

國立中央大學

資訊工程學系  
碩士論文

導航機器人之研製  
The Development of a Navigation Mobile Robot

研究生：楊智翔

指導教授：蘇木春 博士

中華民國 九十九 年 六 月

國立中央大學碩士班研究生

論文指導教授推薦書

資訊工程 學系/研究所 楊智翔 研究生

所提之論文

導航機器人之研製

係由本人指導撰述，同意提付審查。

指導教授 蔡木君 (簽章)

99年7月7日

國立中央大學碩士班研究生  
論文口試委員審定書

資訊工程 學系/研究所 楊智翔 研究生

所提之論文

導航機器人之研製

經本委員會審議，認定符合碩士資格標準。

學位考試委員會召集人

委

員

劉志文

許木君

趙子翔

曾繁貝

黃有進

中華民國 99 年 7 月 7 日

## 摘要

機器人的出現一開始主要是為了代替人類從事一些具高危險性、高精密性或是高重複性的工作，以降低工作的風險與成本。隨著科技的進步，智慧型機器人的發展所涵蓋領域相當廣泛，舉凡軍事、工業、生活、醫療、娛樂…等等應用，人類對於智慧型機器人的需求與日俱增，因此如何設計與研製一台智慧型機器人是我們的首要目標。

對於智慧型機器人而言，導航、定位、避障都是不可或缺的主要功能。因此本論文設計與研製了一台導航機器人，其無障礙的任意走向具有擬人化的行為；在定位方面，採用滑鼠與陀螺儀來估測目前的位置；而即時避障系統採用向量場直方圖(Vector Field Histogram)的技術來達成。因此，可以在未知的環境下，利用沿牆行走演算法進行探索；在已知的環境中透過廣度優先搜尋(Breadth-first search)最短路徑規劃法和即時避障系統順利到達目標地。此外，也可以透過軟體模擬無線遙控器直接對導航機器人下達指令，使其具有多元化功能的智慧型機器人。

關鍵字：機器人、導航、定位、避障、路徑規劃

## Abstract

One of the original application purposes of robots is to replace humans to work in a high risk environment and/or to complete tasks with a request of high precision or repetitive procedures. As modern electronics and information technologies advance, the applications of intelligent robots widely cover the many different domains, such as in the military industry, autonomous industries, life environments, health caring facilities, entertainment, and so on. As the need of an intelligent robot grows day by day, how to design and develop an intelligent robot is a very challenging task.

For intelligent robots, three most essential functionalities are autonomous navigation, localization (positioning) and obstacle avoidance. Therefore, this thesis presents a navigation robot which moves in an omni-directional way just like an anthropomorphic behavior. In positioning, we use a mouse and a gyroscope to estimate the current location of the robot. In addition, we use the Vector Field Histogram algorithm to implement a real-time obstacle avoidance system. While we use a wall-following algorithm to explore an unknown environment, we plan a shortest path in a known environment by the Breadth-first search algorithm. Furthermore, we use a real-time obstacle avoidance system simultaneously to during the navigation tour. In addition, we also develop a software application to simulate a controller to wireless command the robot. Finally, an intelligent robot with multiple functionalities was implemented.

Keyword : robot 、 navigation 、 localization 、 obstacle avoidance 、 path planning

## 誌謝

研究生的兩年確實是過得很快，感覺才剛要進入狀況就要畢業了。不過這兩年的確學到了很多東西。感謝蘇木春教授在這兩年的指導，幫助我在研究所產生的疑惑或問題時的解答，讓我感到受益良多。

此外，感謝大寶、長毛、小捲、良福的幫忙，謝謝博班的引導，讓我適應研究生的生活；感謝嘴砲、BOGGY、小巨人，讓我有一起奮鬥的夥伴；感謝猩猩、豪洨、丸子、嵩陽、阿俊、小鼎學弟妹時常製造歡笑，讓我減壓。真的非常感謝大家的幫忙。

再來，感謝我的家人，那麼用心的把我栽培到現在，提供我食衣住行上所有的協助，再來該是我照顧你們的時候了。

最後感謝我的女友，感謝她不時的為我加油打氣，陪我度過壓力最大的時期。

# 目錄

摘要 .....	i
ABSTRACT.....	ii
誌謝 .....	iii
目錄 .....	iv
圖目錄 .....	vi
表目錄 .....	viii
第一章 緒論.....	1
1-1 研究背景 .....	1
1-2 文獻探討 .....	3
1-3 研究動機與目的 .....	5
1-4 論文架構.....	6
第二章 硬體架構和規格 .....	7
2-1 車體結構設計 .....	8
2-1-1 第一層平台 .....	9
2-1-2 第二層平台 .....	9
2-1-3 第三層平台 .....	9
2-2 核心控制電路-AVR 開發板.....	10
2-2-1 直流馬達方向和速度訊號輸出板 .....	10
2-2-2 紅外線接收板 .....	12
2-3 直流馬達和直流馬達驅動控制板.....	13
2-3-1 IG-42GM 直流馬達.....	13
2-3-2 直流馬達驅動控制板.....	16
2-4 全向輪.....	18
2-5 滑鼠.....	18
2-6 紅外線模組 .....	19
2-7 電池與穩壓電路 .....	21

2-8	陀螺儀.....	22
第三章	機器人系統控制設計及原理.....	24
3-1	機器人控制流程概觀 .....	24
3-2	三輪運動學模型 .....	26
3-3	全方位走向的實現 .....	29
3-4	機器人定位系統 .....	34
第四章	行走模式.....	38
4-1	機器人即時避障系統 .....	39
4-2	沿牆行走.....	44
4-3	BFS 最短路徑規劃 .....	49
第五章	實驗結果.....	52
5-1	機器人的全方位走向實驗.....	52
5-2	沿牆行走實驗.....	56
5-2-1	正方形.....	56
5-2-2	L 型 .....	58
5-2-3	N 型 .....	59
5-3	BFS 最短路徑規劃實驗.....	61
5-3-1	實驗環境 1-無障礙物 .....	61
5-3-2	實驗環境 2-單一障礙物 .....	63
5-3-3	實驗環境 3-多個障礙物 .....	67
第六章	結論與未來展望 .....	71
6-1	結論.....	71
6-2	未來展望.....	72
	參考文獻 .....	73
附錄 1	GP2D12 距離、電壓、ADC 對照表 .....	76



# 圖目錄

圖 1	威力狗清潔機器人.....	2
圖 2	SeQ-1 保全機器人.....	2
圖 3	Upitor 導覽機器人.....	2
圖 4	美國軍方用 BigDog 機器人.....	2
圖 5	MOTOMAN-SDA5D.....	2
圖 6	Actroid 機器人.....	2
圖 7	TOPIO.....	2
圖 8	導航機器人的硬體架構圖.....	7
圖 9	導航機器人車體結構圖.....	8
圖 10	AVR 開發板.....	10
圖 11	ADC 方塊圖.....	12
圖 12	IG-42GM 直流馬達.....	13
圖 13	PWM 方波圖.....	14
圖 14	雙軸直流馬達驅動器圖.....	16
圖 15	全向輪實體圖.....	18
圖 16	Rush FireGlider 滑鼠.....	18
圖 17	具避震系統的 FireGlider 滑鼠.....	19
圖 18	Sharp GP2D12 紅外線.....	20
圖 19	電壓和距離的非線性迴歸曲線.....	21
圖 20	導航機器人使用的電源.....	21
圖 21	穩壓模組電路.....	22
圖 22	IMU 6-DOF v4 模組中的 IDG-500 陀螺儀.....	23
圖 23	系統控制流程圖.....	25
圖 24	三輪式的導航機器人平台.....	26
圖 25	三輪運動學示意圖.....	27
圖 26	各個馬達合力的計算.....	28
圖 27	全方位走向的八個區域.....	30
圖 28	全方位走向的實現流程圖.....	31

圖 29	區域 1 演算法流程.....	32
圖 30	區域 2 演算法流程.....	33
圖 31	雷射滑鼠的成像原理.....	35
圖 32	定位系統初始化流程.....	36
圖 33	定位系統流程圖.....	37
圖 34	機器人的紅外線感測器配置方位圖.....	38
圖 35	VFH 概念圖.....	40
圖 36	Active Window 轉換成 Polar Histogram 概念圖.....	41
圖 37	實際地形模擬圖.....	43
圖 38	實際地形模擬圖相對應的 Polar Histogram H 圖.....	43
圖 39	沿著牆面走.....	44
圖 40	凹角牆面行走.....	44
圖 41	凸角牆面行走.....	45
圖 42	沿牆行走流程圖.....	47
圖 43	BFS 示意圖.....	49
圖 44	新增子目標地示意圖.....	51
圖 45	BFS 最短路徑規劃模式流程圖.....	51
圖 46	BFS 實驗環境.....	61

# 表目錄

表 1	直流馬達 AVR 開發板的傳輸規格.....	10
表 2	直流馬達 AVR 板協定基本格式.....	11
表 3	馬達 Data 格式 .....	11
表 4	紅外線 AVR 開發板的傳輸規格.....	12
表 5	IG-42GM 直流馬達規格.....	13
表 6	馬達和 PWM Duty Cycle 對應表.....	15
表 7	VNH2SP30 規格.....	17
表 8	直流馬達驅動板方向控制表.....	17
表 9	三輪運動學符號表.....	27
表 10	紅外線的方位角度.....	38
表 11	VFH 符號說明.....	42
表 12	各角度行走實驗表.....	52
表 13	實驗表格符號說明.....	54
表 14	任意方向行走數據表.....	54
表 15	正方型地形實驗.....	56
表 16	正方型地形實驗數據.....	57
表 17	L 型地形實驗 .....	58
表 18	L 型地形實驗數據 .....	59
表 19	N 型地形實驗.....	59
表 20	N 型地形實驗數據.....	60
表 21	BFS 無障礙實驗.....	62
表 22	BFS 無障礙實驗數據.....	62
表 23	BFS 單一障礙物實驗 1.....	63
表 24	BFS 單一障礙物實驗 2 和實驗 3.....	64
表 25	BFS 單一障礙物實驗 4 和實驗 5.....	65
表 26	BFS 單一障礙物實驗數據.....	66
表 27	BFS 多障礙物實驗 1.....	67
表 28	BFS 多障礙物實驗 2 和實驗 3.....	68

表 29	BFS 多障礙物實驗 4 和實驗 5.....	69
表 30	BFS 多障礙物實驗數據.....	70

# 第一章緒論

## 1-1 研究背景

智慧型機器人產業是集精密機械、電子、機電控制、電腦計算、感測技術、人工智慧、移動平台、電源技術、系統整合技術、造型設計等多元先進技術於一體的自動化裝備，其涵蓋的產業包括機械、電機、電子、能源、材料、資訊、通訊…等等。據統計，近年來由於智慧型機器人的相關技術成熟，2009 年全球機器人行業總銷售量比 2006 年增長 10% [1]，說明了智慧型機器人的產業市場需求是不斷的在上升，因此各國政府已將機器人的產業發展，列為國內的重要政策。

根據國內報告顯示，工業用機器人主用途分佈：搬運佔 42%，熔接佔 30%，無塵室佔 12%，組裝佔 9%，其他佔 7%；主要應用的產業分佈：汽車業佔 36%，電機電子佔 14%，化工業佔 14% [2]。不斷的顯示智慧型機器人慢慢的與產業密不可分的關係。根據台灣的第六屆全國工業發展會議第七分組：智慧型機器人產業之加速發展策略中提到，行政院於 2005 年產業科技策略會議(SRB)提出『智慧型機器人產業發展與應用策略』，預期在 2015 年以前，將智慧型機器人的發展分成三個階段來推動：1.產業規模化 2.人才國際化 3.產品創意化。期望在 2015 年產值達 2500 億台幣，政府的政策也說明了智慧型機器人的發展已經是全世界主要發展趨勢。

目前國內專門在研究智慧型機器人的單位有不少個，如工研院機械所自行開發的『威力狗清潔機器人』如圖 1；工研院與新光保全合作開發的『新保 2 號』如圖 2、最新一代的是新保五號；精密機械研究發展中心研發的 Upitor 導覽機器人如圖 3；上銀科技，微星科技、明基電通、祥儀、雷虎、松騰實業等等。

放眼國際，各國智慧型機器人產品發展的領域各有不同，在技術能力也各有領先，如美國在國防應用領域如圖 4 和太空應用領域，日本在產業用機器人、人型機器人如圖 5~圖 7、保全導覽機器人，歐洲在醫療、照護用機器人上都各有優勢。因此積極投入智慧型機器人領域是我們首要的目標。



圖 1 威力狗清潔機器人 [3]



圖 2 SeQ-1 保全機器人 [4]



圖 3 Upitor 導覽機器人 [5]



圖 4 美國軍方用 BigDog 機器人 [6]



圖 5 MOTOMAN-SDA5D [7]



圖 6 Actroid 機器人 [7]



圖 7 TOPIO [7]

## 1-2 文獻探討

在智慧型機器人的領域中，關於輪型機器人的設計、導航與避障的研究不勝枚舉，雖然要解的問題都大同小異，但是解決的方法卻是千變萬化，而結果也都很好，我們先對導航和避障作一個簡單介紹如下。

導航一般可以分成兩種類型來討論。第一是全域性的導航，主要應用在已知的環境下，並使用全域路徑規畫法給予起點與終點後找出最佳路徑導航；第二是區域性的導航，主要應用在未知的環境下，使用感測器偵測環境資訊，利用避障系統求出目前(Local)最佳的導航路徑。

避障方面依感測器的類型可以分成兩種。第一是測距型感測器，如紅外線、超音波、雷射測距儀等等；第二是機器視覺型，通常使用攝影機當作偵測環境或目標的感測器。由於測距型感測器在計算速度、系統消耗資源和使用限制上比較有優勢，因此鎖定測距型感測器的研究來討論。我們將在下面對於機器人的設計、導航、避障的相關文獻做一個簡單的介紹。

全方向走位機器人的概念，在早期文獻中就已經可窺其身影 [8]-[9]，具有比兩輪機器人更多樣化行為的特性，不僅僅可以原地旋轉不需要迴轉半徑，也可以在任意角度和方向上行走，由於其高機動性的關係，一直都是被拿來研究的原因。

在機器人導航與避障的領域中，通常偵測外界環境資訊比較簡單的作法是利用距離感測器，而用來判斷導航方式和是否要避障的方法中，通常會用兩種方式來決定：虛擬力場方法(Virtual Force Field) [10]、模糊系統 [11]方法。

虛擬力場的概念就是障礙物會對車子產生排斥力，而目標地會對車子產生吸引力，藉由兩者的合力來決定車子的方向和速度。用虛擬力場的方法來避障的文獻中，比較經典的就屬 J. Borenstein 和 Y. Koren 在 1991 年提出的改良式的虛擬力：Vector Field Histogram(VFH) [12]。在 VFH 中，地圖被表示成

格子狀圖形，而每個格子中都存可能有障礙物的機率值，這種模型稱為 Certainty Grid [10], [13]-[15]。利用超音波偵測到障礙物的距離，根據超音波的特性，在介於感測器與偵測的距離中，將沒有障礙物的範圍在 Certainty Grid 中慢慢遞減；而在偵測到的距離將可能是障礙物的範圍在 Certainty Grid 中累加 [16]，之後再將自定範圍的方格 (Active Window) 中，計算每個區塊(每隔 5 角度劃分一個區塊)的障礙物的密度(polar obstacle density)，並將之平滑化，最後利用平滑化障礙物密度來計算車子前進的速度和轉向。

A. Saffiotti [17]在 1997 年就已經提出用模糊系統來實現避障的機制：以模糊系統來解決外部環境不明確資訊的應用，並建構一個階層式的啟動與協調多種行為機制，最後以模糊系統來決定與整合導航機器人目前的行為。

用模糊系統來實現避障的文獻中，其中一篇比較完整的作法，是由 A. Fatmi 等人 [18]提出以模糊系統為基礎的導航機器人，而其系統概念也是使用 A. Saffiotti [17]所提出的避障機制。A. Fatmi 等人 [18]利用 14 顆紅外線資訊、目標地的角度和目標地的距離來當作整個系統的輸入，並且定義了四種行為模式：emergency situation、obstacle avoidance、wall following、goal reaching，用其所定義的模糊規則，藉由模糊推論引擎來決定哪一種行為要被啟動，再輸入被啟動的行為模式需要的模糊輸入項目，來決定每一個被啟動的行為模式的輸出。以這種方式不斷的切換行為模式直到達目標地為止。



## 1-3 研究動機與目的

科技日新月異，許多國家相繼投入發展機器人產業，而研製機器人的目的在於希望有朝一日機器人能夠除了在軍事和工業應用以外，甚至能夠普及到家庭，為大眾服務，讓生活更便利，替代人類作危險的工作，陪伴人類娛樂…等；以往機器人的應用主要以工業用機器人居多，其目的為代替人類從事一些具高危險性、高精密性或是高重複性的工作，以降低工作的風險與成本；由於先進國家陸續出現高齡化及少子化的現象，又加上機器人的應用技術已達相當水準，機器人在室內自動化的角色扮演上將會越來越重要，不僅是一個移動的平台，更是一個資訊交流的平台，因此其智慧功能的強化是一項需要不斷超越的挑戰。

近年來智慧型機器人的發展已經成為機器人學領域中最受矚目的一個方向。智慧型機器人能夠於人類生活的環境中自由的移動，並且協助人類完成許多事情。因此，機器人的導航與在雜亂環境中的避障，就成為一個十分重要的課題。在智慧型機器人的研發中，導航定位就如同是將人類的思考與辨識能力，配合環境感知的各種訊息，移植到機器人上的實例，由此可知導航定位的重要。而目前大多數的機器人都使用所謂的輪式機器人，輪式機器人有以下幾種基本的優點，如移動快速，前進距離與轉向定位能夠計算較精準，所以本論文決定以輪型的機器人為出發點來研製。

而本論文主要是要設計與製作一個具有在未知環境下探索與自動避障系統的全方位走向導航機器人，希望機器人可以代替人類在不易到達或是有危險的環境下進行偵測；而在已知的環境下，可以透過最短路徑規劃和軟體模擬無線遙控來達到指定的目標地，以期望可以為行動不便或是高重複性的工作帶來便利。

## 1-4 論文架構

本論文總共分為六個章節，目的在於提出研製導航機器人的方法，包含結構設計、系統流程、多種行為模式、及其應用。其架構編排方式如下：

**第一章** 探討介紹機器人相關研究背景和機器人的文獻探討，以及研究動機與目的和整個論文架構。

**第二章** 介紹本論文所提出的導航機器人的結構設計，與硬體元件的規格介紹和使用。

**第三章** 在本章我們將介紹三輪式機器人的基本運動學模式，以及如何克服馬達的限制實現全方位走向。

**第四章** 在本章節我們將利用航機器人的基本功能，來做出在不同情況和環境下的行走模式。

**第五章** 本章節主要是利用前面數章所提出的方法，在真實環境下做實驗，以便驗證導航機器人的實用性。

**第六章** 本實驗的結論以及未來之研究方向。

## 第二章硬體架構和規格

本論文所研製的導航機器人的硬體架構圖如圖 8 所示，本章節基本上可以分為四個部分，第一部分為車體結構設計，介紹本論文因多方考量因素而設計的車體架構；第二部分為 AVR 開發板，在這部分我們將會討論如何接收 IRSharp 紅外線感測器的資料，以及如何利用 PWM 的訊號將我們要的轉速和轉向傳送給馬達驅動板，進而驅動馬達；第三部分為直流馬達驅動板和直流馬達的介紹；第四部分為導航機器人選用的元件介紹。

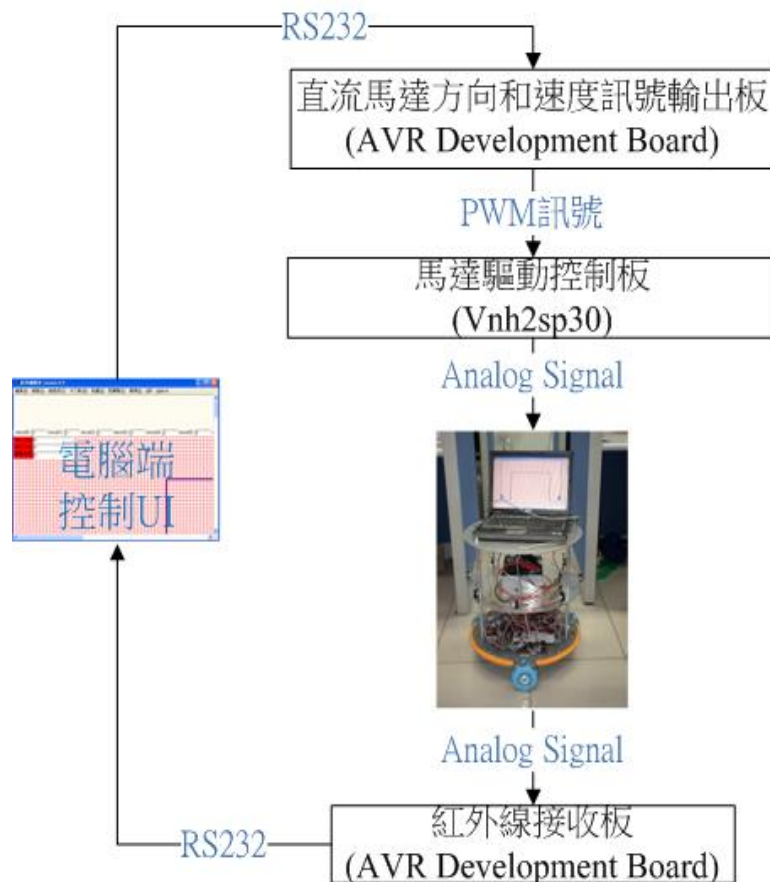


圖 8 導航機器人的硬體架構圖

## 2-1 車體結構設計

本論文所提出的導航機器人車體結構如圖 9 所示，在設計之初，考慮到元件的配置空間、錯綜複雜的電線電路互相纏繞問題、電池重量導致重心偏移的問題、長期重壓致使車體結構或輪軸套件變形以及未來機器人的擴充性；故將機器人整體架構以最容易平衡且最穩固的方式來實現。

車體本身採用三顆全向輪來平均支撐整體的重量，在輪軸套件方面，有別於一般的 L 型固定器較容易變形，我們採用一體成型的 F 型固定器如圖 9 第一層平台區域所示，這是影響馬達是否能夠以最有效率的方式驅動全向輪的關鍵之一，假使固定器變形，則會影響到馬達的軸心沒有辦法正位到輪軸連接器的鎖點。

