

國立雲林科技大學機械工程系

碩士論文

Department of Mechanical Engineering

National Yunlin University of Science & Technology

Master Thesis

光學雷達導引之智慧型自主式車輛研究

Intelligent Autonomous Vehicle with LiDAR Guidance



盧玟攸

Wun-You Lu

指導教授：吳英正 博士

Advisor: Ying-Jeng Wu, Ph.D.

中華民國 104 年 6 月

June 2015

國立雲林科技大學
研究所碩士班學位論文考試委員會審定書

本論文係盧致攸君在本校 機械工程系提論文「光學雷達導引之智慧型自主式車輛研究」合於碩士資格水準，業經本委員會評審認可，特此證明。

口試委員：

吳英正

吳尚德

王威立

吳英正
吳尚德
王威立

指導教授：

吳英正

吳英正

所 長：

機械工程系
代理主任 劉旭光

中 華 民 國 104 年 7 月 28 日

摘 要

本論文使用光學雷達(LiDAR)導引智慧型自主式車輛(IAV)，使智慧型自主式車輛完成自動室內巡航與避開巡航路線上之障礙物功能(cruise and obstacle avoidance)。本論文以提升第一代 Yun-Trip 智慧型自主式車輛之巡航時的車速與避開巡航路線上之障礙物為主。而新開發的 Yun-Trip II 使用原先 Yun-Trip 的車體底盤，硬體部分使用單顆光學雷達(LiDAR)取代原先的電腦視覺(computer vision)，運算核心更換成 ARM 處理器架構之 GUN/Linux 嵌入式系統(ARM-Based embedded system GNU/Linux)，軟體則使用 C++ 程式語言進行開發，計算巡航路線和路線位置判斷。本論文使用 VFH+ 演算法，利用光學雷達所得到的資訊進行運算並增加巡航位置判斷方法，使其能夠完成室內巡航。

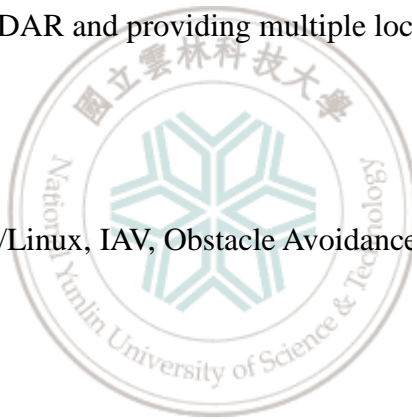
關鍵字：光學雷達、無人自走車、避障、VFH+



ABSTRACT

The thesis presents a method for the automatic interior cruise and obstacle avoidance of Intelligent Autonomous Vehicle (IAV) with the guidance of Light Detection and Ranging sensor (LiDAR). The main point of the thesis is to enhance the cruise speed and obstacle avoidance functions for IAV "Yun-Trip I" and develop its advanced version IAV "Yun-Trip II". The advanced "Yun-Trip II" features chassis from its predecessor, improve computer vision with a single LiDAR as core, adopted ARM-Based embedded system GNU/Linux, and perform cruising calculation and route locating with Microsoft Visual C++. The thesis adopt VFH+ obstacle avoidance algorithm, computing information collected by LiDAR and providing multiple locating methods to help complete interior cruise.

Keywords : LiDAR, GNU/Linux, IAV, Obstacle Avoidance , VFH+.



誌 謝

首先誠摯的感謝吳英正老師這兩年的指導，雖然只有短短兩年，但老師時常教導學生如何建立自我學習與解決問題的能力，使學生的學習方法在這兩年轉變很多。您不但指導我專業知識，更教導了我人生道路上應有的態度及目標，種種的都使學生獲益良多，於此由衷的表示感謝與敬意。

再來感謝實驗室的同學繼中、秉哲和俊誠，感謝你們為實驗室帶來歡樂的氣分，在研究上提供了許多不同的建議，也協助我的實驗測試，讓我的論文更加完整。

最後，感謝我最愛的家人，謝謝你們給予我最大的支持，讓我能夠全心的專注學習與研究，使我完成碩士學位。



目 錄

| | |
|-----------------------|------|
| 摘 要..... | i |
| ABSTRACT | ii |
| 誌 謝..... | iii |
| 目 錄..... | iv |
| 表 目 錄..... | vii |
| 圖 目 錄..... | viii |
| 第 1 章 緒論 | 1 |
| 1.1. 研究動機與目的 | 1 |
| 1.2. 文獻回顧 | 2 |
| 1.3. 研究方法 | 4 |
| 第 2 章 硬體架構介紹 | 5 |
| 2.1. 車體與硬體架構簡介 | 5 |
| 2.2. 運算核心 | 6 |
| 2.3. 感測元件 | 7 |
| 2.3.1. 掃描式雷射測距儀 | 7 |
| 2.4. 驅動元件 | 8 |
| 2.4.1. 轉向伺服機 | 8 |
| 2.4.2. 直流馬達驅動器 | 9 |
| 2.5. 電源供應系統 | 10 |
| 2.5.1. 動力系統 | 10 |
| 2.5.2. 運算電力系統 | 11 |
| 第 3 章 控制法則與路線規劃 | 12 |
| 3.1. 避障演算法 | 12 |

| | | |
|--------------|---|-----------|
| 3.1.1. | 極座標直方圖 | 12 |
| 3.1.2. | 安全空間 | 13 |
| 3.1.3. | 轉向限制 | 18 |
| 3.1.4. | 選擇轉向角度 | 20 |
| 3.2. | 速度演算法 | 22 |
| 3.2.1. | 障礙物密度 | 22 |
| 3.2.2. | 障礙接近率 | 22 |
| 3.2.3. | 碰撞預測 | 24 |
| 3.3. | 巡航路線及路線特徵判斷 | 26 |
| 3.3.1. | 巡航路線 | 26 |
| 3.3.2. | 車體座標 | 27 |
| 3.3.3. | 路線特徵判斷 | 27 |
| 3.3.3.1. | 自走車位於走廊位置 ($\epsilon < \epsilon_l$) | 28 |
| 3.3.3.2. | 自走車位於樓梯位置 ($\epsilon_l < \epsilon < \epsilon_h$) | 30 |
| 3.3.3.3. | 自走車位於轉角位置 ($\epsilon_h < \epsilon < 220^\circ$ or $\epsilon \approx 240^\circ$) | 31 |
| 3.3.3.4. | 自走車位於迴轉位置 | 33 |
| 3.4. | 程式流程圖 | 34 |
| 第 4 章 | 實驗結果與討論 | 35 |
| 4.1. | 實驗討論 | 35 |
| 4.1.1. | 自走車位於走廊位置 | 35 |
| 4.1.2. | 自走車位於轉角位置 | 35 |
| 4.2. | 實驗結果 | 36 |
| 4.2.1. | 自走車位於走廊位置之結果 | 36 |
| 4.2.2. | 自走車位於樓梯位置之結果 | 37 |
| 4.2.2.1. | 樓梯位置位於自走車左側 | 37 |
| 4.2.2.2. | 樓梯位置位於自走車右側 | 39 |

| | |
|---------------------------|-----------|
| 4.2.3. 自走車位於轉角位置之結果 | 41 |
| 4.2.3.1. 自走車左轉轉角位置 | 41 |
| 4.2.3.2. 自走車右轉轉角位置 | 42 |
| 第 5 章 結論與建議 | 43 |
| 5.1. 結論 | 43 |
| 5.2. 建議 | 43 |
| 參考文獻..... | 44 |



表 目 錄

| | |
|------------------------------------|----|
| 表 2-1：Raspberry Pi 規格 | 6 |
| 表 2-2：URG-04LX-UG01 規格..... | 7 |
| 表 2-3：Thunder Tiger DS1015 規格..... | 8 |
| 表 2-4：DC-DC LM2596 可調穩壓模組規格 | 10 |



圖 目 錄

| | |
|---|----|
| 圖 1-1：Yun-Trip 比較圖 | 2 |
| 圖 1-2：巡航路線圖 | 4 |
| 圖 2-1：Yun-Trip II 硬體架構圖 | 6 |
| 圖 2-2：Raspberry Pi 開發板 | 7 |
| 圖 2-3：HOKUYO URG-04LX-UG01 掃描式雷射測距儀 | 8 |
| 圖 2-4：Thunder Tiger DS1015 數位伺服機 | 9 |
| 圖 2-5：Pololu MD01B 直流馬達驅動器 | 9 |
| 圖 2-6：DC-DC LM2596 可調穩壓模組 | 10 |
| 圖 2-7：Kamera Neo-S7 pro 行動電源 | 11 |
| 圖 3-1：光學雷達量測示意圖 | 12 |
| 圖 3-2：極座標直方圖 | 13 |
| 圖 3-3：單值閥與雙值閥過濾結果 | 14 |
| 圖 3-4：雙向遲滯過濾結果 | 15 |
| 圖 3-5：遲滯過濾位元 OR 運算結果 | 16 |
| 圖 3-6：邊界縮減 | 17 |
| 圖 3-7：轉向角度限制 | 18 |
| 圖 3-8：偵測直方圖示意圖 | 19 |
| 圖 3-9：直方圖轉換 | 19 |
| 圖 3-10：邊界分類 | 20 |
| 圖 3-11：障礙物接近率示意圖 | 23 |
| 圖 3-12：碰撞預測示意圖 | 25 |
| 圖 3-13：巡航路線位置圖 | 26 |
| 圖 3-14：車體座標定義 | 27 |
| 圖 3-15：走廊位置地圖 | 28 |
| 圖 3-16：走廊位置圖 | 29 |
| 圖 3-17：樓梯位置圖 | 30 |
| 圖 3-18：左轉轉角路線圖 | 31 |
| 圖 3-19：右轉轉角路線圖 | 32 |
| 圖 3-20：迴轉路線圖 | 33 |
| 圖 3-21：程式流程圖 | 34 |
| 圖 4-1：走廊位置實驗結果 | 36 |
| 圖 4-2：樓梯位於自走車左側實驗環境 | 37 |
| 圖 4-3：樓梯位置補償實驗結果 | 38 |
| 圖 4-4：樓梯位於自走車右側實驗環境 | 39 |

| | |
|-----------------------|----|
| 圖 4-5：樓梯位置補償實驗結果..... | 40 |
| 圖 4-6：左轉轉角位置實驗結果..... | 41 |
| 圖 4-7：右轉轉角位置實驗結果..... | 42 |



