起始指向、終止指向、轉動方向以及平台的移動速度。
- (2) 在「 C_ψ 」階段， C_ψ 將會根據使用者設定的平台指向資訊，計算出平台當前的目標指向。
- (3) 在「Spin or Move + Spin」階段，將會根據 C_ψ 所計算出的平台目標指向以及平台的實際指向來判斷是否需要進行旋轉。另外，系統還會根據目前是否有在移動(Move)，而進行不同性質的旋轉。一開始，因為平台並未開始移動，所以系統僅進行單純的旋轉；而當平台開始移動之後，系統便會進行一邊移動一邊旋轉的動作(僅有全向輪平台可以達成此動作)，以增加平台的動作流暢度。同時，系統會判斷平台的指向是否已經達到目標指向；若未達到則繼續進行旋轉；若已達到，則進入下一階段的動作。
- (4) 在「 C_ϵ 」階段， C_ϵ 將會根據使用者設定的軌跡資訊、速度資訊以及平台的實際位置資訊，計算出平台當前的目標速度及其方向。

(5) 在「MMSP」階段，MMSP 會調整平台的速度，以使平台的實際速度趨近於 C_e 所計算出的目標速度。在此階段，為了維持軌跡追蹤的效率，我們將會限制 MMSP 的調整次數；若 MMSP 在限定的次數之內尚未將實際速度調整至目標速度，此時將會結束此階段的動作，直接回到 C_v 階段。若 MMSP 在限定的次數之內將實際速度調整至目標速度，此時則會進入下一階段「Move」進行定速移動。

(6) 在「Move」階段，因為平台的實際移動速度等同於目標速度，因此平台只需進行定速運動；在持續了一段設定的時間之後，系統會判斷是否已經到達使用者定義的軌跡終止點；若已到達終點，系統則進入下一個階段「Next Path」；若未到達終點，系統則會回到 C_v 階段。

(7) 在「Next Path」階段，因為平台已經到達某一段路徑的終止點，此時系統會判斷使用者是否有指定下一段新的移動路徑，如果有指定新的路徑，系統則會回到「Start」階段以執行新的軌跡追蹤任務，如果沒有指定的話，系統則會進入最後階段「Stop」。

(8) 在「Stop」階段，因為平台已經到達軌跡終止點，因此平台會停止運動而停留在該點。



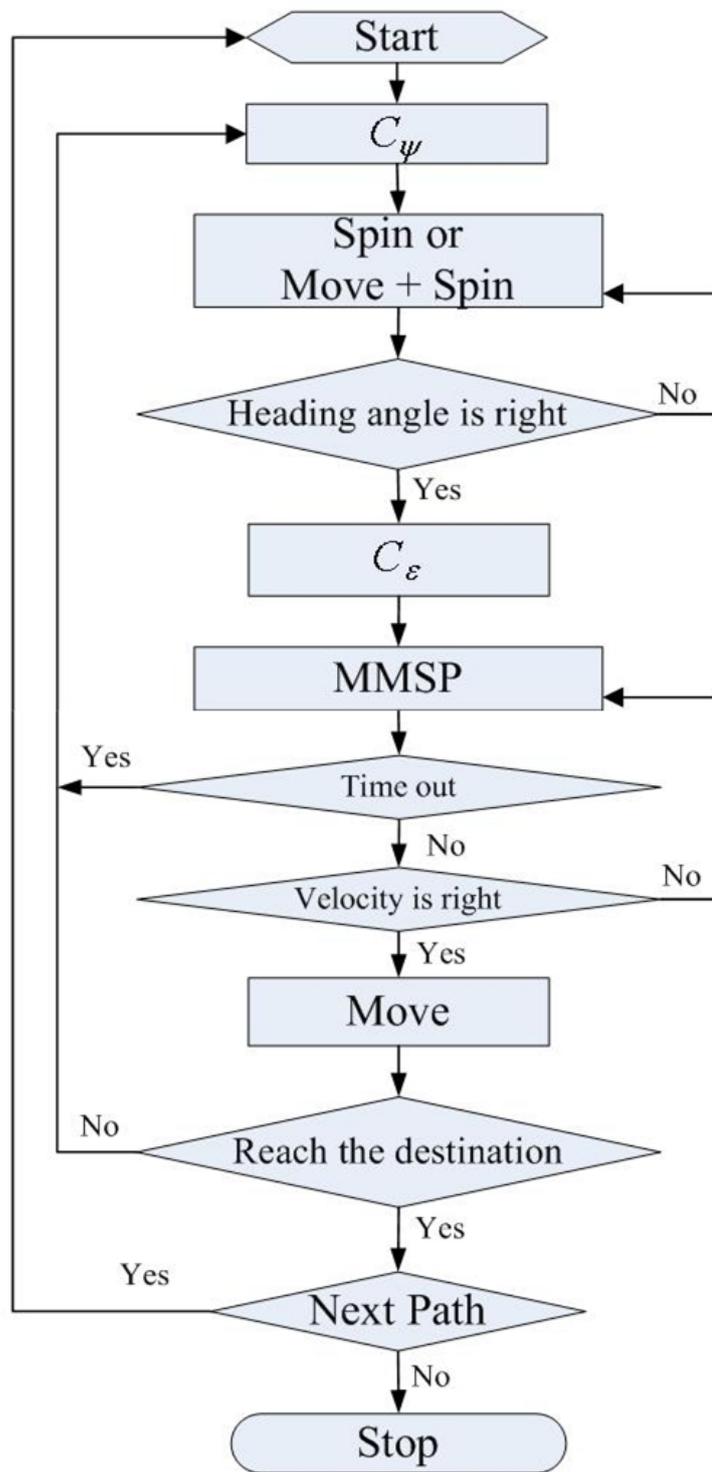


圖 28 RVC-SPTM 的控制流程图

2.2.5 利用 RVC-SPTM 進行碰撞偵測及動態路徑規劃

一般常見的室內輪式機器人通常係使用紅外線感測器、雷射測距儀或是觸碰開關等儀器來偵測機器人是否遇到障礙物而無法前進，但是這些儀器除了價格昂貴之外也很容易損壞。本文則是利用光流影像感測器可以量測輪式機器人即時移動速度的特性，來偵測該輪式機器人是否與路徑上的障礙物發生碰撞。當碰撞發生時，我們所使用的空間性軌跡追蹤法可以即時地規劃一條新的路徑，以使該輪式機器人可以避開障礙物(注意，如果在該避障路線上再次碰撞到障礙物時，此時系統將會即時地規劃另一條新的避障路徑，以使機器人可以順利的避開障礙物)，並回到原先要求的路徑上，繼續往目的地前進。

一開始，我們先來討論當一個運動平台與障礙物發生碰撞時，會對該物體的運動特性產生怎樣的影響。

如圖 29 所示，在理想狀況(兩者接觸面無摩擦力產生)之下，當平台的運動方向與障礙物的接觸面呈垂直狀態時，此時障礙物會給該平台一個與其運動速度方向相反、大小相同的反作用力，該平台會因為合力為零而靜止不動。如圖 30 所示，在理想狀況(兩者接觸面無摩擦力產生)之下，當平台的運動方向與障礙物的接觸面呈非垂直狀態時，此時障礙物會給該平台一個反作用力，而反作用力的方向是在與障礙物接觸面呈垂直的方向上；此時該平台所受的合力方向與原始運動方向不同，因此該平台的運動方向將會改變。

而當考慮實際狀況，也就是運動平台與障礙物之間存在摩擦力的情形下；當平台的運動方向與障礙物的接觸面呈垂直狀態時，此時障礙物會給該平台一個與其運動速度方向相反、大小相同的反作用力，該平台會因為合力為零而靜止不動。如圖 31 所示，當平台的運動方向與障礙物的接觸面呈非垂直狀態時，此時障礙物除了會給該平台一個反作用力，該反作用力的方向是在與障礙物接觸面呈垂直的方向上，而且還會給該平台一個摩擦力，該摩擦力的方向是在與障礙物接觸面呈平行的方向上；此時該平台所受的合力方向與原始運動方向不同，因此該平台的運動方向將會改變。

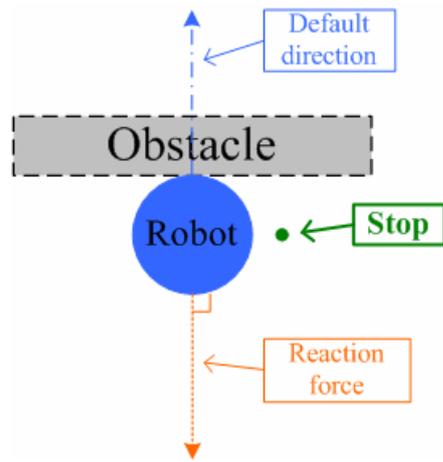


圖 29 碰撞偵測_1

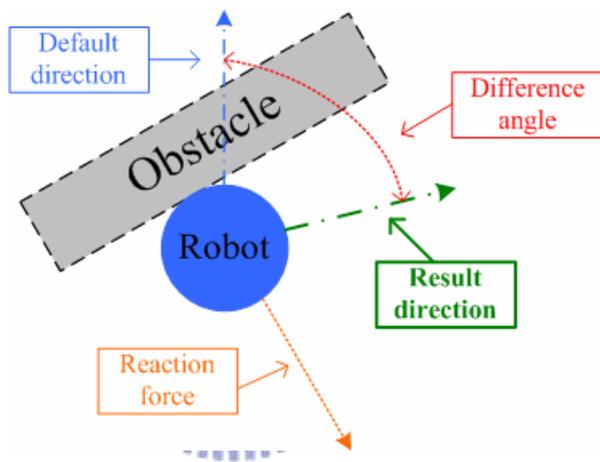


圖 30 碰撞偵測_2

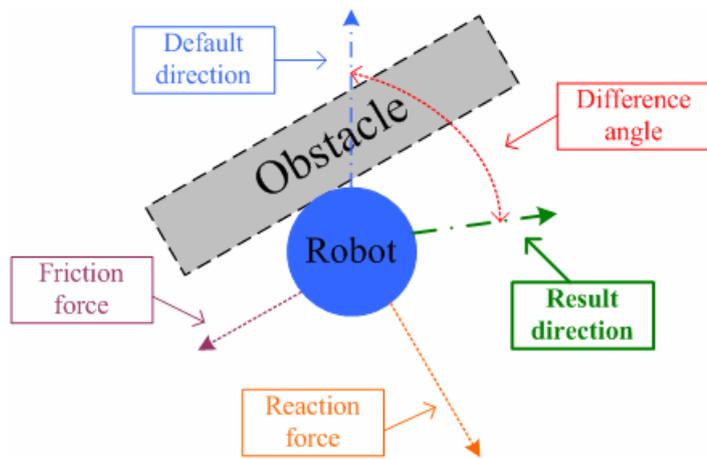


圖 31 碰撞偵測_3

因此，我們在程式中設計了一個用來計算平均速度大小及方向的副程式；當全方向運動平台在預定的軌跡上移動且碰撞到障礙物時，根據上述討論的結果，該副程式將會偵測到此時的平均速度大小為接近零或者是平均速度方向與預設速度方向不同；因而通知主程式停止前進並且進行規劃新移動路徑的動作，以使該運動平台可以繞過障礙物，並回到原先要求的路徑上繼續往目的地前進。必須注意的是，在這裡我們設定當平均速度大小小於 0.5 cm/s，以及平均速度方向與預設速度方向差距 20 度以上時，為碰撞成立的條件值；經由實驗證明，該條件值可以大幅的避免誤判的情形發生。

因為我們使用的空間性軌跡追蹤法是將平台的移動軌跡設定以弧形線段(或直線)為基礎，任何複雜的移動軌跡都可以由弧形線段(或直線)組合而成。同樣的，當運動平台碰撞到障礙物而無法繼續前進時，程式所規劃的避開障礙物之新路徑也必須是由弧形線段(或直線)組合而成，如此才能適用於空間性軌跡追蹤法。接下來，我們將介紹碰撞偵測及動態路徑規劃方法之實現架構，以及加入碰撞偵測及動態路徑規劃方法後 RVC-SPTM 的新控制流程。

如圖 32(a)所示，假設運動平台的預設移動軌跡為一個弧形線段。同時，如圖 32(b)所示，假設有一細長型的障礙物出現在該移動軌跡上，平台在軌跡上移動時會因為碰撞到該障礙物而停止；如圖 32(c)中橙色線段所示，此時系統將會規劃一條避開障礙物的新路徑，而此橙色線段的大小是由 (θ_b, L_b) 這兩個預設的參數值來決定的。而且，如圖 32(d)所示，此橙色線段可以分為「1,2,3,4」四個部分，其中 1 和 3 為弧形線段，而 2 和 4 為直線線段。如圖 32(e)所示，當平台沿著該避障路徑前進時，是有可能再次與該障礙物發生碰撞的(注意，此時發生碰撞的地方是屬於避障路徑中的第 3 部分)。此時，如圖 32(f)中橙色線段所示，系統將會再規劃一條避開障礙物的新路徑，如此一來，該機器人平台將可以順利繞過該障礙物，並回到原先要求的路徑上，繼續往目的地前進。

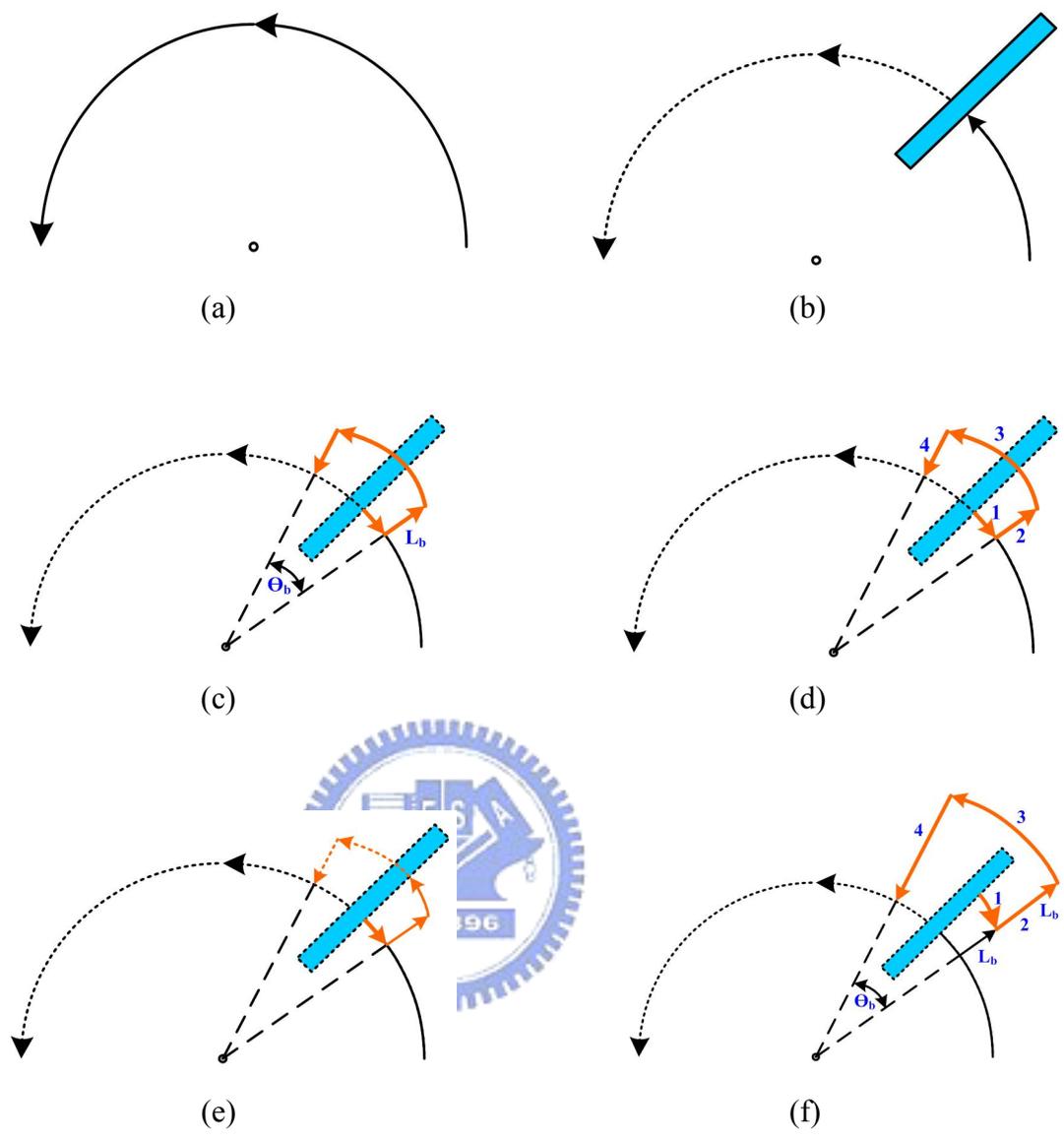


圖 32 動態路徑規劃_1

關於在避障路徑上再次碰撞到障礙物的議題，如果運動平台碰撞到如圖 33(b) 所示，是屬於寬扁型的障礙物時，此時系統所規劃的新的避障路徑將會有所不同。圖 33(a)到(d)與圖 32(a)到(d)所描述的事情是相同的，只是其中的障礙物大小形狀有所不同。如圖 33(e)所示，當平台沿著避障路徑前進時，將會再次與該障礙物發生碰撞(注意，此時發生碰撞的地方是屬於避障路徑中的第 4 部分)。此時，如圖 33(f)中橙色線段所示，系統將會再規劃一條避開障礙物的新路徑，如此一來，該機器人平台將可以順利繞過該障礙物，並回到原先要求的路徑上，繼續往目的地前進。