我們將機器人整體分成三層平台，如圖 9 黃色線所標示區域，每個平台所配置的硬體設備將在下面的章節做更詳細的說明。

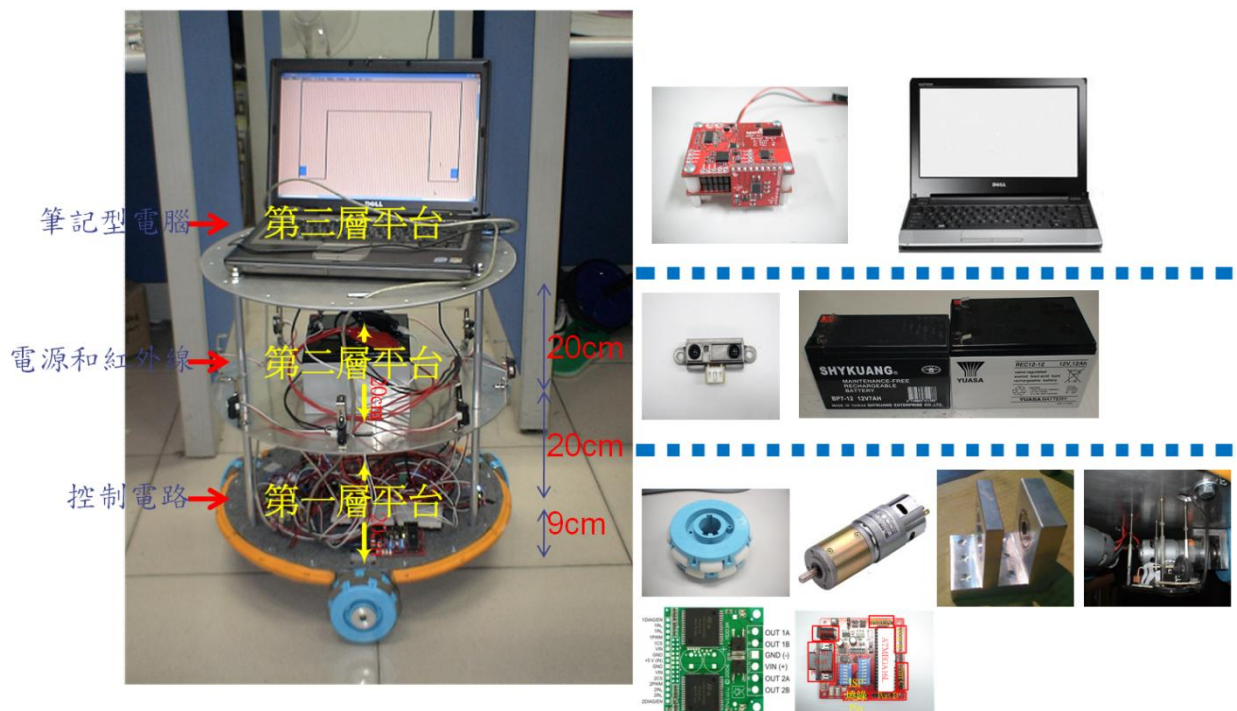


圖 9 導航機器人車體結構圖

### 2-1-1 第一層平台

在此層平台如圖 9 所示，主要放置的硬體設備為兩塊 AVR 開發板、兩塊 Vnh2sp30 雙軸直流馬達驅動器、電源穩壓器以及主要線路的匯集處；一塊 AVR 開發板是用來接收電腦端傳來的馬達速度訊號，另一塊是用來接收 IR Sharp GP2D12 傳回來的類比訊號；而兩塊直流馬達驅動器，則是負責根據 AVR 開發板傳來的 PWM 訊號輸出相對應的電壓來達到控制馬達轉速的功能；電源穩壓器則是把電池的 12V 電源降到 5V 或 3.3V 來提供 AVR 開發板電源使用。

### 2-1-2 第二層平台

在此層平台如圖 9 所示，主要放置的硬體設備為 8 顆 IR Sharp GP2D12 紅外線感測器、兩顆電池負責電源供應，因為電池本身重量不輕，所以為了保持車子的重心盡量平衡，所以我們將電池放置在第二層平台中心，一來電池的重量會由三根鋁柱平均分攤達到重心的平衡，二來可以和第一層平台錯綜複雜的線路分開放，避免線路不正確的接觸而導致嚴重的後果；IR Sharp GP2D12 紅外線感測器分別每隔 45 度角放置於第二層平台的外緣，可以用來感測周遭環境的資訊。

### 2-1-3 第三層平台

在此層平台如圖 9 所示，主要放置的硬體設備為筆記型電腦，負責分析處理感測器的資料以做出最適合的行為。

## 2-2 核心控制電路-AVR 開發板

我們使用兩塊實驗室自行開發的 AVR 開發板如圖 10 所示，來達成輸出直流馬達的 PWM 訊號和 IR Sharp 紅外線的接收，使用此開發板的目的是在於可以節省開發的時間和電路的體積；我們只需要插上電源再把我們自行撰寫的程式透過 ISP 燒錄 Pin 寫入 ATMEGA16L 單晶片即可，在下面的小節我們將更詳細的討論 AVR 開發板如何和電腦端之間互相溝通。

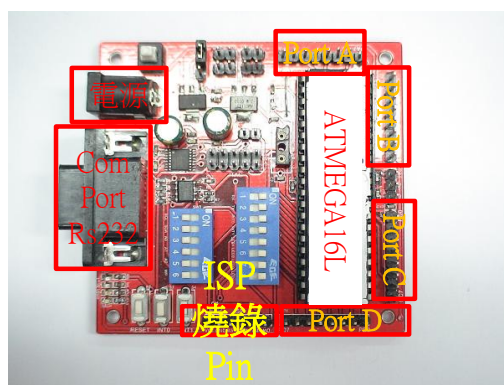


圖 10 AVR 開發板

### 2-2-1 直流馬達方向和速度訊號輸出板

直流馬達 AVR 開發板利用 RS-232 介面與電腦端控制 UI 溝通，本論文使用的直流馬達專用的 RS-232 設定如表 1：

表 1 直流馬達 AVR 開發板的傳輸規格

鮑率	起始位元	資料長度	同位元檢測	結束位元
9600 bps	1 bit	8 bits	無	1 bit

使用直流馬達開發板的主要目的在於把電腦端控制 UI 傳過來的指令轉換成馬達驅動控制板的輸入指令；為了使電腦端控制 UI 可以與直流馬達 AVR 開發板順利溝通，我們使用了自行定義的通訊協定如表 2，利用此通訊協定來傳輸馬達的方向和速度的資料給直流馬達 AVR 開發板。

表 2 直流馬達 AVR 板協定基本格式

Length	1 byte	9 bytes	1 byte
Format	Header	Data	Footer

表 3 馬達 Data 格式

Length	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte
Format	馬達 1 編號	馬達 1 方向	馬達 1 速度	馬達 2 編號	馬達 2 方向	馬達 2 速度	馬達 3 編號	馬達 3 方向	馬達 3 速度

在表 3 的 Data 格式中：馬達 1 編號、馬達 2 編號、馬達 3 編號分別為 X、Y、Z；而馬達的方向有 F(順時針轉)、B(逆時針轉)、S(不轉)三種；而馬達的速度則分為 23 段分別為 1~9、A、B、C、D、E、F、G、H、I、J、K、L、M、N。

利用 Header 和 Footer 判斷由電腦端控制 UI 傳輸過來的資料和長度是否正確；若為一筆正確的資料，則將方向轉換為 Pin 腳的 Low 或 High，再將速度轉換為 PWM 的訊號傳給馬達驅動控制板，關於輸出的 Pin 腳和 PWM 訊號的轉換我們將在 2-3 的章節做更詳細的討論。

## 2-2-2 紅外線接收板

紅外線 AVR 開發板利用 RS-232 介面與電腦端控制 UI 溝通，本論文使用的紅外線專用的 RS-232 設定如表 4：

表 4 紅外線 AVR 開發板的傳輸規格

鮑率	起始位元	資料長度	同位元檢測	結束位元
38400 bps	1 bit	8 bits	無	1 bit

使用紅外線開發板的主要目的在於將 IR Sharp GP2D12 的類比訊號電壓轉換為數位訊息(ADC)，再將此數位訊息轉換為實際偵測到的距離；然後將八顆紅外線的距離值利用 RS-232 介面傳送給電腦端控制 UI。

因為 ATMEGA16L 單晶片本身已經有提供 ADC 的功能，所以我們直接照著文件上的相關暫存器設定就可以使用，圖 11 為 ATMEGA16L 的 ADC 方塊圖，圖 11 的 ADC 0~ADC 7 就是圖 10 的 Port A(PA0~PA7)Pin 腳，也就是我們紅外線訊號線的實際接腳位置。

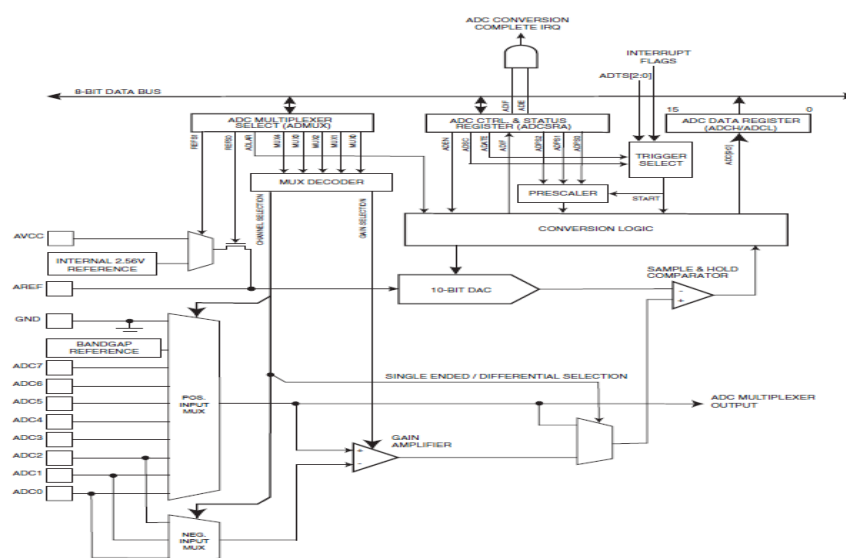


圖 11 ADC 方塊圖 [19]



## 2-3 直流馬達和直流馬達驅動控制板

一般而言，直流馬達有以下的優點：1.同樣的體積下，相較於其他的馬達(例如交流馬達)可以輸出較大功率 2.直流馬達轉速不受電源頻率限制可以製做出高速馬達 3.速度控制只要控制電壓比較簡單容易 4.馬達可輸出的最大扭力可用減速裝置來提高 5.價格便宜，這就是為什麼我們要使用直流馬達的原因。

找到我們適合的馬達之後，接下來就需要馬達驅動控制板來控制馬達的轉向和速度，我們將在以下章節分別討論直流馬達和直流馬達驅動控制板。

### 2-3-1 IG-42GM 直流馬達

本論文所使用的馬達為祥儀企業股份有限公司所生產，型號為 IG-42GM 的直流馬達如圖 12 所示，相關規格如表 5：



圖 12 IG-42GM 直流馬達 [20]

表 5 IG-42GM 直流馬達規格

最大驅動電壓	12V
最大驅動電流	5.5A
定格扭力	20 kg-cm
定格回轉數	63 rpm
減速比	1 比 104
保證正常運作時數	200 hrs

脈衝寬度調變 (Pulse Width Modulation, PWM) 是一種連續的高低電位方波，其中PWM 波型中的高電位時間寬度用來定位，而高低電位的時間寬度比例則是用來控制速度。一般轉換後脈波的週期固定，但脈波的Duty Cycle會依類比訊號的大小而改變。也就是說將供應馬達的電源在一個固定週期做ON及OFF 的控制，若ON 的時間越長，馬達的轉速越快，反之越慢。

此種 ON 與 OFF 比例控制速度的方法，ON 的期間稱為工作週期(duty cycle)，以百分比表示。許多微控制器都內建 PWM 功能，當然我們使用的 ATMEGA16L 也有提供這項功能。

PWM 常用於直流馬達的控制，是控制直流馬達轉速最常見的方法，其工作原理如圖 13 所示：

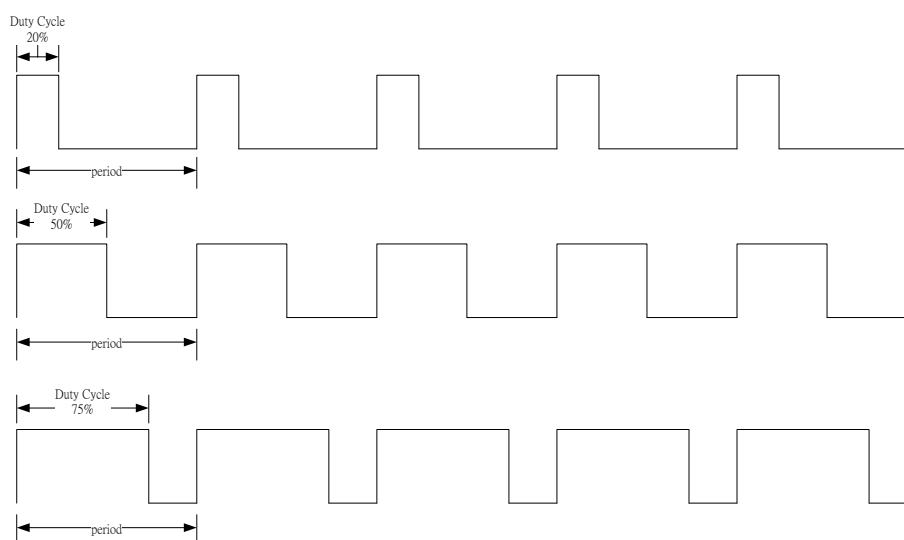


圖 13 PWM 方波圖

因為 IG-42GM 非精密馬達，依實際測量發現，各顆馬達在給予相同的訊號時的轉速有些許差異，而 PWM 訊號至少 Duty Cycle 要給 13% 以上才會讓馬達運轉，而馬達的轉速和 PWM 的 Duty Cycle 並沒有呈現線性的關係，因此在實際應用上，需要建表對應以確保輸出的轉速是符合所需，如表 6 所示：

表 6 馬達和 PWM Duty Cycle 對應表

Duty Cycle Unit : %	馬達編號-1 Unit : cm/s	馬達編號-2 Unit : cm/s	馬達編號-3 Unit : cm/s
13.7254902	3.395532	3.799762	4.092051
14.11764706	3.395532	3.1918	4.092051
14.50980392	3.507473	3.395532	4.199737
14.90196078	3.711395	3.586292	4.433056
15.29411765	3.892439	3.546444	4.495493
15.68627451	4.625797	4.559714	4.987188
16.07843137	3.940494	3.892439	4.910462
16.47058824	4.092051	3.940494	5.066349
16.8627451	4.145195	4.372329	5.232459
17.25490196	4.433056	4.433056	5.409831
17.64705882	5.319667	5.148065	5.599649
19.60784314	5.6996	6.25843	6.51387
21.56862745	6.791064	6.649583	7.092889
23.52941176	7.599524	7.422791	7.9795
23.92156863	7.092889	7.254091	8.184103
24.31372549	7.092889	7.599524	8.184103
24.70588235	7.422791	7.422791	8.399474
25.09803922	7.599524	7.599524	8.866111
25.49019608	8.399474	8.399474	9.387647
27.45098039	9.387647	8.626486	9.387647
29.41176471	9.974375	10.29613	10.63933
31.37254902	10.29613	9.974375	11.39929
33.33333333	11.39929	11.39929	11.82148
35.29411765	12.27615	12.27615	13.29917
37.25490196	13.29917	13.29917	14.50818
39.21568627	14.50818	14.50818	15.19905
41.17647059	15.19905	14.50818	15.959

49.01960784	17.73222	18.77529	18.77529
58.82352941	22.79857	22.79857	22.79857
68.62745098	26.59833	26.59833	26.59833
78.43137255	31.918	31.918	31.918
88.23529412	39.8975	39.8975	39.8975
98.03921569	45.59714	45.59714	45.59714

## 2-3-2 直流馬達驅動控制板

我們選擇 Pololu Robotics and Electronics 公司所生產的 Dual VN12SP30 Motor Driver Carrier MD03A 雙軸直流馬達驅動控制板如圖 14 所示，詳細規格如表 7，這塊區棟板的優點在於它具有高 PWM 頻率、高功率、體積小、電力反接保護的 FET。

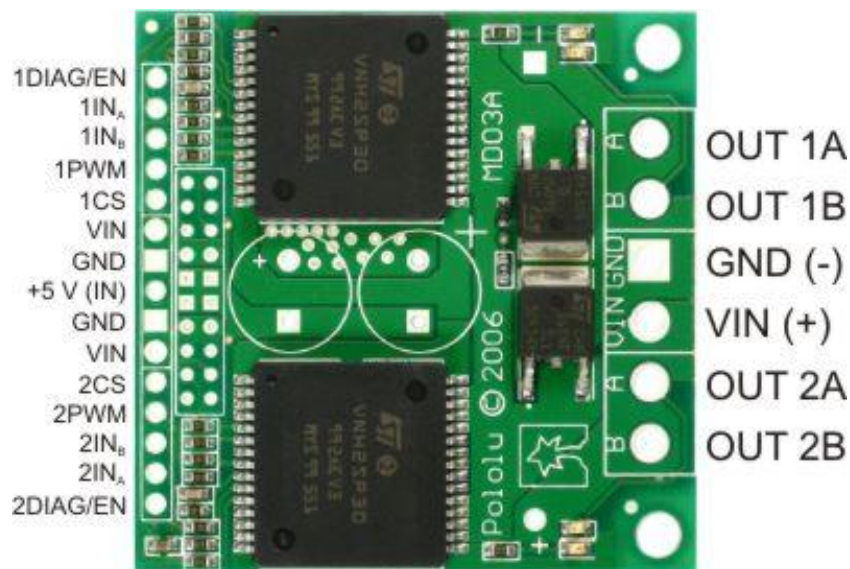


圖 14 雙軸直流馬達驅動器圖 [21]

表 7 VNH2SP30 規格

項目	VNH2SP30
操作電壓	5.5 – 16 V
最大輸出電流	30 A
MOSFET 電阻(每隻腳)	19 mΩ
最大 PWM 切換頻率	20 kHz
回饋電流偵測值	大約 0.13 V/A
過電壓斷路	最小 16V(一般 19V)
在 20 安培過熱時間	35 秒
在 15 安培過熱時間	150 秒
無限制運轉時間的電流	14 A

我們只需要在圖 14 的 1PWM Pin 或 2PWM Pin 接上由我們直流馬達 AVR 開發板所發出的 PWM 訊號，就可以控制馬達的轉速；而馬達的方向則是由 1 IN<sub>a</sub> 和 1 IN<sub>b</sub> 或 2 IN<sub>a</sub> 和 2 IN<sub>b</sub> 的 Pin 腳來控制，如表 8 所示。

表 8 直流馬達驅動板方向控制表

IN <sub>a</sub>	IN <sub>b</sub>	EN <sub>A</sub>	EN <sub>B</sub>	OUT <sub>A</sub>	OUT <sub>B</sub>	方向
1	1	1	1	H	H	煞車
1	0	1	1	H	L	順時針
0	1	1	1	L	H	逆時針
0	0	1	1	L	L	煞車

## 2-4 全向輪

全向輪是具有兩個自由度的輪子，而我們採用的是由 Kornylak Corporation 公司所生產型號為 4202KX 的全向輪，因為輪型機器人在行走的時候，勢必會跟地面造成很多摩擦力，為了讓機器人的行動更有效率，我們使用全向輪的機構如圖 15 所示。



圖 15 全向輪實體圖

## 2-5 滑鼠

滑鼠是由 SHARKOON Technologies GmbH 所生產型號為 Rush FireGlider 的雷射滑鼠如圖 16 所示



圖 16 Rush FireGlider 滑鼠

我們使用 FireGlider 滑鼠來當作我們導航機器人的定位系統，我們必須要改裝滑鼠機構，以方便我們安裝在機器人上面；如果只是單純的把滑鼠安裝在車體下，會因為地勢高低不平而造成跟地面互相摩擦產生極大的外力而影響到機器人的行走方向，所以我們在車體和滑鼠的連接部分加裝了避震器系統如圖 17 所示，可以有效的減少與地面摩擦的外力影響。

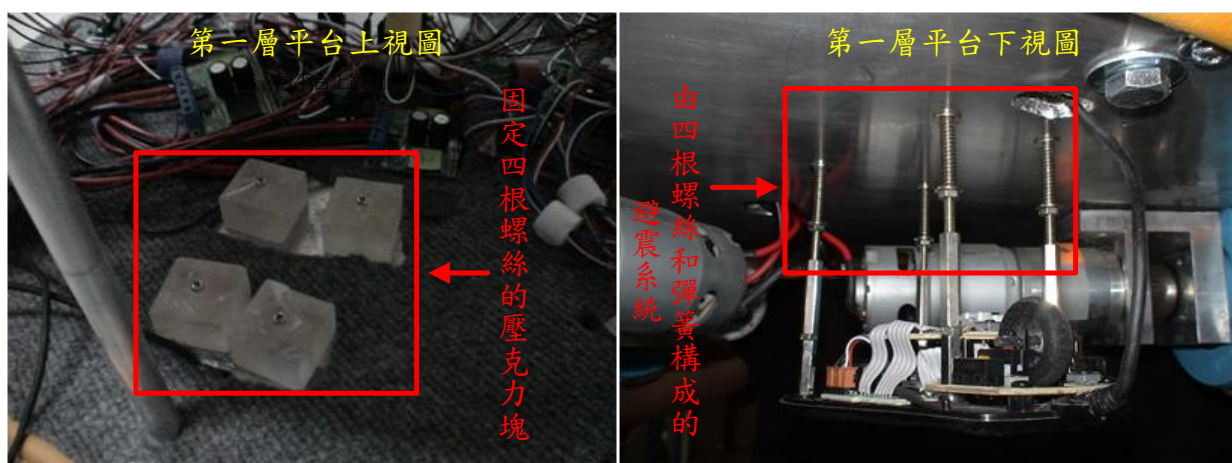


圖 17 具避震系統的 FireGlider 滑鼠

## 2-6 紅外線模組

IR Sharp GP2D12 紅外線模組是由 Sharp 公司所生產的如圖 18，此模組本身回傳的是類比的訊號，所以必須透過 ADC(Analog to Digital Converter)將類比的訊號轉換成數位的訊號，再根據其電壓和距離的特性，來判斷障礙物的距離，進而達到沿牆行走或是避障的行為。

Sharp GP2D12 本身的電壓和距離呈現非線性的關係，經由實驗測量結果如附錄 1 所示，我們得到電壓、ADC 值和距離的對應關係。



圖 18 Sharp GP2D12 紅外線

本實驗使用 ATMEGA16L 內建的 ADC 功能，其具有 10 bits 的解析度，我們使用的參考電壓為 2.56V，換算的公式如式(2.1)下：

$$\text{ADC 值} = \frac{\text{IR Sharp 輸入的類比電壓}}{2.56} \times 1024 \quad (2.1)$$

因為類比電壓輸入的值(0.43V~2.54V)是連續的，而我們實際測量的資料(10cm~80cm)是以 1 cm 為單位屬於離散的資料，為了讓我們得到的電壓可以完全得到一個對應的距離，我們使用非線性迴歸的逼近方法來求出最佳解；假設我們的總平方誤差是式(2.2)，我們使用的數學模型是  $y = f(\vec{x}_i, \vec{\theta})$  如式(2.3)，其中  $\vec{x}_i$  是輸入向量，也就是電壓值； $\vec{\theta}$  是可變的非線性函數； $y$  是輸出變數，也就是距離值；我們利用附錄 1 當作資料樣本，求出一條曲線如圖 19 所示。

$$E(\vec{\theta}) = \sum_{i=1}^m [y_i - f(\vec{x}_i, \vec{\theta})]^2 \quad (2.2)$$

$$f(\vec{x}_i, \vec{\theta}) = a_1 \times e^{\lambda_1 \times x_i} + a_2 \times e^{\lambda_2 \times x_i} \quad (2.3)$$

其中  $a_1 = 197.1$ ， $\lambda_1 = -3.24$ ， $a_2 = 38.81$ ， $\lambda_2 = -0.5305$ ， $e \approx 2.71828$ 。



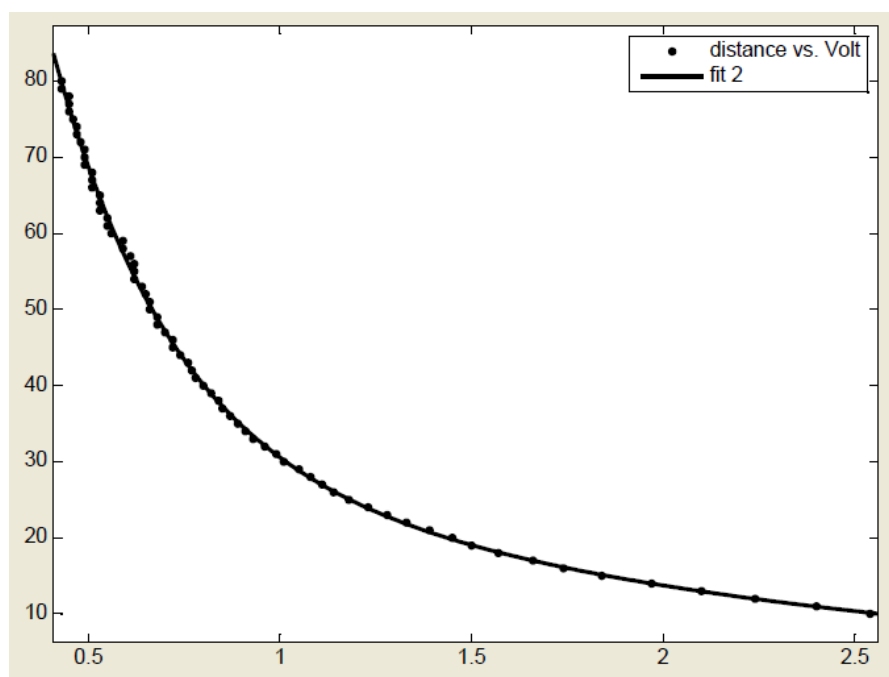


圖 19 電壓和距離的非線性迴歸曲線

## 2-7 電池與穩壓電路

選擇一個適當的電源才足以供應整個導航機器人的系統，我們分離馬達和穩壓電路的電源，才不會因為馬達所需的電流過高而影響到穩壓電路的運作，本實驗使用的電源有兩顆電池如圖 20，一顆為為 Shykuang 公司所生產的鉛酸蓄電池，其蓄電量為 7AH，可以提供 12V 的電壓，供 AVR 開發板和直流馬達驅動板使用；一顆為湯淺電池股份有限公司生產的鉛酸蓄電池，其蓄電量為 12AH，可以提供 12V 的電壓，供馬達使用。