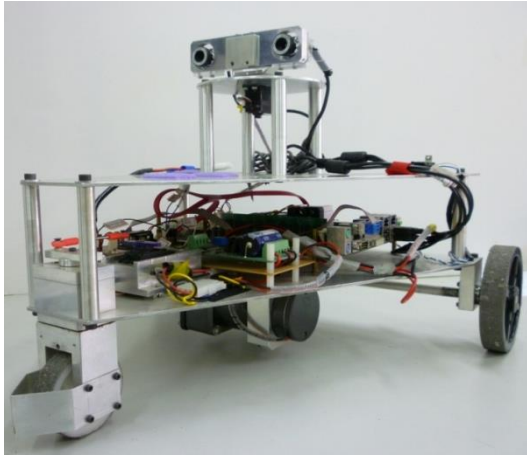
第1章 緒論

1.1. 研究動機與目的

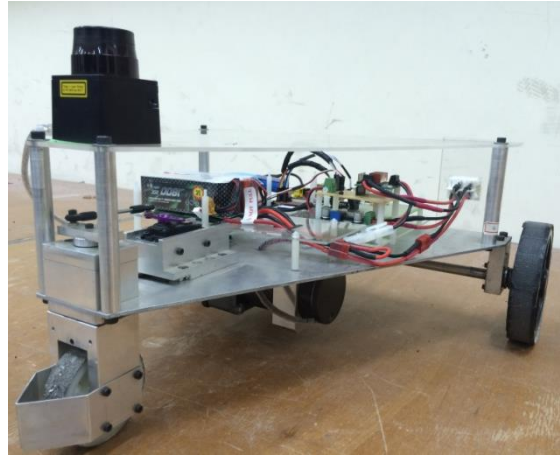
近年來美國國防部高等研究計劃署 (Defense Advanced Research Projects Agency, DARPA)所舉辦的無人車輛競速大賽(DARPA Grand Challenge)場地是在一個充滿砂礫的戈壁中，且無任何建築，2003 年無人能完成任務，2005 年由史丹佛大學拿到冠軍，因為比賽場地道路崎嶇，所以使用五顆雷達掃描前方路面，並使用六顆 Pentium M 處理器搭配 GUN/Linux 作業系統做開發，2007 年比賽場地更改在市區內，參賽隊伍的自主式車輛必須能分辨道路上的停止線、斑馬線及交通號誌等等。而 Google 公司在 2011 年時也發表了自行研發的無人自走車，人工智慧的發展讓自主式無人載具成為目前廣為研究的課題，而基於這些影響，本實驗室自 2005 年也開始投入研究無人自走車的領域。

本實驗室目前已開發一台 Yun-Trip 自走車，如圖 1-1(a)所示，使用工業用電腦主機板搭配 Windows XP 作業系統作為決策中心，加上電腦視覺所需要的攝影機，其重量較重且影像處理速度所需時間較長，造成巡航速度降低與耗電量增加。而本論文使用原先 Yun-Trip 的車體底盤開發了第二代實驗平台 Yun-Trip II，如圖 1-1 (b)所示，其使用單顆光學雷達 (LiDAR) 取代電腦視覺，以嵌入式單板電腦搭配 GNU/Linux 作業系統，並且簡化了所需要的週邊配備，使得體積與重量大幅降低。

本論文所開發的自主式車輛 Yun-Trip II 實驗平台，使用光學雷達(LiDAR)量測環境的變化，同時規劃出安全的路線以避開障礙物，避免碰撞。



(a)Yun-Trip



(b)Yun-Trip II

圖 1-1：Yun-Trip 比較圖

1.2. 文獻回顧

在自走車的相關研究領域中，使用電腦視覺的自走車大多都使用工業用電腦主機板搭配 Windows XP 作業系統，原因為電腦視覺的運算量龐大，必須使用運算速度較快的工業用主機板；本論文所開發的 Yun-Trip II 使用單顆光學雷達（LiDAR）來取代電腦視覺。由於所需運算量不再龐大，進而將原本的工業主機板更換成嵌入式單板電腦，並搭配 GNU/Linux 作業系統。

陳依璟(2012,[8]) 使用光學雷達(LiDAR,掃描式雷射測距儀)量測周遭環境，來得知自走車巡航的周遭環境情形，做出室內定位，同時使用超音波偵測障礙物及避障；本論文實驗地點為走道走廊，可利用光學雷達（LiDAR）來得知周遭環境，由於使用光學雷達（LiDAR）搭配超音波來判別障礙物，會使功能重複，進而增加系統及硬體負擔。

劉忻武(2014,[4])，使用 Vector Field Histogram Plus (VFH+ (1998,[5]))演算法，將光學雷達 (LiDAR) 量測到的資訊建立一極座標直方圖 (Polar Histogram)，並計算出機器人可通過的安全空間，進而避開障礙物，其中也改善了 VFH+遲滯過濾 (Hysteresis)效果與修正障礙物誤判等功能；由於本論文自走車巡航時必須避開路線上之障礙物，適合使用改善後的 VFH+演算法作為避障演算法，並與室內環境定位之功能做整合，完成室內巡航。

本實驗室所開發的自主式車輛 Yun-Trip 實驗平台，是由顏大翔(2008,[1])從第一代 Yun-Trip 車體的概念初期的設計及加工著手，在車體上方裝載雙軸的雙 CCD 攝影機構，使用 DirectShow 撰寫影像處理程式，擷取日光燈罩做為特徵物，並完成夜間巡航，由於白天光線的影響造成環境過亮，導致使影像不易擷取出環境的特徵，黃平雲(2012,[2])使用動態調整曝光度控制進光量，加以改善先前的環境過亮的問題，接著由詹鈞博(2013,[3])使用五顆超音波，判斷路線上的障礙物，其避障功能較繁瑣複雜；本論文使用單顆光學雷達 (LiDAR) 取代電腦視覺與超音波測距儀，並整合室內定位與避障功能，使機器人完成巡航。

1.3. 研究方法

Yun-Trip II 必須完成室內巡航，並且避開巡航路線上的障礙物，本論文使用光學雷達（LiDAR）量測機器人巡航時的環境變化，並在無障礙物情況下巡航於走廊中央；巡航路線由起點 331 巡航至 310 後折返回原處，巡航路線圖如圖 1-2 所示。

在前往 310 的過程中，會遇到四種特徵位置，走廊位置(圖 1-2 虛線路線)、樓梯位置(圖 1-2 (a)處)、轉彎位置(圖 1-2 (b)處)與迴轉位置(圖 1-2 (c)處)，可由這四個特徵得知巡航位置。

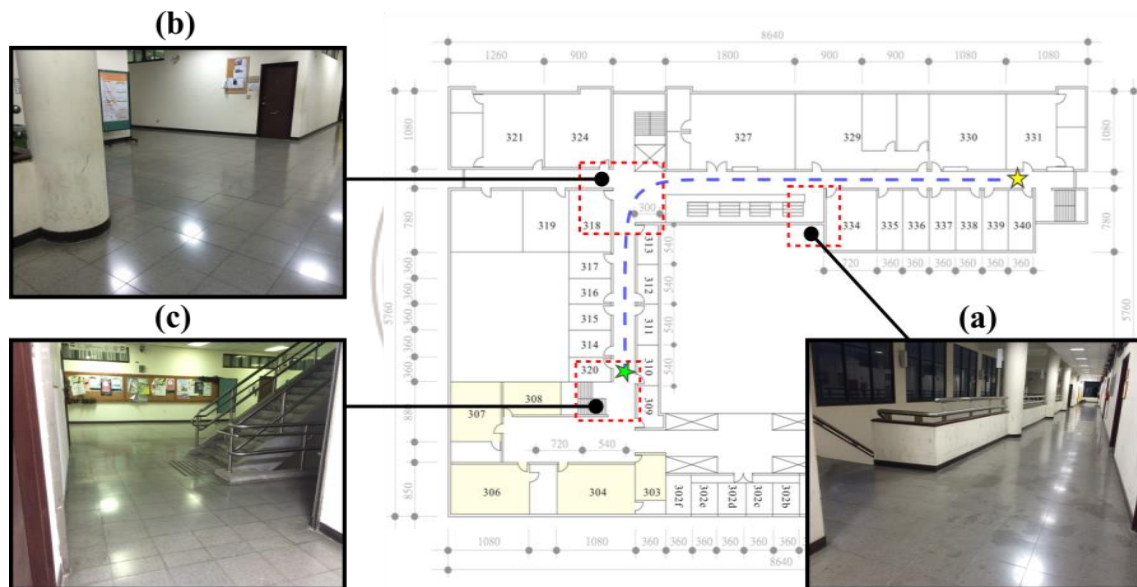


圖 1-2：巡航路線圖

第2章 硬體架構介紹

2.1. 車體與硬體架構簡介

本論文所開發的 Yun-Trip II 車體是汽車以底盤架構作為雛型並加以修改成三輪車體架構。三輪車體架構可使自走車擁有較小的迴轉半徑，Yun-Trip II 後輪間距為 48 公分，同時也是最小迴輪半徑，靈活度較使用四輪架構的車體好，機構設計也較簡單。實驗環境主要為室內，較少有凹凸地面，所以採用三輪車體架構作為車體，除了車體架構相對簡單外，為了減低控制的複雜度，在後輪間加裝一差速器，差速器內部為一行星齒輪機構，此機構設計使自走車在轉彎時，允許內側輪與外側輪擁有不同的轉速，可以避免在轉向時產生滑移現象，造成失控，並使用皮帶輪與皮帶轉動差速器，進而驅動後輪，前輪則使用一顆伺服馬達控制轉向機構。