因此，我們所設計的動態路徑規劃方法可以經由多次與障礙物發生碰撞而動態地更改其避障路徑，以增加該避障功能的執行效率及成功率。

另外，圖 34 為加入碰撞偵測及動態路徑規劃方法後的 RVC-SPTM 新控制流程，其中 VM 代表速度量測(Velocity Measure)方法。

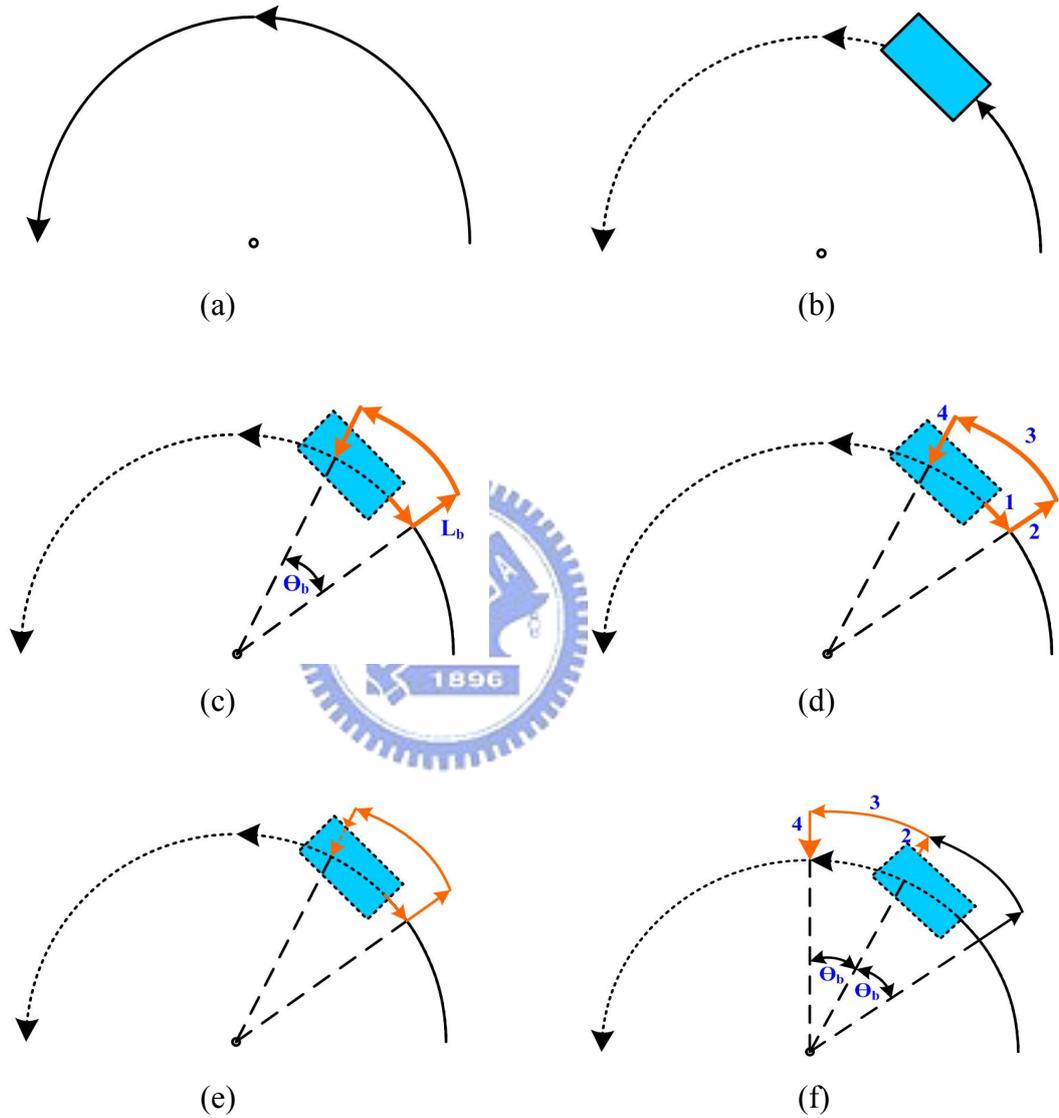


圖 33 動態路徑規劃_2

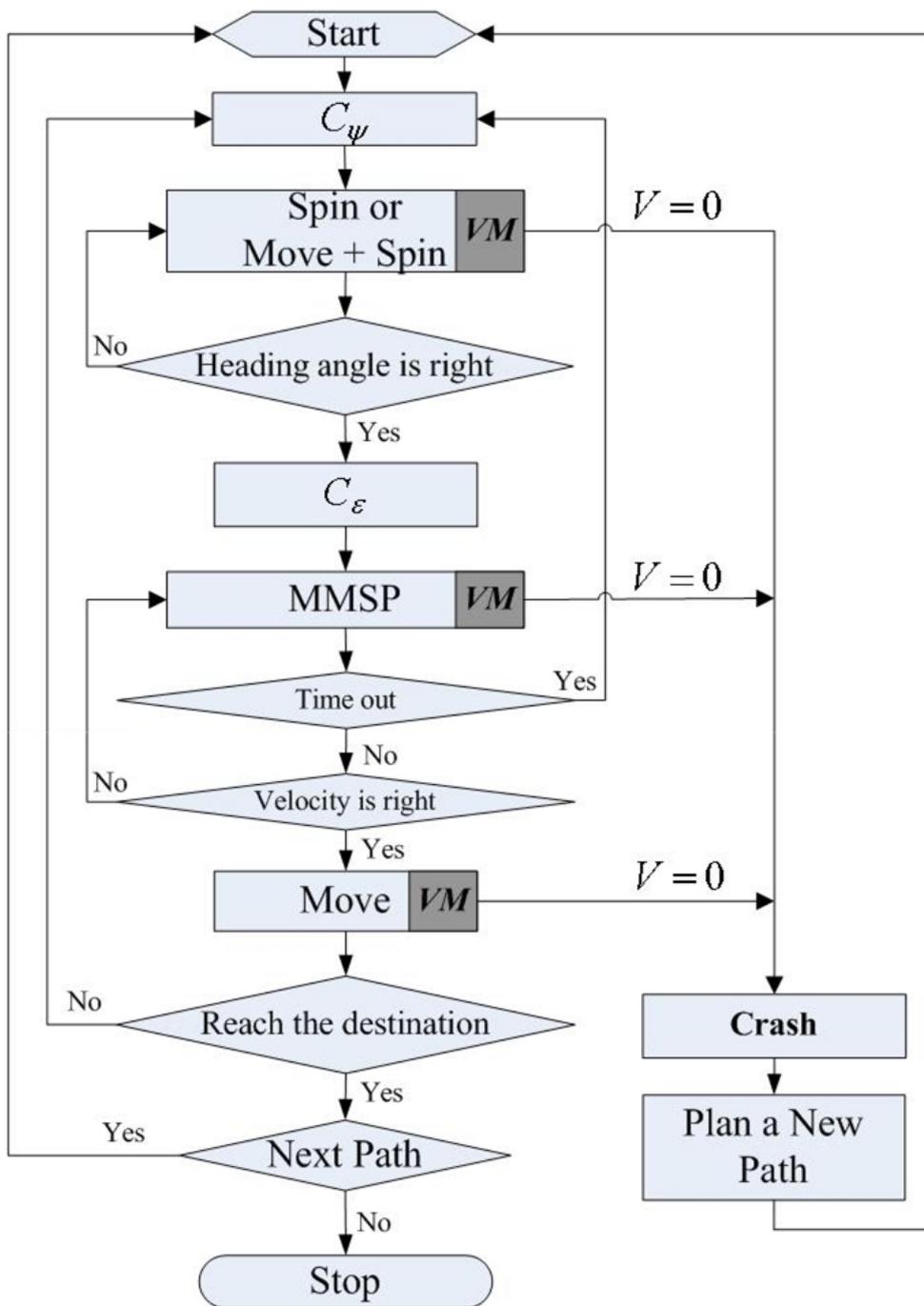


圖 34 含碰撞偵測及動態路徑規劃之 RVC-SPTM 控制流程圖

第三章 光流影像偵測定位法

3.1 光流(Optical Flow)的定義

所謂的光流(Optical Flow)是指，對於兩張以上的連續影像中之物體，預測其運動方向及大小。光流是透過比較兩張影像之間的影像強度而獲得，通常該兩張影像之間必須擁有許多相同的內容，以保證該方法可以正常運作。

假設將攝影機裝設在某運動物體上，攝影機朝著地面進行拍攝，而該物體正在地面上進行移動；當運動物體往某個方面移動某個距離時，攝影機所拍攝到的地面影像則會有相對的變動，其移動的方向相反、距離成正比關係(視裝置狀況決定)。因此，可以利用「光流」方法來偵測影像中物體的運動，因而推算出該運動物體的真实運動狀況。

之所以會以光流做為計算運動位移的工具，主要是因為光流方法比較不容易受到外界環境因素的影響，且不需要事先了解追蹤的物體與當時的環境狀況，因此比較適合用於未知的環境來進行運動位移偵測。

圖 35 為一個光流向量場的示意圖，其中物體分別往正 X 軸和正 Y 軸方向各移動了一個像素(pixel)的距離[11]。

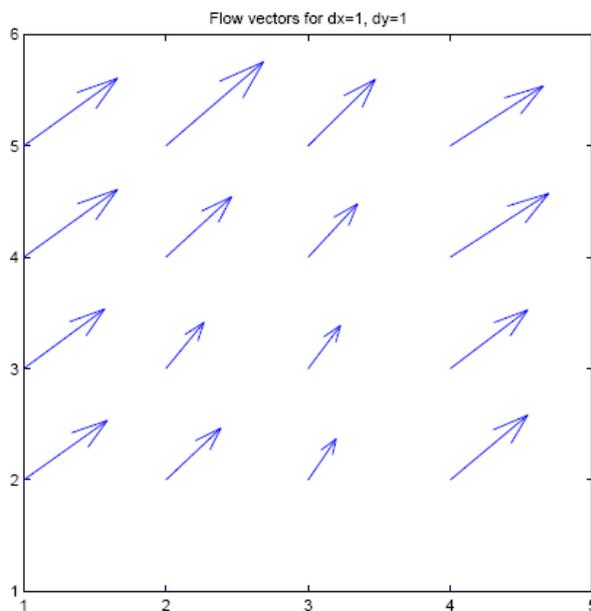


圖 35 光流向量場

3.2 光流影像偵測定位法之原理[11]

在這裡我們使用兩個光流影像感測器來偵測該運動平台的位移資訊(包含 x 軸、y 軸的位移量，和 z 軸的旋轉量)；將這兩個光流影像感測器放置在不同的位置，觀察它們與平台之間的幾何關係，我們可以得到下列的運動學方程式：

$$\begin{aligned}
 V_{r,x} &= V_{1,x} + \omega_r \cdot r_{1,y} \\
 V_{r,y} &= V_{1,y} - \omega_r \cdot r_{1,x} \\
 V_{r,x} &= V_{2,x} + \omega_r \cdot r_{2,y} \\
 V_{r,y} &= V_{2,y} - \omega_r \cdot r_{2,x}
 \end{aligned} \tag{26}$$

方程式中的 $V_{r,x}$ 和 $V_{r,y}$ 是機器人平台中心點的移動速度值， ω_r 是平台的旋轉角速度， $V_{i,x}$ 和 $V_{i,y}$ 是第 i 個感測器所偵測到的移動速度值，而 $r_{i,x}$ 和 $r_{i,y}$ 是第 i 個感測器與平台中心的距離，如圖 36 所示。

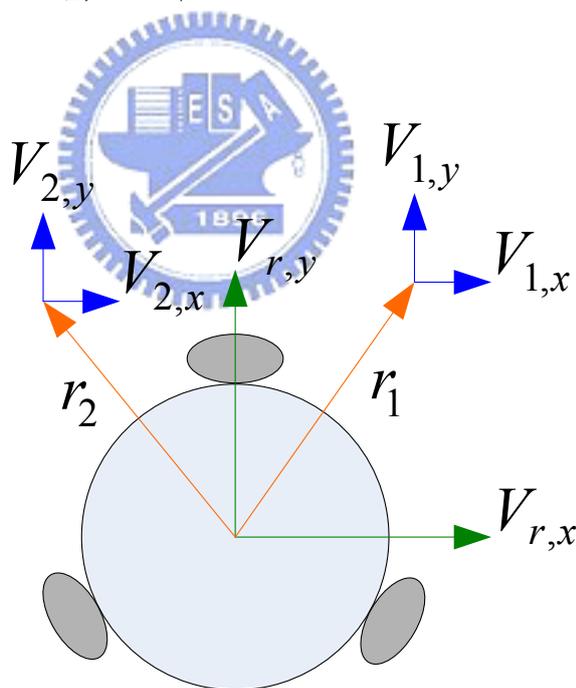


圖 36 光流感測器與平台的幾何關係圖

為了得到該平台的速度和角速度值，我們可以將上述式子改寫成矩陣向量的形式：

$$\begin{bmatrix} 1 & 0 & -r_{1,y} \\ 0 & 1 & r_{1,x} \\ 1 & 0 & -r_{2,y} \\ 0 & 1 & r_{2,x} \end{bmatrix} \cdot \begin{bmatrix} V_{r,x} \\ V_{r,y} \\ \omega_r \end{bmatrix} = \begin{bmatrix} V_{1,x} \\ V_{1,y} \\ V_{2,x} \\ V_{2,y} \end{bmatrix} \quad (27)$$

可以知道(27)式是一個過度決定(over-determine)的系統，因此我們需要使用最小平方誤差法(least square error method)來求得此方程式的最佳近似解，也就是此平台的速度與角速度值($V_{r,x}$ ， $V_{r,y}$ ， ω_r)；在得到平台速度與角速度之後，我們便可以利用積分運算來得到此平台的移動距離和旋轉量：

$$\begin{aligned} \theta_{robot} &= \int (\omega_r) dt \\ X_{robot} &= \int (V_{r,x} \cos \theta_{robot} - V_{r,y} \sin \theta_{robot}) dt \\ Y_{robot} &= \int (V_{r,x} \sin \theta_{robot} + V_{r,y} \cos \theta_{robot}) dt \end{aligned} \quad (28)$$

在(28)式之中， θ_{robot} 代表的是機器人平台的轉動角度， X_{robot} 是X軸方向的位移， Y_{robot} 是Y軸方向的位移。如圖 37 所示，可以在電腦視窗中看出機器人移動的路徑以及目前方向。

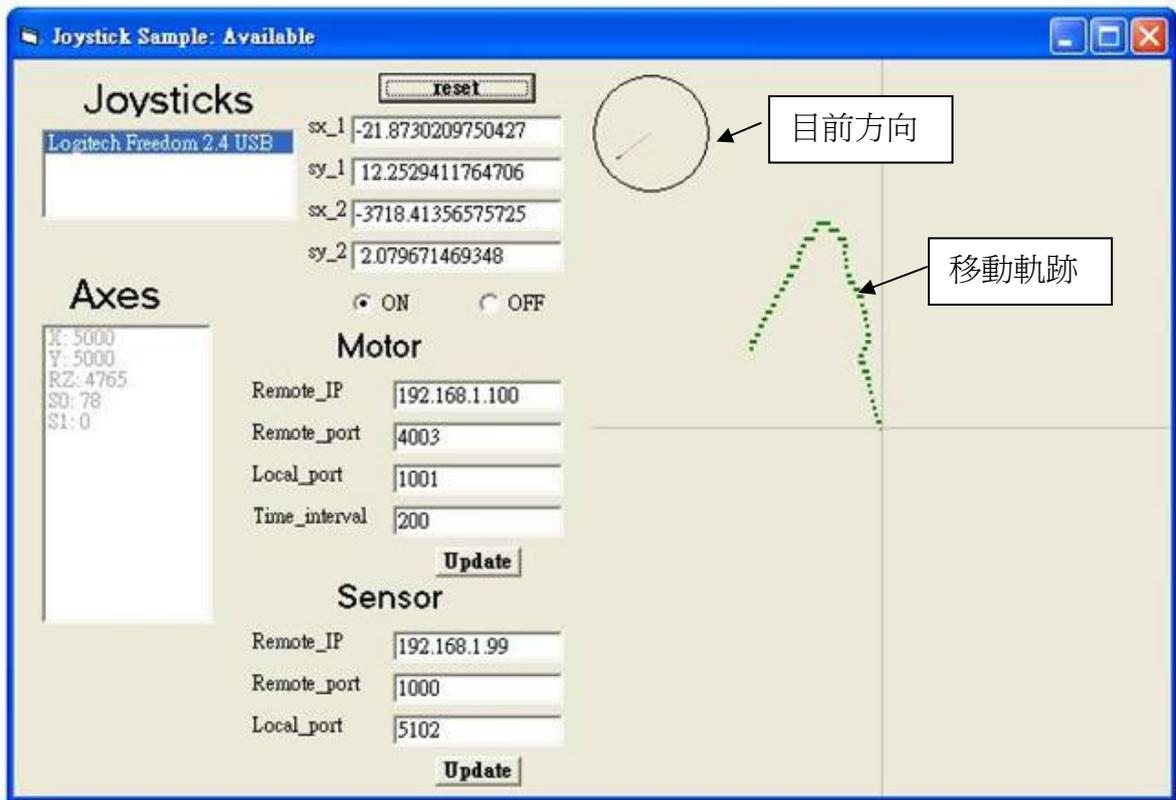


圖 37 光流軌跡視窗



第四章 全方位運動平台

4.1 簡介

全方位運動平台使用特殊結構的輪子，一般稱為全向輪(omni-directional wheel or omni-wheel)；它可以將地面作用於輪子上的摩擦力之中平行於輪軸的部分，藉由滾子的轉動而分散掉；因而，造就了全向輪平台所擁有的獨特運動模式。