圖 20 導航機器人使用的電源

我們使用實驗室自行開發的穩壓模組電路如圖 21，其主要使用了兩個元件，LM317 和 LM7805；LM317 是美國國家半導體公司的三端子可調整穩壓 IC，輸出的電壓範圍是 1.2V 至 37V，負載電流最大 1.5A，使用上非常簡單方便，僅需要兩個外接電阻來設置輸出電壓，此外它的線性調整率和負載調整率也比標準的固定穩壓器好，在此處我們使用來提供藍芽模組的電源；LM7805 是美國國家半導體公司生產的三端固定穩壓集成電路，用於將輸入的電壓穩壓集為 5V 後提供給有關電路，其應用相當廣泛，在此處我們使用來提供 AVR 開發板和直流馬達驅動板的電源。

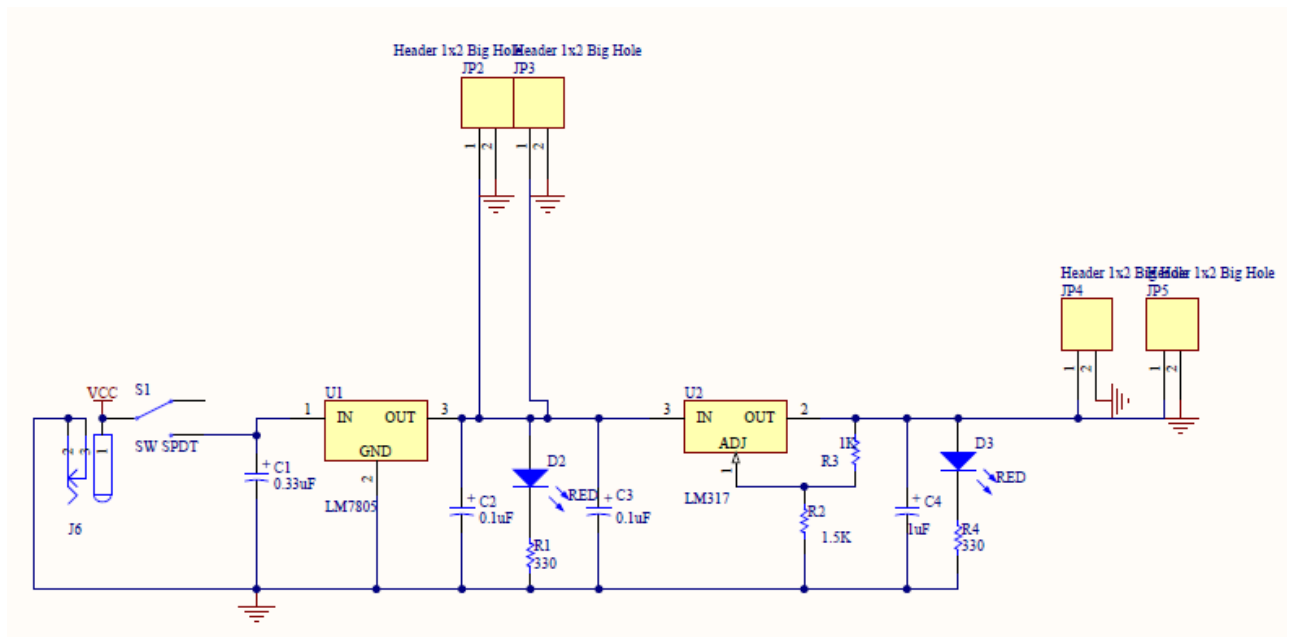


圖 21 穩壓模組電路

## 2-8 陀螺儀

陀螺儀的使用非常的廣泛，有的是拿來做步態偵測，有的是拿來做空中滑鼠的移動座標依據，如果是手機的話，也可以用來判斷使用者是否有旋轉手機，藉此旋轉螢幕角度或是供遊戲動作偵測使用；而在本論文中，我們使

用 Inven Sense 公司所生產型號為 IDG-500 的陀螺儀如圖 22 所示，主要是用來偵測導航機器人的朝向角度，我們將 IDG-500 裝在機器人身上，藉由 IDG-500 傳回的類比電壓經過 ADC 轉換後得到角速度的值，透過藍芽模組傳回電腦端控制 UI，再積分得到導航機器人旋轉的角度。

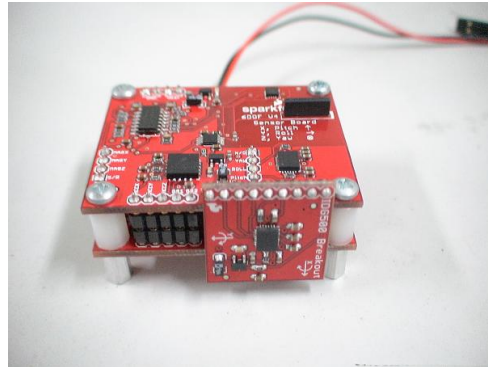


圖 22 IMU 6-DOF v4 模組中的 IDG-500 陀螺儀

## 第三章 機器人系統控制設計及原理

在本章節裡，我們分成四個部分來討論；首先，我們先介紹導航機器人的整體控制流程；再來介紹三輪運動學的原理，以此原理為基礎，才能使導航機器人在任意方向上行走；第三部分，我們將討論如何克服經由三輪運動學計算出來馬達的速度過快的問題，而我們使用的 IG-42GM 馬達並無法提供這麼快的轉速，用軟體的方式來將我們的單位時間內所需要的速度可以達到的距離，轉換成我們馬達可以提供的最大穩定速度，在此速度下要達到所指定的距離，所需的時間；第四部分，提出如何利用滑鼠和陀螺儀來做導航機器人的定位系統。

### 3-1 機器人控制流程概觀

本論文提出的導航機器人的系統控制流程如圖 23 所示；我們將控制系統分成三種模式下操作，第一種模式是沿牆行走，使得導航機器人可以在未知的空間下，探索環境；第二種模式是到達指定的目標地，並擁有即時避障的功能；第三種模式是使用軟體模擬無線遙控器來控制導航機器人的行為，而我們使用藍芽的無線技術來達到微處理器和電腦端 UI 控制介面的溝通。

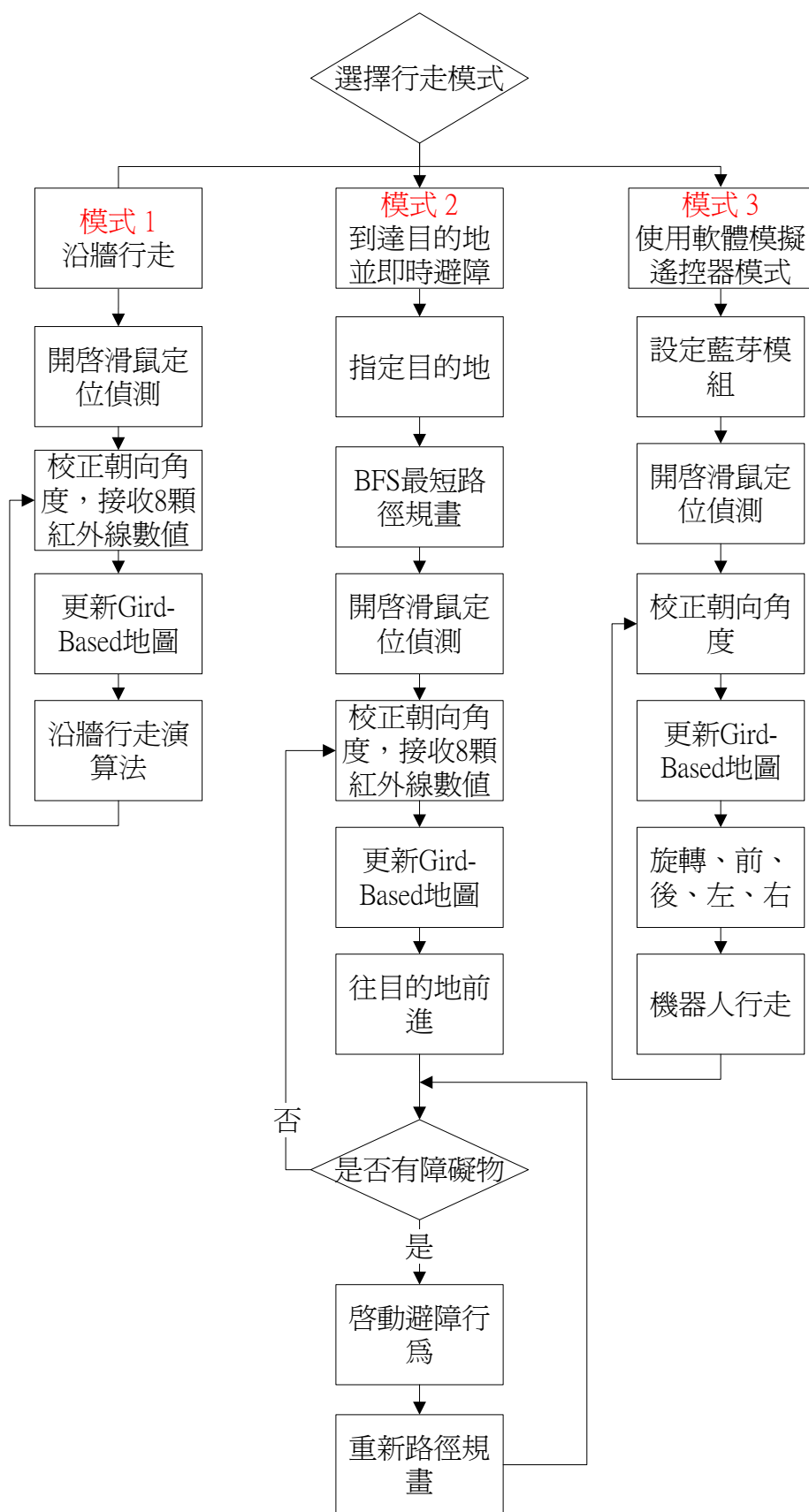


圖 23 系統控制流程圖

## 3-2 三輪運動學模型

本論文使用的導航機器人所使用的行走方式是三顆全向輪的架構如圖 24 所示，使用三輪的好處在於它可以任意方向的行走，而且不需要迴轉半徑，相較於二輪所需要的迴轉半徑，而四輪需要更多的硬體成本；而使用全向輪是為了減少與地面的摩擦力；所以我們選擇使用三顆全向輪的方式來實現我們的導航機器人的運動。

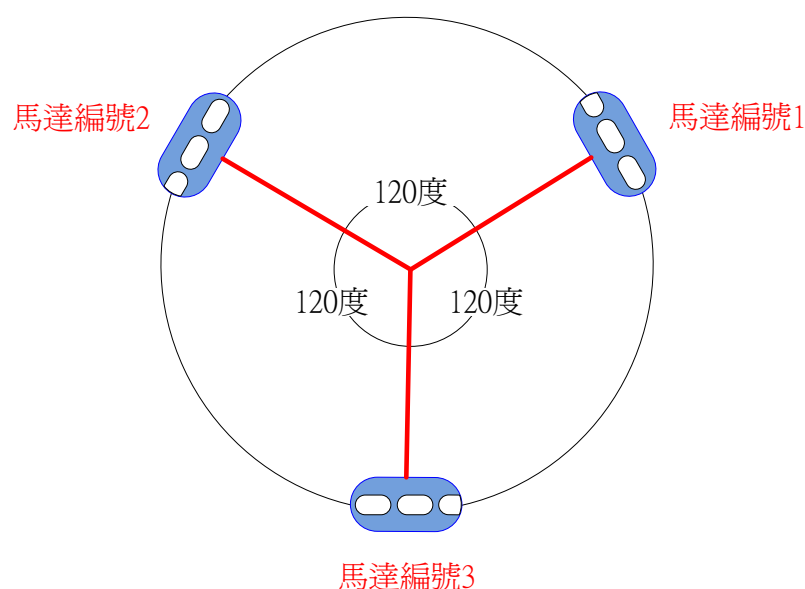


圖 24 三輪式的導航機器人平台

當機器人要往任意方向前進時，首先我們必須要知道三顆全向輪的轉速會造成機器人往什麼樣的方向前進；所以假設我們要移動一個方向，我們會給定一個距離和角度，藉由這個距離和角度來計算出三顆全向輪所應該具備的轉速，下方表 9 為三輪運動學示意圖的符號表以及三輪運動學示意圖如圖 25 和推導方程式：

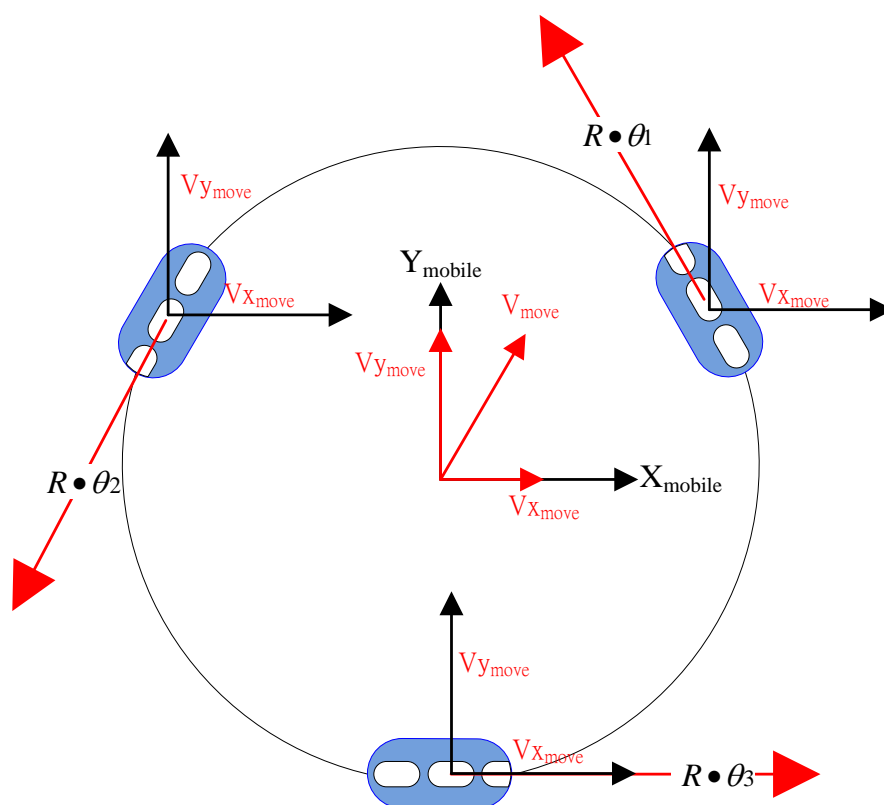


圖 25 三輪運動學示意圖

表 9 三輪運動學符號表

$Y_{mobile}$	機器人自身的 Y 座標
$X_{mobile}$	機器人自身的 X 座標
$V_{move}$	機器人想要移動的方向和速度
$V_{y_{move}}$	$V_{move}$ 在 Y 座標上的分量
$V_{x_{move}}$	$V_{move}$ 在 X 座標上的分量
$R$	全向輪的半徑
$\theta_1$	馬達編號 1 的角速度
$\theta_2$	馬達編號 2 的角速度
$\theta_3$	馬達編號 3 的角速度
$\phi$	30 度

圖 25 中的  $Y_{\text{mobile}}$  和  $X_{\text{mobile}}$  是以馬達編號三為 270 度所定義的導航機器人自身的 X 座標和 Y 座標， $V_{\text{move}}$  是我們指定導航機器人要移動的方向，所以我們要計算的是三顆馬達在 X 座標和 Y 座標上的合力要分別等於  $V_{X_{\text{move}}}$  和  $V_{Y_{\text{move}}}$  的合力，如圖 26 所示，推導公式如下：

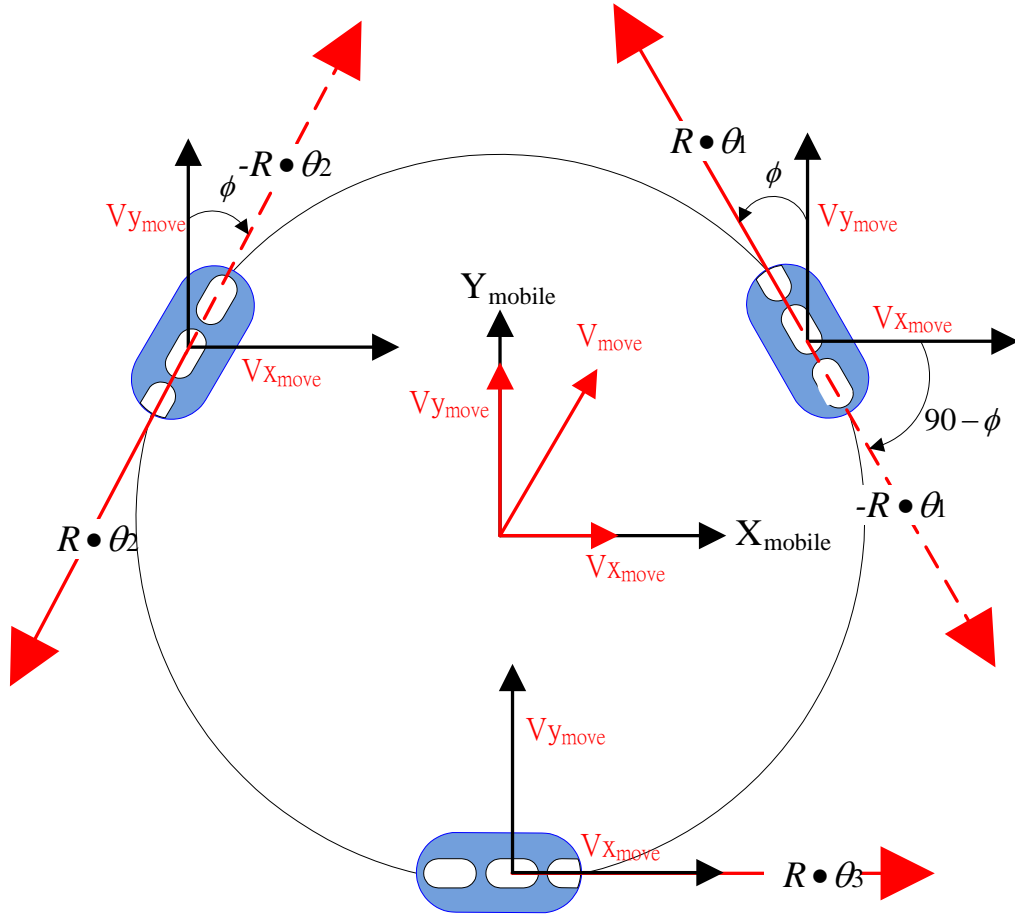


圖 26 各個馬達合力的計算

$$R \cdot \theta_1 = -\cos(90 - \phi) \times V_{X_{\text{move}}} + \cos(\phi) \times V_{Y_{\text{move}}} \quad (3.1)$$

$$= -\sin(\phi) \times V_{X_{\text{move}}} + \cos(\phi) \times V_{Y_{\text{move}}} \quad (3.2)$$

$$R \cdot \theta_2 = -\cos(90 - \phi) \times V_{X_{\text{move}}} - \cos(\phi) \times V_{Y_{\text{move}}} \quad (3.3)$$

$$= -\sin(\phi) \times V_{X_{\text{move}}} - \cos(\phi) \times V_{Y_{\text{move}}} \quad (3.4)$$

$$R \cdot \theta_3 = V_{X_{\text{move}}} \quad (3.5)$$



我們將以上的公式，用矩陣來表示：

$$\begin{bmatrix} R \cdot \theta_1 \\ R \cdot \theta_2 \\ R \cdot \theta_3 \end{bmatrix} = \begin{bmatrix} -\sin(\phi) & \cos(\phi) \\ -\sin(\phi) & -\cos(\phi) \\ 1 & 0 \end{bmatrix} \begin{bmatrix} V_{x_{move}} \\ V_{y_{move}} \end{bmatrix} \quad (3.6)$$

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} -\sin(\phi) & \cos(\phi) \\ -\sin(\phi) & -\cos(\phi) \\ 1 & 0 \end{bmatrix} \begin{bmatrix} V_{x_{move}} \\ V_{y_{move}} \end{bmatrix} \quad (3.7)$$

其中符號 $R$ 、 $\theta_i$ 、 $\phi$ 、 $V_{x_{move}}$ 和 $V_{y_{move}}$ 如同表 9 所示。

由式(3.7)我們可以得知，只要給定系統 $V_{x_{move}}$ 和 $V_{y_{move}}$ ，我們就可以求知相對應的三顆馬達的角速度，當我們給馬達各自的速度之後，導航機器人就會往  $V_{move}$  方向行走。

### 3-3 全方位走向的實現

由三輪運動學的公式可以得知，我們只要給定  $V_{move}$ ，則機器人就可以自身座標為基準朝任一個方向行走，但是在實際的應用中卻是有限制的，若我們給定的  $V_{move}$  的距離夠遠的話，則計算出來的馬達的轉速會非常高，但是我們的馬達並不支援這麼高的轉速；由於 IG-42GM 馬達本身的最高轉速為 63 rpm，相當於 33.49752 cm/s，這是官方給的規格，比我們實際測量的稍慢如表 6，但卻是一個可以維持穩定的高轉速的狀態，所以在我們自己定義的 23 段速度中，最高的速度 Duty Cycle 為 78.4% 的 PWM 訊號，也就是 31.918 cm/s 的速度，是屬於接近最高速度的穩定狀態；因為全向輪的外徑為 10.16 cm，而 31.918 cm/s 相當於馬達的轉速為 6.283 徑度/s 如式(3.8)~(3.10)。

$$R \cdot \theta_i = V_i, \quad i = 1,2,3 \quad (3.8)$$

$$\theta_i = \frac{V_i}{R}, \quad i = 1,2,3 \quad (3.9)$$

將  $R = 5.08$  ,  $V_i = 31.918$  代入 式(3.9)中，我們可以得到角速度：

$$\theta_i = \frac{31.918}{5.08} = 6.283, \quad i = 1,2,3 \quad (3.10)$$

其中符號 $R$ 和 $\theta_i$ 如同表 9 所示， $V_i$ 代表在各顆馬達方向的速度。

也就是說 IG-42GM 馬達，在我們定義的最高轉速下，能支援的最高角速度為 6.283 徑度/s，所以為了克服馬達角速度不支援的情況下，我們提出了利用時間換取速度的演算法，也就是假設原本若需要(6.283\*2) 徑度/s 的角速度，我們將用 (6.283 徑度/s) \* 2(s)的時間去行走。首先我們定義八個區域如圖 27 所示，演算法流程如圖 28~圖 30 所示；經由式子(3.8)我們可以知道，當我們把最大馬達角速度 $\theta_{max} = 6.283$ 和全向輪半徑 $R = 5.08$ 代入，可以求得最大的 $V_{move}$ 的距離，我們定義 $V_{move}$ 的最長距離為 31.91764 cm 標示為  $V_{max_{move}}$ ，以確保三顆馬達計算出來的角速度都不會超過 6.283 徑度/s。