Yun-Trip II 使用開源開發平台 Raspberry Pi 與 GNU/Linux 作業系統作為運算核心。而 Yun-Trip II 的動力來源為一直流馬達，並由前方伺服馬達控制轉向。由於 Raspberry Pi 開發板可直接輸出 PWM 訊號，因此不需要伺服馬達控制器便可直接控制伺服馬達，另外將輸出的 PWM 訊號利用馬達驅動器放大後也可直接控制直流馬達。Raspberry Pi 也可直接輸出 3.3V 與 5V 準位的 IO 訊號，因此另外裝置了 2 顆白光 LED 做為狀態指示燈，以顯示車輛運作狀況。環境感測器使用 HOKUYO 公司所生產的 URG-04LX-UG01 光學雷達(LiDAR，掃描式雷射測距儀)，使用 USB 介面與運算核心連接，硬體架構如下圖 2-1 所示。

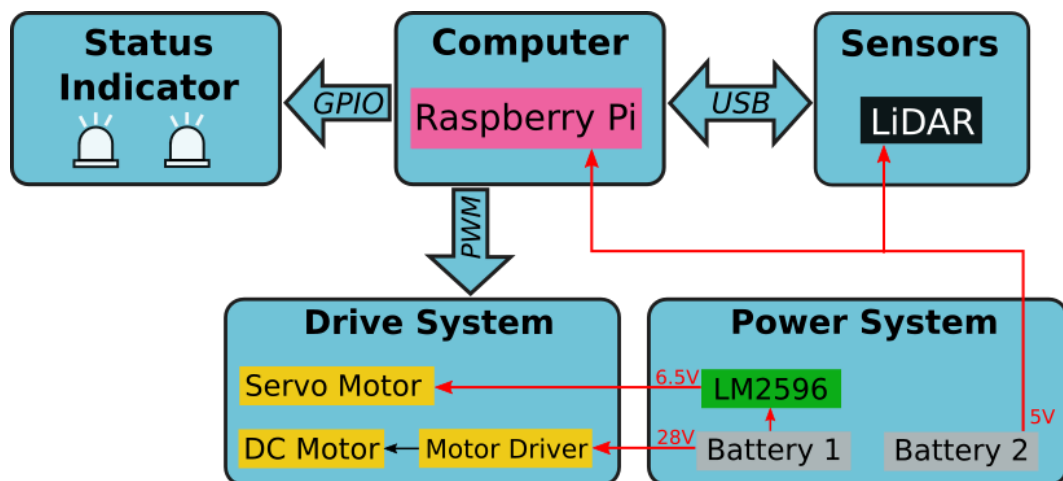


圖 2-1：Yun-Trip II 硬體架構圖

2.2. 運算核心

為了減輕重量，Yun-Trip II 使用 Raspberry Pi 開發板取代原先的工業用主機板作為運算核心。Raspberry Pi 為開放原始碼之開發平台，Raspberry Pi 具有低耗功、體積小、成本低等優點，其規格如表 2-1 所示，外觀如圖 1-2：巡航路線圖所示。

表 2-1：Raspberry Pi 規格

| | |
|---------|----------------------------------|
| CPU | 700MHz single-core ARM11176JZF-S |
| Memory | 512MB |
| Storage | SD Card |
| IO | 2 x 13 Pin Header |
| OS | GNU/Linux |

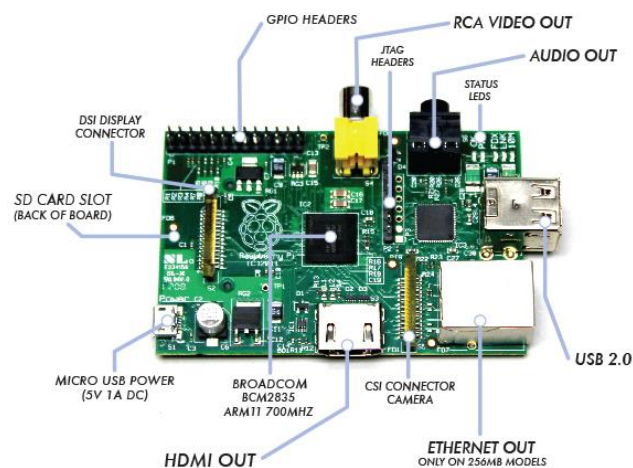


圖 2-2：Raspberry Pi 開發板

來源：www.pcmag.com

2.3. 感測元件

2.3.1. 掃描式雷射測距儀

Yun-Trip II 使用 HOKUYO 公司所生產的 URG-04LX-UG01 掃描式雷射測距儀，又稱光學雷達（LiDAR），來量測週遭環境的變化，並規劃路線避開障礙物。其規格如下表 2-2 所示，外觀如下圖 2-3 所示。

表 2-2：URG-04LX-UG01 規格

| | |
|-------|------------------------------------|
| 光源 | 半導體雷射 ($\lambda = 785\text{nm}$) |
| 輸入電壓 | 5V DC \pm 5% (USB Power) |
| 輸入電流 | 500mA （最大 800mA） |
| 量測距離 | 20mm ~ 4000mm |
| 距離解析度 | 1mm |
| 掃描範圍 | $\pm 120^\circ$ |
| 角度解析度 | 0.36° |
| 取樣頻率 | 10Hz |



圖 2-3：HOKUYO URG-04LX-UG01 掃描式雷射測距儀

來源：www.hokuyo-auti.jp

2.4. 驅動元件

2.4.1. 轉向伺服機

伺服馬達採用 Thunder Tiger 公司所生產的 DS1015 數位伺服機構做為轉向之動力來源。其規格如下表 2-3：Thunder Tiger DS1015 規格所示，外觀如下圖 2-4 所示。

表 2-3：Thunder Tiger DS1015 規格

| | |
|-----------|------------------------|
| 電壓範圍 | 4.8V ~ 6V |
| 扭力@ 6V | 14.5kg-cm |
| 重量 | 66g |
| 速度@ 6V | 0.108sen/60° |
| 尺寸（長×寬×高） | 41.8mm × 20.6 × 39.6mm |
| 角度範圍 | 180° |

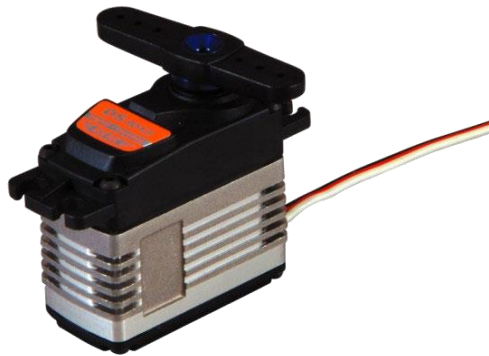


圖 2-4：Thunder Tiger DS1015 數位伺服機

來源：Thunder Tiger 官方網站

2.4.2. 直流馬達驅動器

由於 Raspberry Pi 輸出的 PWM 訊號為 5V 準位，且最大電流約只有 6mA，無法直接驅動直流馬達，所以使用 Pololu 公司生產 MD01B 直流馬達驅動器將訊號放大，其外觀如下圖 2-5 所示。

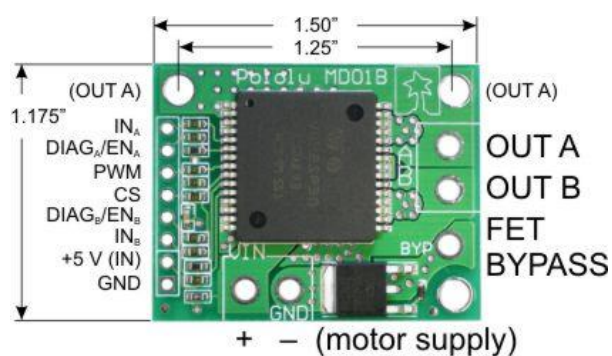


圖 2-5：Pololu MD01B 直流馬達驅動器

來源：Pololu 官方網站

2.5. 電源供應系統

2.5.1. 動力系統

Yun-Trip II 使用 28V 的鋰電池做為動力系統電力來源（直流馬達與伺服機），其中伺服機需要 6V 的電壓源，所以當中使用飆機器人公司所生產的 DC-DC LM2596 可調穩壓模組來供應不同的電壓源。其規格如下表 2-4 所示，外觀如下圖 2-6 所示。

表 2-4：DC-DC LM2596 可調穩壓模組規格

| | |
|------|------------|
| 輸入電壓 | 4 ~ 40V |
| 輸出電壓 | 1.25 ~ 37V |



圖 2-6：DC-DC LM2596 可調穩壓模組

來源：飆機器人官方網站

2.5.2. 運算電力系統

由於 Raspberry Pi 電源輸入端是 micro USB 介面，所以本論文使用 Kamera 公司所生產的 Neo-S7 pro 行動電源，來做為運算核心的電力來源。其外觀如下圖 2-7 所示。



圖 2-7：Kamera Neo-S7 pro 行動電源

來源：Kamera 官方網站

第3章 控制法則與路線規劃

本章先介紹避障時所使用的演算法，VFH+避障演算法、速度演算法與碰撞預測，由以上三種演算法計出算最佳轉向與偵測機器人巡航時之速度及機器人是否即將發生碰撞，最後在介紹巡航路線及路線特徵判斷方法。

3.1. 避障演算法

VFH+ 使用四個階段計算逐一減少資訊量並找出最佳轉向角度 β_t ，前三個階段著重於機器人的拘束找出可通過的方向，第四階段為計算最佳轉向。

3.1.1. 極座標直方圖

先由光學雷達所量測到的資訊以 d_i 與 θ_i 表示， d_i 為 i 個量測到的距離， θ_i 則為 d_i 所對應之角度，如下圖 3-1 所示。

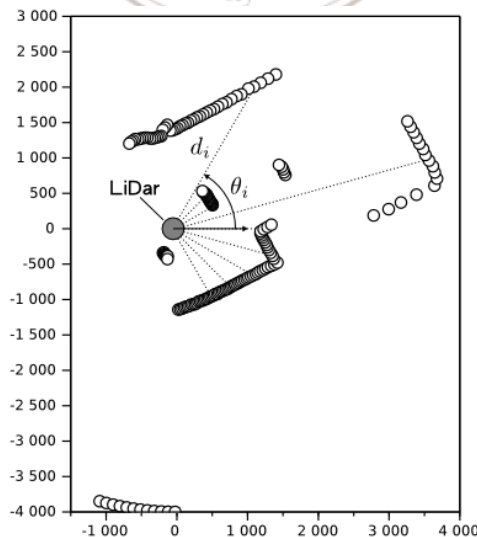
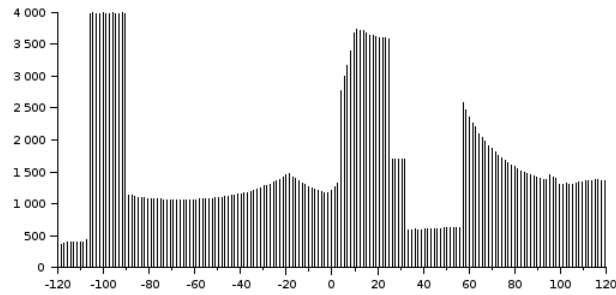