實際上，全向輪的構想早在 1910 年就發表於美國的專利文件中，截至目前，為了提高效率及改善震動的缺點，已有許多改良的全向輪被發明使用，而現今也已被使用在搬運車、輪椅或輪式足球機器人等方面[12-13]。圖 38 中共列出三種不同型式的全向輪，(c)是目前我們正在使用的全向輪。從平行於輪軸的方向看過去，其形狀就像是一般傳統的輪子。

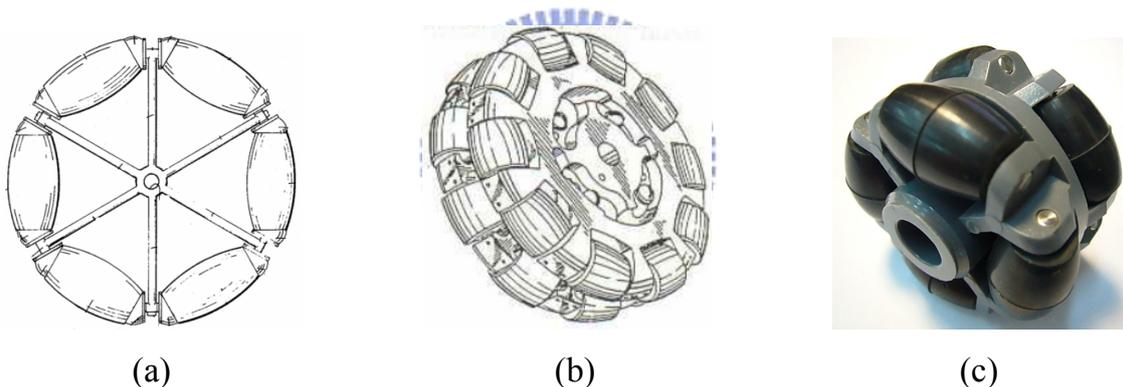


圖 38 三種不同型態的全向輪

我們的機器人平台共使用三顆全向輪，該平台較使用傳統輪子的二輪或四輪式的機器人擁有較高的運動自由度；經由三顆輪子所產生的多種合力組合，共可以產生多達五種運動模式，包括：原地旋轉、對頭直行、對頭差速轉彎、平移、平移且自轉。

4.2 運動學分析

本平台使用三顆全向輪，如圖 39 所示，三輪之間各夾 120 度角，為這個移動平台最大的一個特點，這種構造使得這個平台得以呈現較一般輪式平台更多樣化的運動模式，在此將詳細探討這種構造的運動學理論[14-17]。

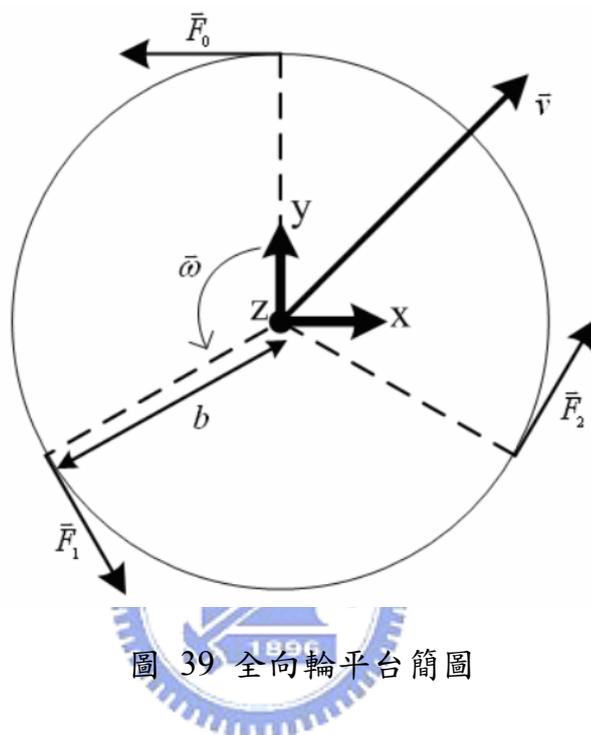


圖 39 全向輪平台簡圖

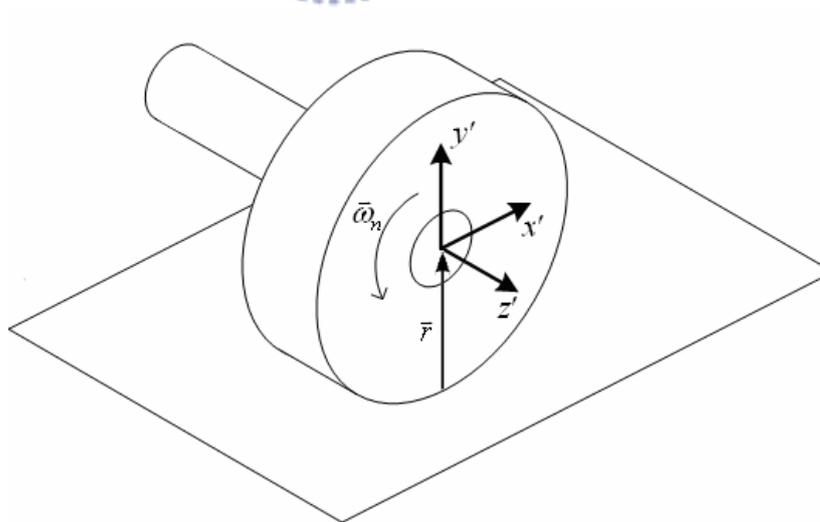


圖 40 全向輪簡圖

在圖 39 及圖 40 中，各項符號的意義表示如下：[14-17]

- (a) x 、 y 、 z ：定義在平台上的座標系，以下推導中的向量皆相對於本座標系。 x 、 y 如圖所示，而 z 軸為指出紙面為正。
- (b) \bar{v} ：希望平台移動之速度向量。
- (c) $\bar{\omega}$ ：希望平台轉動之角速度向量，且定義沿著反時針方向旋轉為正。
- (d) \bar{F}_n ：垂直於各輪軸之單位向量， $n=1,2,3$ 。

$$\begin{aligned}\bar{F}_0 &= (-1, 0, 0) \\ \bar{F}_1 &= \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}, 0 \right) \\ \bar{F}_2 &= \left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0 \right)\end{aligned}\quad (29)$$

- (e) \bar{b}_n ：從 n 號輪軸心端點到平台中心的距離向量。
- (f) \bar{P}_n ：代表 n 號輪軸心相對於平台座標的速度向量。

令建立於各輪之上的座標系為 $x' y' z'$ ，如圖 40 所示(圖 40 中皆表其正向)：

- (g) \bar{r} ：從輪子接觸地面之點到輪子軸心的距離向量。
- (h) $\bar{\omega}_n$ ：各輪相對於其軸心之轉動角速度向量，令逆時針方向旋轉為正(沿正 z' 軸方向)。
- (i) \bar{Y}_n ：各輪之軸心在輪沿地面滾動時的速度向量於 x' 軸方向上的分量。

在考慮 $x y z$ 座標系與 $x' y' z'$ 座標系之間的轉換，可知 x' 軸方向與 \bar{F}_n 方向永遠一致，因此在 $x y z$ 座標系中看 x' 軸上的向量，為：

$$(x' \text{ 軸上正向大小}) \cdot \bar{F}_n$$

考慮此平台一個最廣泛的運動模式：平台本體具有一個自旋轉的角速度 $\bar{\omega}$ ，並且以速度向量 \bar{v} 移動；如何求得每一伺服機所需運作的角速度 $\bar{\omega}_n$ ，其運動學推導如下：

在 $x y z$ 座標上，各輪子之軸心端點的速度等於平台中心點的速度加上平台自旋所產生的角速度：

$$\vec{P}_n = \vec{v} + (\vec{b}_n \times \vec{\omega}) \quad (30)$$

(30)式也可以寫成

$$\vec{P}_n = \vec{v} + (b \cdot \omega) \cdot \vec{F}_n \quad (31)$$

而在 $x' y' z'$ 座標上，各輪子之軸心端點速度為：

$$\vec{Y}_n = \vec{r} \times \vec{\omega}_n \quad (32)$$

(32)式也可以寫成

$$Y_n = r \cdot \omega_n \quad (33)$$

在 $x' y' z'$ 座標中的 x' 軸正向即為 xyz 座標中的 \vec{F}_n 單位向量的方向，因此我們可知，同是表示各輪軸心端點速度的 \vec{Y}_n 跟 \vec{P}_n ，其關係如下：

$$Y_n = \vec{P}_n \cdot \vec{F}_n \quad (34)$$

綜合(33)、(34)式，可得

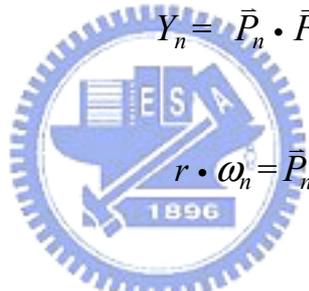
$$r \cdot \omega_n = \vec{P}_n \cdot \vec{F}_n \quad (35)$$

將(31)式代入(35)式可得：

$$r \cdot \omega_n = [\vec{v} + (b \cdot \omega) \cdot \vec{F}_n] \cdot \vec{F}_n \quad (36)$$

$$\omega_n = [\vec{v} + (b \cdot \omega) \cdot \vec{F}_n] \cdot \vec{F}_n / r \quad (37)$$

由此可知，如果要讓此平台以角速度 ω (逆時針為正)作自轉、加上以速度向量 \vec{v} 作平移，將 ω 以及 \vec{v} 代入(37)式，即可求出各輪所需要轉動的角速度 ω_n ，亦即各伺服機所需要提供之角速度。

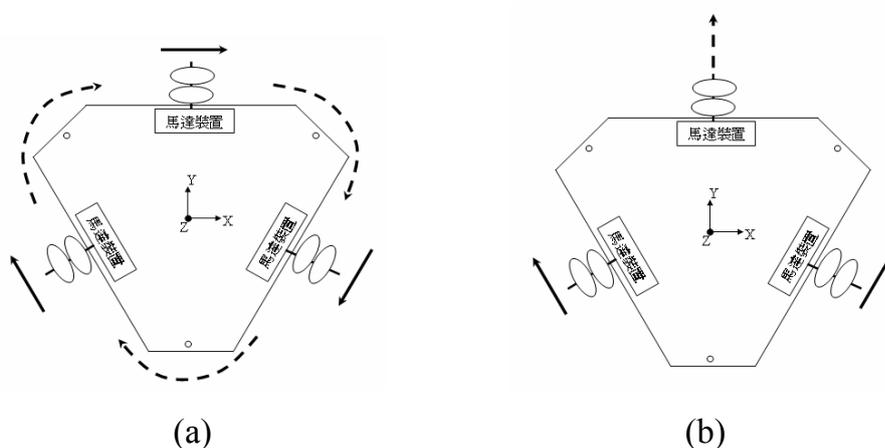


4.3 運動模式

經過前一小節的分析，可以知道此一移動平台有能力呈現較一般兩輪平台更為複雜、多樣化的運動模式，由簡至繁共有以下五種：

- (a). 原地旋轉：三個輪子的角速度保持一定且相等、轉向相等，則本平台可作原地等角速度旋轉。
- (b). 對頭直行：當三輪中其中一輪不運轉，其他兩輪以相同的角速度旋轉但轉向相反時，則本平台會朝不運轉的輪子方向(一個「頭」)前進。
- (c). 對頭差速轉彎：以對頭直行模式為基礎，轉動的兩輪若轉速不同，則本平台會進行轉彎的動作，此時「頭」的指向會改變。這種運動模式與一般兩輪平台差速轉彎的道理類似。
- (d). 平移：所謂平移即是本體指向不改變的移動，而且移動方向可以任意選擇。這種運動方式正是普通輪式平台無法呈現的運動模式。
- (e). 平移且自轉：這是本平台所能呈現最複雜的一種運動模式，一邊朝著固定方向作平移，一邊又以固定角速度作自轉，這也是普通輪式平台無法呈現的運動模式。

圖 41 為上述幾種運動模式的示意圖，圖中的實線代表輪子轉動所產生的前進速度方向，長度則表示相對的速度大小；而圖中的虛線則代表平台的移動或轉動方向。(a)圖為「原地旋轉」，(b)圖為「對頭直行」，(c)圖為「對頭差速轉彎」，(d)圖為「平移」，(e)圖為「平移且自轉」的示意圖。



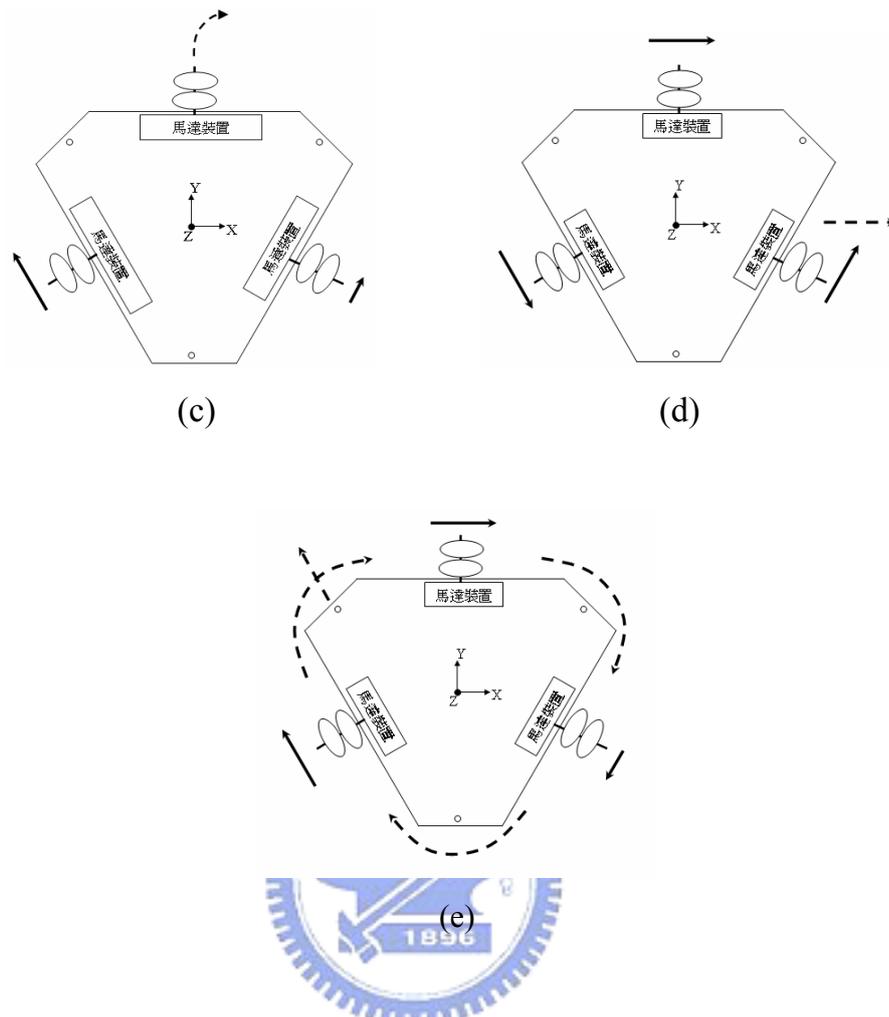


圖 41 運動模式示意圖

第五章 軟硬體設計與實現

5.1 系統架構

圖 42 為整個系統的架構圖，主要分成兩個部分：個人電腦端(PC)以及海星號(Starfish)，而兩者之間則是透過無線網路機制來進行訊息的溝通傳遞。接下來的兩個小節將分別針對兩者(PC and Starfish)進行詳細的介紹。