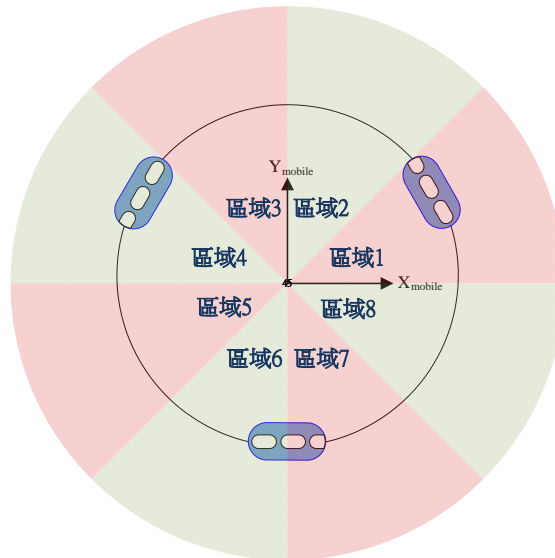


圖 27 全方位走向的八個區域

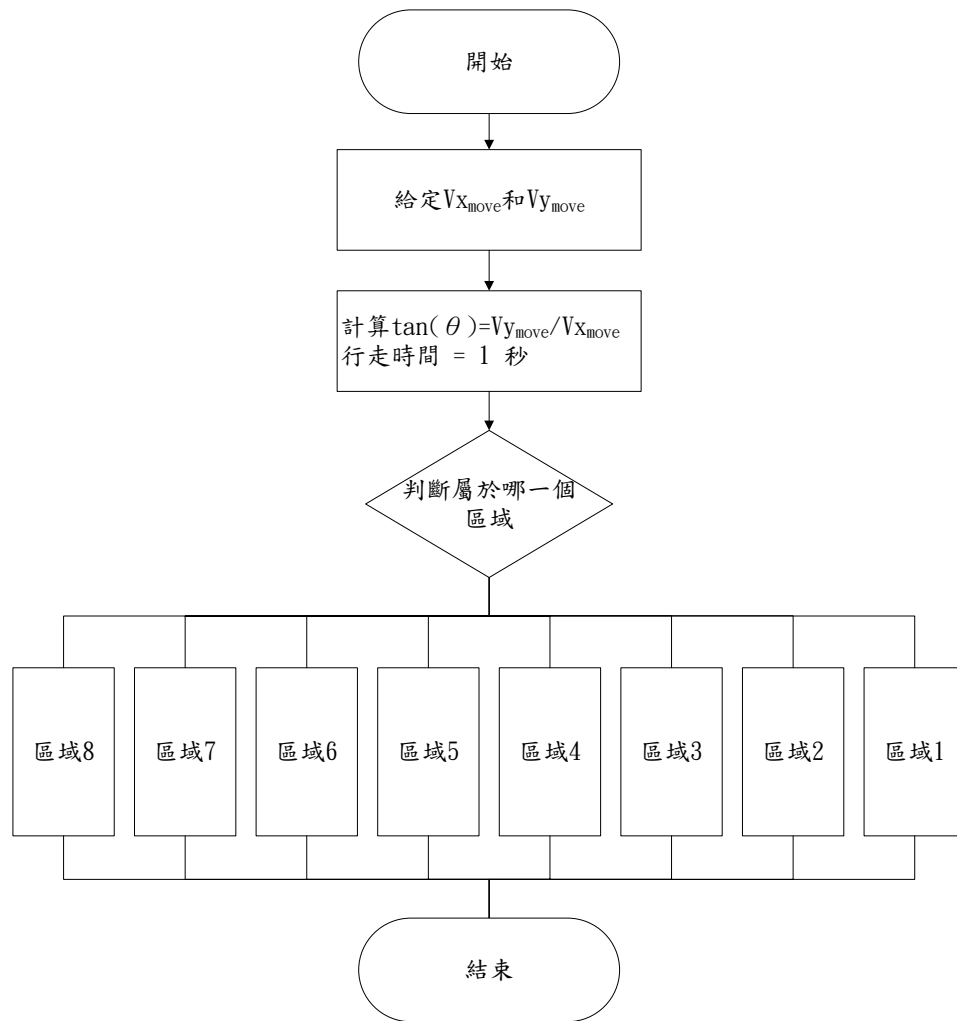


圖 28 全方位走向的實現流程圖

為了方便表示我們在各個區域的動作，我們定義函式：

$\text{Movement}(X\text{位移}, Y\text{位移}, T\text{時間})$

代表我們要機器人在自身的 X 座標上位移  $X$  的距離，在自身的 Y 座標位移  $Y$  的距離，之後在延遲了  $T$  秒鐘後，發出煞車的訊號，所以  $T$  就是機器人行走的時間。

因為區域 1、區域 4、區域 5、區域 8 的演算法流程是一樣的，差別在於  $X$  位移和  $Y$  位移的正負號不同而已；同理，區域 2、區域 3、區域 6、區域 7 也是一樣，差別在正負號；所以我們只需要列舉出區域 1 和區域 2 的演算法即可如圖 29、圖 30 所示。

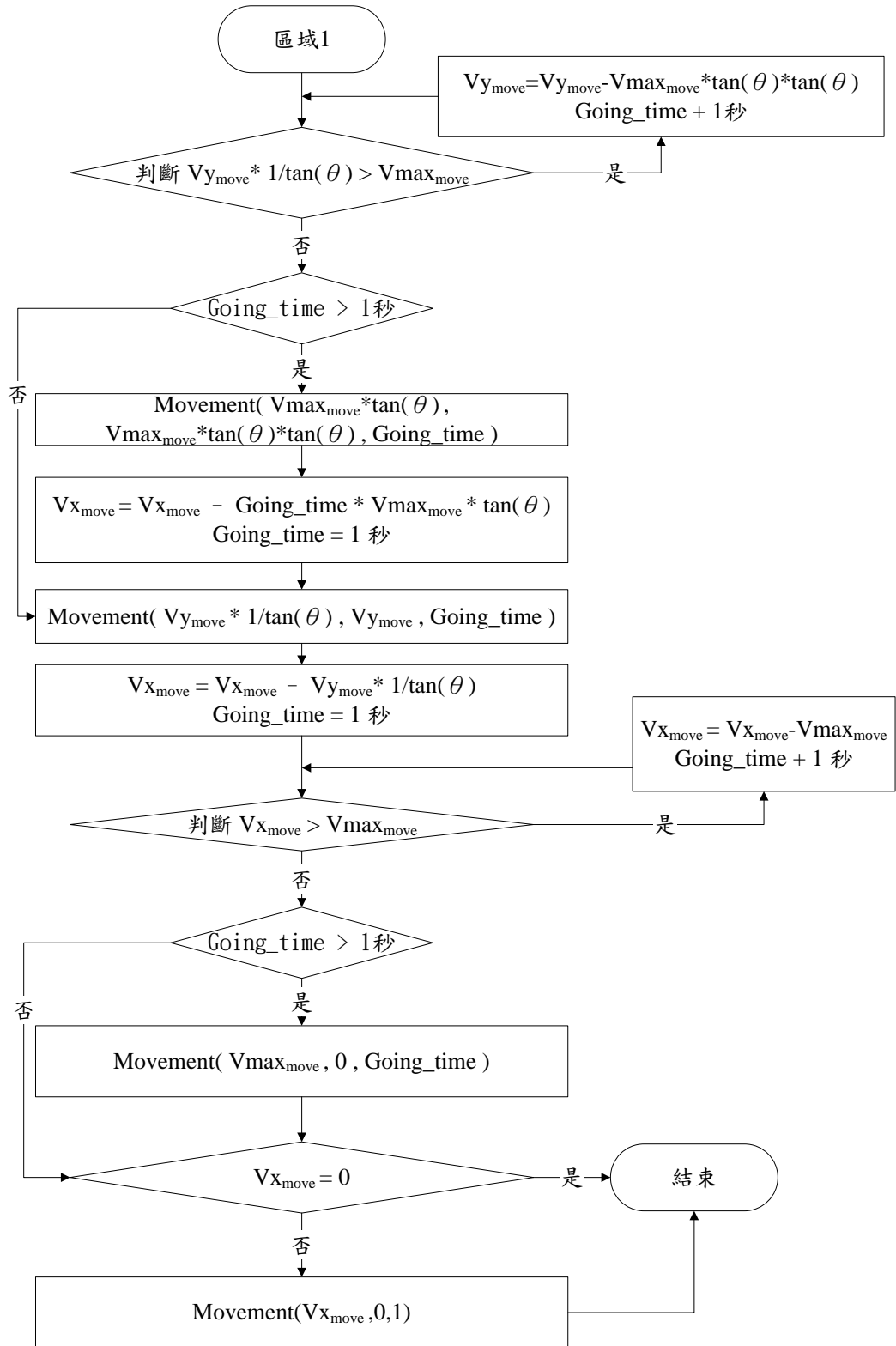


圖 29 區域1 演算法流程

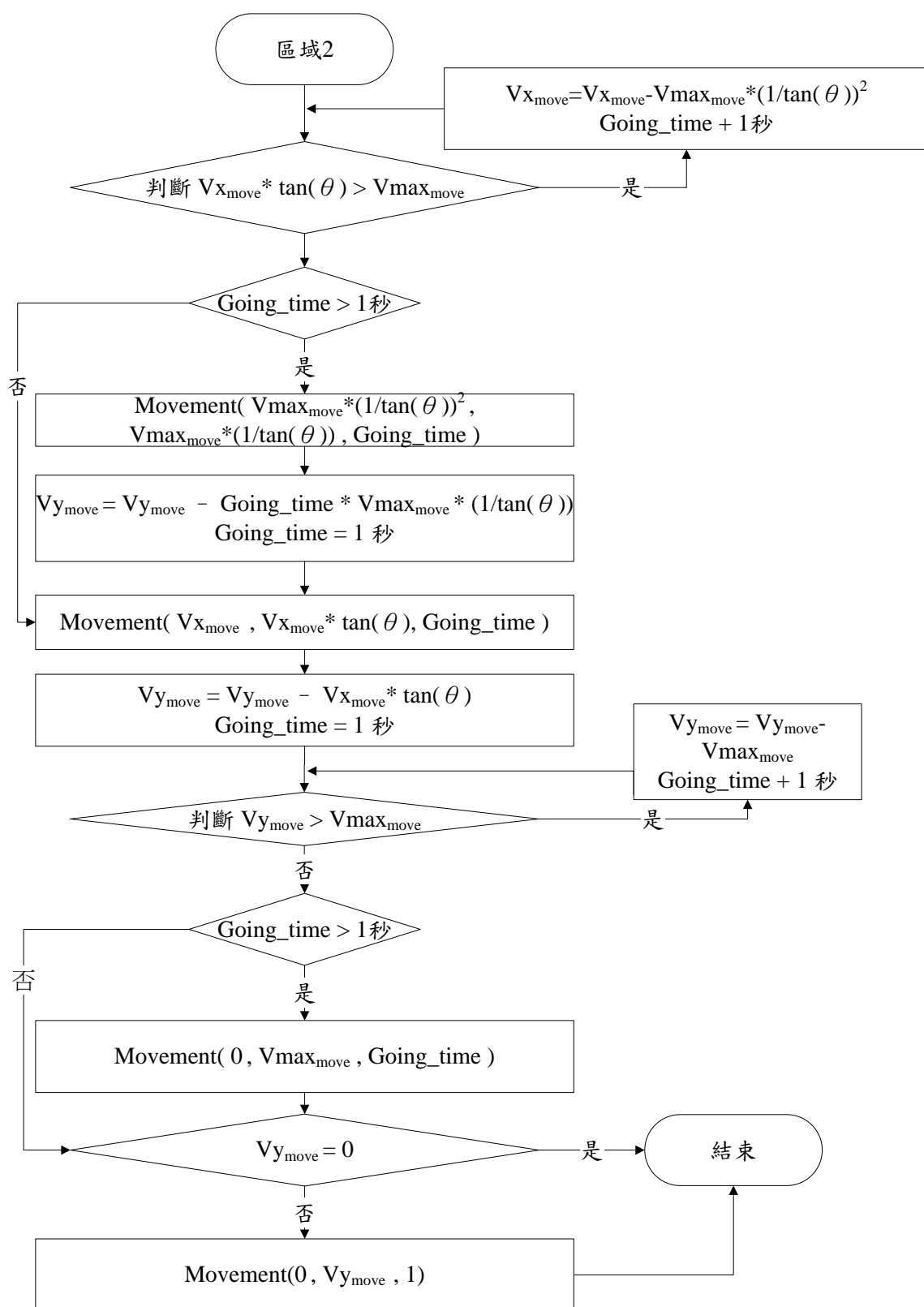


圖 30 區域2演算法流程

### 3-4 機器人定位系統

本論文提出的導航機器人的定位系統是使用滑鼠和陀螺儀透過 USB 和藍芽傳輸模組連結到筆記型電腦接收感測器訊號值來實現的，和一般傳統所使用的定位方法不同；傳統上，一般都是使用馬達的編碼器或是光學編碼器來測量目前馬達的轉速和位置，藉由馬達的資訊再加上機器人的運動學模型，如此就可以利用航位推算法(Dead Reckoning，簡稱 DR)來推算出目前所在的位置 [22]，但是 DR 的缺點在於，因為這一刻的位置是由上一刻位置所推導出來的，在推算的過程中，誤差是會不斷的被累積起來的，因此推算出來的誤差會隨著時間而增長，而造成此誤差的最主要原因在於因為地面的材質或是障礙物造成輪子在地面空轉或滑動，實際上機器人的位置不變，但是馬達編碼器的資訊卻已經被拿來推算下一刻的位置了。

使用滑鼠和陀螺儀來定位雖然也會因為感測器偵測值的準確性而產生誤差，但是最主要的好處在於以下幾點：

1. 滑鼠所偵測到的移動距離，所得到的值並不會因為滑動或空轉而有所改變，也就是偵測到的是實際上相對位置的值。
2. 陀螺儀偵測到的角度變化，也是實際上相對角度的變化，也就是說並不會受到外力影響，例如：電子羅盤的值常常會受到金屬或是磁場的變化而有很大的誤差。
3. 若感測器或是微機電系統的技術更成熟後，準確性會相對提升，在未來非常具有前瞻性。
4. 因為定位系統本身所使用的資訊，皆為模組化元件(滑鼠、陀螺儀)所提供的，因此在移植到不同平台上非常具有彈性。

雷射滑鼠的偵測原理是先利用雷射二極體的光源，照射在物體表面上後，產生干涉條紋形成的光斑點，然後反射到成像感測器上如圖 31 所示，藉由前

後兩張影像的像素比對計算出  $x$  和  $y$  軸的位移量，因此物體表面的材質會影響到成像感測器的成像，進而影響到  $x$  和  $y$  軸的位移量，也就是說，在不同的物體表面上機器人移動相同的距離，滑鼠所偵測到的  $x$  和  $y$  的位移量會不同；因此我們在使用滑鼠定位系統之前，我們會先做單位像素和實際距離比值的初始化動作如圖 32 所示，也就是我們在此物體表面上偵測到的一個單位的像素所對應到實際上的移動距離的數值。其作法是我們先經由實驗得知讓機器人行走實際距離 90 公分的指令，接著任意設定一個正實數  $\text{correspond}$  變數，接下來開啟滑鼠偵測，然後對機器人行行走實際距離 90 公分的指令，再計算滑鼠偵測到的像素變化量與 90 之間的比值，接下來再把這個比值乘上原本的  $\text{correspond}$  變數就是新的  $\text{correspond}$  變數。而經由實驗發現，我們在以紙為材質所鋪的平坦地面上定位出來的誤差會比較小，其定位效果較穩定。

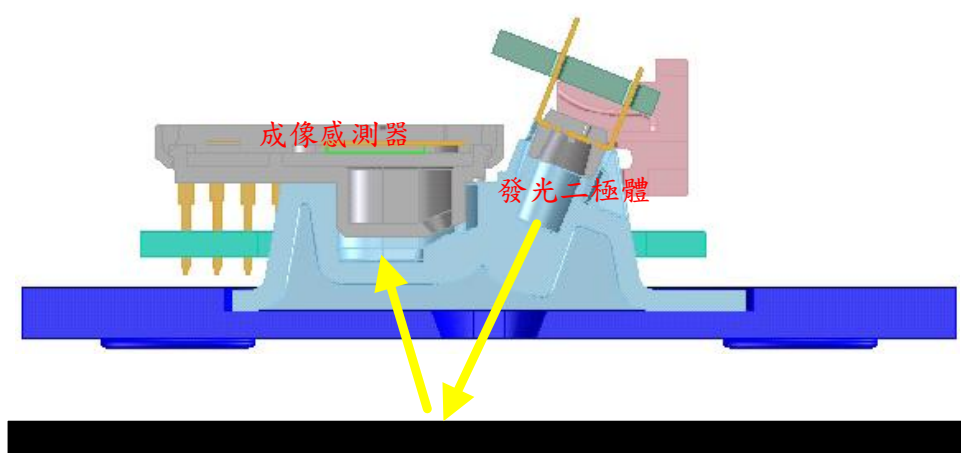


圖 31 雷射滑鼠的成像原理 [23]

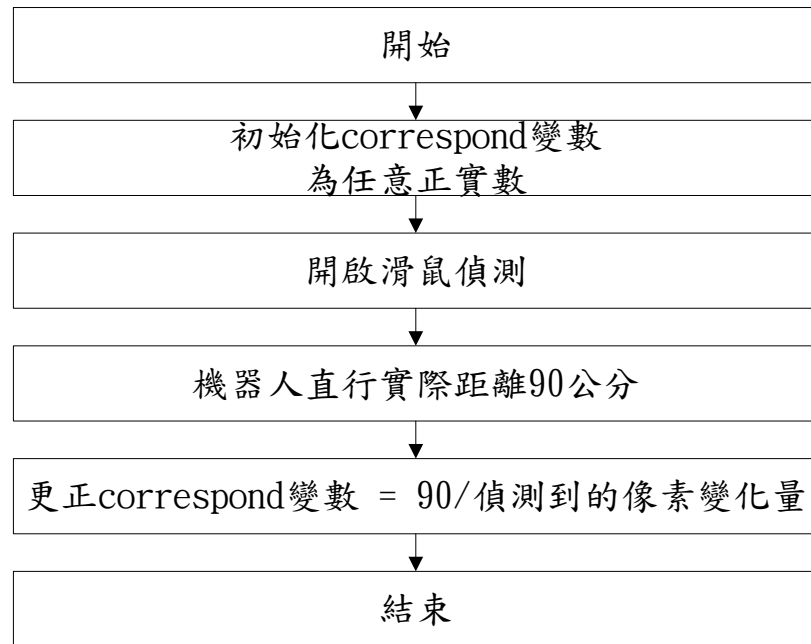


圖 32 定位系統初始化流程

經過初始化之後，就可以開始用滑鼠定位系統來幫機器人做位置推算，我們必須先把 Windows 作業系統的桌面解析度調成 1024x768，另外我們定義 previous\_x 和 previous\_y 代表上次偵測到的 X 座標和 Y 座標，而 x 和 y 代表目前偵測到的 X 座標和 Y 座標。首先我們透過偵測滑鼠目前的 XY 座標來與上一次的 XY 座標相減計算出 X 像素和 Y 像素的位移，再根據陀螺儀目前計算出來的朝向角度當作我們的二維座標轉換公式的旋轉角作計算，將 XY 像素的位移量乘上 correspond 變數就是我們實際 XY 移動的距離。整個定位系統的流程如圖 33 所示。



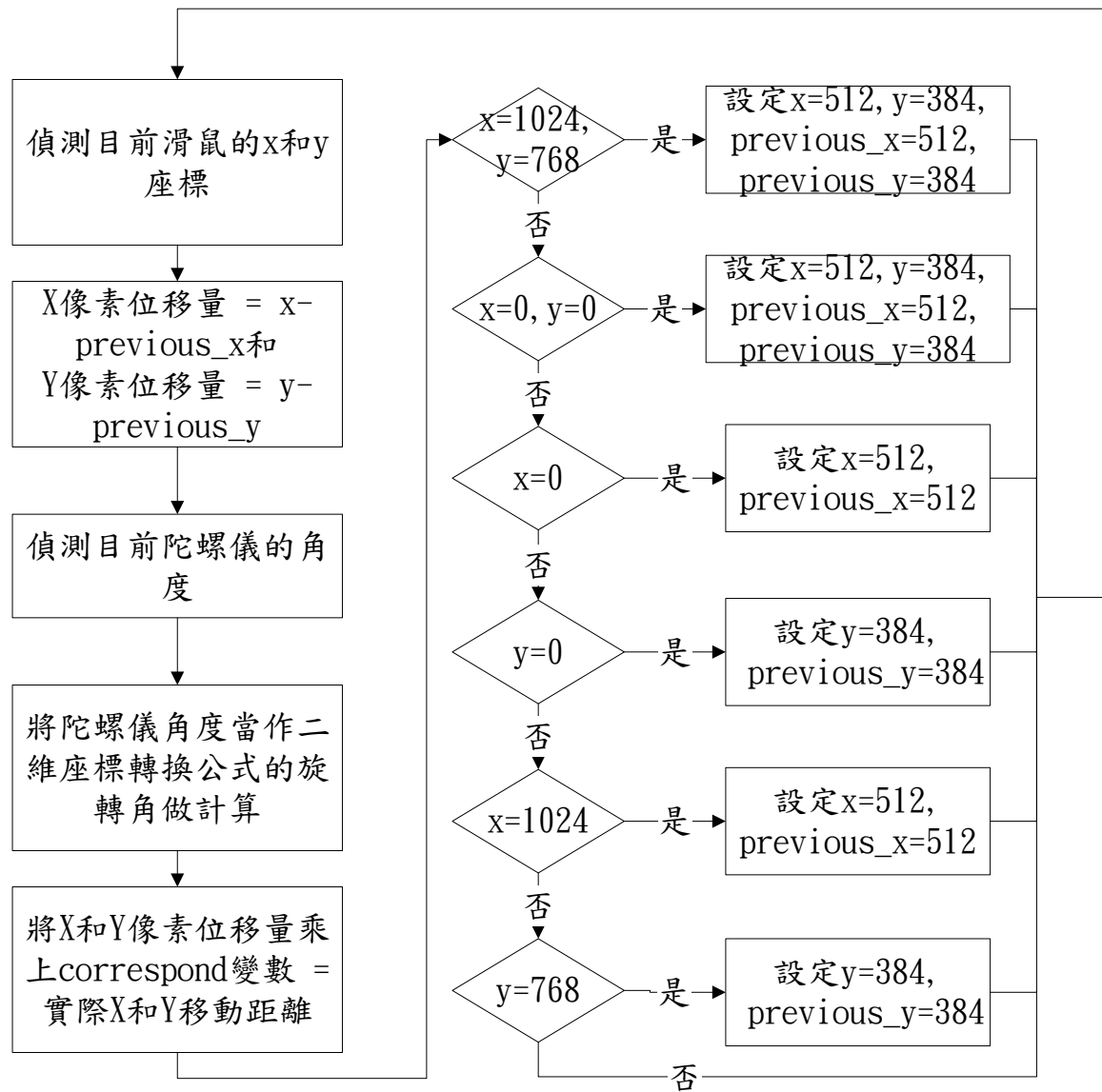


圖 33 定位系統流程圖

## 第四章行走模式

在本章節我們將介紹導航機器人的四種行走模式：即時避障系統、沿牆行走、BFS 最短導航路徑規劃、軟體模擬無線遙控器；這四種模式大致上可以利用已知環境和未知環境來分類，即時避障和沿牆行走屬於在未知環境下的模式，透過紅外線感測器來接收現在所處環境的資訊如圖 34 所示，各顆紅外線的方位角度如表 10 所示；而 BFS 最短路徑規劃和軟體模擬無線遙控器是屬於在已知環境下的操作，但是比較不一樣的是 BFS 最短路徑規劃裡面也包含即時避障系統，可以即時反應出環境的變動而重新路徑規劃，在下面的章節我們將會更詳細的介紹。