圖 3-1：光學雷達量測示意圖

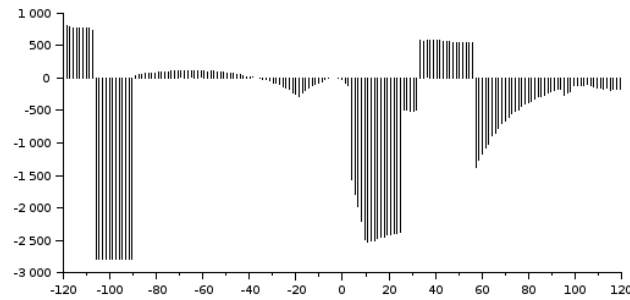
接著使用光學雷達所得到的資訊建立一極座標直方圖 P_i ：

$$P_i = a - b \cdot d_i \quad 3-1$$

a 與 b 皆為正值，使用者可調整 a 與 b 各值，來調整出 VFH+所要偵測與計算的範圍。以下圖 3-2 為例子 $a = 1200$ 、 $b = 1$ 的設定下，所量測到之環境產生的 d_i 與 P_i 示意圖。



(a) d_i 示意圖



(b) P_i 示意圖

圖 3-2：極座標直方圖

3.1.2. 安全空間

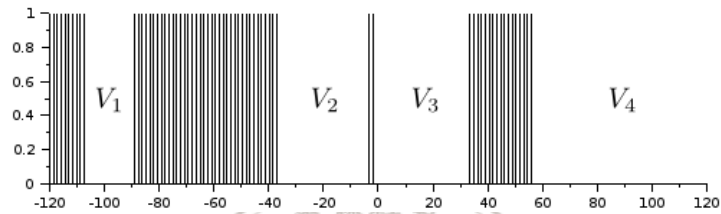
而一段連續的安全距離就代表一個可通過的安全空間 V_j ，如下圖 3-3 所示，由一組邊界向量 $(B_L, B_R)_j$ 定義，定義其左邊界與右邊界的角度 θ 與距離 d ，視為機器

人所能通過的方向：

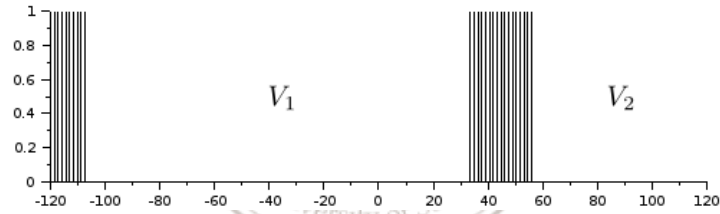
$$B_L = [\theta_l, d_l] \quad 3-2$$

$$B_R = [\theta_r, d_r] \quad 3-3$$

然而只使用單一閾值可能會過濾出許多不連續的空間，造成許多不必要的方向選擇出現，讓機器人在決定方向時產生左右搖擺的現象。因此，VFH+使用雙值閾 τ_{max} 和 τ_{min} 與雙向遲滯(Hysteresis)效果過濾掉不必要的空間。



(a) 單值閾過濾結果



(b) 雙值閾過濾結果

圖 3-3：單值閾與雙值閾過濾結果

以下圖 3-4 為範例，設定 $\tau_{min} = 0, \tau_{max} = 450$ ，接著在由直方圖的左側至右側進行計算，從 P_i 中得到二元直方圖(Binary Histogram) H_i ，如式 3-4 所示，計算出 H_i ，如圖 3-4(a) 所示。

$$H_i = \begin{cases} 1 & \text{if } P_i \geq \tau_{max} \\ 0 & \text{if } P_i \leq \tau_{min} \\ H_{i-1} & \text{otherwise} \end{cases} \quad 3-4$$

再由右側至左側進行相同計算如式 3-5 所示，計算出 H'_i ，過濾出所有視為安全的距離，如圖 3-4(b) 所示。

$$H'_i = \begin{cases} 1 & \text{if } P_i \geq \tau_{max} \\ 0 & \text{if } P_i \leq \tau_{min} \\ H'_{i+1} & \text{otherwise} \end{cases} \quad 3-5$$

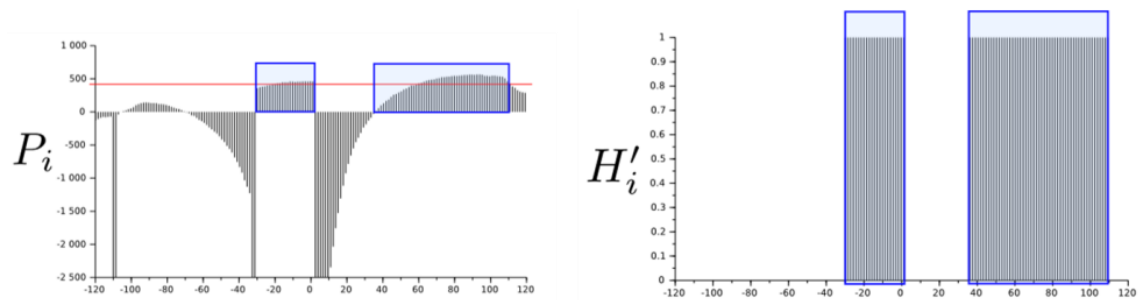
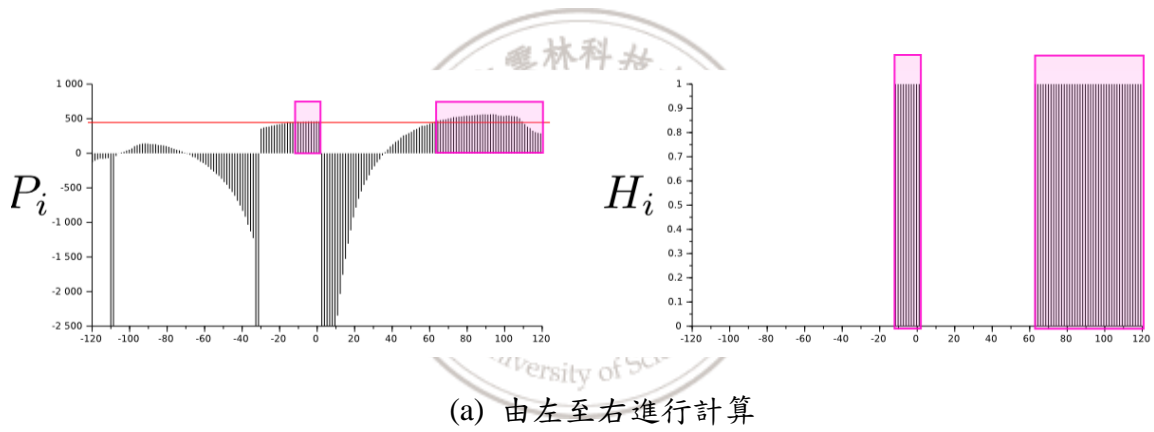


圖 3-4：雙向遲滯過濾結果

接著將 H_i 與 H'_i 做位元 OR 運算，結合兩個方向的過濾結果，找出正確的安全空間 H''_i ，如式 3-6 所示，而結果如圖 3-5 所示。

$$H''_i = H'_i \text{OR} H_i \quad 3-6$$

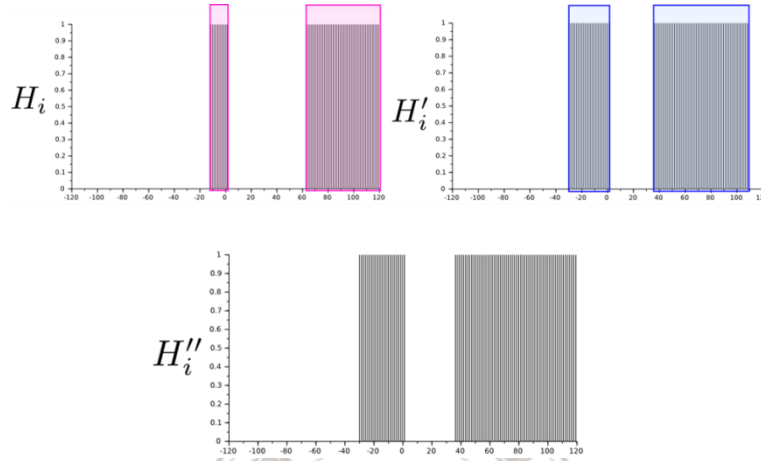


圖 3-5：遲滯過濾位元 OR 運算結果

為了考慮機器人本身的幾何尺寸，使用縮小安全空間 V_j 邊界的方式來增加幾何大小，此時便可將機器人視為一質點。假設機器人之尺寸為半徑 ω_s 的圓，則將 V_j 之邊界同樣縮小 ω_s 後便可將機器人視為一點，如下圖 3-6 所示。縮減後的安全空間 $\hat{V}_j = (\hat{B}_L, \hat{B}_R)_j$ 可由 3-7 式和 3-8 式計算：

$$\hat{B}_L = [\theta_l - \delta_l \quad d_l \cos \delta_l], \quad \delta_l = \sin^{-1} \left(\frac{\omega_s}{d_l} \right) \quad 3-7$$

$$\hat{B}_R = [\theta_r - \delta_r \quad d_r \cos \delta_r], \quad \delta_r = \sin^{-1} \left(\frac{\omega_s}{d_r} \right) \quad 3-8$$