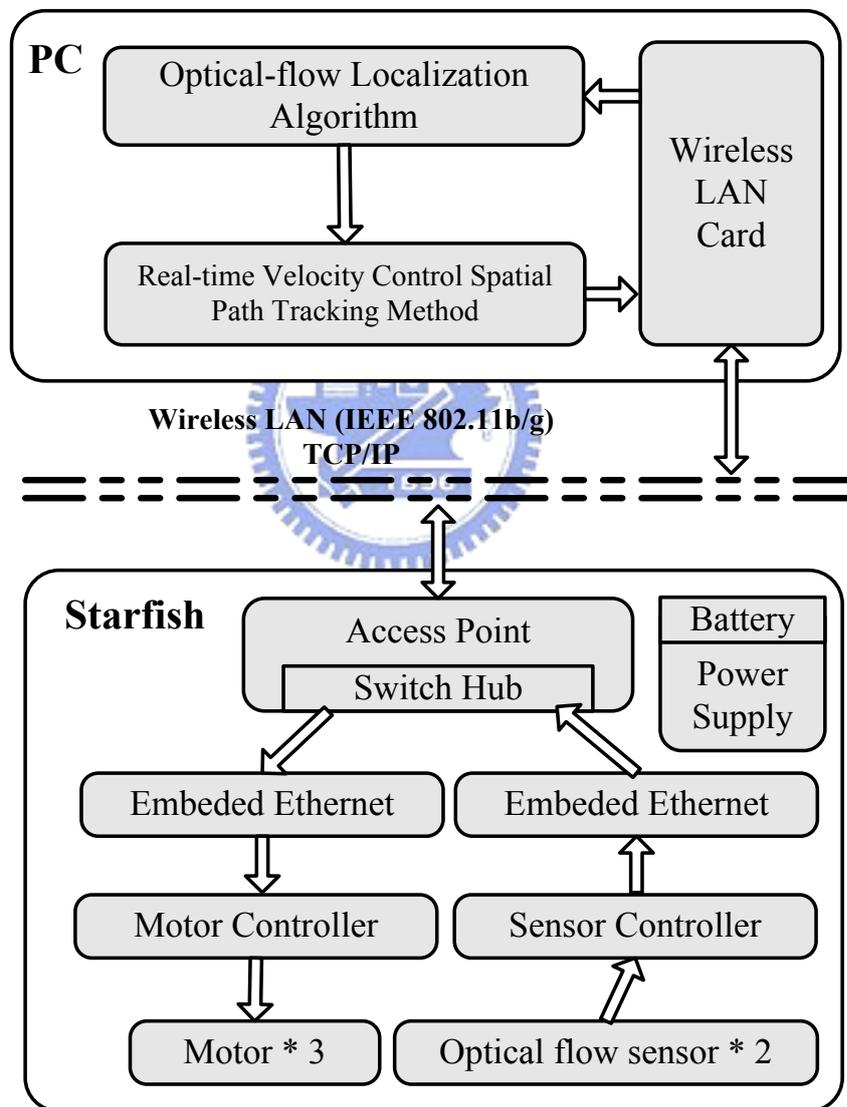


圖 42 完整系統架構

5.2 全方向運動平台-海星號(Starfish)之架構

全方向運動平台-海星號的底盤下方裝置了 2 個光流影像偵測器、3 個馬達伺服機以及 3 顆全向輪；光流影像偵測器是用來感測地面影像，並使用地面影像的光流變化來量測該平台的運動狀態；馬達是用來提供該平台運動動力，可以透過微控制器使用 PWM 訊號來控制該馬達的正反轉向和轉速快慢；全向輪則是提供該平台全方向運動的能力。

此平台底盤的上方則放置平台的系統電路；此系統電路包含 1 組 Access Point(無線網路基地台且包含 Switch Hub)、2 組嵌入式網路系統板(Embedded Ethernet)、1 組馬達控制板(Motor Controller)、1 組感測器控制板(Sensor Controller)、1 組充電電池組以及 1 組電力供應系統板。

如圖 42 所示，Access Point 是用來建構無線網路溝通路徑；嵌入式網路系統板則是藉由 Access Point 的無線網路傳輸功能將 PC 端馬達控制程式所傳送出來的馬達控制指令傳達至馬達控制板以進行馬達控制的動作，或是將感測器控制板從光流影像偵測器所得到的平台移動資訊傳達至 PC 端控制程式以進行運動軌跡的偵測和記錄；充電電池和電力供應控制板則是提供此平台運動和計算所需的所有電力。

此平台除了底盤之外，還有一個上蓋可以跟底盤組合；可以保護該底盤上所有的設施器材免於因碰撞而造成損壞，並可以提供該平台擁有裝載貨物的能力及擴充架構的可能性；此上蓋的形狀架構以及跟底盤組合後的透視圖請參考圖 43。

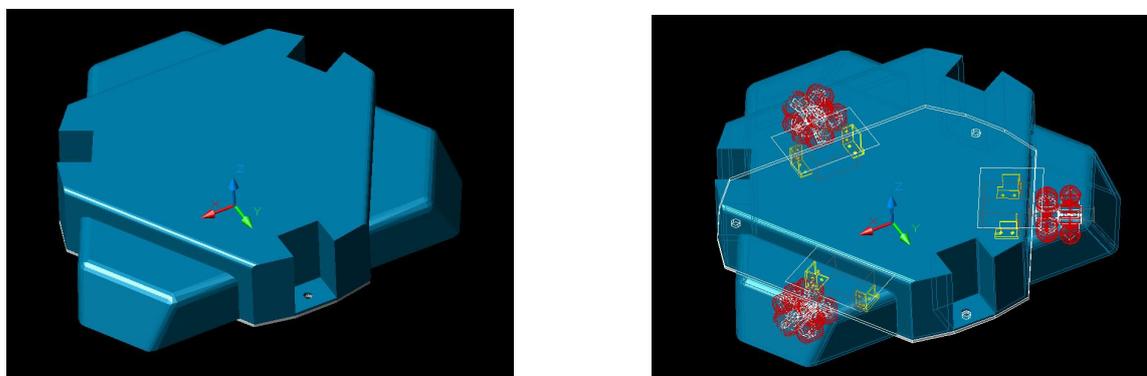


圖 43 (左)上蓋, (右)上蓋與底盤組合透視圖

5.2.1 電路功能介紹

底下將逐一介紹海星號上面各個電路架構的詳細功能，可參照圖 42 之完整系統架構圖。

(1)嵌入式網路(Embedded Internet)板，如圖 44 所示；它可以將網路訊息封包解開，並將資料內容從 RS-232 port 傳送出去；或是將從 RS-232 port 所收到的訊息進行封包化，再經由網路端傳送出去。而嵌入式網路的概念最早是在 1998 年 5 月於網路計算期刊(Journal of Internet Computing)中一個關於嵌入式乙太(Embedded Ethernet)技術的特殊議題所提出的[18]；這種技術可以支援許多儀器免於被傳統桌上型電腦的操作環境所限制；例如，像手機、PDA 以及一些手持電子裝置都會使用到這種技術，而且是屬於比較需要藉由 PC 協助使用及非即時性應用的一類；另外，像工廠自動化流程、機械控制器以及電信通訊設備也都需要使用這種技術，而且是屬於重視即時性運作以及不需藉由 PC 協助便可獨立運作的一類[19][20]。在這裡我們所使用到技術主要是屬於分散式多層網路架構，其相關理論架構詳細敘述於[21~25]；而且已經廣泛延伸應用於[26~33]。

(2)馬達控制(Motor Control)板，如圖 45 所示；它會根據 RS-232 port 所接收到的控制指令內容去調整每一顆馬達裝置的 PWM 控制訊號。

(3)感測器控制(Sensor Control)板，如圖 46 所示；它會解讀光流影像偵測器輸出的 Quadrature waveform 訊號，將之轉換成累積位移資訊，並且將此資訊內容從 RS-232 port 輸出。

(4)將 Futaba S9402 伺服機改裝後便成為全方向平台所使用的馬達裝置，如圖 47 所示。原本伺服機只能控制輸出角度，但經由將內部的可變電阻移除換上兩顆固定電阻且磨除齒輪上的防過轉門後，便可依照 PWM 訊號來控制馬達轉速。

(5)Access Point 以及 Switch Hub 係使用 ASUS 出產的 WL-530g，如圖 48 所示，它的最大傳輸率為 54Mbps，符合 IEEE802.11g 的規格。

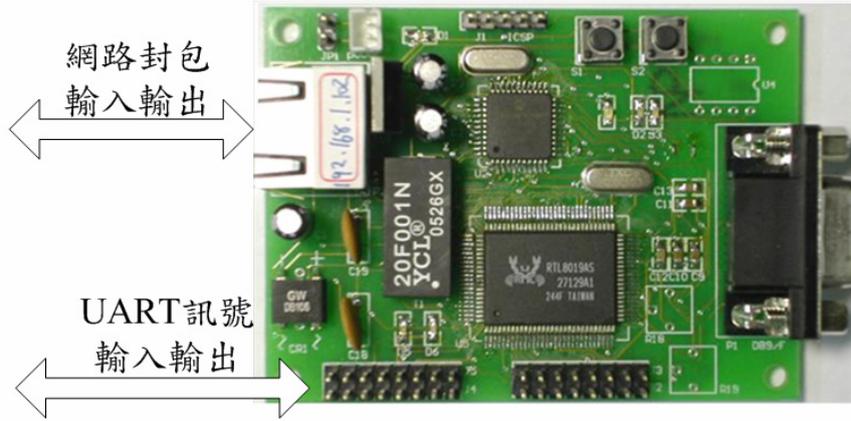


圖 44 嵌入式網路板

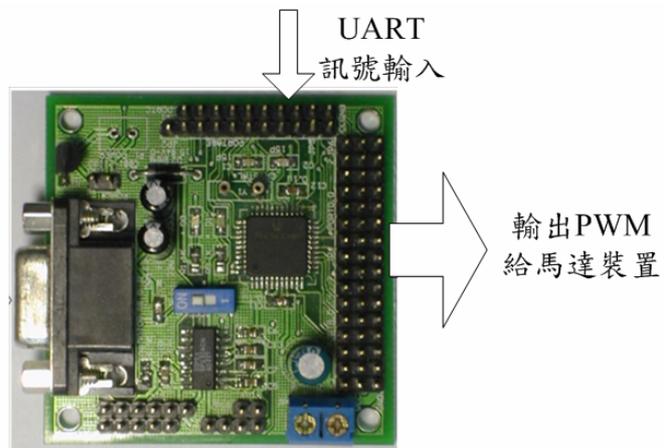


圖 45 馬達控制板

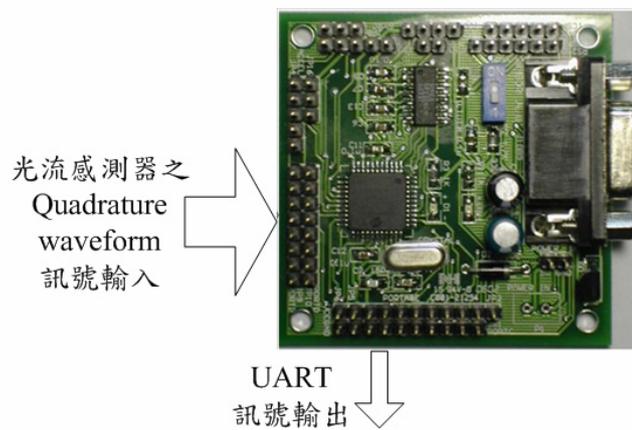


圖 46 感測器控制板

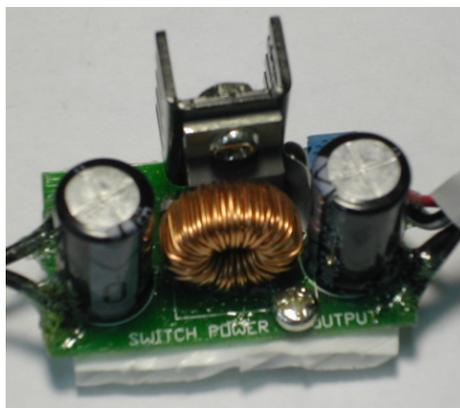


圖 47 馬達裝置



圖 48 AP(包含 Switch Hub)

(6)電源模組(Power module)採用三個 Step-down 穩壓模組，相較於一般常用的 78 系列降壓 IC，Step-down 穩壓模組能更有效率地轉換能量以減少損失。電池(Battery)使用三顆 7.4V 鋰聚合電池，比起鎳鎘、鎳氫電池，它的重量較輕、體積較小而且容量較大。



(a)



(b)

圖 49 降壓模組與電池組

(7)在這裡我們使用逸盛公司出產的迷你雷射滑鼠作為光流影像感測器(Optical flow sensor)；這個感測器擁有 1600 cpi (counts per inch)的解析度，並且相較於普通的光學滑鼠，雷射滑鼠有幾項優點：1. 解析度較高 2.可在玻璃等透明物質上使用 3.不需與地面接觸。滑鼠改裝後將如圖 50 所示。

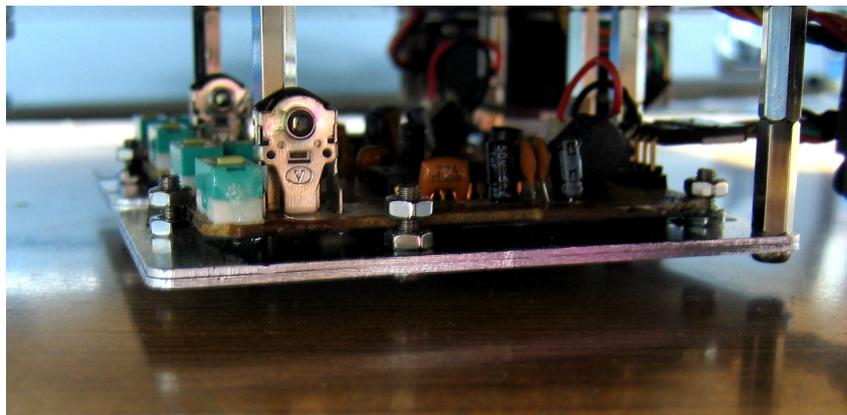


圖 50 光流感測器

(8)利用這個感測器所提供的「Quadrature Mode Output Waveform」，我們便可以得知該感測器在 X 軸和 Y 軸上面的移動資訊。以下列出 Quadrature Mode 的相關資訊圖表：