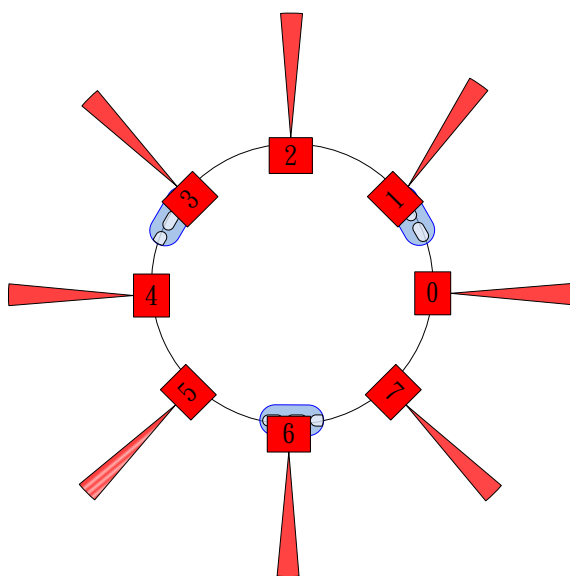


圖 34 機器人的紅外線感測器配置方位圖

表 10 紅外線的方位角度

紅外線編號	0	1	2	3	4	5	6	7
角度	0	45	90	135	180	225	270	315

## 4-1 機器人即時避障系統

避開障礙物(或是稱為 Local Path Planning)是智慧型機器人最基本的功能，也是最重要的功能之一。而避障的研究一直以來都是機器人領域中一個非常重要的議題，很多人用不同的方法(例如：模糊系統、增強式學習法 Q-learning、虛擬力、邊緣偵測)提出了相異的避障方式。由於邊緣偵測的效果有限，而 Q-learning 又需要建立大量的表格，非常消耗記憶體 [24]，模糊系統在訂定模糊規則與模糊集合必須確保使用者的邏輯正確性才能定出一套合理的系統。因此，在這邊我們使用 J. Borenstein 和 Y. Koren 在 1991 年提出的改良式的虛擬力：Vector Field Histogram(VFH) [12]，其演算法不需太複雜的計算，我們只須將紅外線感測器的值讀入就可以計算障礙物的可能方向(障礙物密度)，對於避障功能來說非常適用。我們將在下面做一個詳細的介紹。

我們可以將 VFH 的流程方法分成三個階段。在第一和第二階段中，主要是將作資料處理；在第三階段主要是輸出車子的轉向和速度，但是由於我們只使用 VFH 資料處理部分，在此我們只分別介紹兩階段的工作如下。

### 第一階段

首先地圖模型的建立方式是使用二維 Cartesian 座標的 histogram grid C 如圖 35 所示，本論文使用的 histogram grid 的大小是 100X100 個 grid，而每個 grid 的大小為 10cmX10cm。圖中所標示的 Active Window 是由自訂的 Window Size \* Window Size 所圍起來的正方型區域，其代表感測器所能偵測的有效範圍，也就是說在 Active Window 外的距離不會受到影響。而在本論文中使用的 Window Size 是 80cm。而 Active Window 的中心也是機器人的幾何位置中心，由於機器人一直處於行走的狀態，所以 Active Window 在地圖中也不斷的在變換位置。標示為 Certainty Value 的數字，則代表障礙物存在的可能性，Certainty Values 的值越高代表障礙物存在的可能性越大。裝載在導航機器人平台上的

八顆紅外線即時偵測障礙物的距離資訊，並且將相對應偵測距離的方格 (Active Cell) 其 certainty values 累加(一次加一)，而介於障礙物和機器人中間的距離，依照紅外線的發射角為範圍所圍起來的區域，在區域中的方格 (Active Cell) 遞減其 certainty values (一次減一)。以此方式將我們所接收到的紅外線值對應到整張地圖，第一階段的資料處理就已經完成。

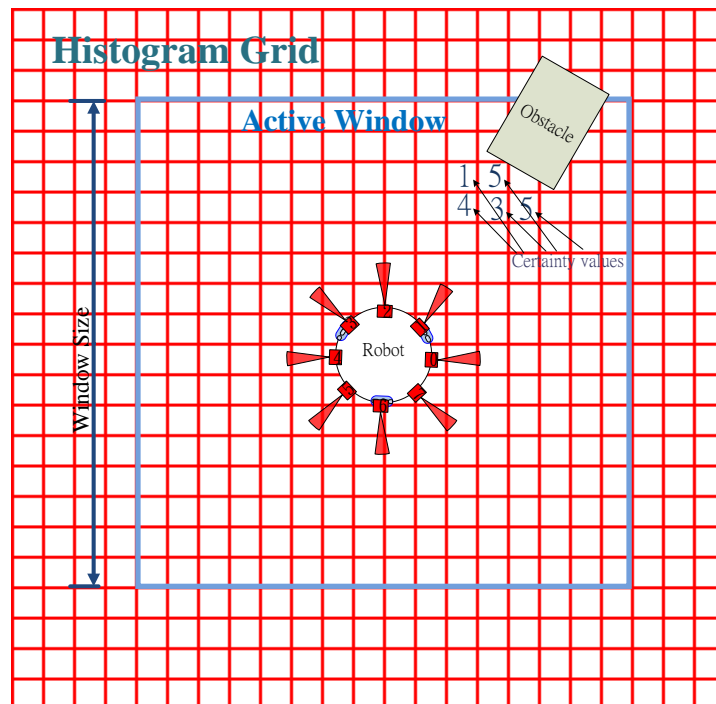


圖 35 VFH 概念圖

## 第二階段

在此階段的主要工作是將我們在第一階段中所建立的二維地圖中的 Active Window 轉換成一維的 polar histogram  $H$  的極座標長方條圖資訊。首先我們必須根據目前機器人所在的位置，以機器人為中心所建立的 Active Window，將 Active Window 以每隔  $\alpha$  度為一個區塊 (Sector) 分成  $360/\alpha$  塊 (Sector) 如圖 36 所示；接下來計算每個區塊 (Sector) 中的方格 (Active Cell) 其  $m_{ij}$  值，其中  $i, j$  代表方格在 histogram grid 中的 X 和 Y 座標，再將區塊中的每個方格的

$m_{ij}$  值全部加總就是這個區塊的障礙物密度(Polar Obstacle Density,POD) $h_k$  值，其中  $k$  代表區塊(Sector)的編號；之後再將所有區塊的  $h_k$  值作平滑化得到每個區塊的平滑化障礙物密度(Smoothed Polar Obstacle Density,SPOD) $h'_k$  值，詳細的計算過程如式(4.1)~(4.5)。

有了每個區塊的 SPOD 值之後，我們就可以將 Active Window 轉換成 polar histogram  $H$  如圖 37、圖 38 所示。接下來利用自定的閾值(Threshold)，篩選出可以走的方向(Safe Direction Area)，然後讓機器人往最適合的方向前進。若我們將閾值設的過小則容易導致車子停滯不前，造成沒有可走的角度；反之，若我們將閾值設的過大則容易誤判有障礙物的方向是可走的角度。所以我們經由實驗結果選定閾值為 5 是最適合目前所處的環境中的門檻值，並且讓機器人以低於閾值下可走的角度區間中，具有最大連續可走區間的區塊方向前進。

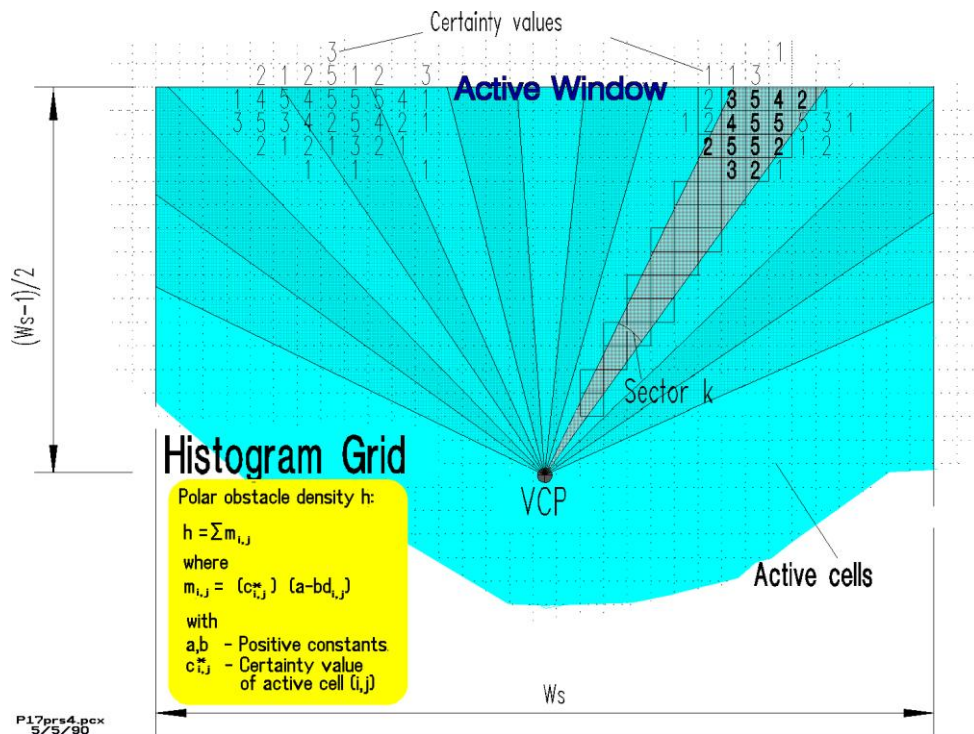


圖 36 Active Window 轉換成 Polar Histogram 概念圖 [12]

障礙物密度的計算式子：

$$\beta_{i,j} = \tan^{-1} \left( \frac{y_j - y_0}{x_i - x_0} \right) \quad (4.1)$$

$$m_{i,j} = (c_{i,j}^*)^2 \times (a - b \times d_{i,j}) \quad (4.2)$$

$$k = \text{INT} \left( \frac{\beta_{i,j}}{\alpha} \right) \quad (4.3)$$

$$h_k = \sum_{i,j} m_{i,j} \quad (4.4)$$

$$h'_k = \left( \frac{h_{k-l} + 2 \times h_{k-l+1} + \dots + (l+1) \times h_k + \dots + 2 \times h_{k+l-1} + h_{k+l}}{2 \times l + 1} \right) \quad (4.5)$$

其中符號說明如表 11：

表 11 VFH 符號說明

符號	說明
a,b	為正的常數
$c_{i,j}^*$	Active Cell(i,j)的 Certainty Values
$d_{i,j}$	機器人的幾何中心與 Active Cell(i,j)的距離
$m_{i,j}$	在 Active Cell(i,j)位置的障礙物向量(Obstacle Vector)大小
$x_0, y_0$	機器人的幾何中心位置
$x_i, y_j$	Active Cell 的座標位置
$\beta_{i,j}$	機器人幾何中心指向 Active Cell 的角度
k	區塊(Sector)的編號
$\alpha$	區塊(Sector)的角度
INT(·)	轉換成整數的函數
$l$	平滑化的參數
$h_k$	障礙物密度(Polar Obstacle Density)值
$h'_k$	平滑化後的障礙物密度(Polar Obstacle Density)值

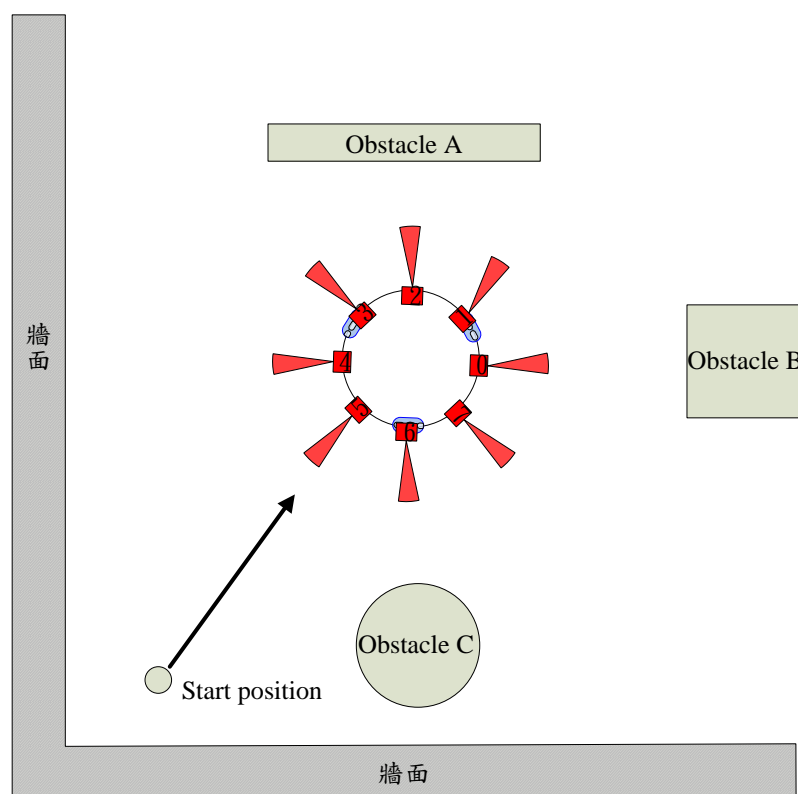


圖 37 實際地形模擬圖

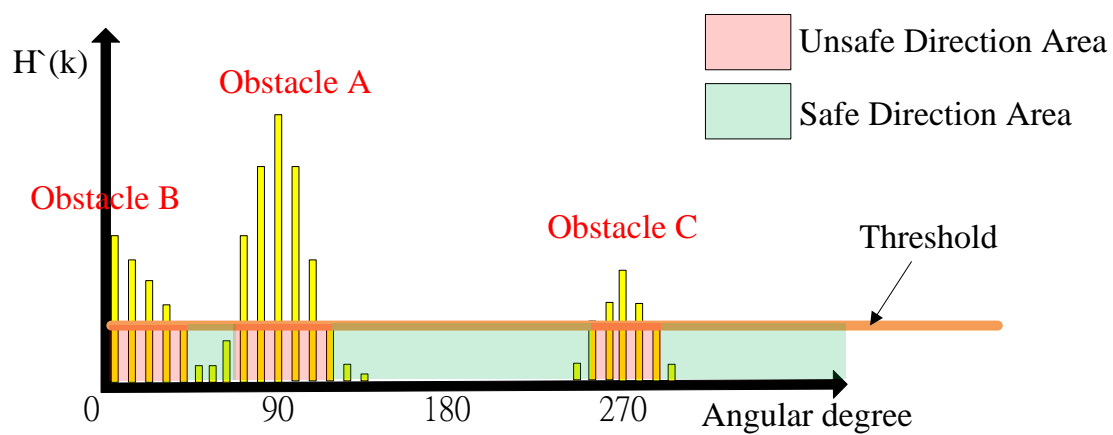


圖 38 實際地形模擬圖相對應的 Polar Histogram  $H$  圖

## 4-2 沿牆行走

沿牆行走的主要目的在於未知的環境下進行探索的工作，其原理就類似避障系統，可以想像牆就是一塊很大的障礙物，只是我們是與這個障礙物保持安全距離，然後沿著它的邊緣行走。有一點要特別說明，就是我們在實際沿牆行走的時候並沒有使用即時避障系統的功能。

沿牆行走大致上可以看成四種行為不斷的切換：自動搜尋牆面、沿著牆面走、凹角牆面行走、凸角牆面行走，如圖 39~圖 41 所示。

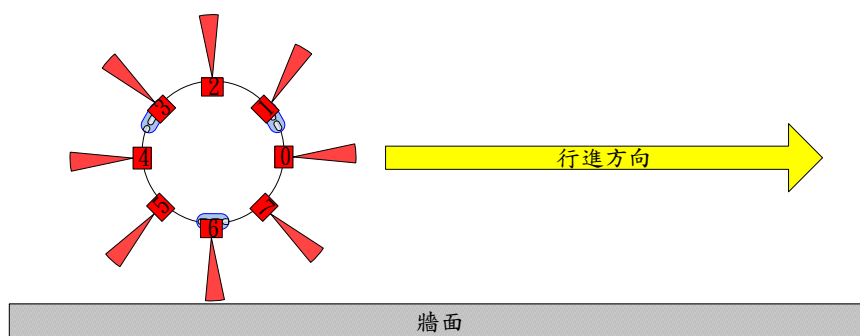


圖 39 沿著牆面走

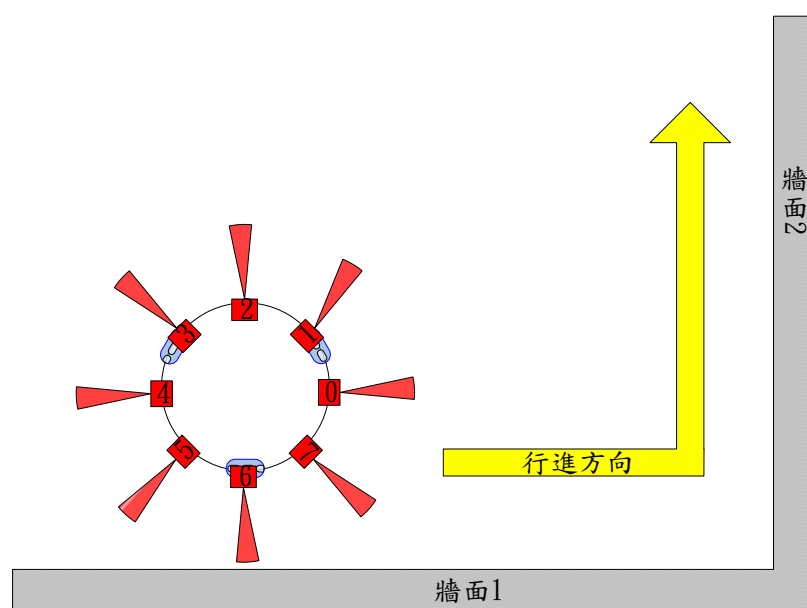


圖 40 凹角牆面行走



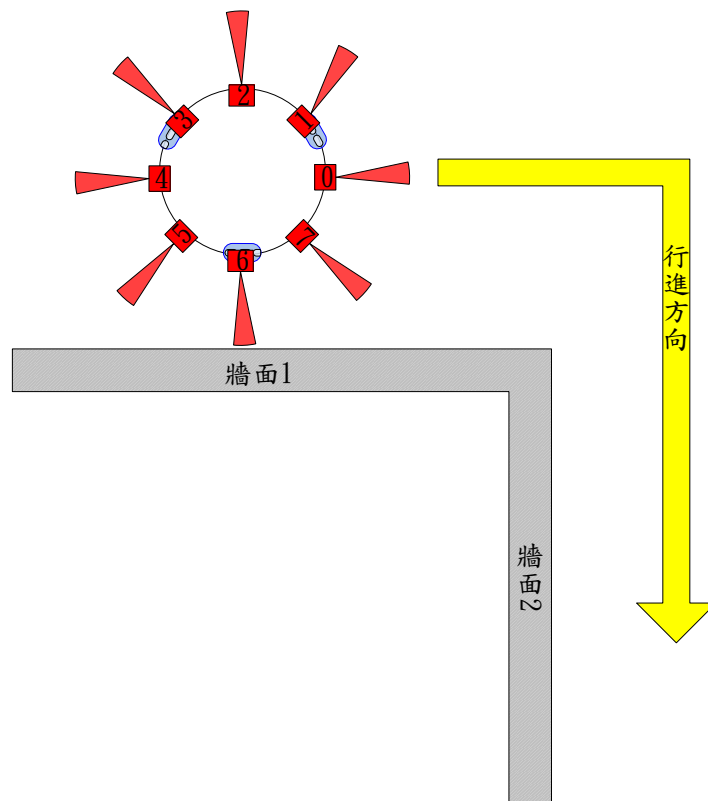


圖 41 凸角牆面行走

### 自動搜尋牆面：

當一開始機器人沒有沿著任何牆面走的時候，我們會自動往紅外線距離值最近的方向移動，並沿著該方向的牆面行走；倘若所有紅外線的訊號都偵測不到時(當距離大於 80cm，則紅外線訊號會偵測不到)，我們自動以車子的右邊走，直到搜尋到牆面。

### 沿著牆面走：

以圖 39 為例子，我們要沿著車體下方的牆面走時，我們會用到離牆面最近的三顆紅外線編號 5、編號 6、編號 7 來偵測跟牆面的關係，利用靠近行進方向的那顆紅外線編號 0 來偵測行進中是否有障礙物；並且以編號 6 來偵測與牆面是否處於安全距離範圍，而編號 5 和編號 7 則是用來調整車身的角度是否與牆面平行。

### 凹角牆面行走：

以圖 40 為例子，原本是沿著牆面 1 行走，當我們發現編號 0 的紅外線低於我們設定的閾值之後，我們會判斷是凹角的地形；此時我們要讓機器人沿著牆面 2 行走，在此刻我們將原本負責偵測的紅外線編號 5、編號 6、編號 7、編號 0 改換成利用編號 7、編號 0、編號 1、編號 2；而編號 0 是負責偵測與牆面是否處於安全距離範圍，編號 7 和編號 1 是負責用來調整車身角度是否與牆面平行，編號 2 是用來偵測行進方向是否有障礙物。

### 凸角牆面行走：

以圖 41 為例子，原本是沿著牆面 1 行走，用來偵測與牆面安全距離的紅外線是編號 6，負責調整車身角度的是編號 5 和編號 7，偵測行進中的障礙物是編號 0，當我們遇到凸角的牆面之後，首先我們編號 7 的紅外線訊號會消失，但是編號 6 的訊號不會消失，此時我們判斷是在進入凸角牆面的開始，讓機器人依然沿著原本方向前進，等到編號 6 和編號 7 同時消失後，我們就會讓機器人改變行進方向，接著沿著牆面 2 行走，與凹角牆面同理，此時我們用來偵測的紅外線換成編號 3、編號 4、編號 5、編號 6。

將幾個與牆面的情況釐清後，我們可以建構出完整的沿牆行走的演算法流程如圖 42 所示。

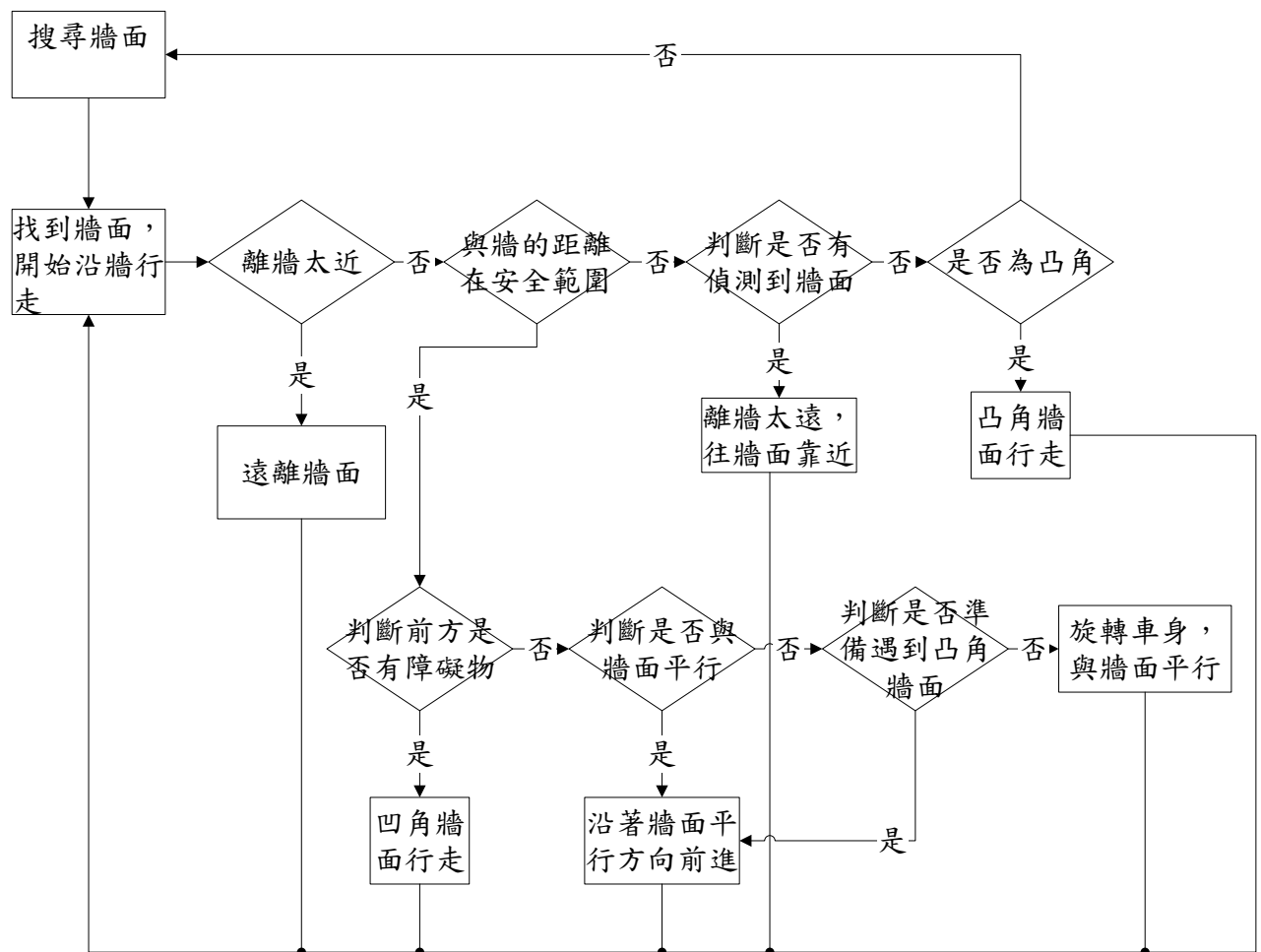


圖 42 沿牆行走流程圖

我們定義 `sensor[center]` 代表目前用來偵測與牆面是否在安全距離範圍的紅外線，`sensor[up]` 代表靠近行進方向的那顆紅外線，`sensor[center_up]` 代表是介於偵測安全距離的紅外線與偵測行進方向的紅外線之中的紅外線，而 `sensor[center_down]` 就是用來調整車身角度的，除了 `sensor[center_up]` 以外的另一顆紅外線。