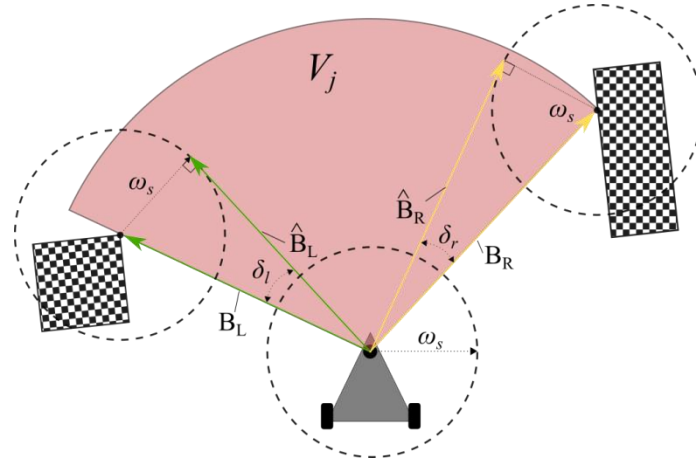


圖 3-6：邊界縮減



3.1.3. 轉向限制

使用機器人的最小迴轉半徑 R_s 與機器人尺寸 ω_s ，計算出被障礙物影響的轉向角度限制 (ϕ_l, ϕ_r) ，除去無法達成的轉向角度。將迴轉半徑 R_s 加入機器人尺寸 ω_s 計算，形成與最小迴轉半徑之行進軌跡同心但半徑為 $R_s + \omega_s$ 的圓，如下圖 3-7 所示。

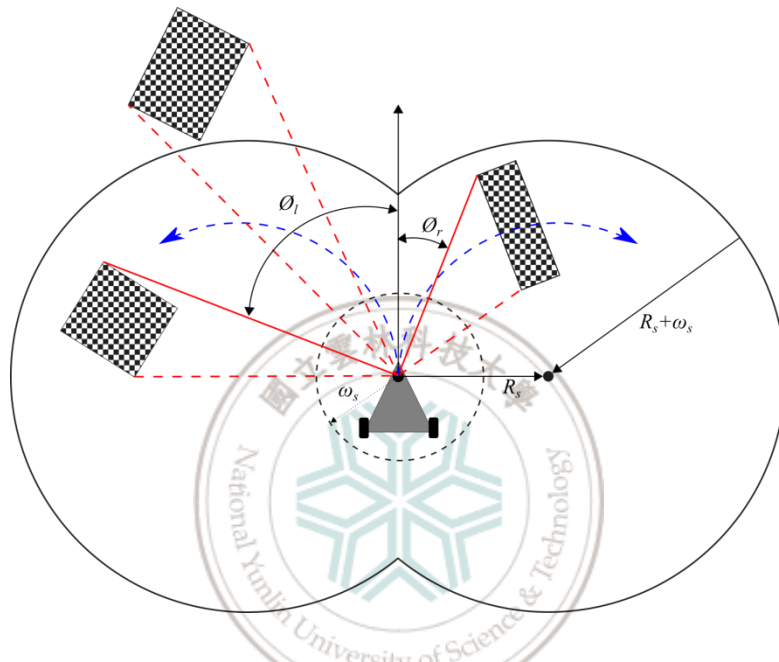


圖 3-7：轉向角度限制

接著找出 (ϕ_l, ϕ_r) ，使用與光學雷達相同的 θ_i 建立一偵測用的直方圖 D_i ，偵測障礙物是否阻擋了轉向角度，如下圖 3-8 所示。其可由 3-9 式計算。

$$D_i = |R_s \sin \theta_i| + \sqrt{R_s^2 \sin^2 \theta_i + \omega_s^2 + 2R_s \omega_s} \quad 3-9$$

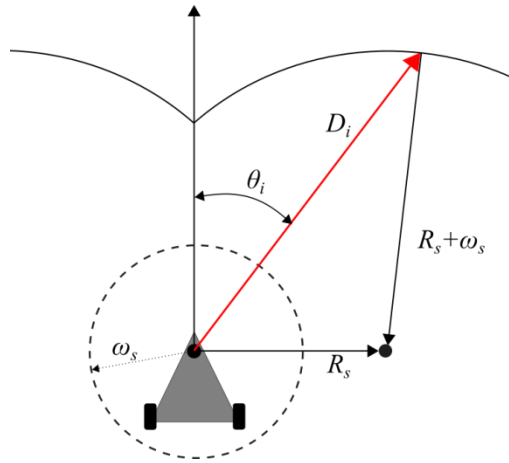


圖 3-8：偵測直方圖示意圖

接著將光學雷達測得的 d_i 與 D_i 相減，得到遮罩直方圖 M_i ：

$$M_i = d_i - D_i$$

3-10

若 M_i 的值小於 0，代表 θ_i 方向有障礙物位於機器人的最小迴轉範圍內，此時必須改變 (ϕ_l, ϕ_r) 限制轉向角度，以排除無法達成的轉向方向。環境與這些直方圖之間的轉換如下圖 3-9 所示。

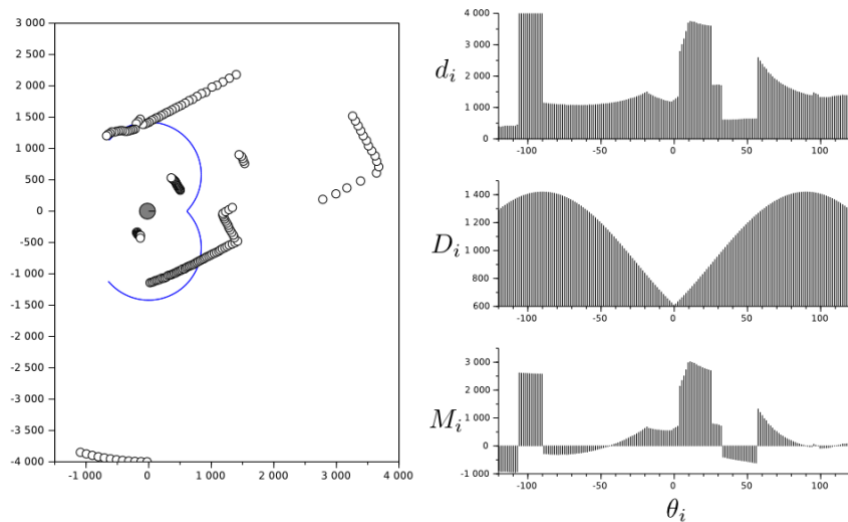


圖 3-9：直方圖轉換

利用 M_i , (ϕ_l, ϕ_r) 可以非常快速的被計算出來：

1. 首先設 $\phi_l = \pi$ 、 $\phi_r = -\pi$
2. 對所有 i ，若 $M_i < 0$ ：
 - (a) 若 $\theta_i < 0$ 且 $\theta_i > \phi_r$ ，則將 ϕ_r 設為 θ_i
 - (b) 若 $\theta_i > 0$ 且 $\theta_i > \phi_l$ ，則將 ϕ_l 設為 θ_i

3.1.4. 選擇轉向角度

根據安全空間 \hat{V}_j 的寬度，可以從每個 \hat{V}_j 中找到一至兩個候選方向 β ，接著使用成本函數於這些候選方向中找出最佳解。對於安全空間的寬度，本論文使用邊界向量的角度差 $\epsilon = \theta_l - \theta_r$ 將安全空間分為兩種情況：交疊與未交疊，如下圖 3-10 所示。

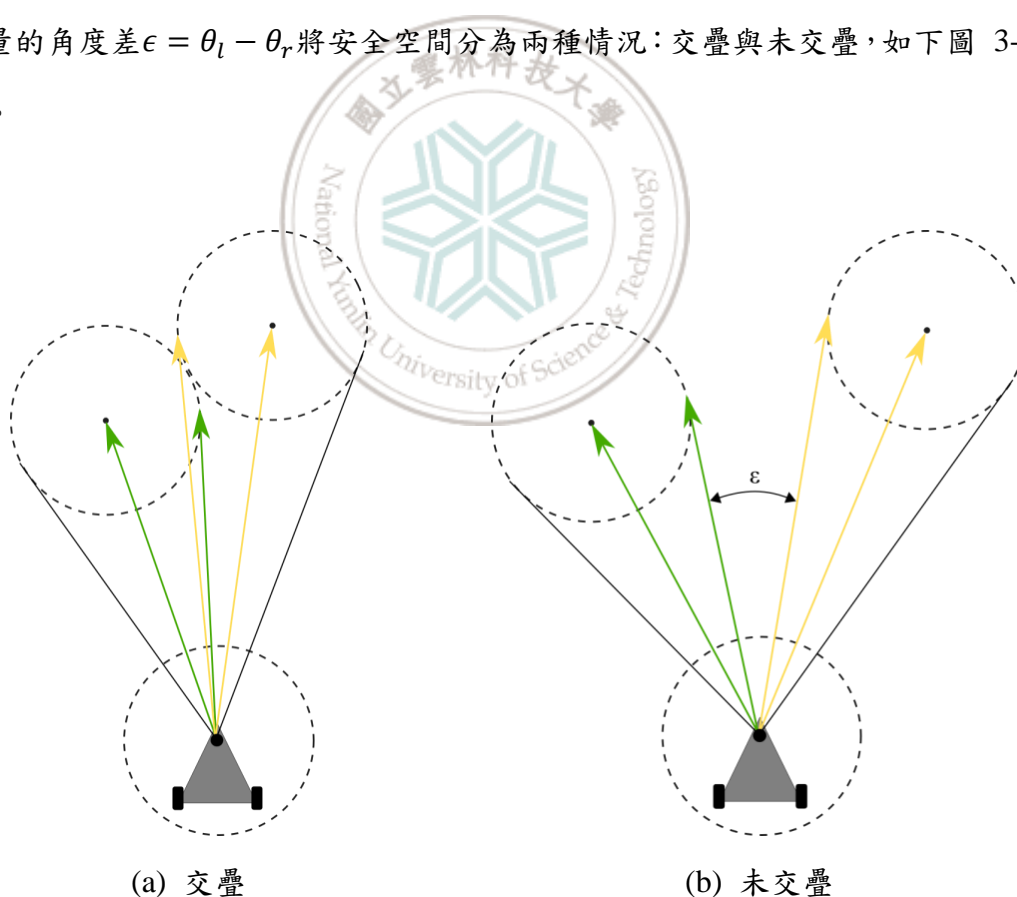


圖 3-10：邊界分類

由於光學雷達在量測時是逆時針方向從 -120° 掃描至 120° ，在搜尋安全空間時也是同一方向，因此每個安全空間的右邊界角度一定小於左邊界角度。因此，若是右邊界角度大於左邊界代表發生了邊界交疊，此時該安全空間將會被捨棄，不會產生任何候選方向，如圖 3-10(a)所示。