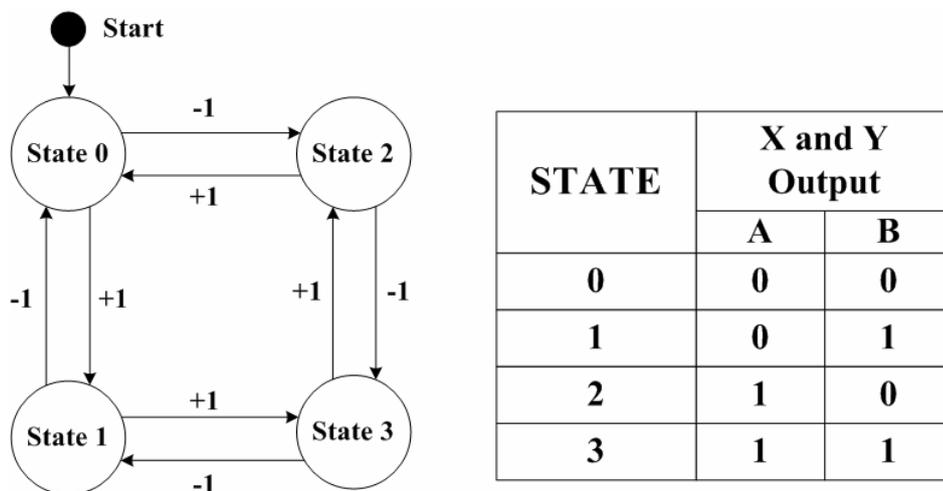


圖 51 Quadrature Mode Output state

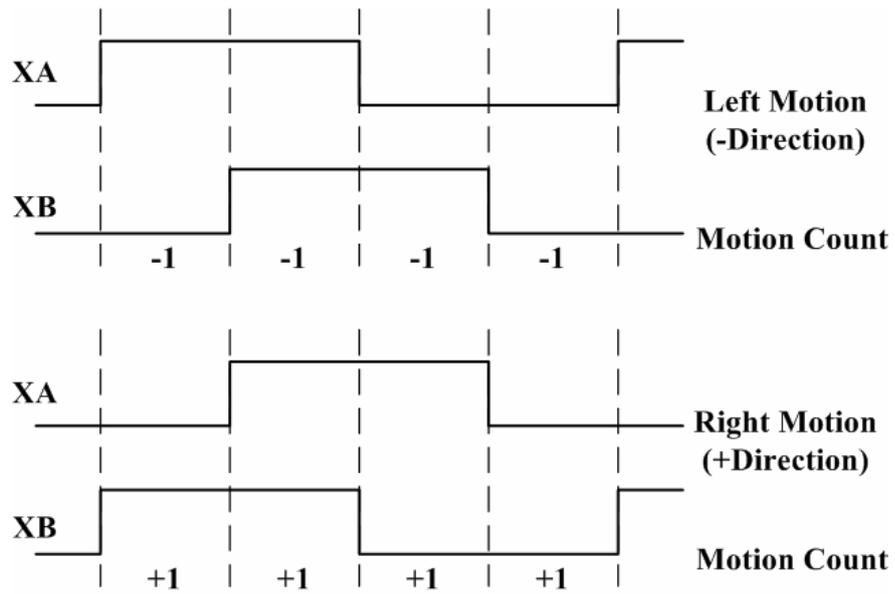


圖 52 Quadrature Mode Output waveform

最後，組裝完成後的嵌入式網路全方向運動平台內部電路如圖 53 所示，從底部拍攝的樣子如圖 54 所示。

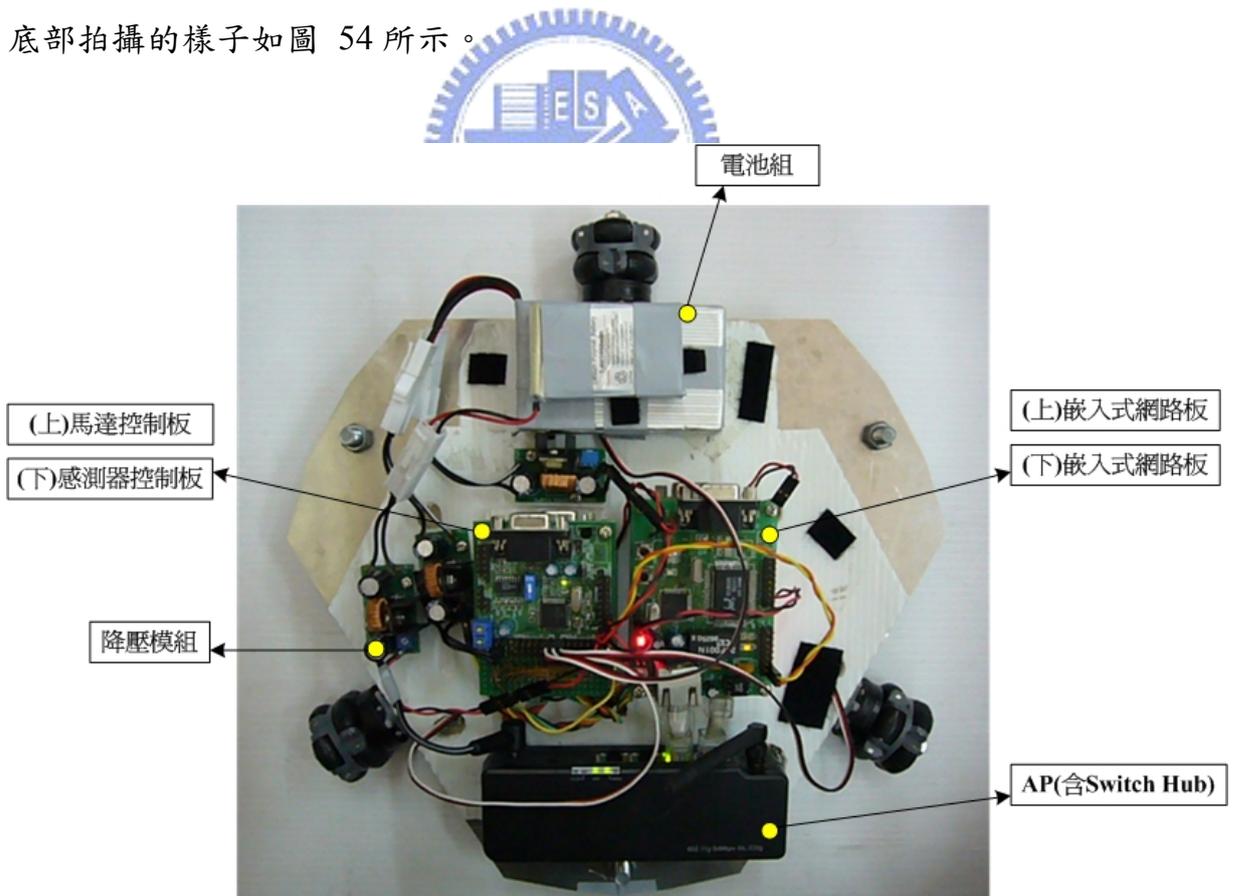


圖 53 內部電路圖

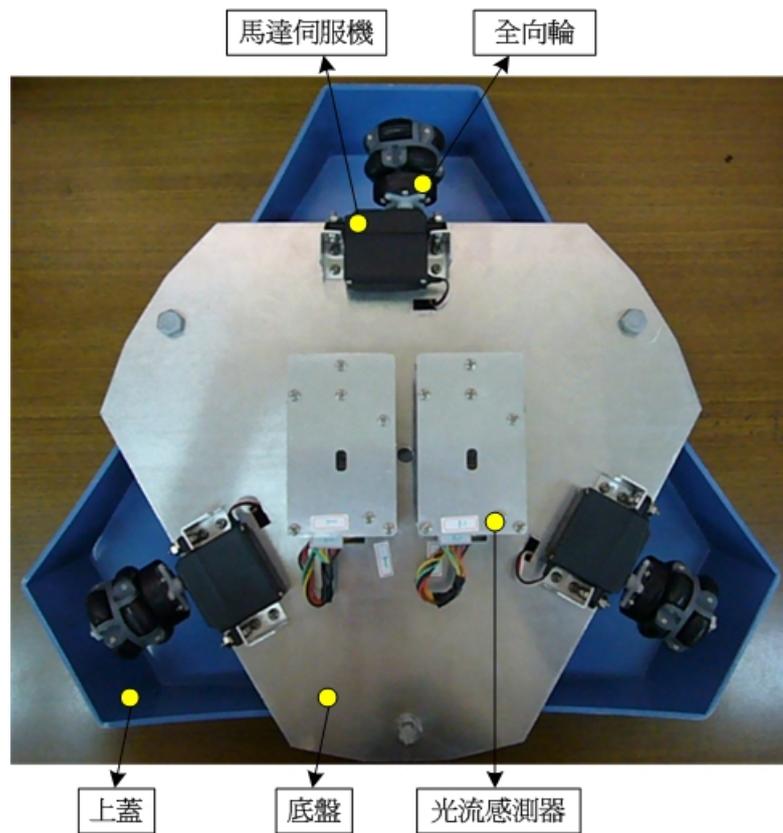


圖 54 底部照片

5.3 主機端架構

對照圖 42 中的 PC 端部分，PC 端程式中的 Real-time Velocity Control Spatial Path Tracking Method 部分是用來實現第二章所介紹的即時速度控制空間性軌跡追蹤法、第四章所介紹的全方向運動平台演算法，而 Optical-flow Localization Algorithm 部分則是用來實現第三章所介紹的光流影像定位演算法。由於該機器人平台擁有無線網路的機制，所以可以利用分散式運算的概念，將大部分的運算工作都放在 PC 端來處理。

如圖 55 所示，我們在 PC 端使用 Visual Basic 軟體來建立使用者圖形介面以及整個程式主體架構；另外如圖 56 至圖 58 所示，我們透過動態資料交換(DDE)的技術將大部分的量測資料傳送至 MATLAB 軟體進行記錄以及繪製相關圖表然後再回傳至 Visual Basic 端，以增進開發過程中除錯及驗證的效率。

程式中的 Optical flow Localization Algorithm 負責從網路接收兩個光流感測器所量測到的平台之單位時間位移資訊，經過運算後，便可以將運動平台的位移距離、旋轉角度、移動速度以及旋轉角速度等即時資訊傳達給程式中的 Real-time Velocity Control Spatial Path Tracking Method(RVC-SPTM)。而 RVC-SPTM 則會計算出接下來平台的移動速度大小及方向，並透過無線網路傳輸該控制指令以控制此全方向運動平台。

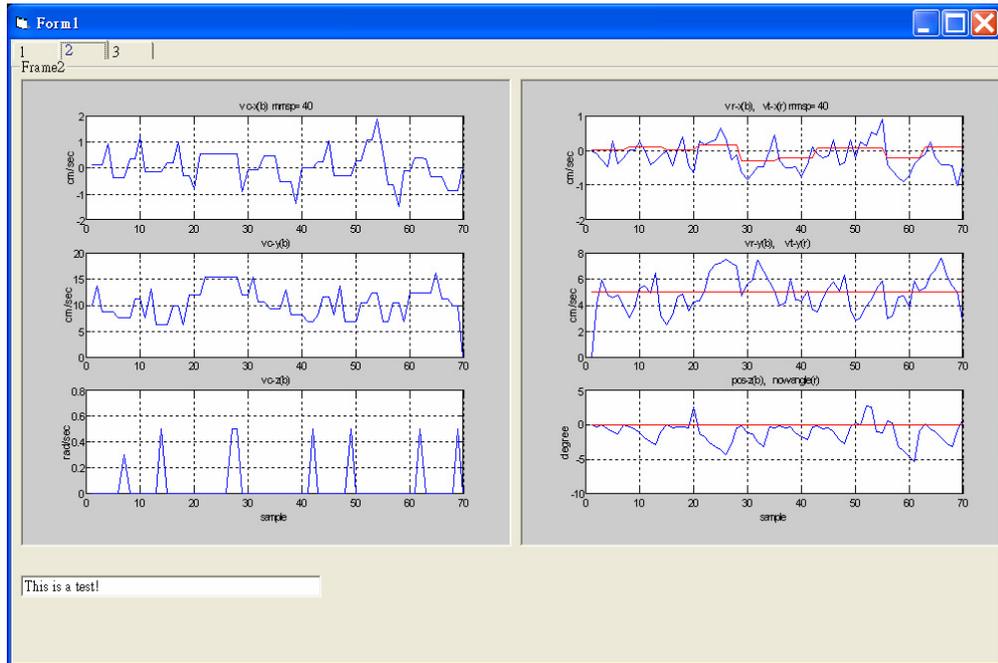


圖 57 利用 MATLAB 所繪製的速度及指向圖

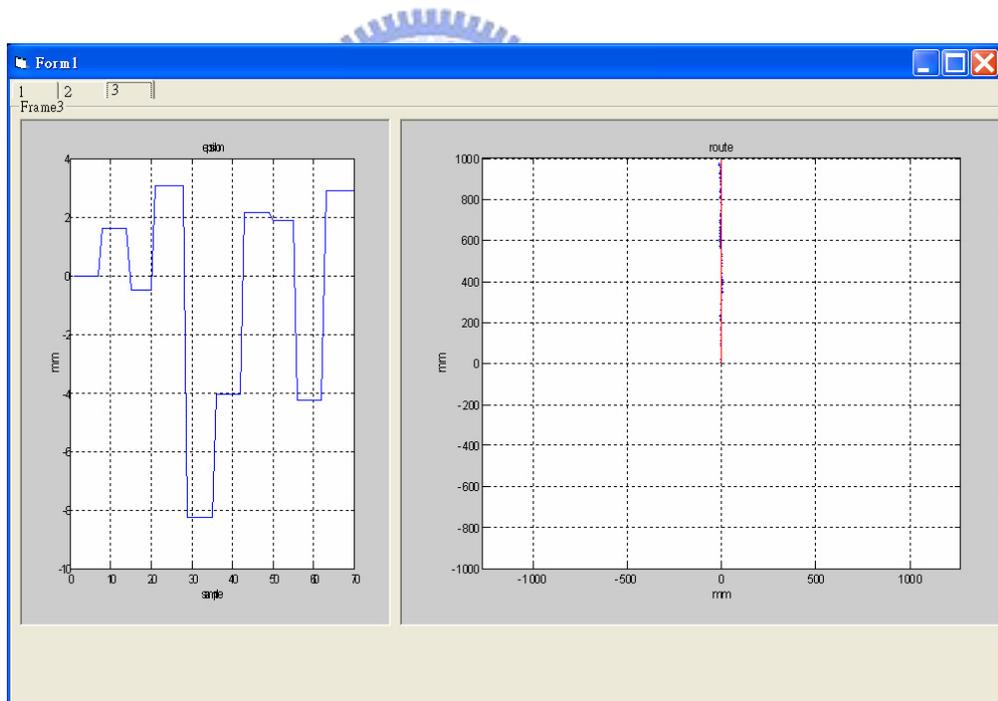


圖 58 利用 MATLAB 所繪製的誤差及軌跡圖

第六章 實驗結果與討論

6.1 光流影像定位實驗結果

本節將分析四種不同移動情況下，光流影像定位的累積誤差情形。作法是將全方向運動平台從(0,0)點位置出發，移動一段已知的距離或旋轉一個已知的角度；然後將 PC 端程式上所量測到的移動距離和旋轉角度數值與實際已知的數值相減，而得到該光流影像定位的累積誤差值。

下面列出四種不同的移動狀況及其結果，每種實驗均進行 5 次，然後將其數據平均後呈現。移動情形以軌跡圖和示意圖來表示，並以表格列出誤差情形。軌跡圖以水平軸為 X 軸且往右為正向，垂直軸為 Y 軸且往上為正向，而角度以正 Y 方向為 0 度，正 X 方向為-90 度。

實驗 1：

運動平台指向 Y 軸，往正 Y 軸移動 1 公尺(100cm)。

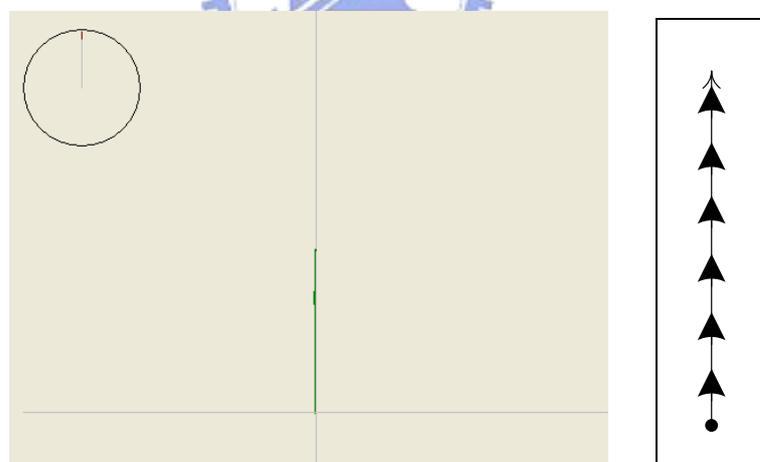


圖 59 光流定位軌跡圖(左)及平台移動示意圖(右)-實驗 1

表 1 光流定位誤差表-實驗 1

誤差		
X 軸(cm)	Y 軸(cm)	角度(degree)
-0.37	0.14	0

實驗 2：

運動平台指向 Y 軸，往正 X 軸移動 1 公尺(100cm)。

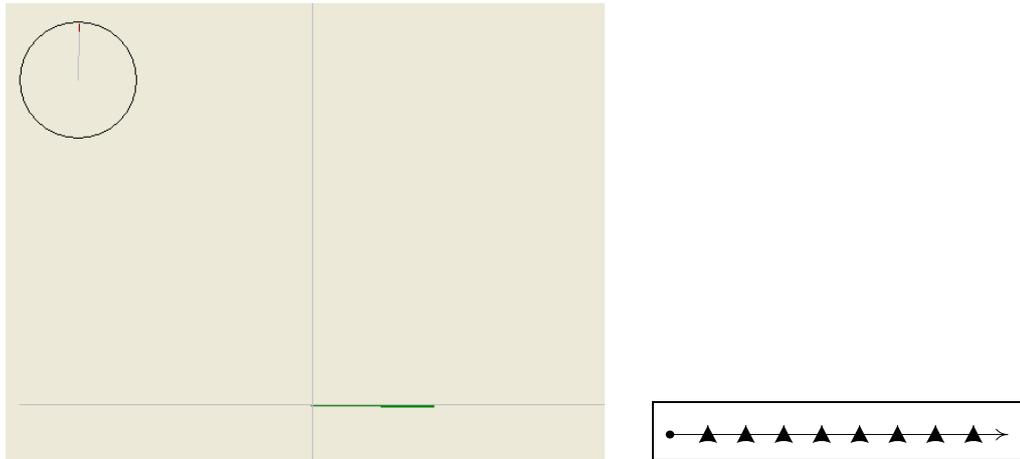


圖 60 光流定位軌跡圖(左)及平台移動示意圖(右)-實驗 2

表 2 光流定位誤差表-實驗 2

誤差		
X 軸(cm)	Y 軸(cm)	角度(degree)
3.19	1.00	-1

實驗 3：

運動平台指向 Y 軸，往 25.6 度方向移動 1 公尺(往負 X 軸移動 43.3cm，往正 Y 軸移動 90.3cm)。



圖 61 光流定位軌跡圖(左)及平台移動示意圖(右)-實驗 3

表 3 光流定位誤差表-實驗 3

誤差		
X 軸(cm)	Y 軸(cm)	角度(degree)
2.23	2.06	-4

實驗 4：

運動平台一開始指向 Y 軸，接著往 X 軸方向旋轉 90 度。

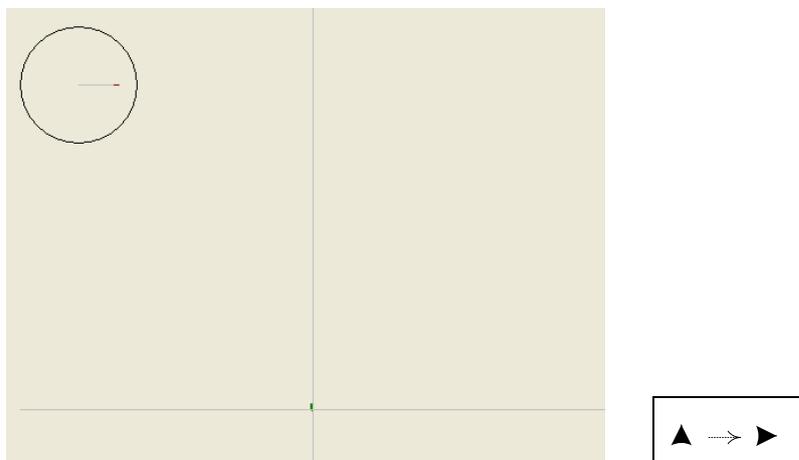


圖 62 光流定位軌跡圖(左)及平台移動示意圖(右)-實驗 4

表 4 光流定位誤差表-實驗 4

誤差		
X 軸(cm)	Y 軸(cm)	角度(degree)
-1.32	2.14	1

經由上述的實驗數據，我們可以發現利用光流影像進行運動平台的定位所產生的誤差是很小的；直線移動 1 公尺(100 公分)，所產生的累積量測誤差平均僅有 3.2 公分(3.2%)；原地旋轉 90 度，所產生的累積量測誤差平均僅有 1 度(1.1%)。

產生誤差的原因有很多種類，操作者的觀測誤差、光流感測器的裝置誤差、光流感測程式參數(cpi 值)的設定誤差以及光流影像定位公式中所使用的最小平方誤差求解法(least square error method)所產生的計算誤差…等，都會對光流影像定位法造成一定程度上的影響。其中，光流感測器的裝置誤差可以使用精密的雷射板件切割方法而大幅降低，光流感測程式參數(cpi 值)的設定誤差則可以進行量測校準實驗而有所改進，而最小平方誤差求解法所產生的誤差可以藉由調整光流感測器的位置而改善[11]；但是，整體下來還是會有不可避免的誤差存在，因而造成該定位方法會產生累積性的量測誤差。