以下為沿牆行走的虛擬程式碼(pseudo code)：

```

void Wall_Following( )
{
    IF sensor[center]離牆太近 THEN
        遠離牆面
    ELSE IF sensor[center]介於安全距離 THEN
        IF sensor[up]小於等於閾值上限 THEN
            判斷為凹角牆面，變換沿走牆面繼續呼叫 Wall_Following( )
        ELSE sensor[up]大於閾值上限 THEN
            IF sensor[center_up]小於閾值下限 THEN
                旋轉車身以保持與牆面平行
            ELSE IF sensor[center_up]介於安全距離 THEN
                沿著原本行走方向前進
            ELSE IF sensor[center_up]大於閾值上限且訊號沒消失 THEN
                旋轉車身以保持與牆面平行
            ELSE
                IF sensor[center_down]小於閾值下限 THEN
                    旋轉車身以保持與牆面平行
                ELSE IF sensor[center_down] 介於安全距離 THEN
                    判斷為剛進入凸角牆面情況，沿著原本行進方向繼續前進
                ELSE
                    旋轉車身以保持與牆面平行
                END IF
            END IF
        END IF
    ELSE IF sensor[center]大於閾值上限且訊號沒消失 THEN
        靠近牆面
    ELSE
        IF sensor[center_up]訊號消失 THEN
            判斷為已經進入凸角牆面，變換沿走牆面繼續呼叫 Wall_Following( )
        ELSE
            不明狀況，重新搜尋最靠近的牆面，繼續呼叫 Wall_Following( )
        END IF
    END IF
}

```

## 4-3 BFS 最短路徑規劃

Breadth-First-Search(廣度優先搜尋演算法，簡稱 BFS)，是一種圖形搜索演算法 [25]，從根節點開始，沿著樹的寬度遍歷樹的節點，如果所有節點均被訪問，則演算法中止如圖 43 所示。

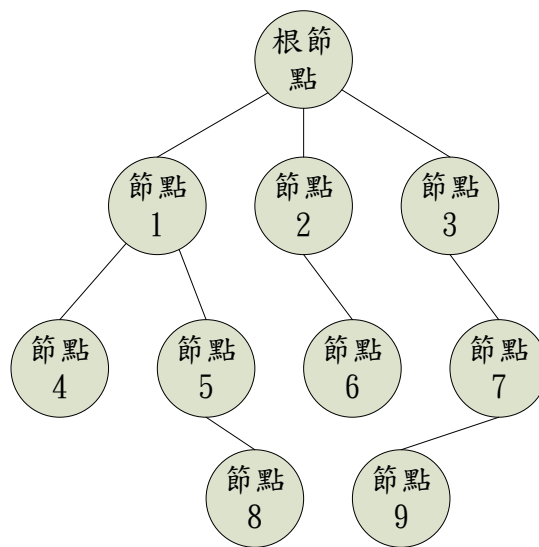


圖 43 BFS 示意圖

以圖 43 為例子，從根節點開始拜訪，接著拜訪根節點相鄰的鄰居，假設我們拜訪的順序都是由該節點左邊鄰居到右邊鄰居，則我們拜訪順序為節點 1、節點 2、節點 3，等到根節點的鄰居都拜訪完後，再依剛剛拜訪的節點順序，依序拜訪該節點鄰居，也就是我們會先拜訪節點 1 的鄰居節點 4 和節點 5，接下來拜訪節點 2 的鄰居節點 6…等等以此類推，直到所有節點都拜訪完為止。

基於 BFS 演算法的完全性，也就是說不論圖形長什麼樣子，BFS 都能夠完全搜索到每一個節點，非常適合用於本實驗當作最短路徑規劃的方法，但是在使用 BFS 之前，我們必須要假設幾個前提條件，如此才能保證 BFS 能求出最短路徑：

1. 環境是已知的，且為封閉的空間(實驗場地皆為室內空間)，如此才能保證 BFS 會收斂(停止)。
2. 把環境畫分成格子狀地圖，相鄰的格子(前、後、左、右、左上、右上、左下、右下)的權重都是一致的，也就是一步的距離，有了這項依據，BFS 求出的路徑才能保證是最短的。
3. 必須先指定起點(也就是機器人的出發點)。

首先我們會先輸入地圖的資訊，然後指定導航機器人的起點，再來指定目標地的位置，目標地必須是在地圖中，接下來使用 BFS 做最短路徑的規劃，然後開始讓機器人往目標地行走，若途中有遇到障礙物則進入避障系統開始避障，一但進入避障系統後，就要重新最短路徑規劃，如此才能在變動的環境下順利的走到目標地，整個流程如圖 45 所示。在這邊我們要特別說明，本論文所指的最短路徑的規劃是指依目前機器人所在的位置與目前環境的狀態距離目標地的最短路徑，而不是從起點到目標地的最短路徑，原因在於環境是變動的，導致機器人遇到障礙物的時候會更新目前環境資訊並啟動即時避障系統，此時的環境狀態已經與最初輸入的已知環境狀態不同。

當我們在做最短路徑規劃之後，我們會把從起點到目標地的規劃路徑中，加入幾個子目標，由機器人依序到達這幾個子目標後，最後到達目標地；而子目標的劃分是依據行走的方向是否一致來決定的，也就是這一步走的方向和下一步走的方向是否相同，若不同，則新增一個子目標地，若相同則繼續往下檢查直到與目標地距離 10 公分以內為止，則我們認為到達目標地，如圖 44 所示。

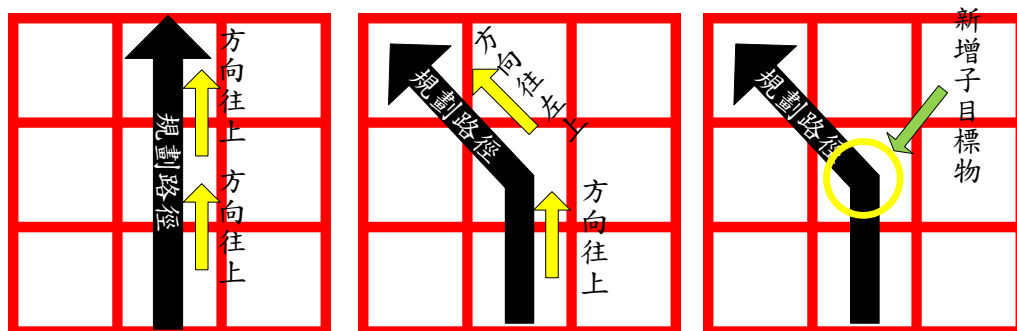


圖 44 新增子目標地示意圖

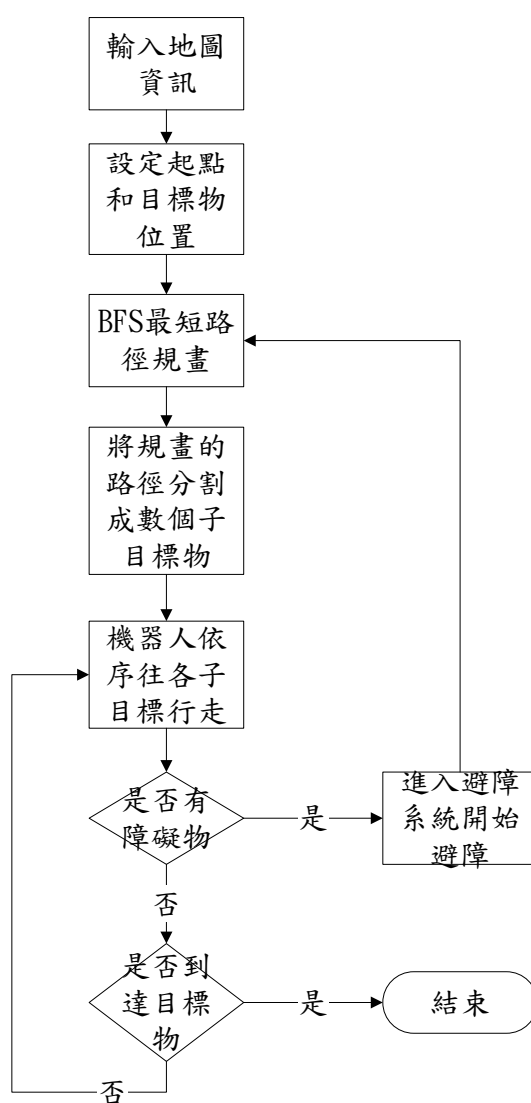


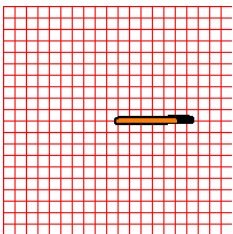
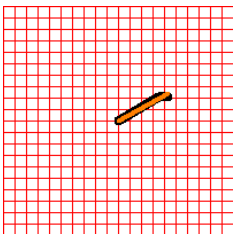
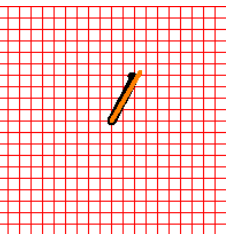
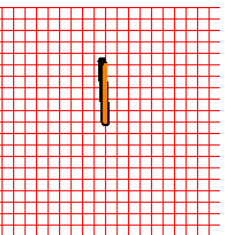
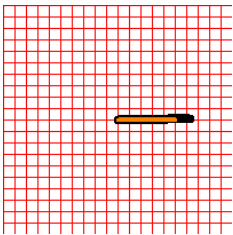
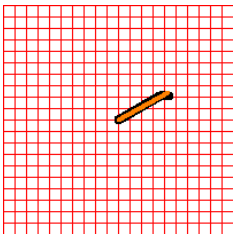
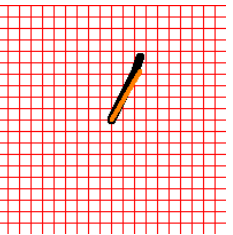
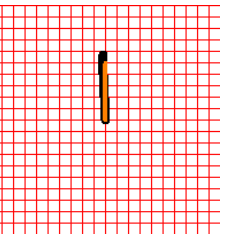
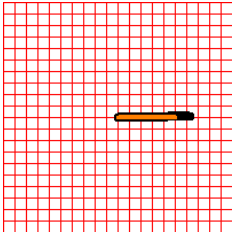
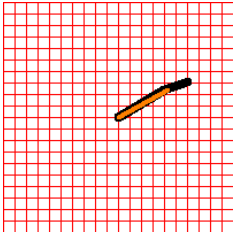
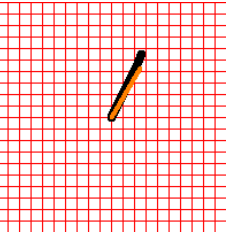
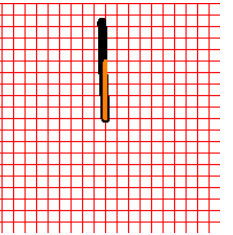
圖 45 BFS 最短路徑規劃模式流程圖

## 第五章實驗結果

### 5-1 機器人的全方位走向實驗

為了測試機器人的任意方向行走能力和定位系統的準確性，我們以機器人的第三顆馬達為 270 度當作基準，指定行走距離為 50 公分，每次以相同原點出發，每隔 30 角度做三次的測試。在電腦端的地圖上畫出指定角度的 50 公分直線，並開啟滑鼠偵測移動距離，再實際量測機器人的行走距離，以驗證其對任意方向行走和定位系統的精準度。實驗結果如表 12、表 14 所示。表 12 中圖片部分，橘色的線為事先畫出的指定角度 50 公分直線，而黑色的線則代表滑鼠偵測機器人的移動軌跡，由實驗可知，機器人在任意角度上的行走是符合我們所指定的角度。

表 12 各角度行走實驗表

角度	0	30	60	90
實驗 1				
實驗 2				
實驗 3				



角度	120	150	180	210
實驗 1				
實驗 2				
實驗 3				
角度	240	270	300	330
實驗 1				
實驗 2				
實驗 3				

表 14 是相對應表 12 各實驗中滑鼠偵測值與實際量測值的數據，並計算滑鼠和實際量測值之間的誤差與指定距離(50 公分)和實際量測值之間的誤差。關於實驗數據的符號說明如表 13：

表 13 實驗表格符號說明

符號	說明
指定角度(實驗編號)	相對應圖中指定角度的第一個實驗數據
$d_{\text{實際}}$	實際量測機器人移動距離值，單位 cm。
$d_{\text{滑鼠}}$	滑鼠偵測機器人移動距離值，單位 cm。
$\Delta d_{\text{滑鼠}}$	實際量測值與滑鼠偵測值之間的誤差，單位 cm。
$\Delta d_{\text{指定}}$	實際量測值與指定距離之間的誤差，單位 cm。

表 14 任意方向行走數據表

實驗編號	0(1)	0(2)	0(3)	30(1)	30(2)	30(3)	60(1)	60(2)	60(3)
$d_{\text{實際}}$	48	48	48	53.1	55.2	52.1	31.3	46.5	46.5
$d_{\text{滑鼠}}$	63.5	63.4	63.9	53.9	50.3	69.3	44.4	61.6	61.9
$\Delta d_{\text{滑鼠}}$	15.5	15.4	15.9	0.8	4.9	17.2	13.1	15.1	15.4
$\Delta d_{\text{指定}}$	2	2	2	3.1	5.2	2.1	18.7	3.5	3.5
實驗編號	90(1)	90(2)	90(3)	120(1)	120(2)	120(3)	150(1)	150(2)	150(3)
$d_{\text{實際}}$	43.5	44	65.5	42.6	45.8	46	46.1	60.6	59.8
$d_{\text{滑鼠}}$	52.2	57.1	84.1	61	63.8	64.1	81.1	78.6	77
$\Delta d_{\text{滑鼠}}$	8.7	13.1	18.6	18.4	18	18.1	35	18	17.2
$\Delta d_{\text{指定}}$	6.5	6	15.5	7.4	4.2	4	3.9	10.6	9.8

實驗編號	180(1)	180(2)	180(3)	210(1)	210(2)	210(3)	240(1)	240(2)	240(3)
d <sub>實際</sub>	49.5	75.1	75.1	81.3	66.2	80.9	31.2	31.2	31.3
d <sub>滑鼠</sub>	51.7	78.9	77.7	50.7	37.2	49.7	34.7	34.8	35.4
$\Delta d_{滑鼠}$	2.2	3.8	2.6	30.6	29	31.2	3.5	3.6	4.1
$\Delta d_{指定}$	0.5	25.1	25.1	31.3	16.2	30.9	18.8	18.8	18.7
實驗編號	270(1)	270(2)	270(3)	300(1)	300(2)	300(3)	330(1)	330(2)	330(3)
d <sub>實際</sub>	44	44	44	46.9	30.4	47.4	51.7	50.3	52.6
d <sub>滑鼠</sub>	44.6	41.9	38.6	49.4	36.2	49.3	52.6	53.3	52.5
$\Delta d_{滑鼠}$	0.6	2.1	5.4	2.5	5.8	1.9	0.9	3	0.1
$\Delta d_{指定}$	6	6	6	3.1	19.6	2.6	1.7	0.3	2.6

由表 14 計算可得知， $\Delta d_{滑鼠}$  平均誤差為 11.4cms，標準差為 9.636894cms； $\Delta d_{指定}$  平均誤差為 9.5cms，標準差為 8.883208cms。經由現場觀測電腦螢幕中滑鼠的移動情況發現，造成誤差的原因主要有三個，第一當程式啟動的動作與判斷過多的時候，會造成滑鼠的在計算像素與實際距離的程式有稍微延遲的現象。第二由於地面不完全是水平的，而滑鼠有所謂的 lift distance 的限制，lift distance 就是滑鼠離地或平面多遠的距離依然可以偵測移動的情況，所以在地面有稍微凹陷的部分時，有可能造成滑鼠已超過 lift distance 的距離而少算了當下的移動距離。第三，當滑鼠有多算距離的情況出現時，代表我們在初始化定位系統的時候受到第二個因素的影響，而提高了像素與實際距離的對應值，進而造成誤差。

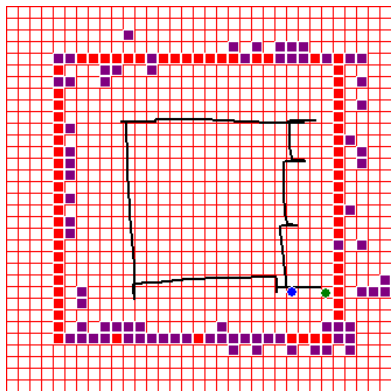
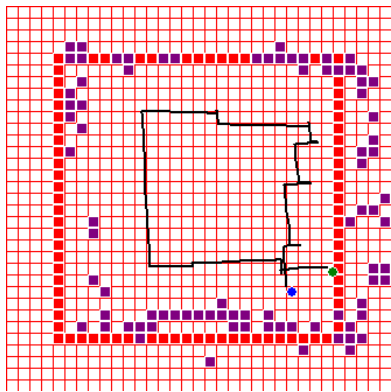
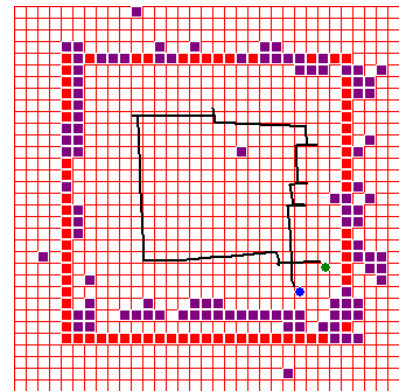
## 5-2 沿牆行走實驗

為了測試機器人在未知環境下的探索能力，我們設計了三種地形：正方形、L 型、N 型，並且在各地形以繞完整一圈地形為一次實驗，個別做了 9 次實驗，並且紀錄一次實驗所需的時間、總共行走的路徑長與感測器的偵測值並將之顯示於地圖上。地形的設計主要是為了測試機器人在沿牆行走時，是否能夠測試到沿牆行走的所有基本功能：沿上方牆往左邊走、沿上方牆往右邊走、沿下方牆往左邊走、沿下方牆往右邊走、沿左邊牆往上走、沿左邊牆往下走、沿右邊牆往上走、沿右邊牆往下走、凹角牆面變換方向、凸角牆面變換方向。此外，我們以紅色格點建立實際測試地形，利用紅外線感測器偵測環境資訊並將之以紫色格點顯示於地圖上，與實際測試地形做一個參照，以便了解其探索環境的能力。

### 5-2-1 正方形

在正方型的地形下，我們要測試機器人在凹角牆面的變換行走能力與基本的沿牆行走能力，實驗結果如表 15 所示。

表 15 正方型地形實驗

實驗 1	實驗 2	實驗 3
		

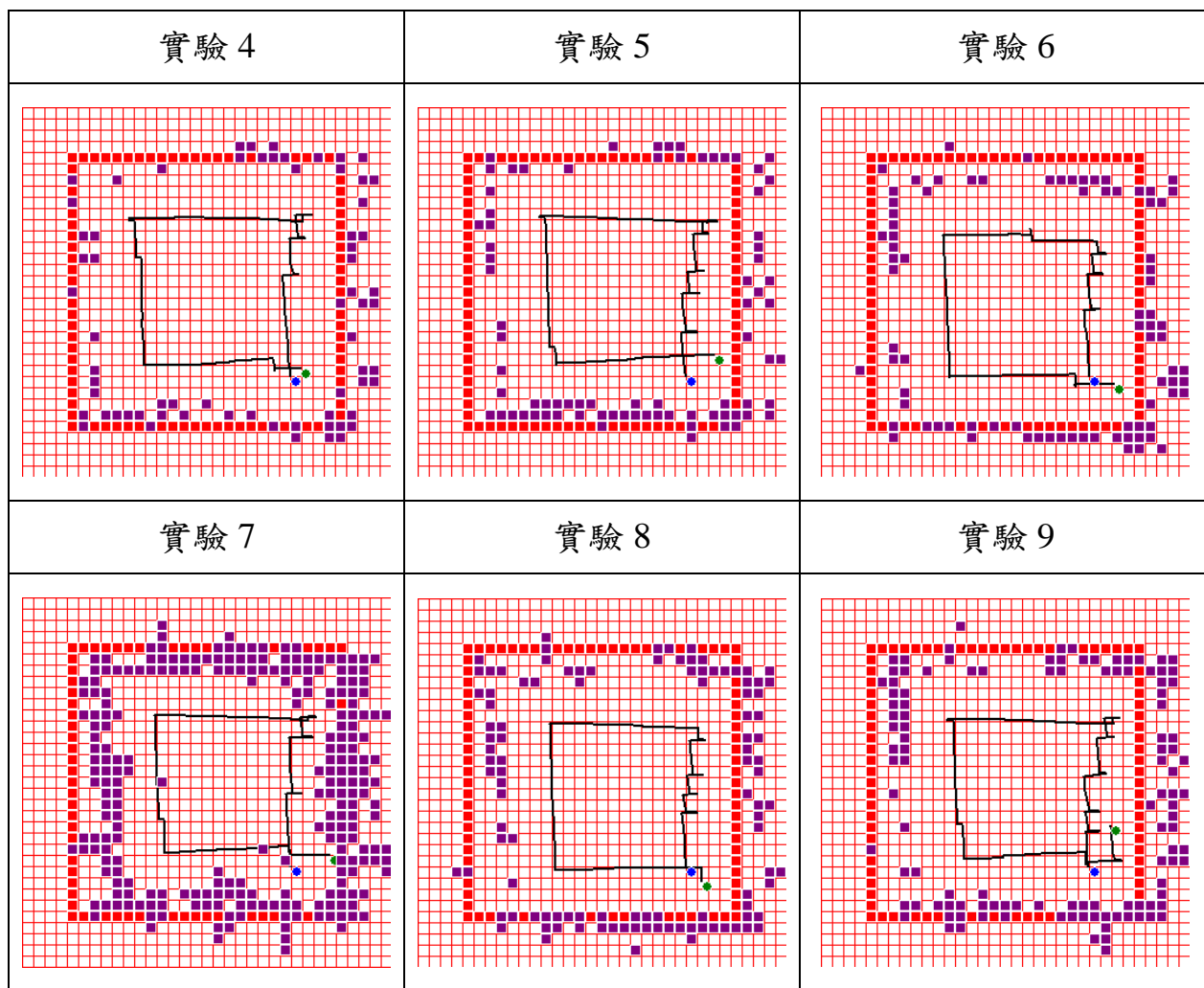


表 16 是表 15 各實驗紀錄的數據，包含行走的時間、行走的距離，以了解其探索的效率。

表 16 正方形地形實驗數據

實驗編號	1	2	3	4	5	6	7	8	9	平均	標準差
行走時間(秒)	151	126	129	141	152	134	121	139	165	139	19.88
行走距離(cm)	740	692	680	670	695	632	641	622	737	678	51.63