圖 3-10(b)為可通過的安全空間，此時將正中央視為候選方向：

$$\beta_n = \frac{\theta_l + \theta_r}{2} \quad 3-11$$

最後使用成本函數 G 於這些候選轉向角 β 中找出最佳解 β_t ：

$$G(\beta) = \mu_1 \cdot (|\beta|) + \mu_2 \cdot (|\beta - \beta_{t-1}|) \quad 3-12$$

在 3-12 式中， $(|\beta|)$ 項代表候選方向與目前車輛方向的差距。這些候選角都是以車身座標系做為參考，所以車輛本身方位角將永遠是 0° ，而候選方向越大代表機器人將會偏離目前的方向越多，進而增加成本。

$(|\beta - \beta_{t-1}|)$ 項代表候選角與前一次選擇的最佳轉向 β_{t-1} 之間的差距。這個差距越大代表機器人的越容易偏離目前的航向，造成擺盪的現象。 μ_1, μ_2 則是相對應的權重係數，藉由調整這兩個係數之間的相對大小，機器人的巡航特性就能夠被調整，由成本函數中產生最小值的候選方向即為最佳方向 β_t 。

3.2. 速度演算法

3.2.1. 障礙物密度

VFH+ 使用障礙物密度 Borenstein and Koren (1991,[7]) 來計算機器人於該環境狀態下的速度。此密度可使用光學雷達的量測結果 d_i 與最大量測距離 d_{max} 計算，以一函數 $D(d_i)$ 表示：

$$D(d_i) = 1 - \frac{1}{N} \sum_{i=1}^N \frac{d_i}{d_{max}} \quad 3-13$$

此函數所計算之值將會落在 0 與 1 之間，越接近 1 代表環境中的障礙物距離越近或數量較多(密度較高)；反之則代表障礙物距離越遠或數量較少(密度較低)。因此，若設定機器人之最大速度 v_{max} 與最小速度 v_{min} ，其速度 v 可依下式計算：

$$v = v_{min} + (1 - D(d_i)) \cdot (v_{max} - v_{min}) \quad 3-14$$

雖然使用障礙物密度來計算速度是相當有效的方式，但此密度僅使用當時所處的環境進行計算，並沒有考慮機器人當下的速度。然而在進行路線規劃時，若是機器人的速度過快，則可能因反應不及無法迴避，而造成碰撞。因此機器人當下的速度也是速度控制中，一個相當重要的因素。

3.2.2. 障礙接近率

障礙物接近率(Obstacle Approaching Rate) δ ，其概念為假設機器人位於一靜態環境中(障礙物的位置不隨時間而改變)，則若是量測到的環境變化相當大，代表機器人的速度較高；反之則代表速度較低。而環境變化又以正前方的變化最為重要，因為這代表機器人接近正前方障礙物的速度，如下圖 3-11 所示。因此，設定一角度範圍 θ_r ，例如正前方左右各 20° ，對所有位於此範圍內的光學雷達量測值 d_j ，其接近率 δ 可依下式計算：

$$\delta = -\frac{1}{M} \sum_j \frac{(d_j)_t - (d_j)_{t-1}}{T} \quad 3-15$$

其中 $(d_j)_t$ 代表當下的環境量測值； $(d_j)_{t-1}$ 代表上一次的環境量測值； T 代表兩次量測的時間間隔； M 代表範圍內的量測值個數。

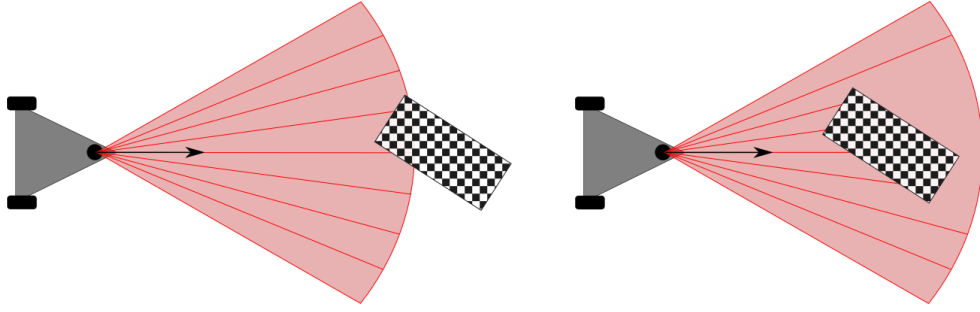


圖 3-11：障礙物接近率示意圖

巡航時，最重要的是以高速接近障礙物時能夠成功的減速，對加速的要求則相對較低。因此，在計算環境變化率時，只將變化率 $d_t - d_{t-1}$ 為負值(障礙物接近機器人)的方向納入計算，將變化率為正值的排除在外，得 δ_a ：

$$\delta_a = -\frac{1}{M} \sum_j \frac{\Delta((d_j)_t - (d_j)_{t-1})}{T} \quad 3-16$$

其中

$$\Delta(d) = \begin{cases} d & \text{if } d < 0 \\ 0 & \text{otherwise} \end{cases} \quad 3-17$$

為了讓障礙物接近率與環境密度能夠一起計算，必須讓障礙物接近率的範圍落在 0 與 1 之間。因此，將上式算出的 δ_a 除以機器人的最大速度 v_{max} 便可得到落在 0 與 1 之間的障礙物接近率 δ_n ：

$$\delta_n = -\frac{1}{M \cdot v_{max}} \sum_j \frac{\Delta((d_j)_t - (d_j)_{t-1})}{T} \quad 3-18$$

δ_n 之值越接近 1 代表障礙物以較高的速度接近機器人；反之則代表障礙物以較低的速度接近。

最後，將 δ_n 列入考慮後，速度 v 可依下式計算：

$$v = v_{min} + (1 - D(d_i) + \delta_n) \cdot (v_{max} - v_{min}) \quad 3-19$$

為了達成平滑的速度輸出， $D(d_i) + \delta_n$ 之值會被限制在 1 以下。

3.2.3. 碰撞預測

為了避免機器人在巡航時發生碰撞，造成損壞，因此使用碰撞預測(Collision Prediction) 做為巡航停止與否的指標，在巡航過程中不斷偵測碰撞是否即將發生，進而控制機器人的行動。

碰撞預測可分為兩階段：第一階段使用設定的機器人尺寸，偵測障礙物是否已經發生碰撞；第二階段是偵測機器人的行進方向是否有障礙物存在，偵測碰撞是否即將發生。只要其中一階段偵測到碰撞即代表碰撞發生，機器人將會停止前進。

第一階段：為了安全起見，設定機器人尺寸時，通常會讓設定的尺寸比真實尺寸還大。因此，第一階段的「已經發生碰撞」實際上代表的是，障礙物已經進入了設定的尺寸範圍內，而非真的發生了碰撞。而由於在計算時已經將機器人的尺寸簡化為半徑為 ω_s 的圓形，因此只要判斷光學雷達的量測值 d_i 是否小於 ω_s ，即可判定碰撞是否發生，如圖 3-12(a)所示。

第二階段：若單純只使用機器人尺寸來偵測碰撞，在機器人速度較快時可能無法及時停止，讓機器人與障礙物實際發生碰撞，造成損壞。因此，第二階段設定一距離 d_c ，利用車身尺寸 ω_s 計算出一角度 $\theta_c = \tan^{-1} \frac{\omega_s}{d_c}$ ，預先偵測行進路線上的障礙物，如圖 3-12(b)所示。

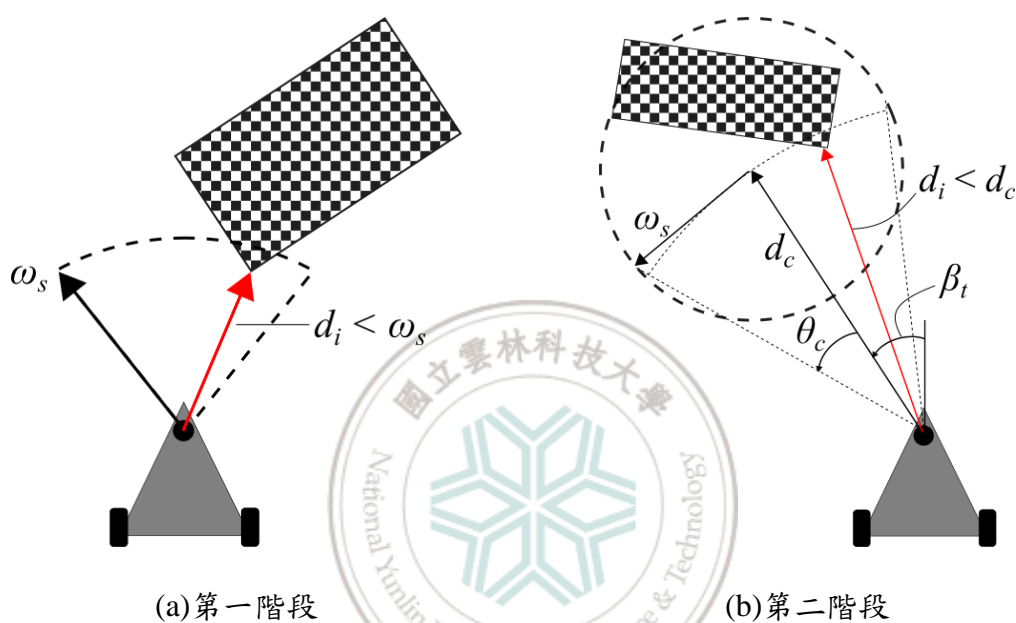


圖 3-12：碰撞預測示意圖

得到 θ_c 後，利用轉向方向 β_t 可得到一角度範圍 $(\beta_t + \theta_c, \beta_t - \theta_c)$ 。在此角度範圍內的光學雷達量測值 d_i 若小於預先設定的距離 d_c ，如圖 3-12(b)所示，代表即將發生碰撞。