6.2 速度控制及軌跡追蹤實驗結果

本節將分析在四種不同性質的地面上，使用即時速度控制之空間性軌跡追蹤法(RVC-SPTM)進行全方向運動平台的速度控制及軌跡追蹤任務。

下列的圖形大致上分為三種；第一種是「速度及指向角度圖」，其中記錄了運動平台 X 軸方向和 Y 軸方向上的理論速度值(紅色曲線)和實際速度值(藍色曲線)，以及運動平台 Z 軸方向上的理論指向角度值(紅色曲線)和實際指向角度值(藍色曲線)；第二種則是「軌跡圖」，其中記錄了運動平台中心點在絕對座標中的理論移動軌跡(紅色曲線)和實際移動軌跡(藍色曲線)；第三種則是「移動誤差圖」，其中記錄了運動平台理論移動軌跡和實際移動軌跡之間在圓弧半徑方向上的距離誤差值，即 ε (epsilon) 的數值。必須提醒的是，這裡所謂的實際移動軌跡是指使用光流影像感測法所量測到的平台相對移動軌跡，它與真實的平台絕對移動軌跡之間是有差距的，兩者之間的誤差值可以由 6.1 節的分析獲得。

一開始，我們先列出一組對照組，它們係利用時間性軌跡追蹤法來進行直線軌跡以及弧形軌跡的追蹤任務，必須注意它們並無使用平台的移動速度及位移資訊來進行迴授控制。

接下來，我們讓全方向運動平台在四種不同性質的地面上，使用即時速度控制之空間性軌跡追蹤法(RVC-SPTM)來進行下列兩種運動行為：(1)走直線軌跡且平台指向改變以及(2)走弧形軌跡且平台指向改變，以驗證即時速度控制之空間性軌跡追蹤法(RVC-SPTM)在多種不同性質的地面上皆能擁有不錯的速度控制及軌跡追蹤效能。相關實驗圖形請參照圖 63 至圖 92。