## 5-2-2 L 型

在 L 型的地形下，我們要測試機器人在凹角牆面和凸角牆面的變換行走能力與基本的沿牆行走能力，實驗結果如表 17、表 18 所示。

表 17 L 型地形實驗

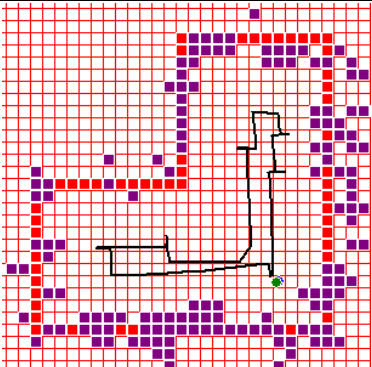
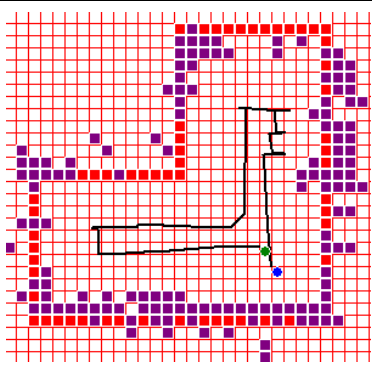
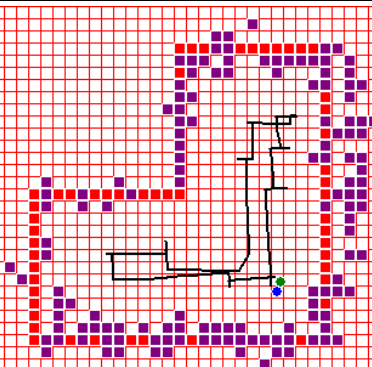
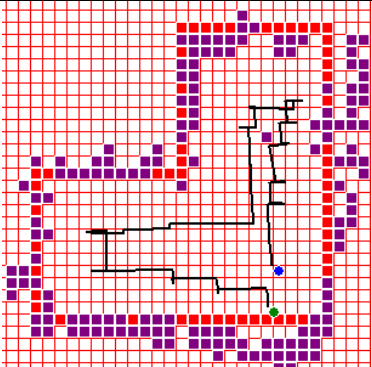
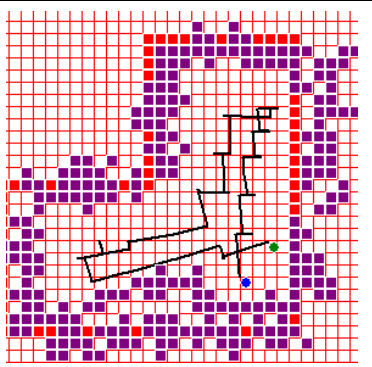
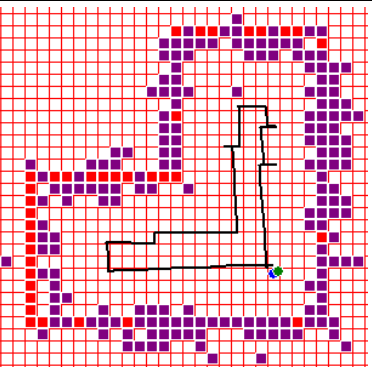
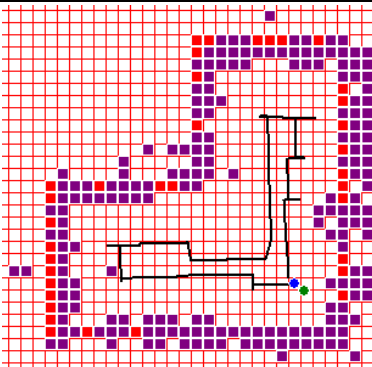
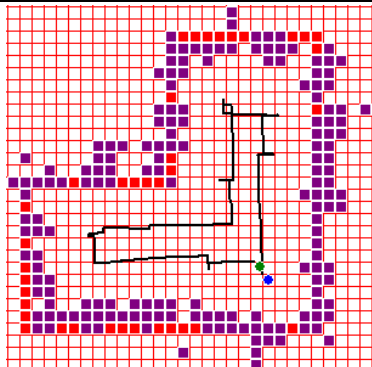
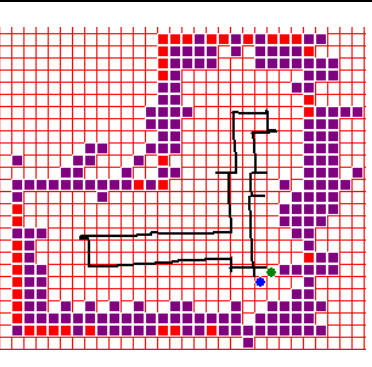
實驗 1	實驗 2	實驗 3
		
實驗 4	實驗 5	實驗 6
		
實驗 7	實驗 8	實驗 9
		

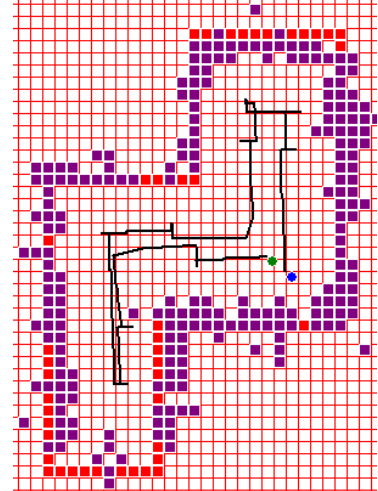
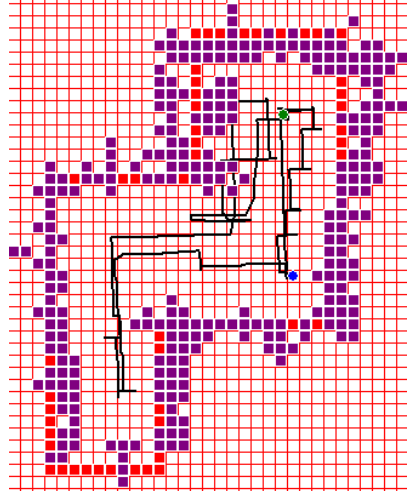
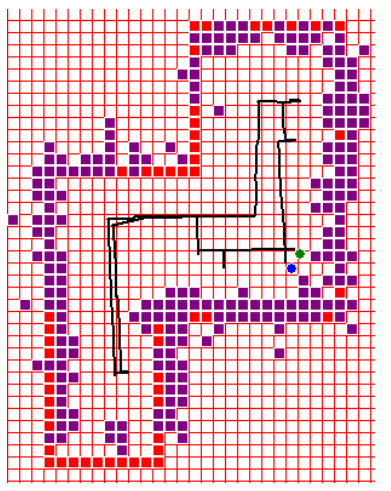
表 18 L 型地形實驗數據

實驗編號	1	2	3	4	5	6	7	8	9	平均	標準差
行走時間(秒)	135	127	153	197	177	142	147	135	146	151	21.09
行走距離(cm)	681	622	734	814	786	607	720	661	694	702	66.63

### 5-2-3 N 型

在 N 型的地形下是屬於 L 型的推廣，也就是說若我們可以在 N 型的地形下正常進行探索，則可以探索任意個由 L 型所組合的地形，實驗結果如表 19、表 20 所示。

表 19 N 型地形實驗

實驗 1	實驗 2	實驗 3
		

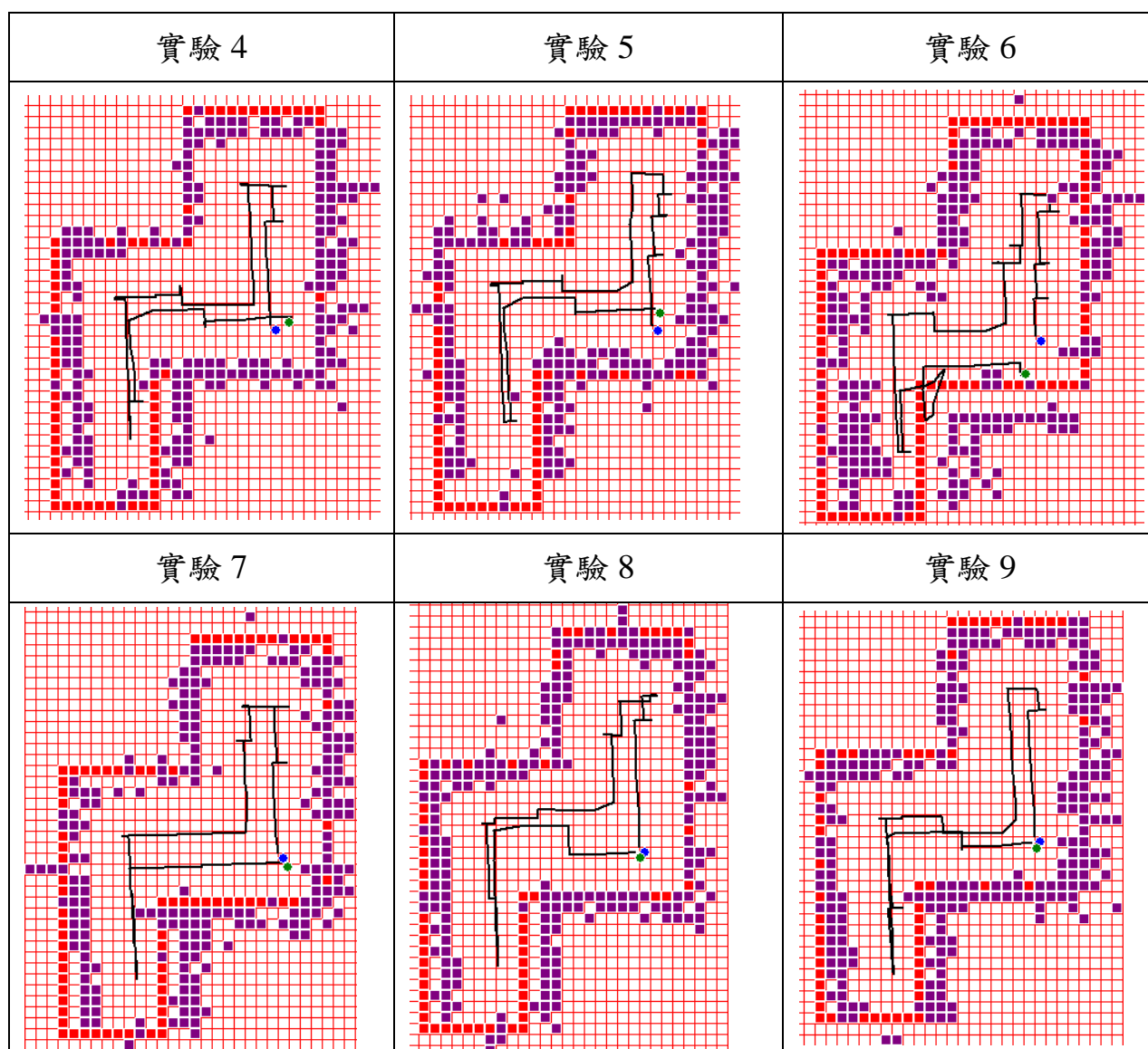


表 20 N 型地形實驗數據

實驗編號	1	2	3	4	5	6	7	8	9	平均	標準差
行走時間(秒)	209	339	198	217	210	288	192	181	189	224	53.43
行走距離(cm)	950	1565	921	933	914	1050	856	899	914	1000	206.74



## 5-3 BFS 最短路徑規劃實驗

在已知的環境中，本論文提出三種實驗情境：無障礙物環境、單一障礙物環境、多障礙物環境。目的是為了測試導航機器人可以在沒有障礙物的阻礙下透過 Breadth-first search 計算出來的最短路徑順利達到目標地；若環境有所變動，多了一個障礙物，則機器人會透過即時避障系統閃開障礙物，並根據紅外線感測器接收到的新環境資訊重新路徑規劃，藉由新的路徑達到目標地；若環境的變動更複雜時，障礙物之間可走的區域非常有限的情況下，機器人依然可以避開障礙物順利達到目標地。我們在三種實驗情境中，個別做了五次實驗，在每一次實驗中，紀錄機器人在到達目標地的過程中計算出來的每一條最短路徑規劃軌跡、行走距離與行走時間。

### 5-3-1 實驗環境 1-無障礙物

我們在一塊 510cm X 180cm 的走道下當作實驗的環境如圖 46 所示，起點設在離右邊牆壁 60cm 的距離，目標地則設在起點的正前方 410cm 的距離，而我們設定離牆壁或障礙物的距離若小於 25cm 則啟動避障模式。經由實驗可知，導航機器人確實可以到達目標地。

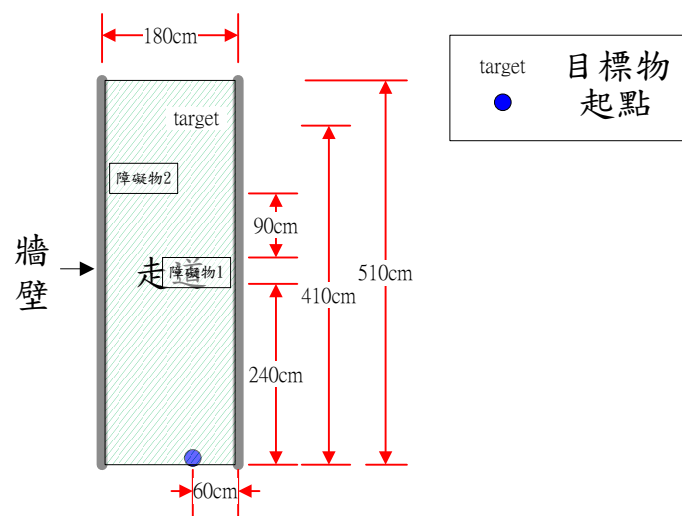


圖 46 BFS 實驗環境

表 21 為五次實驗的結果，其數據如表 22。圖片的部分，黑色實線是機器人的移動軌跡，橘色實線是 BFS 最短路徑規劃的軌跡，藍色的圓點是起點，綠色的圓點是目前機器人所在的位置，淺紅色的方格點為實際地形的輪廓，而深紅色的方格點則是機器人利用紅外線感測器偵測到的環境資訊。

表 21 BFS 無障礙實驗

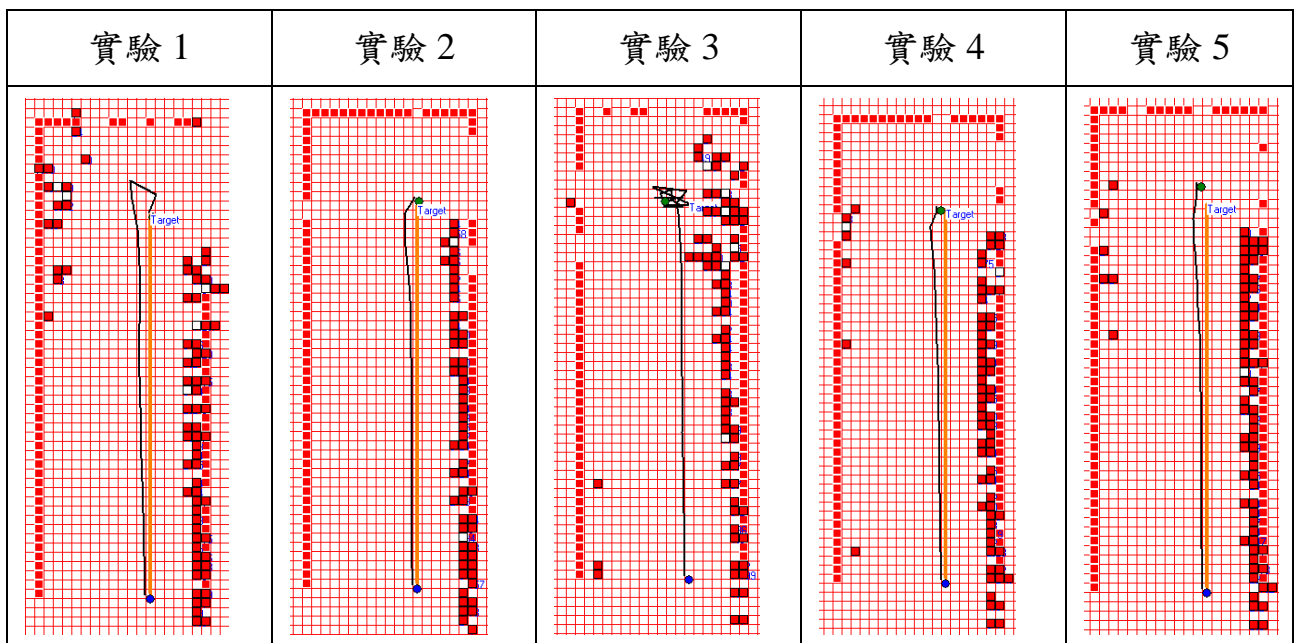


表 22 BFS 無障礙實驗數據

實驗編號	1	2	3	4	5	平均	標準差
行走時間(秒)	30	34	56	42	29	38.2	10
行走距離(cm)	505.7	424.9	771.4	418.6	435.5	511	134.61

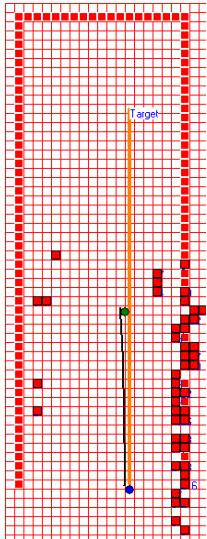
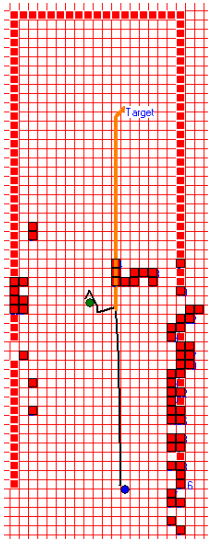
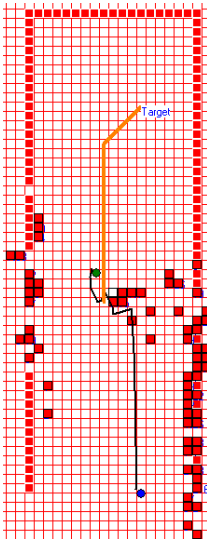
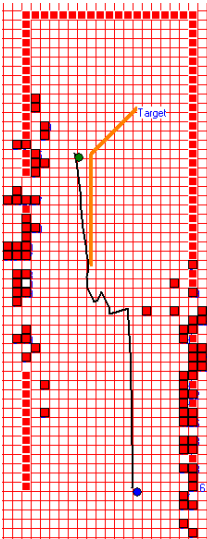
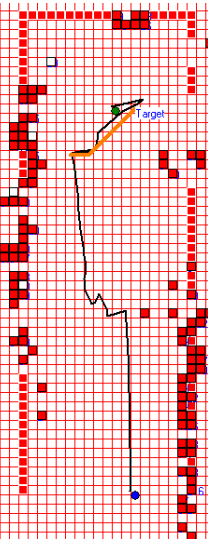
由表 21、表 22 可以得知實驗 3 所偵測的地圖誤差較大，其原因是陀螺儀值不夠精準而造成朝向角與實際不符所造成的，因而造成位置的誤判，實際位置已經非常靠近牆面，但是地圖顯示卻離牆壁有些距離，導致不斷的啟

動避障系統，與往目標地行走的模式之間不斷切換，致使行走時間與距離都被拉長。

### 5-3-2 實驗環境 2-單一障礙物

在 BFS 實驗環境中，加入一個障礙物，使機器人可以即時偵測到障礙物並順利避障，再依新的最短路徑規劃往目標地前進。由於新增障礙物的關係，機器人會偵測到障礙物啟動避障系統，所以會有新的最短路徑規劃產生。我們針對五次實驗中，將各實驗從起點到終點間曾經啟動避障系統並重新規劃路徑的過程呈現出來，實驗結果如表 23~表 26。

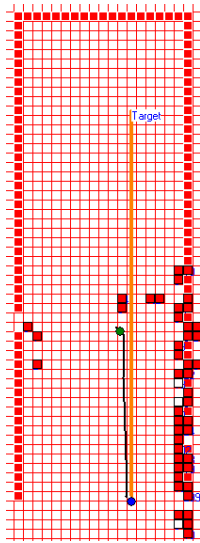
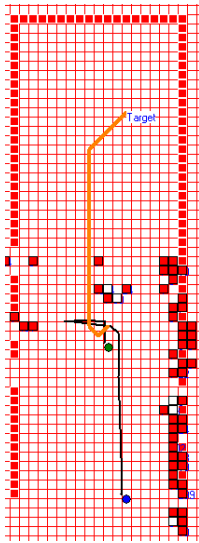
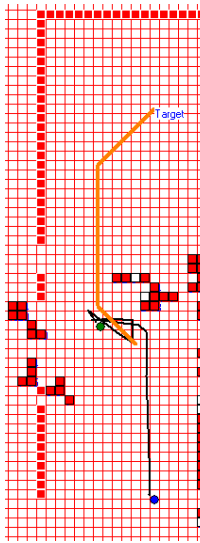
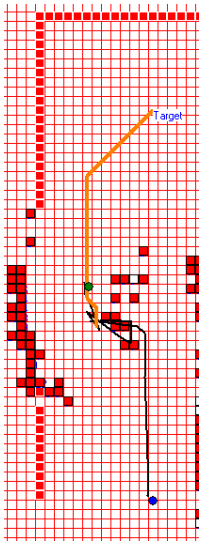
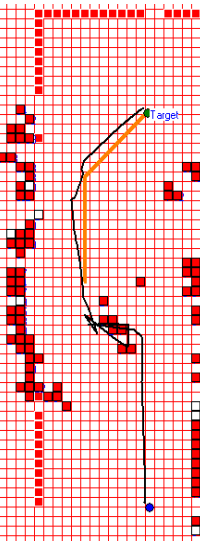
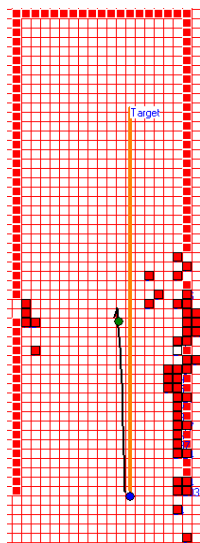
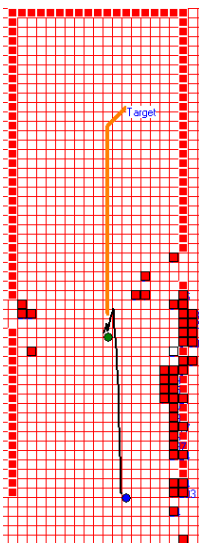
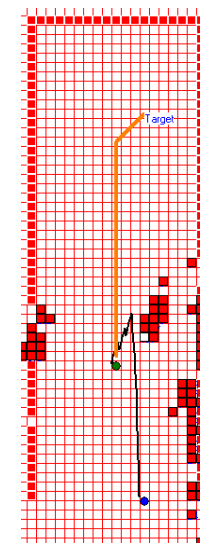
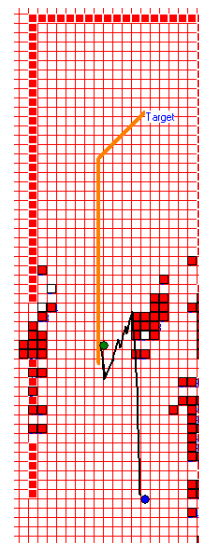
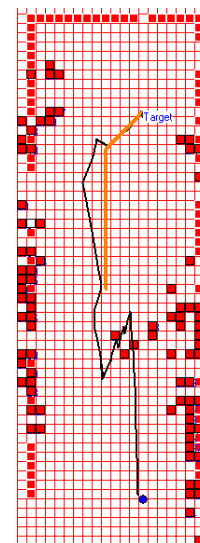
表 23 BFS 單一障礙物實驗 1