若是在無候選方向的情況下未偵測到碰撞，則使用上一次計算得到的目標方向 β_{t-1} 做為轉向角度，並將速度設定為最低速度，直到再次計算出 β_t 或偵測到碰撞。

3.3. 巡航路線及路線特徵判斷

3.3.1. 巡航路線

巡航路線由起點 331 巡航至 310 後折返回原處，巡航路線圖如下圖 3-13 所示。在前往 310 的過程中，有四種情況分為走廊位置（如下圖 3-13 虛線路線）、樓梯位置（如下圖 3-13 (a)處）、轉角位置（如下圖 3-13 (b)處）以及迴轉位置（如下圖 3-13 (c)處），巡航過程中會使用兩角度閾值與邊界向量的角度大小判斷機器人位置，並完成巡航。

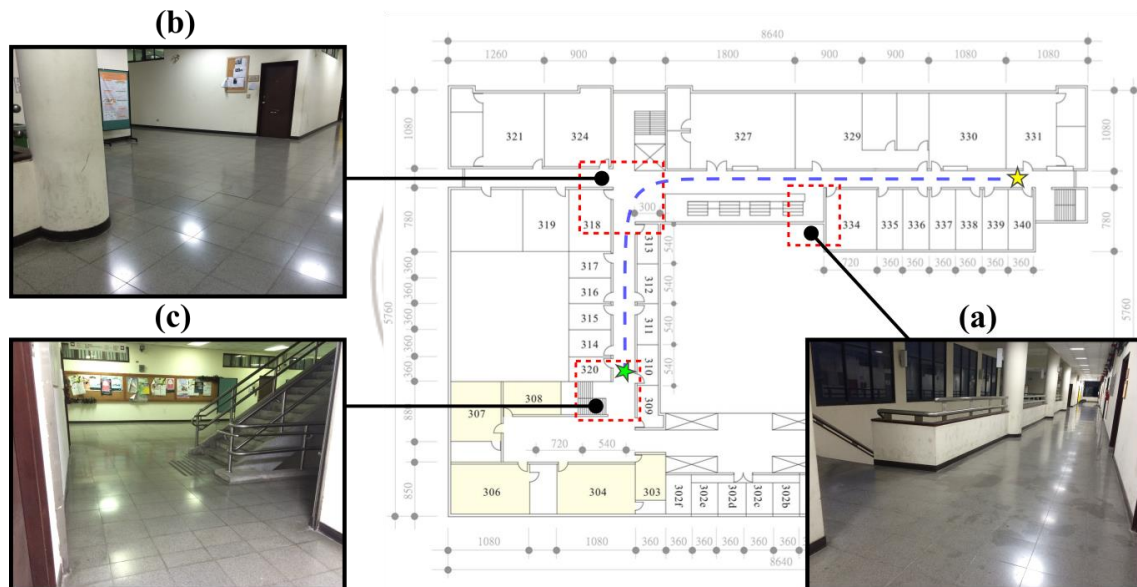


圖 3-13：巡航路線位置圖

3.3.2. 車體座標

車體座標是依照光學雷達掃描範圍擬定，如下圖 3-14 所示，光學雷達角度 0° 位置為車體前方的正中央，角度負值為車體右方 (Right)，角度正值為車體左方 (Left)， \hat{V}_j 、 \hat{B}_L 和 \hat{B}_R 為前一小節介紹的安全空間與邊界向量， ϵ 為車體巡航時邊界向量的角度差， β_t 為候選方向，候選角方向為兩邊界的正中央。

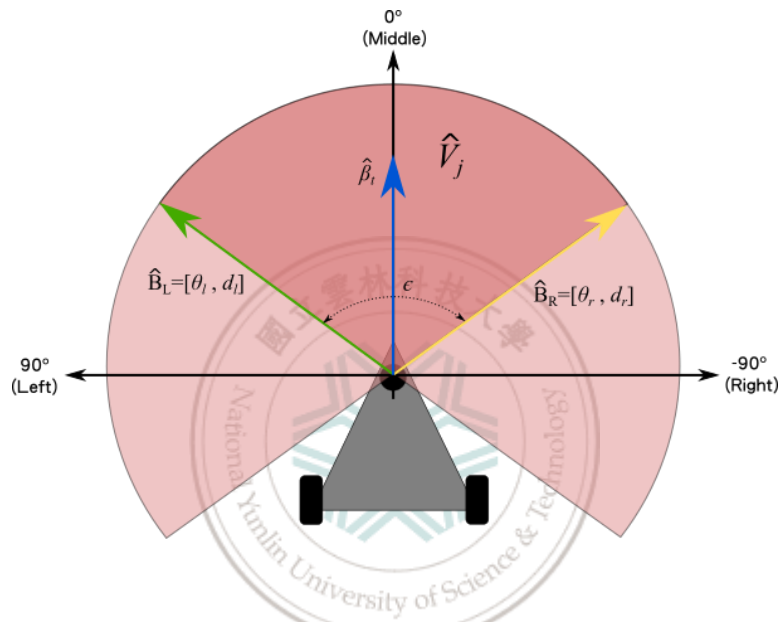
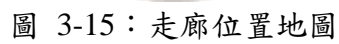


圖 3-14：車體座標定義

3.3.3. 路線特徵判斷

本論文使用邊界向量的角度差 $\epsilon = \theta_l - \theta_r$ 做為巡航路線特徵判斷標準，並使用兩角度閾值 ϵ_l 和 ϵ_h 與比較邊界向量的角度值 θ_l 與 θ_r 判斷目前巡航位置，巡航位置分為四種：走廊位置、樓梯位置、轉角位置與迴轉位置。

依照地圖將走廊位置分成三段，第一段為下圖一號紅框、第二段為下圖二號紅框和第三段為下圖三號紅框，走廊位置地圖如下圖 3-15 所示。



走廊有三種不同寬度 α ，如下圖 3-16 所示，分別為 2300、2400、2700(mm)，依照寬度最寬的計算巡航於走廊時大至的角度差值 ϵ 。

$$\theta_r = \sin^{-1} \left(\frac{\alpha_{max}/2}{d_r} \right) \approx 35^\circ \quad 3-20$$

$$\epsilon = 2\theta_r \approx 70^\circ \quad 3-21$$

為了確保機器人是巡航於中央，設定一走廊角度閾值 ϵ 為 80° ，若 ϵ 小於 ϵ_l 表示機器人巡航於走廊中央。

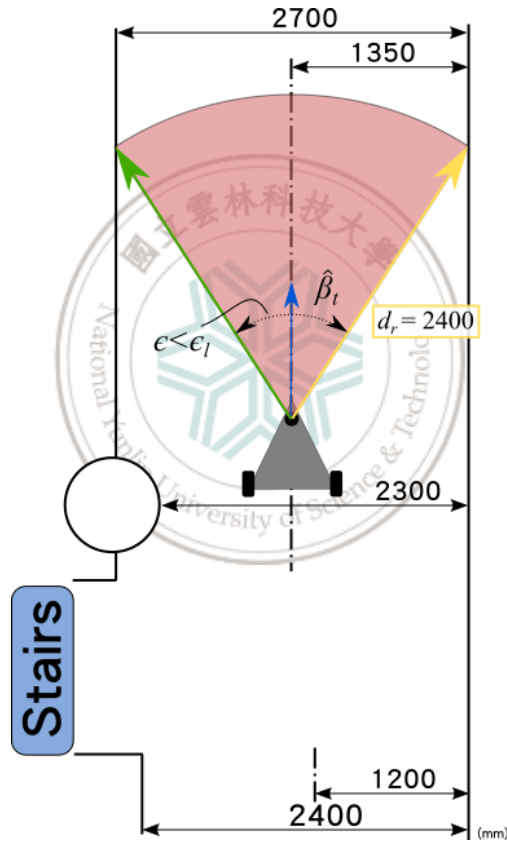


圖 3-16：走廊位置圖

3.3.3.2. 自走車位於樓梯位置 ($\epsilon_l < \epsilon < \epsilon_h$)

當巡航至樓梯位置時，角度差值會增加，最大約為 130° ，為了確保機器人已巡航至樓梯處，所以設定一樓梯角度閾值 ϵ_h 為 140° ，若角度差小於樓梯角度閾 ϵ_h (140°) 且大於走廊角度閾 ϵ_l (80°) 表示機器人已巡航至樓梯位置；接著確認樓梯位於機器人方向，此時樓梯方向會有兩種情況，第一種情況為樓梯位置在左側，如下圖 3-17 (a) 所示，比較兩邊界向量角度的大小，若 $|\theta_l| > |\theta_r|$ ，表示樓梯位置在左側。若 $|\theta_l| < |\theta_r|$ ，為第二種情況樓梯位置在右側，如下圖 3-17 (b) 所示。

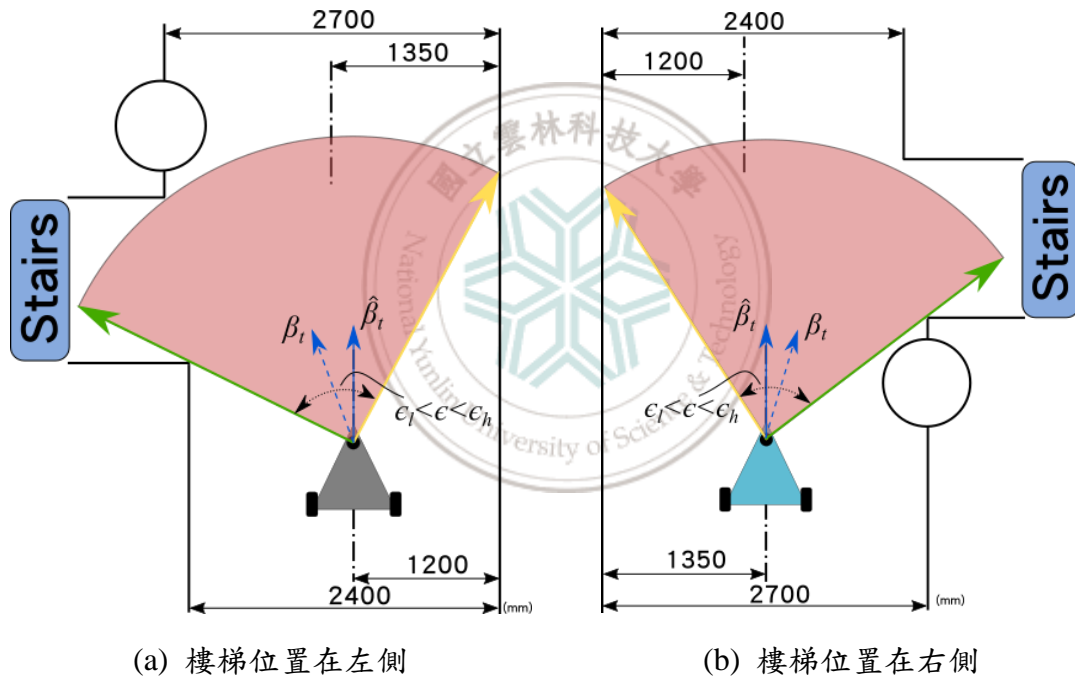


圖 3-17：樓梯位置圖

由於機器人巡航至樓梯位置時，候選角會因為安全空間擴大偏向樓梯處，如圖 3-17 中的 β_t ，可能導致機器人左右擺盪，甚至轉向樓梯，為了讓機器人繼續向前行駛，不讓機器人轉向樓梯處，因此，當機器人巡航至樓梯位置時，將候選角乘上一比例值 Kp 後為 $\hat{\beta}_t$ ，降低機器人轉向樓梯處的比例。