對照組：

(1) 平台於水平狀態下的木板上走直線軌跡且平台指向保持不變。

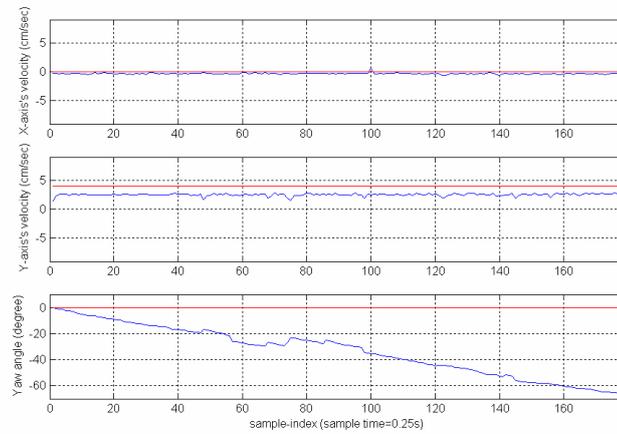


圖 63 對照組 1:於木板上進行直線軌跡追蹤任務之速度及指向圖

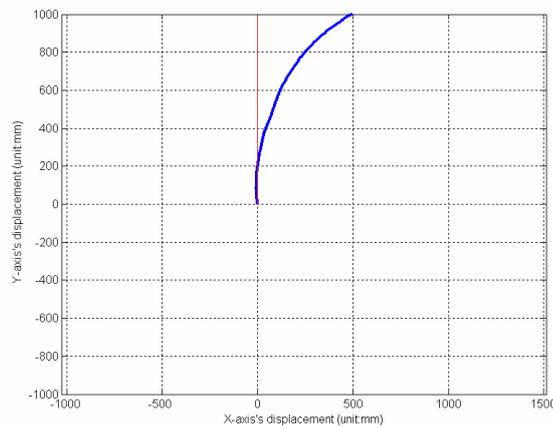


圖 64 對照組 1:於木板上進行直線軌跡追蹤任務之軌跡圖

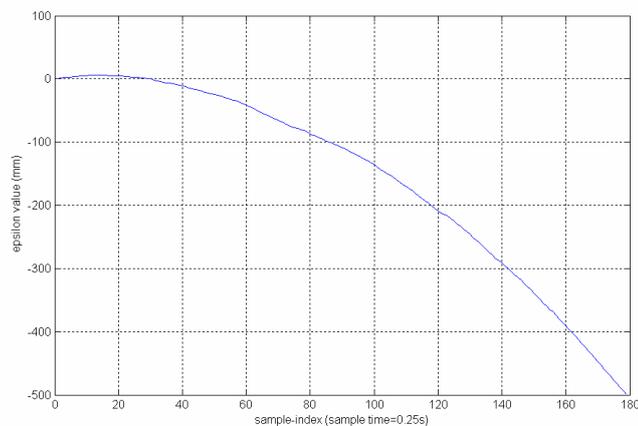


圖 65 對照組 1:於木板上進行直線軌跡追蹤任務之移動誤差圖

對照組：

(2) 平台於水平狀態下的木板上走弧線軌跡且平台指向保持不變。

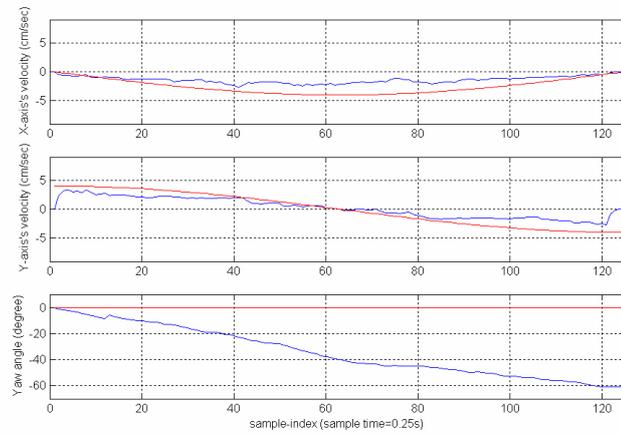


圖 66 對照組 2:於木板上進行弧線軌跡追蹤任務之速度及指向圖

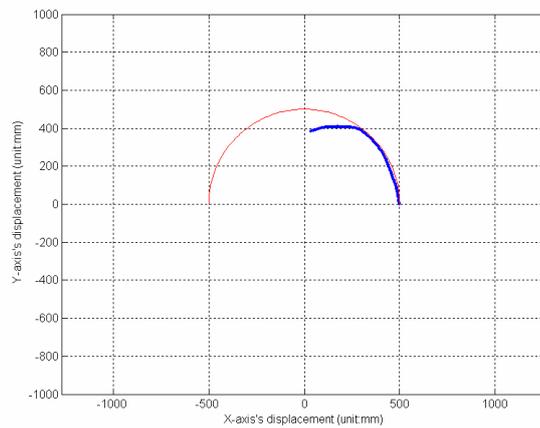


圖 67 對照組 2:於木板上進行弧線軌跡追蹤任務之軌跡圖

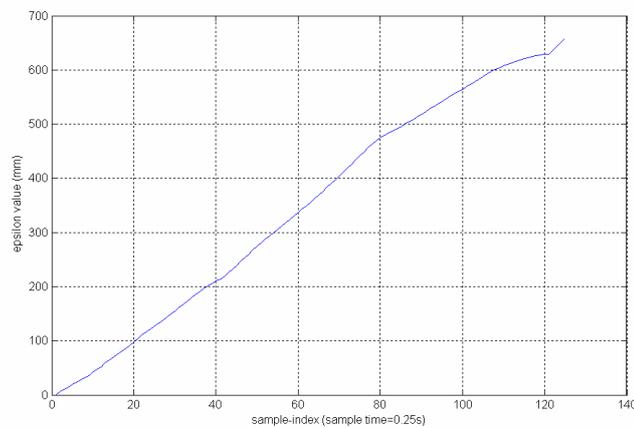


圖 68 對照組 2:於木板上進行弧線軌跡追蹤任務之移動誤差圖

實驗組 1：

(1) 平台於水平狀態下的木板上走直線軌跡且平台指向從 0 度變化至-25 度。

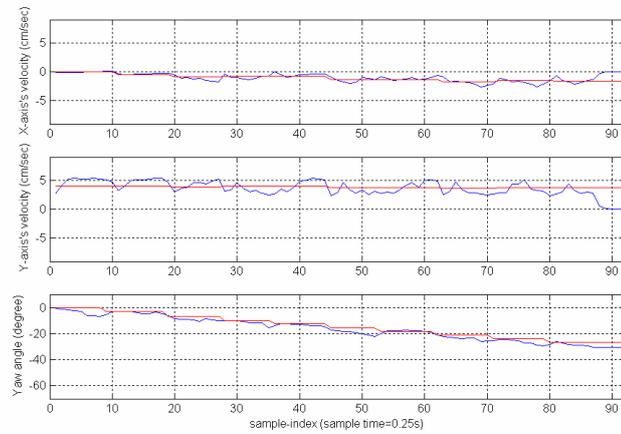


圖 69 實驗組 1-1:於木板上進行直線軌跡追蹤任務之速度及指向圖

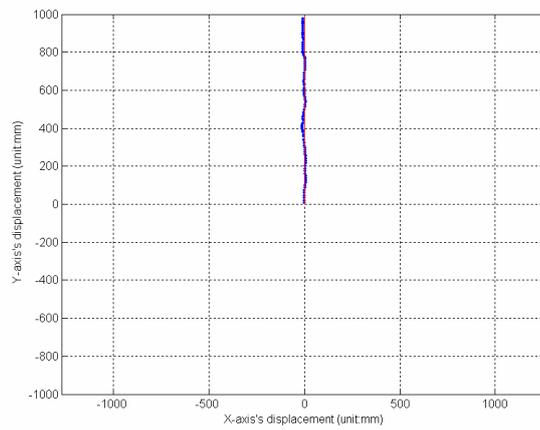


圖 70 實驗組 1-1:於木板上進行直線軌跡追蹤任務之軌跡圖

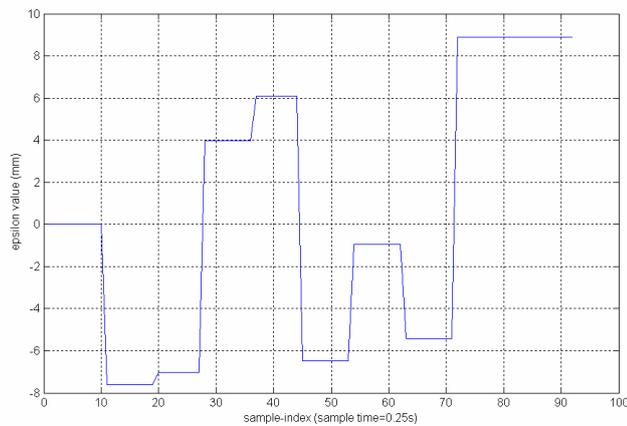


圖 71 實驗組 1-1:於木板上進行直線軌跡追蹤任務之移動誤差圖

(2) 平台於水平狀態下的木板上走弧線軌跡且平台指向從 0 度變化至 -45 度。

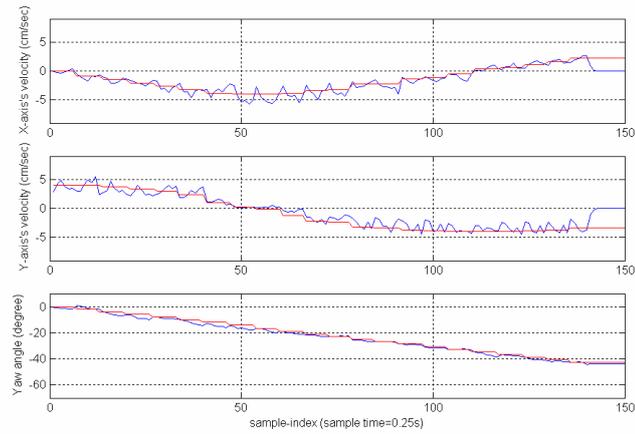


圖 72 實驗組 1-2:於木板上進行弧線軌跡追蹤任務之速度及指向圖

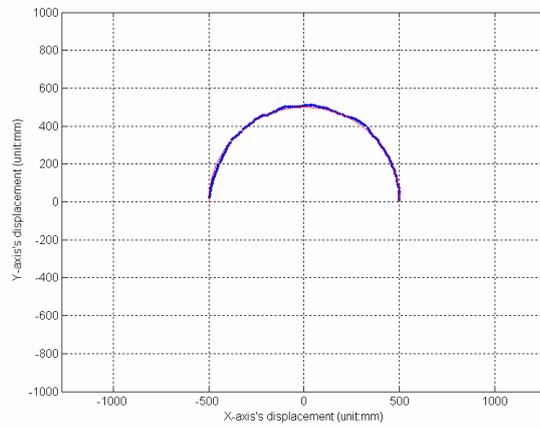


圖 73 實驗組 1-2:於木板上進行弧線軌跡追蹤任務之軌跡圖

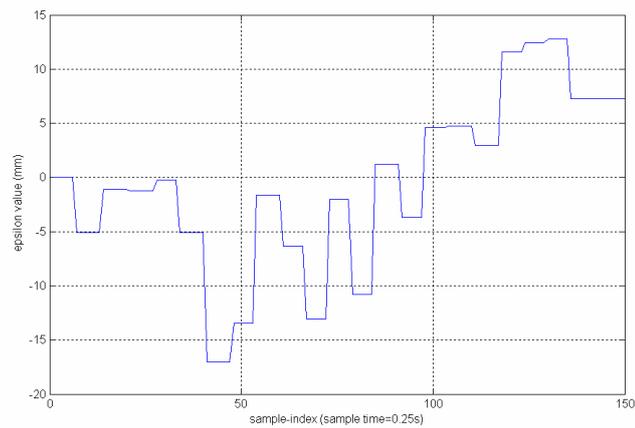


圖 74 實驗組 1-2:於木板上進行弧線軌跡追蹤任務之移動誤差圖

實驗組 2：

(1) 平台於水平狀態下的地毯上走直線軌跡且平台指向從 0 度變化至 -45 度。

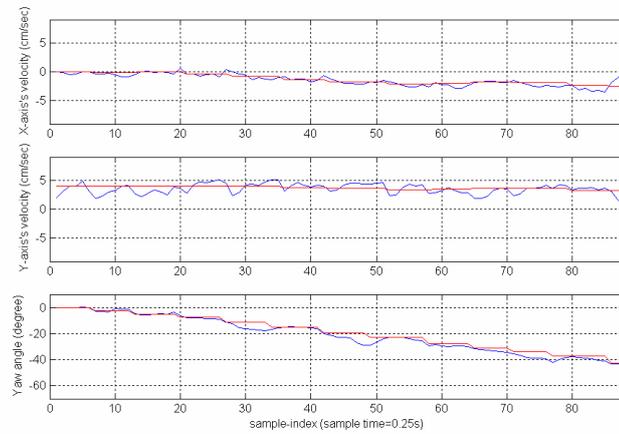


圖 75 實驗組 2-1:於地毯上進行直線軌跡追蹤任務之速度及指向圖

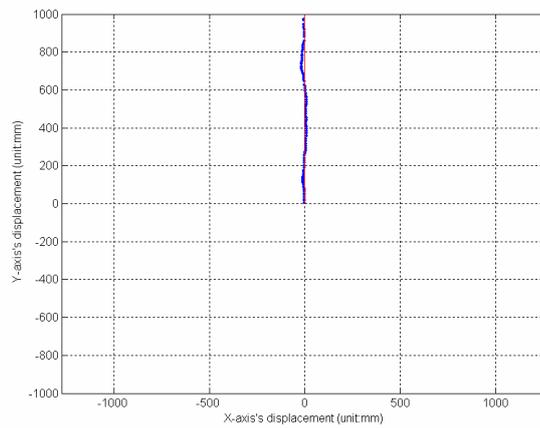


圖 76 實驗組 2-1:於地毯上進行直線軌跡追蹤任務之軌跡圖

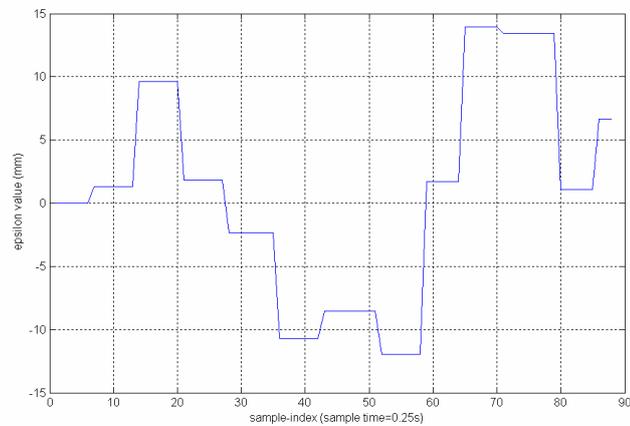


圖 77 實驗組 2-1:於地毯上進行直線軌跡追蹤任務之移動誤差圖

(2) 平台於水平狀態下的地毯上走弧線軌跡且平台指向從 0 度變化至 -45 度。

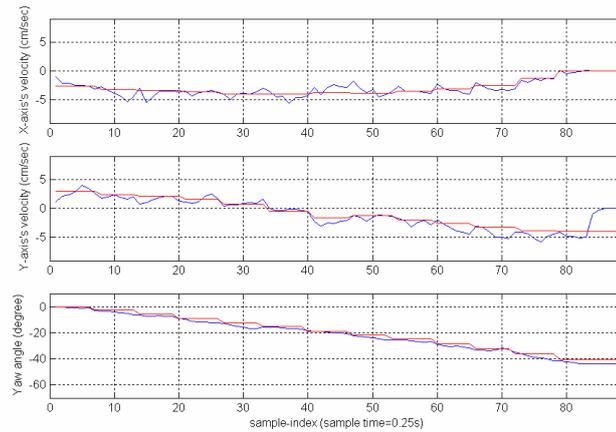


圖 78 實驗組 2-2:於地毯上進行弧線軌跡追蹤任務之速度及指向圖

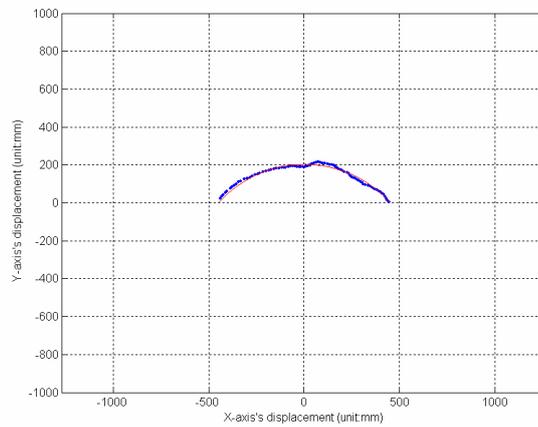


圖 79 實驗組 2-2:於地毯上進行弧線軌跡追蹤任務之軌跡圖

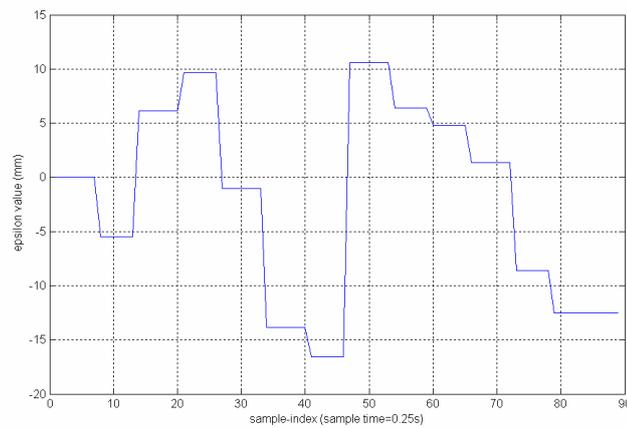


圖 80 實驗組 2-2:於地毯上進行弧線軌跡追蹤任務之移動誤差圖

實驗組 3：

(1) 平台於水平狀態下的磨石地磚上走直線軌跡且平台指向從 0 度變化至 -45 度。

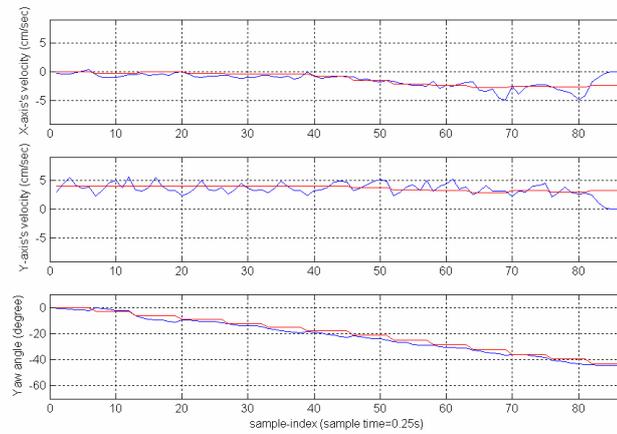


圖 81 實驗組 3-1:於磨石地磚上進行直線軌跡追蹤任務之速度及指向圖

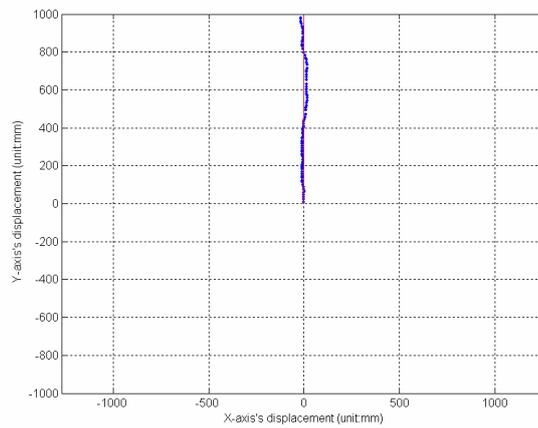


圖 82 實驗組 3-1:於磨石地磚上進行直線軌跡追蹤任務之軌跡圖

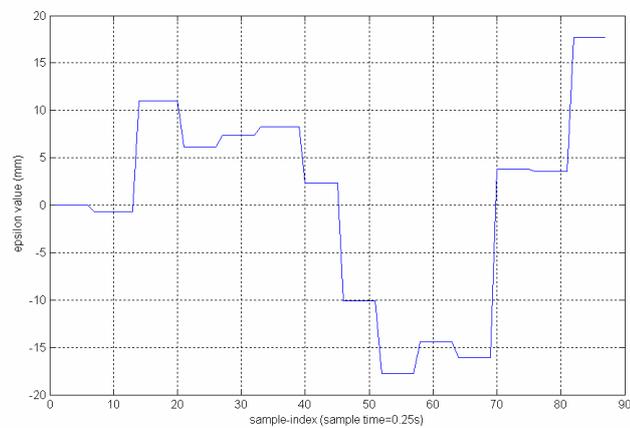


圖 83 實驗組 3-1:於磨石地磚上進行直線軌跡追蹤任務之移動誤差圖

(2) 平台於水平狀態下的磨石地磚上走弧線軌跡且平台指向從 0 度變化至-60 度。

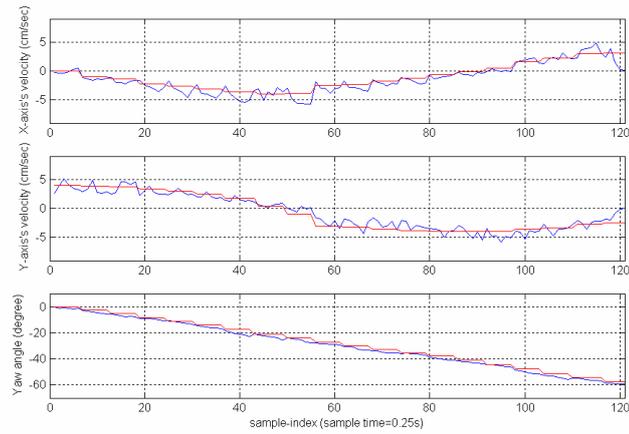


圖 84 實驗組 3-2:於磨石地磚上進行弧線軌跡追蹤任務之速度及指向圖

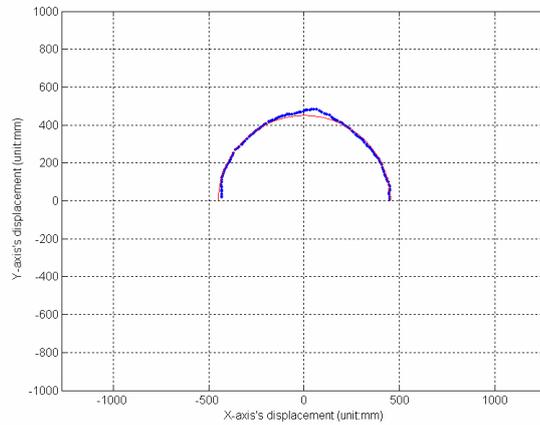


圖 85 實驗組 3-2:於磨石地磚上進行弧線軌跡追蹤任務之軌跡圖

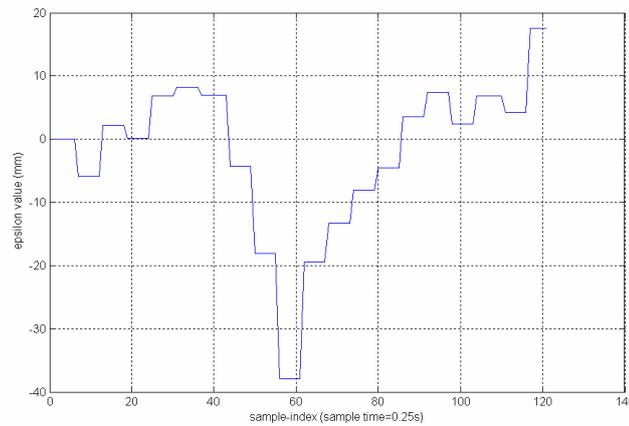


圖 86 實驗組 3-2:於磨石地磚上進行弧線軌跡追蹤任務之移動誤差圖

實驗組 4：

(1) 平台於傾斜 9 度的木板上走直線軌跡且平台指向從 0 度變化至 -60 度。

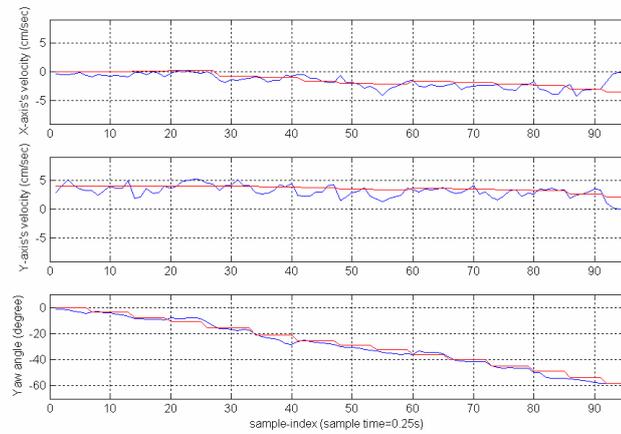


圖 87 實驗組 4-1:於傾斜木板上進行直線軌跡追蹤任務之速度及指向圖

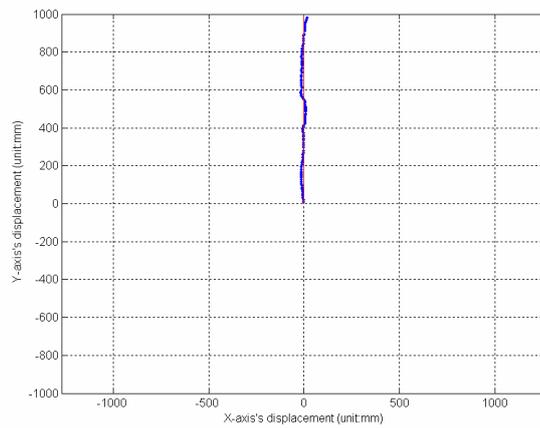


圖 88 實驗組 4-1:於傾斜木板上進行直線軌跡追蹤任務之軌跡圖

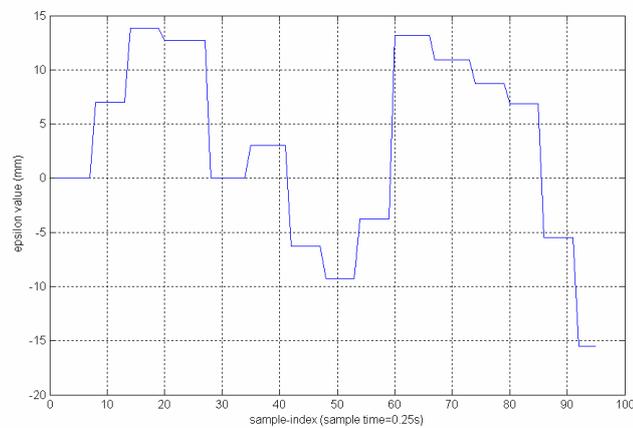


圖 89 實驗組 4-1:於傾斜木板上進行直線軌跡追蹤任務之移動誤差圖

(2) 平台於傾斜 9 度的木板上走弧線軌跡且平台指向從 0 度變化至 -60 度。

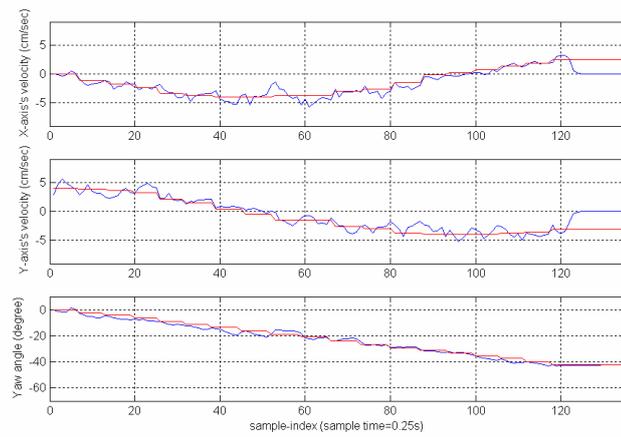


圖 90 實驗組 4-2:於傾斜木板上進行弧線軌跡追蹤任務之速度及指向圖

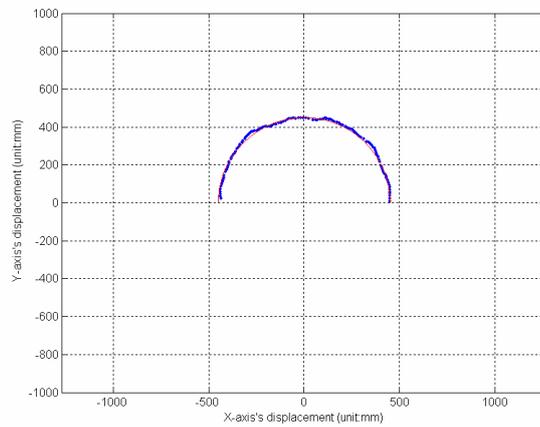


圖 91 實驗組 4-2:於傾斜木板上進行弧線軌跡追蹤任務之軌跡圖

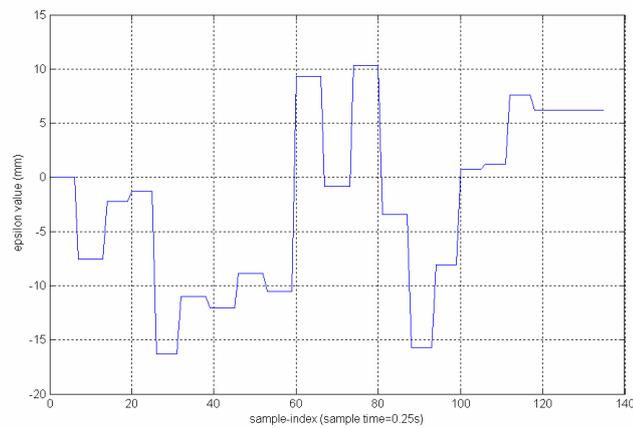


圖 92 實驗組 4-2:於傾斜木板上進行弧線軌跡追蹤任務之移動誤差圖

經由上述的實驗數據，我們可以發現與時間性軌跡追蹤法比較起來，即時速度控制之空間性軌跡追蹤法(RVC-SPTM)在多種不同地形上，包括：木板、地毯、磨石地磚以及傾斜的木板，皆能擁有不錯的速度控制及軌跡追蹤效能。

6.3 碰撞偵測及動態路徑規劃實驗結果

本節將使用即時速度控制之空間性軌跡追蹤法(RVC-SPTM)進行全方向運動平台的速度控制及軌跡追蹤任務，同時測試該方法中碰撞偵測及動態路徑規劃的功能。

為了測試方便起見，我們讓全方向運動平台在木板上以一個設定的速度值，分別進行直線及圓弧形狀的軌跡追蹤任務，然後在軌跡上放置障礙物，觀察該平台是否可以執行碰撞偵測以及動態路徑規劃的任務。相關實驗圖形請參照圖 93 至圖 98。

下列的圖形大致上分為二種；第一種是「速度及指向角度圖」，其中記錄了運動平台 X 軸方向和 Y 軸方向上的理論速度值(紅色曲線)和實際速度值(藍色曲線)，以及運動平台 Z 軸方向上的理論指向角度值(紅色曲線)和實際指向角度值(藍色曲線)；第二種則是「軌跡圖」，其中記錄了運動平台中心點在絕對座標中的理論移動軌跡(紅色曲線)和實際移動軌跡(藍色曲線)；另外，我們在軌跡圖中標示出障礙物的實際大小(240mm*200mm)和位置，用以說明平台遇到障礙物的實際狀況。必須注意的是，因為軌跡圖中記錄的是運動平台「中心點」在絕對座標中實際移動資訊，而運動平台是有體積的物體，因此在軌跡圖中是不會看到運動軌跡與障礙物接觸的現象發生的。