實驗 1				
過程 1	過程 2	過程 3	過程 4	過程 5
				

由實驗 1 的過程 3 中觀察，機器人的移動軌跡似乎和障礙物的格點重疊在一起，並不是代表機器人碰到障礙物的關係，而是因為我們設定地圖每個方格為 10 公分，當機器人與障礙物靠很近時(低於 10 公分時)，有可能會將目前機器人的位置和障礙物的位置分在同一個方格中。另一個因素是機器人實

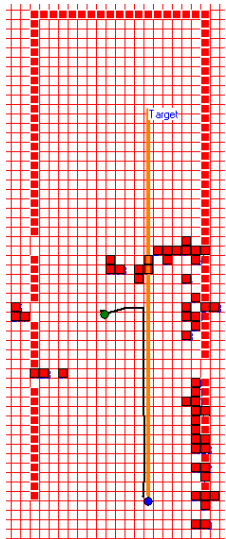
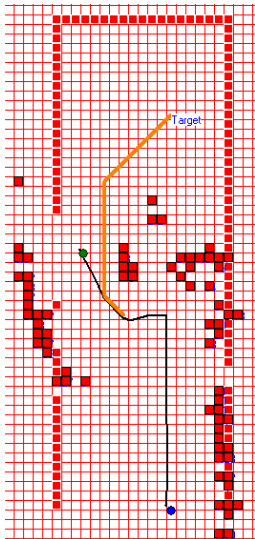
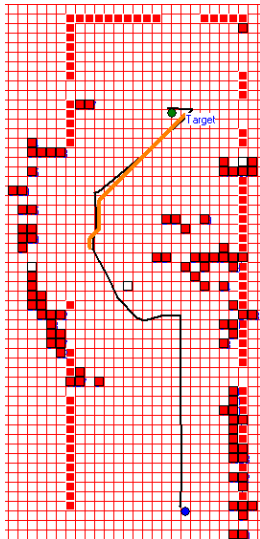
際偏移的角度與陀螺儀計算出來的朝向角相差太多時，會導致紅外線偵測到的障礙物方位與實際障礙物方位有偏差，進而造成這種現象。

表 24 BFS 單一障礙物實驗 2 和實驗 3

實驗 2				
過程 1	過程 2	過程 3	過程 4	過程 5
				
實驗 3				
過程 1	過程 2	過程 3	過程 4	過程 5
				

由實驗 3 的過程 4 和过程 5 可以發現，在过程 4 偵測到的障礙物，在过程 5 的時候深紅色格點消失了，由於紅外線的特性在 50~80 公分之間的值是跳動很大，所以可能障礙物原本在 50 公分的位置若偵測為 60 公分的話，就會將介於 50cm~60cm 的障礙物格點消除。

表 25 BFS 單一障礙物實驗 4 和實驗 5

實驗 4		
過程 1	過程 2	過程 3
		

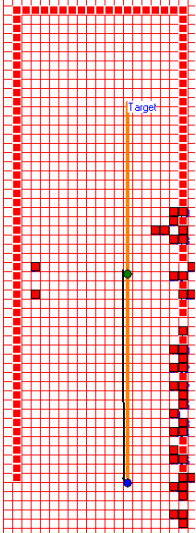
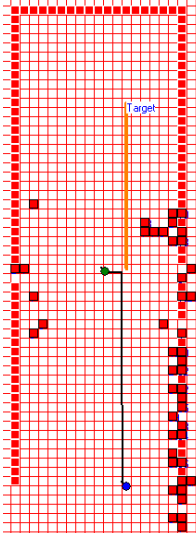
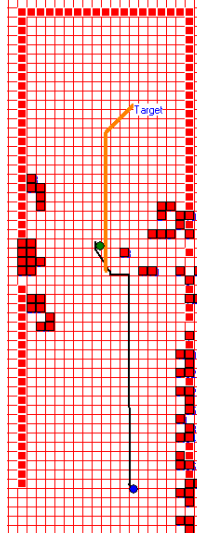
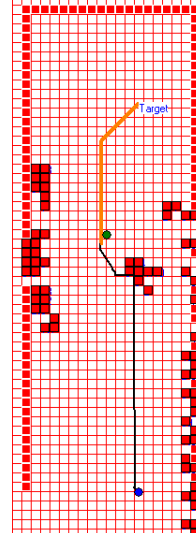
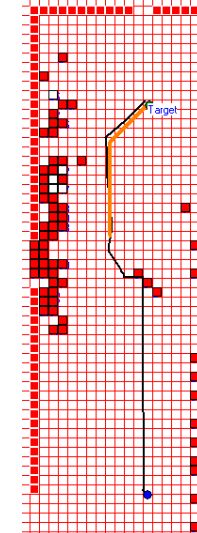
實驗 5				
過程 1	過程 2	過程 3	過程 4	過程 5
				

表 26 BFS 單一障礙物實驗數據

實驗編號	1	2	3	4	5	平均	標準差
行走時間(秒)	78	64	72	50	53	63.4	10.72
行走距離(cm)	677.3	703	700.5	574.4	471.2	625.2	90.81

### 5-3-3 實驗環境 3-多個障礙物

本次實驗放置了兩個障礙物，而兩障礙物之間的垂直距離為 90cm，並且設計擺設的方式是機器人必須通過第一個障礙物之後，必定需要閃開第二個障礙物才能到達目標地。由於放置兩個障礙物會導致避障系統啟動頻率非常頻繁，因此我們只擷取較重要最短路徑規劃的過程圖，實驗結果如表 27~表 30 所示，而實驗影片可在計算式智慧暨人機互動實驗室網站下載觀看 [26]。

表 27 BFS 多障礙物實驗 1

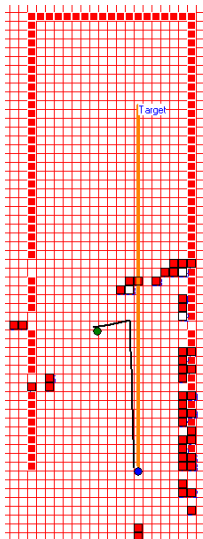
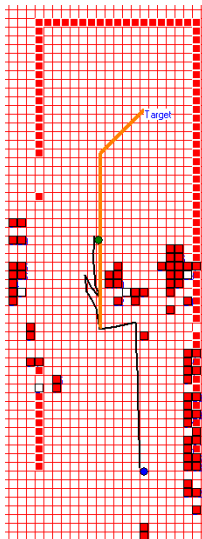
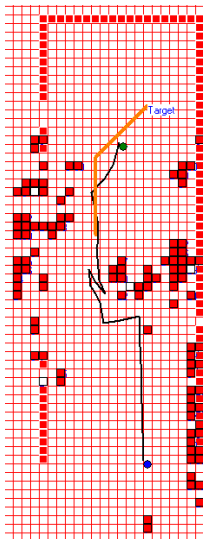
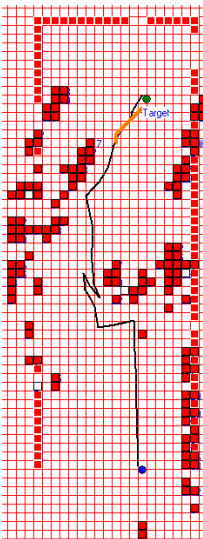
實驗 1			
過程 1	過程 2	過程 3	過程 4
			

表 28 BFS 多障礙物實驗 2 和實驗 3

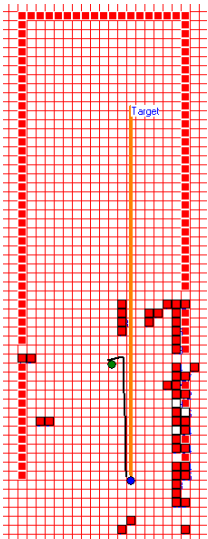
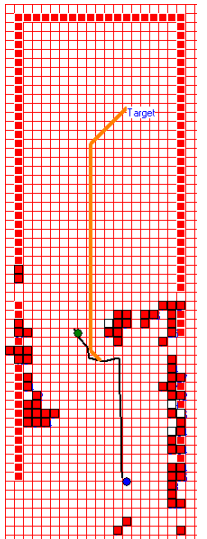
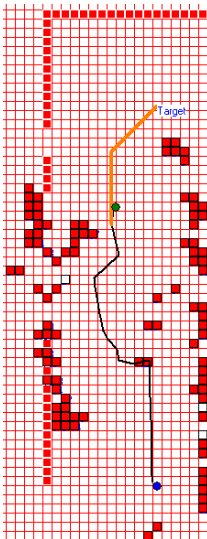
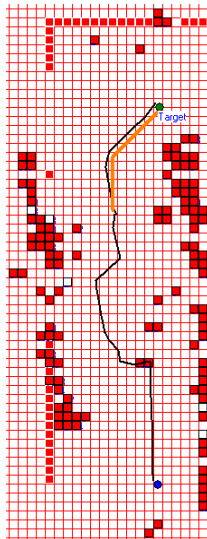
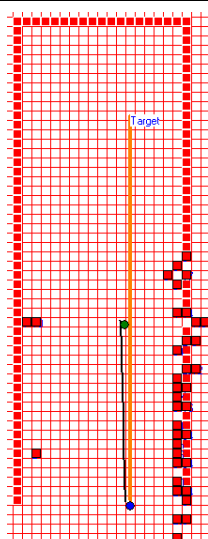
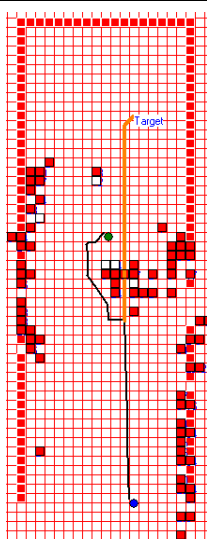
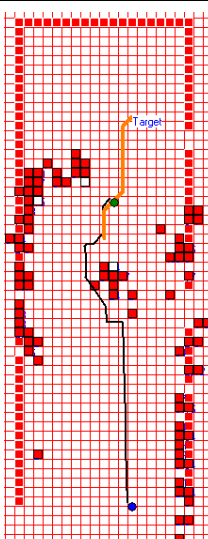
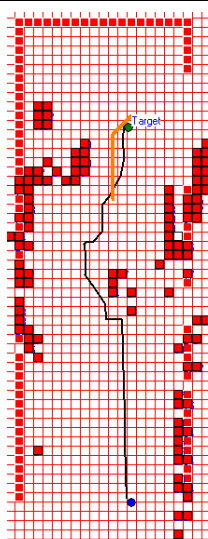
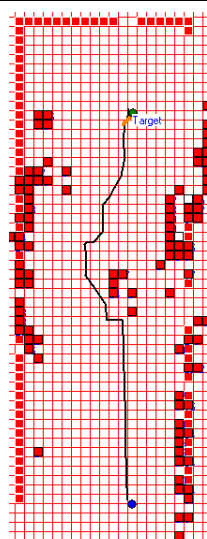
實驗 2				
過程 1	過程 2	過程 3	過程 4	
				
實驗 3				
過程 1	過程 2	過程 3	過程 4	過程 5
				



表 29 BFS 多障礙物實驗 4 和實驗 5

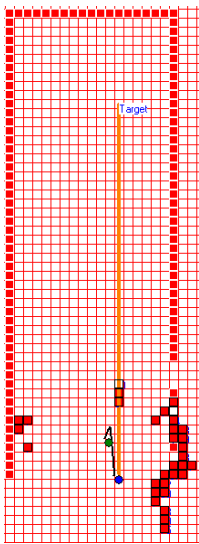
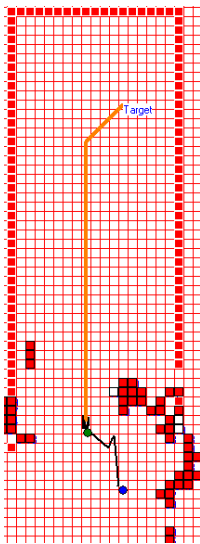
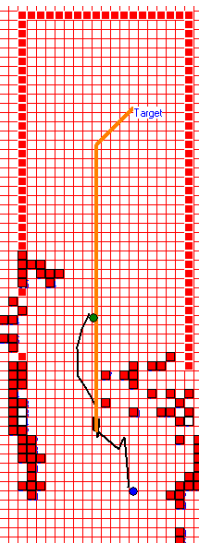
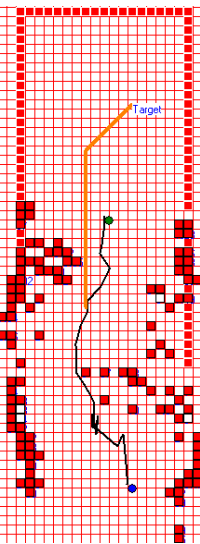
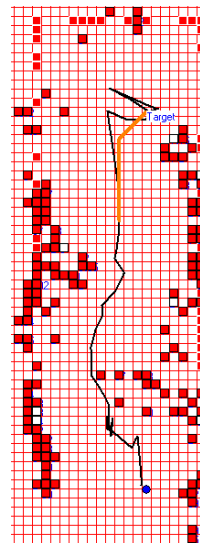
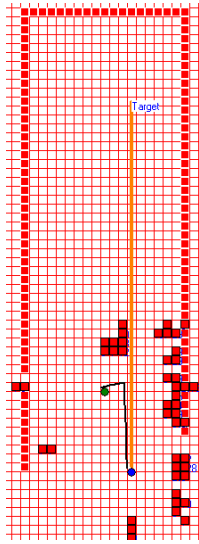
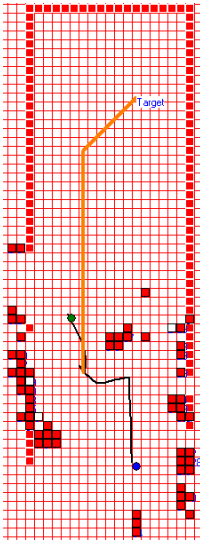
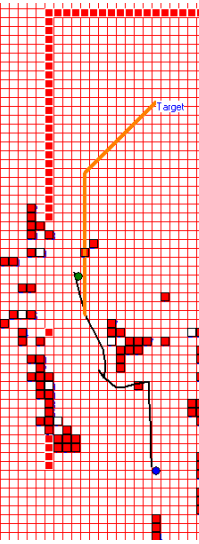
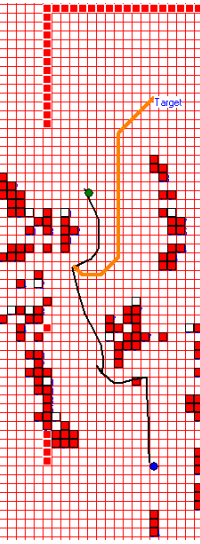
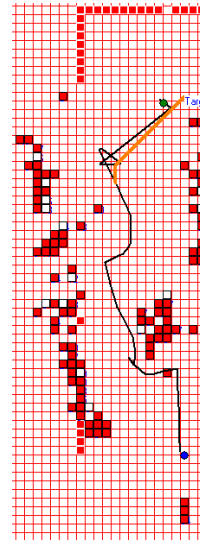
實驗 4				
過程 1	過程 2	過程 3	過程 4	過程 5
				
實驗 5				
過程 1	過程 2	過程 3	過程 4	過程 5
				

表 30 BFS 多障礙物實驗數據

實驗編號	1	2	3	4	5	平均	標準差
行走時間(秒)	54	56	65	93	90	71	19.08
行走距離(cm)	581	517	478	831	855	652	160.87

## 第六章結論與未來展望

### 6-1 結論

本論文提出的導航機器人，在機構設計的部分，我們使用三顆全向輪來平衡整個車身的平台，透過三輪運動學，使得我們的機器人可以任意方向行走，旋轉不需迴轉半徑。基於全向輪的特性，讓機器人在行走時可以有效地降低與地面的摩擦力。

在系統設計的部份，由於馬達的最高轉速是固定的，為了解決給定運動學方程式的距離夠遠的情況下，計算出來的馬達轉速會非常高的問題，我們提出了利用時間換取速度的全方位走向演算法來克服這個問題，讓我們在操控導航機器人不會因馬達而受到限制。

在定位系統中，我們使用別於傳統利用編碼器的方式，而使用滑鼠和陀螺儀來做位置的推算，不管是在移植還是使用上都極其方便，甚至不限於特定的滑鼠或是陀螺儀的規格，非常具有彈性。

在行走模式中，有分為在已知環境或是未知環境下的操作。若是在未知環境下，我們可以使用沿牆行走來探索環境，以對環境有一個大概的了解；如果是在已知的環境下，我們可以利用 BFS 最短路徑規劃或是軟體模擬無線遙控器的模式來達到目標地。

本論文研製的導航機器人已具備最基本且重要的功能，在應用的層面上，有相當大的發展空間。

## 6-2 未來展望

導航機器人在軟體模擬無線遙控器模式中，是透過藍芽模組無線傳輸的，而藍芽的最遠傳輸距離只能達到 100 公尺，而且訊號的傳輸隨著距離的拉長而不穩定；若能在筆記型電腦中撰寫透過 WiFi 無線通訊協定的軟體模擬遙控器，來控制機器人，則可遙控的範圍就更廣更穩定了。

由於類似玻璃的地板對於我們的定位系統而言影響甚大，其原因在於滑鼠在偵測玻璃地板的技術尚不成熟，因此是否能夠透過不斷的校正來減少定位誤差或是使用對表面材質更靈敏的滑鼠將會是一個重要的議題。

目前我們只使用 8 顆紅外線的感測器來偵測外界的環境資訊，對於環境的偵測上有很大的死角，我們可以加多一點紅外線感測器，盡量減少死角偵測不到的可能性，但是畢竟這是一時的權宜之計，若能再加上攝影機，結合影像處理的技術，做邊緣偵測、道路辨識、用立體視覺判斷可行距離與物體辨識，再搭配紅外線的使用，相信可以建置一套更完善的探索環境和避障機制。

## 參考文獻

- [1] 網路資料：慧典市場研究報告網：2010-2012 年全球機器人產業發展研究分析報告。2010 年 6 月 14 日，取自  
<http://big5.hdcmr.com/jiqiren63691BaoGao.html>。
- [2] 網路資料：工業技術研究院，白忠哲：中台灣發展智慧型機器人產業分析。2010 年 6 月 14 日，取自  
[http://www.roboat.org.tw/upload\\_files/test/1233821271\\_1.pdf](http://www.roboat.org.tw/upload_files/test/1233821271_1.pdf)。
- [3] 網路資料：工業技術研究院：工業技術與資訊 199 期 2008 年 5 月號。2010 年 6 月 14 日，取自 <http://college.itri.org.tw/TopicLearn.aspx?id=140>。
- [4] 網路資料：電子工程專輯：工研院與新光保全成功開發保全機器人 SeQ-1。2010 年 6 月 14 日，取自  
[http://www.eettaiwan.com/ART\\_8800460772\\_676964\\_NT\\_4d488069.HTM](http://www.eettaiwan.com/ART_8800460772_676964_NT_4d488069.HTM)。
- [5] 網路資料：財團法人精密機械研究發展中心，劉憲鋒：導覽機器人機械手機構概念設計－以 Upitor 為例。2010 年 6 月 14 日，取自  
[http://www.robotworld.org.tw/index.htm?pid=10&News\\_ID=3754](http://www.robotworld.org.tw/index.htm?pid=10&News_ID=3754)。
- [6] Boston Dynamics. [online]. Available:  
[http://www.bostondynamics.com/img/BigDog\\_Overview.pdf](http://www.bostondynamics.com/img/BigDog_Overview.pdf) June 14,2010  
[data access]
- [7] Pink Tentacle. [online]. Available:  
<http://pinktentacle.com/2009/11/photos-international-robot-exhibition-2009/>  
June 14,2010 [data access]
- [8] S. M. Killough and F. G. Pin, "Design of an Omnidirectional and holonomic wheeled platform prototype," in *1992 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 84-90, May 12-14, 1992.
- [9] W. K. Loh, K. H. Low, and Y. P. Leow, "Mechatronics design and kinematic modelling of a singularityless omni-directional wheeled mobile robot," in *IEEE International Conference on Robotics and Automation*, vol.3, pp. 3237-3242, September 14-19, 2003.

- [10]J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,” *IEEE Transactions on Systems Man and Cybernetics*, vol.19, no.5, pp.1179-1187, September 1989.
- [11]蘇木春、張孝德, “機器學習:類神經網路、模糊系統以及基因演算法則”, 修訂二版, 全華圖書股份有限公司, 台北市, 民國九十三年。
- [12]J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278-288, June 1991.
- [13]A. Elfes, “Sonar-based real-world mapping and navigation,” *IEEE Journal of Robotics and Automation*, vol.3, no.3, pp. 249-265, June 1987.
- [14]H. P. Moravec, “Sensor Fusion in Certainty Grids for Mobile Robots,” *AI Magazine*, pp. 61-74, 1988.
- [15]H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 116-121, March 1985.
- [16]J. Borenstein and Y. Koren, “Histogramic in-motion mapping for mobile robot obstacle avoidance,” *IEEE Transactions on Robotics and Automation* , vol. 7, no. 4, pp. 535-539, August 1991.
- [17]A. Saffiotti, “The uses of fuzzy logic for autonomous robot navigation,” *Soft Computing*, vol. 1, no. 4, pp. 180-197, 1997.
- [18]A. Fatmi, A. A. Yahmadi, L. Khriji, and N. Masmoudi, “A Fuzzy Logic Based Navigation of a Mobile Robot,” in *Proceedings of World Academic of Science, Engineering and Technology*, vol. 15, October 2006.
- [19]Atmel Corporation. [online]. Available:  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc2466.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf) June 20 2010[data access]
- [20]SHAYANG YE INDUSTRIAL CO.,LTD. [online] Available:  
<http://www.shayye.com.tw/pdf/IG42-01&02-2006.pdf> June 23,2010 [data access]

- [21]Pololu Robotics and Electronics. [online]. Available:  
<http://www.pololu.com/picture/view/0J121> June 27,2010 [data access]
- [22]G. P. Roston, and E. P. Krotkov, “Dead Reckoning Navigation For Walking Robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 607-612, July 7-10, 1992.
- [23]Avago Technologies. [online]. Available:  
<http://www.avagotech.com.tw/docs/AV02-1358EN> June 27,2010 [data access]
- [24]C. Chen, H. X. Li , and D. Dong , “Hybrid Control for Robot Navigation - A Hierarchical Q-Learning Algorithm,” *IEEE Robotics & Automation Magazine*, vol. 15, no. 2, pp. 37-47, June 2008.
- [25]T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.
- [26]計算式智慧暨人機互動實驗室-導航機器人實驗影片  
<http://cilab.csie.ncu.edu.tw/students/Tally/>

## 附錄 1 GP2D12 距離、電壓、ADC 對照表

距離 (單位：Cm)	電壓(單位：V)	ADC 值
10	2.54	1016
11	2.4	960
12	2.24	896
13	2.1	840
14	1.97	788
15	1.84	736
16	1.74	696
17	1.66	664
18	1.57	628
19	1.5	600
20	1.45	580
21	1.39	556
22	1.33	532
23	1.28	512
24	1.23	492
25	1.18	472
26	1.14	456
27	1.11	444
28	1.08	432
29	1.05	420
30	1.01	404
31	0.99	396
32	0.96	384
33	0.93	372
34	0.91	364
35	0.89	356
36	0.87	348
37	0.85	340
38	0.84	336
39	0.82	328



40	0.8	320
41	0.78	312
42	0.77	308
43	0.76	304
44	0.74	296
45	0.72	288
46	0.72	288
47	0.7	280
48	0.68	272
49	0.68	272
50	0.66	264
51	0.66	264
52	0.65	260
53	0.64	256
54	0.62	248
55	0.62	248
56	0.62	248
57	0.61	244
58	0.59	236
59	0.59	236
60	0.56	224
61	0.55	220
62	0.55	220
63	0.53	212
64	0.53	212
65	0.53	212
66	0.51	204
67	0.51	204
68	0.51	204
69	0.49	196
70	0.49	196
71	0.49	196
72	0.48	192
73	0.47	188

74	0.47	188
75	0.46	184
76	0.45	180
77	0.45	180
78	0.45	180
79	0.43	172
80	0.43	172