實驗 1：

進行直線軌跡追蹤任務，並執行碰撞偵測以及動態路徑規劃之功能。

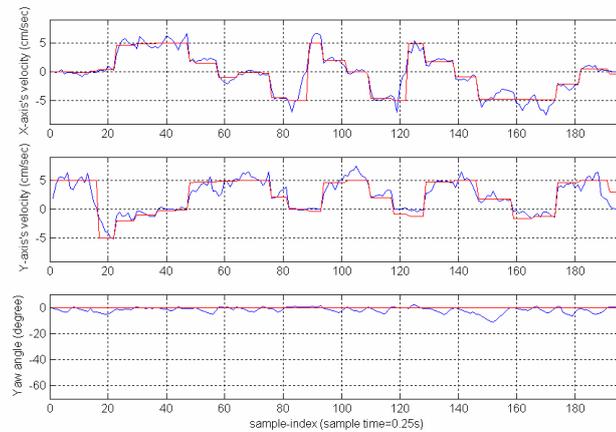


圖 93 實驗 1:碰撞偵測及動態軌跡規劃之速度及指向圖

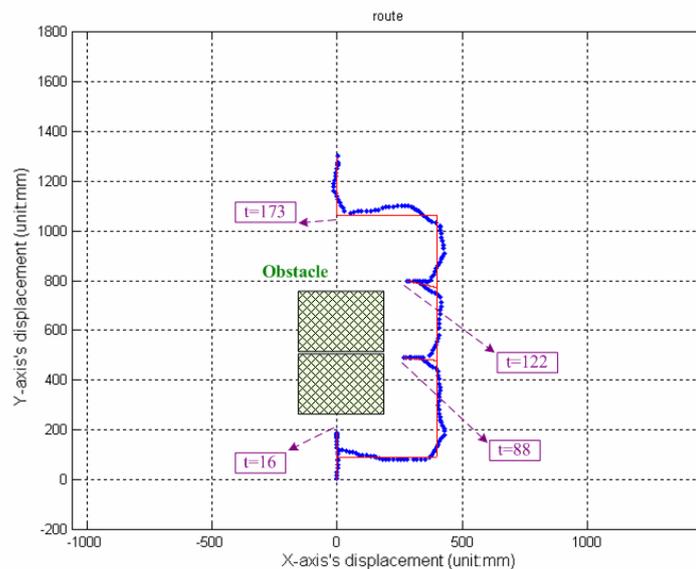


圖 94 實驗 1:碰撞偵測及動態軌跡規劃之軌跡圖

由圖 93 可知，平台於 sample_index=16 時碰撞到障礙物而停止，此時平台開始規劃並執行一條避障路徑以避開障礙物；但平台分別於 sample_index=88 及 122 時再次碰撞到障礙物而停止，此時平台將會再次規劃並執行另一條新的避障路徑；於 sample_index=173 時，平台回到原本預定的路徑上，並且繼續往終點前進。實際的行進路線圖形可參照圖 94。

實驗 2：

進行直線軌跡追蹤任務，並執行碰撞偵測以及動態路徑規劃之功能。

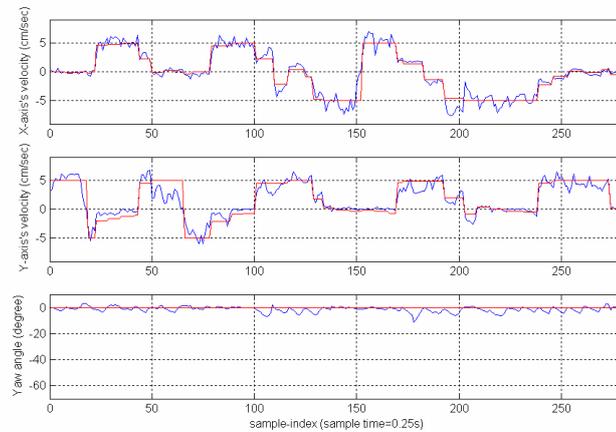


圖 95 實驗 2:碰撞偵測及動態軌跡規劃之速度及指向圖

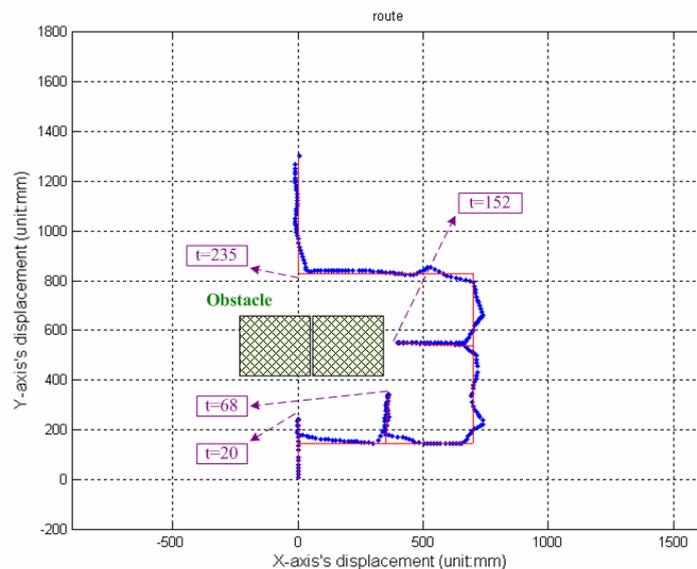


圖 96 實驗 2:碰撞偵測及動態軌跡規劃之軌跡圖

由圖 95 可知，平台於 $\text{sample_index}=20$ 時碰撞到障礙物而停止，此時平台開始規劃並執行一條避障路徑以避開障礙物；但平台分別於 $\text{sample_index}=68$ 及 152 時再次碰撞到障礙物而停止，此時平台將會再次規劃並執行另一條新的避障路徑；於 $\text{sample_index}=235$ 時，平台回到原本預定的路徑上，並且繼續往終點前進。實際的行進路線圖形可參照圖 96。

實驗 3：

進行弧線軌跡追蹤任務，並執行碰撞偵測以及動態路徑規劃之功能。

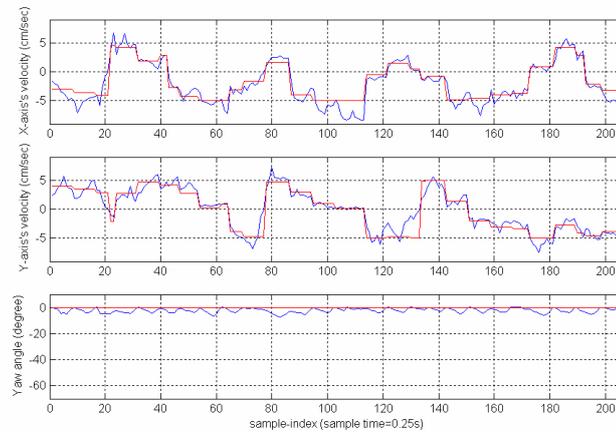


圖 97 實驗 3:碰撞偵測及動態軌跡規劃之速度及指向圖

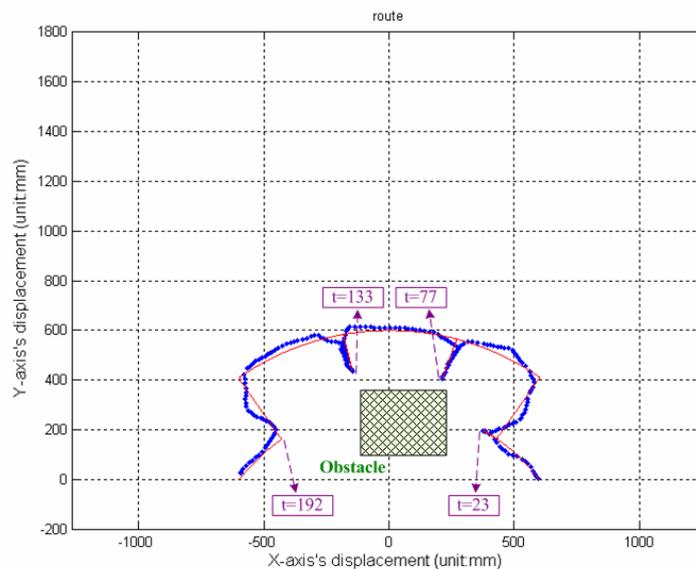


圖 98 實驗 3:碰撞偵測及動態軌跡規劃之軌跡圖

由圖 97 可知，平台於 $\text{sample_index}=23$ 時碰撞到障礙物而停止，此時平台開始規劃並執行一條避障路徑以避開障礙物；但平台分別於 $\text{sample_index}=77$ 及 133 時再次碰撞到障礙物而停止，此時平台將會再次規劃並執行另一條新的避障路徑；於 $\text{sample_index}=192$ 時，平台回到原本預定的路徑上，並且繼續往終點前進。實際的行進路線圖形可參照圖 98。

經由上述的實驗數據，可以發現我們所設計的即時速度控制之空間性軌跡追蹤法(RVC-SPTM)中的碰撞偵測及動態路徑規劃功能，擁有經由多次與障礙物發生碰撞而動態地更改其避障路徑的能力；因此，不論預設的移動軌跡是直線還是弧形線段，以及在不知道障礙物的位置和大小的情況下，該方法均能透過多次與障礙物發生碰撞而動態地更改其避障路徑，以使平台機器人可以順利的繞過該障礙物，並回到原先要求的路徑上，繼續往目的地前進。



第七章 結論

7.1 研究成果

在第六章中，我們已經分別列出光流影像定位、使用 RVC-SPTM 進行速度控制及軌跡追蹤、以及使用 RVC-SPTM 進行碰撞偵測及動態路徑規劃等實驗結果及分析數據。

經由這些實驗數據，我們可以知道光流影像是一種低成本卻擁有高精確度的相對定位方法；而 RVC-SPTM 是一個可以在多種未知性質的地面上進行運動平台的速度控制、軌跡追蹤、碰撞偵測以及動態路徑規劃等任務的適應性控制方法。結合這兩者的優點並配合上全向輪所擁有的高運動自由度之特性，我們可以建構出一個低成本且擁有高精確度定位能力的全方向輪式運動平台，且該運動平台擁有在多種不同且未知性質的地面上進行速度控制、軌跡追蹤、碰撞偵測以及動態路徑規劃的能力。同時我們也整合了嵌入式網路技術，以達成分散式運算及遠端操控機器人之目的。

擁有上述特點的全方向輪式運動平台非常適合應用在家庭機器人(Home Robot)、工廠無人搬運車(AGV)、醫療服務機器人…等方面上，利用其優越的定位能力及操控性能，以提供一個安全性高且操控便利的全方向輪式運動平台架構。

7.2 未來展望

光流影像定位方法與其他定位方法比較起來，雖然擁有較高的精確度，但實際上還是會有量測誤差出現；而且量測誤差是會累積的，因而對整個定位方法造成很大的影響。因此，未來將可以考慮結合 landmarks、maps 或是 beacons 等絕對定位方法，以改善該相對定位方法的準確性。

另外，為了開發及除錯的便利性，目前的定位及控制方法均建構在 PC 端的 Robot Agent 上；未來將可以考慮把這些定位及控制方法移植整合到平台機器人的系統控制板上，以增加該平台機器人的獨立性和完整性。未來，使用者只要身處

於無線網路的環境中，便可以透過無線網路使用專屬遙控器、手機、PDA 或是 Notebook 來控制該平台進行動作。

另外，平台機器人目前可以透過執行多次的碰撞偵測及動態路徑規劃以避開出現在預設路徑上的障礙物，但是其效率稍為不足；因此，未來將可以考慮加裝其他感測器，如：超音波測距儀、紅外線感測器…等，當平台機器人偵測到碰撞時，可以利用這些儀器對障礙物進行分析，幫助該平台機器人規劃一條正確且有效率的避障路徑，讓平台可以有效率地避開該障礙物並回到預設的軌跡上進行運動。



Reference:

- [1] J. Borenstein, H. R. Everett, and L. Feng, *Navigating Mobile Robots: Systems and Techniques*, Wellesley, 1996.
- [2] Georgiev, and P.K. Allen, "Localization methods for a mobile robot in urban environments," *IEEE Trans. Robotics*, vol. 20, pp. 851-864, Oct. 2004.
- [3] C. D. McGillem and T. S. Rappaport, "Infra-red location system for navigation of autonomous vehicles," *IEEE Int. Conf. Robotics and Automation*, pp.1236-1238, 1988.
- [4] Ohya, A. Kosaka, and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *IEEE Trans. Robotics and Automation*, vol. 14, no. 6, pp. 969-978, Dec 1998.
- [5] C. C. Tsai, et al., "Kinematics Control of an Omnidirectional Mobile Robot," in *Proceedings of CACS Automatic Control Conference*, 2005.
- [6] The home page of iRobot company, [Online]. Available: <http://www.irobot.com/>
- [7] K. B. Ariyur and M. Krstic, *Real-time Optimization by Extremum-Seeking Control*, John Wiley & Sons, 2003.
- [8] K.B. Ariyur, M. Krstic, "Analysis and design of multivariable extremum seeking," *Proceedings of the American Control Conference*, May 8-10, 2002.
- [9] Y. Zhang, "Stability and Performance Tradeoff with Discrete Time Triangular Search Minimum Seeking," *Proceedings of the American Control Conference*, Chicago, June 2000.
- [10] M. Davidson and V. Bahl, "The Scalar ε -Controller: A Spatial Path Tracking Approach for ODV, Ackerman, and Differentially-Steered Autonomous Wheeled Mobile Robots," *Proceedings of IEEE International Conference on Robotics & Automation*, 2001.
- [11] D. K. Sorensen, "On-Line Optical Flow Feedback for Mobile Robot Localization /Navigation," Master Thesis, Texas A&M University, USA, 2003.
- [12] J. F. Blumrich, "Omnidirectional Wheel", US Patent 3,789,947, Feb 5, 1974.

- [13] Mark E. Smith, "Introduction to a wheelchair with omni-direction wheels", [Online]. Available: <http://www.wheelchairjunkie.com/page10.html>
- [14] 蔡寬騰, 蔡清元(指導教授), 「全方向輪式機器人之建構及其利用立體視覺與超音波資訊之導航與避障」, 國立成功大學, 碩士論文, 民國 94 年。
- [15] P.F. Muir, C.P. Neuman, "Kinematic modeling of wheeled mobile robots," *J. Robotic Systems*, 4(2):281-340, 1987.
- [16] S. M. Killough, F.G. Pin, "Design of an Omnidirectional Holonomic Wheeled Platform Prototype," *IEEE Int. Conf. Robotics and Automation*, pp. 84-90, May, 1992.
- [17] Y.P. Leow, K.H. Low and W.K. Loh, "Kinematic Modeling and Analysis of Mobile Robots with Omni-Directional Wheels," *7th International Conference on Control, Automation, Robotics and Vision (ICARCV'02)*, Dec, 2002.
- [18] Lee, B.H., "Embedded internet systems: Poised for Takeoff," *IEEE Internet Computing*, vol. 2, pp. 24 -29, 1998.
- [19] Filman, R.E., "Embedded internet systems come home," *IEEE Internet Computing*, vol. 5, Issue: 1, pp. 52 -53, 2001.
- [20] Axelson, J., *Embedded Ethernet and Internet Complete: Designing and Programming Small Devices for Networking*, ISBN#1-931448-00-0, 2003.
- [21] Li-Wei Wu, Jwu-Sheng Hu, "Distributed Embedded Real-Time Ethernet Platform for Robots Control," *IEEE International Conference on Mechatronics*, July 10-12, 2005.
- [22] Li-Wei Wu, Jwu-Sheng Hu, "Embedded Platform for Complex Intelligent Machines," Contest on Embedded Software Design, Ministry of Education, Taipei, Taiwan, R. O. C., 2003.
- [23] Li-Wei Wu, Jwu-Sheng Hu, "Achieving Internet Synchronization Clock Error Detection with HRTP," *Proceedings R. O. C. Automatic Control Conference*, Taipei, 2003.
- [24] Jwu-Sheng Hu and Li-Wei Wu, "Robot Control System Using Embedded Network," 中華民國專利申請案號(4/6): 93109415, 專利公告號: I238622。
- [25] Wu, Li-Wei and Hu, Jwu-Sheng, "Distributed Embedded Control Platform for Robots Using Real-Time Ethernet: Design Concept, System Architecture, and

Implementation,” in revision, *IEEE Robotics and Automation Magazine*, 2006.

- [26] Li-Wei Wu, Jung-Hung Cheng, Yung-Jung Chang and Jwu-Sheng Hu, “An Omi-directional Mobile Platform Using Embedded Network Control and Optical Flow Motion Measurement,” *Automatic Control Conference (CACCS 2005)*, Nov. 18-19, 2005.
- [27] 吳立偉，廖憲政 “以腦電波控制的嵌入式網路機器人架構”，中華民國專利申請案號 (12/10) : 93138348(民 93)。
- [28] Li-Wei Wu, Hsien-Cheng Liao, “Architecture of an Embedded Internet Robot System Controlled by Brain Waves,” US Patent Application No. 11/296,217 (2005).
- [29] 吳立偉，鄭榮煌，張永融，胡竹生，“運用對地面光流影像偵測法定位全方向運動之網路控制平台設計”，中華民國專利申請案號(11/4): 94138828 (民 94)。
- [30] Li-Wei Wu, Jung-Hung Cheng, Yung-Jung Chang and Jwu-Sheng Hu, “The Design of Network Control Omni-directional Motion Platform Based on Optical Flow Navigation/Localization Technique,” US Patent Application No.:11/370,929.
- [31] Li-Wei Wu, Hsien-Cheng Liao, Jwu-Sheng Hu, Pei-Chen Lo, “Brain-Controlled Robot Agent: an EEG-Based eRobot Agent,” in revision, *IEEE Transactions on Systems, Man, and Cybernetics--Part C : Applications and Reviews*, 2006.
- [32] Wu, Li-Wei and Hu, Jwu-Sheng, “A Split-Type Robot: Explorer,” submitted to *IEEE Robotics and Automation Magazine*, 2006.
- [33] Wu, L.-W., Hu, J.-S. Demo Video: A Distributed Embedded Control Platform for Robots Using Real-Time Ethernet. [Online]. Available: <http://xlab.cn.nctu.edu.tw/Liwei/eRobot.htm>.