3.3.3.3. 自走車位於轉角位置 ($\epsilon_h < \epsilon < 220^\circ$ or $\epsilon \approx 240^\circ$)

當自走車巡航至轉角位置時，有兩種情況，分別由左轉轉角位置與右轉轉角位置表示。

左轉：巡航至左轉轉角位置時，會有三種情形，如下圖 3-18 所示，當機器人巡航至(a)的位置時角度差會落在大於 ϵ_h 小於 220° ，且 $|\theta_l| > |\theta_r|$ ，此時為了讓機器人轉向至第三段走廊，所以乘上一 Kc 值，使候選角偏向左側行駛，當機器人巡航至(b)時，因為空間過大，角度差會相當於 240° ，使候選角度趨近於 0° (為機器人正前方)，當機器人巡航至(c)時，角度差小於 80° ，此時已進入第三段走廊。

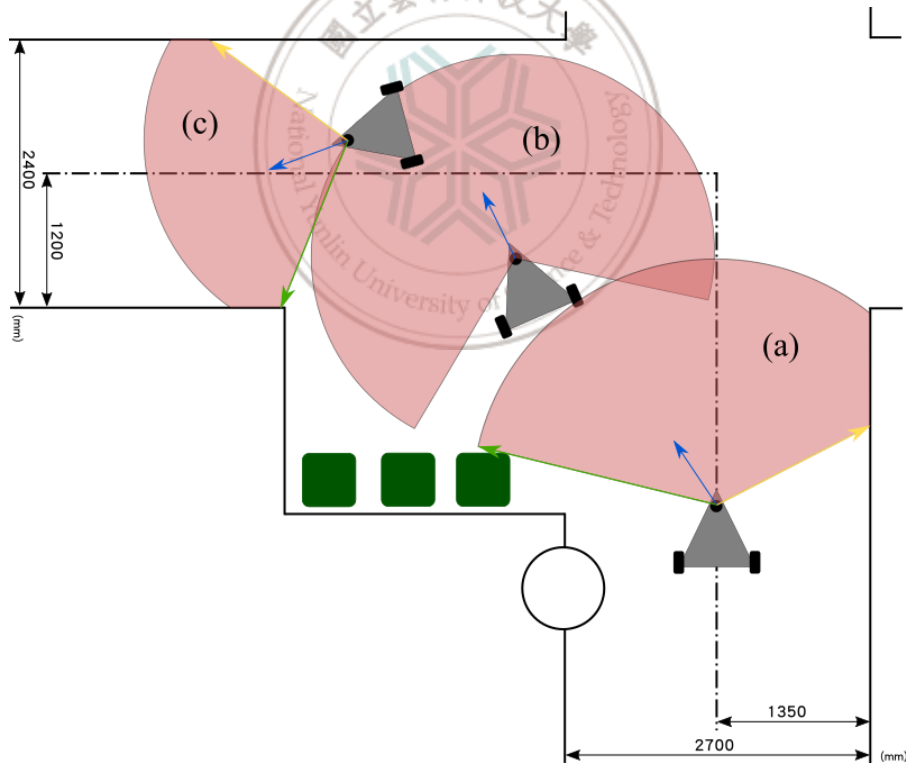


圖 3-18：左轉轉角路線圖

右轉：右轉轉角處與左轉相同共有三種情況，如下圖 3-19 所示，當機器人巡航至(a)的位置時角度差會大於 ϵ_h 小於 220° ，且 $|\theta_l| < |\theta_r|$ ，此時為了讓機器人轉回至第二段走廊，所以乘上一 Kd 值，使候選角偏向右側行駛，當機器人巡航至(b)時，角度差與左轉轉角位置相同，使候選角度為 0° (為機器人正前方)，當機器人巡航至(c)時，角度差小於 80° ，此時已回到第二段走廊。

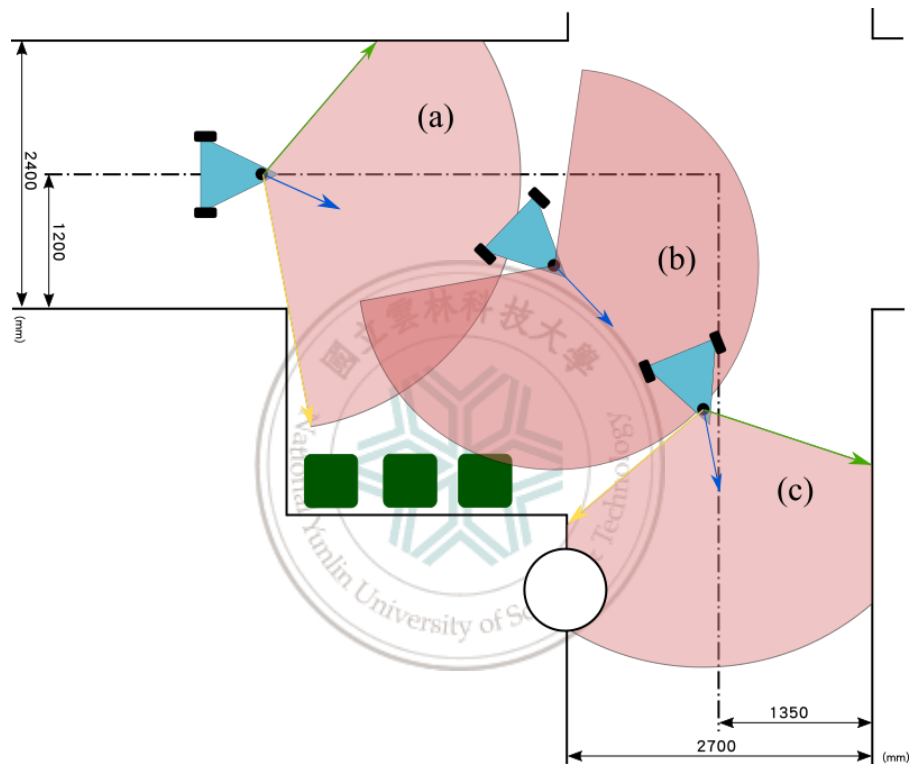


圖 3-19：右轉轉角路線圖

3.3.3.4. 自走車位於迴轉位置

當機器人巡航至迴轉位置時，如下圖 3-20 (a)位置所示，角度差會落於 ϵ_l 與 ϵ_h 之間（與樓梯位置相同），並且會遇到第一次右邊界角度大於左邊界角度 ($|\theta_r| < |\theta_l|$)，此時進行迴轉動作，並記錄已抵達目的 310（計數右邊界角度大於左邊界角度次數），接著轉向回程路線，如下圖 3-20 (b)位置所示。

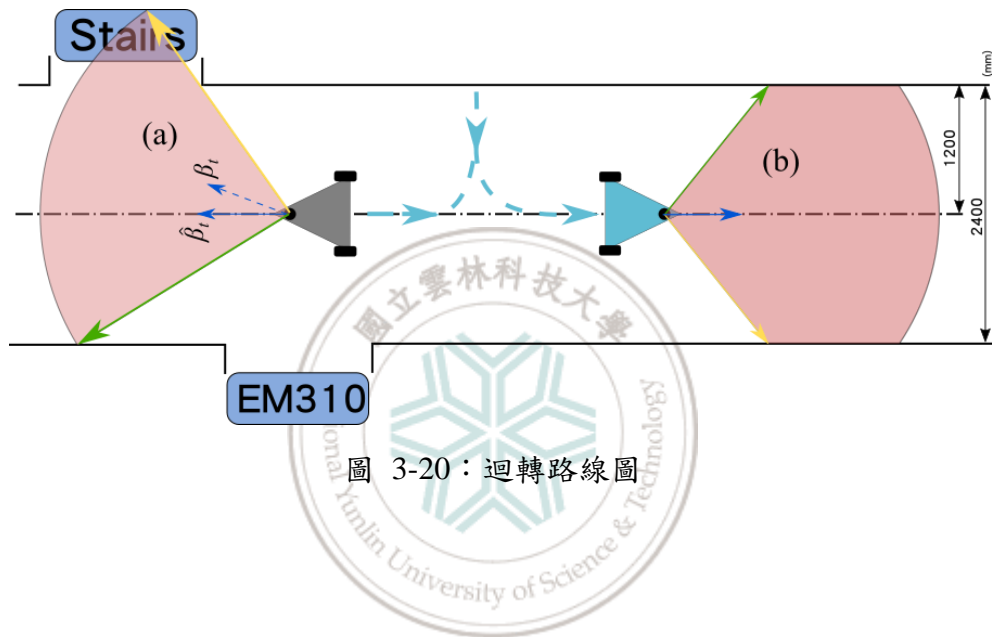


圖 3-20：迴轉路線圖

3.4. 程式流程圖

綜合以上之演算法與判斷方法，機器人的程式流程圖如下圖 3-21 所示：

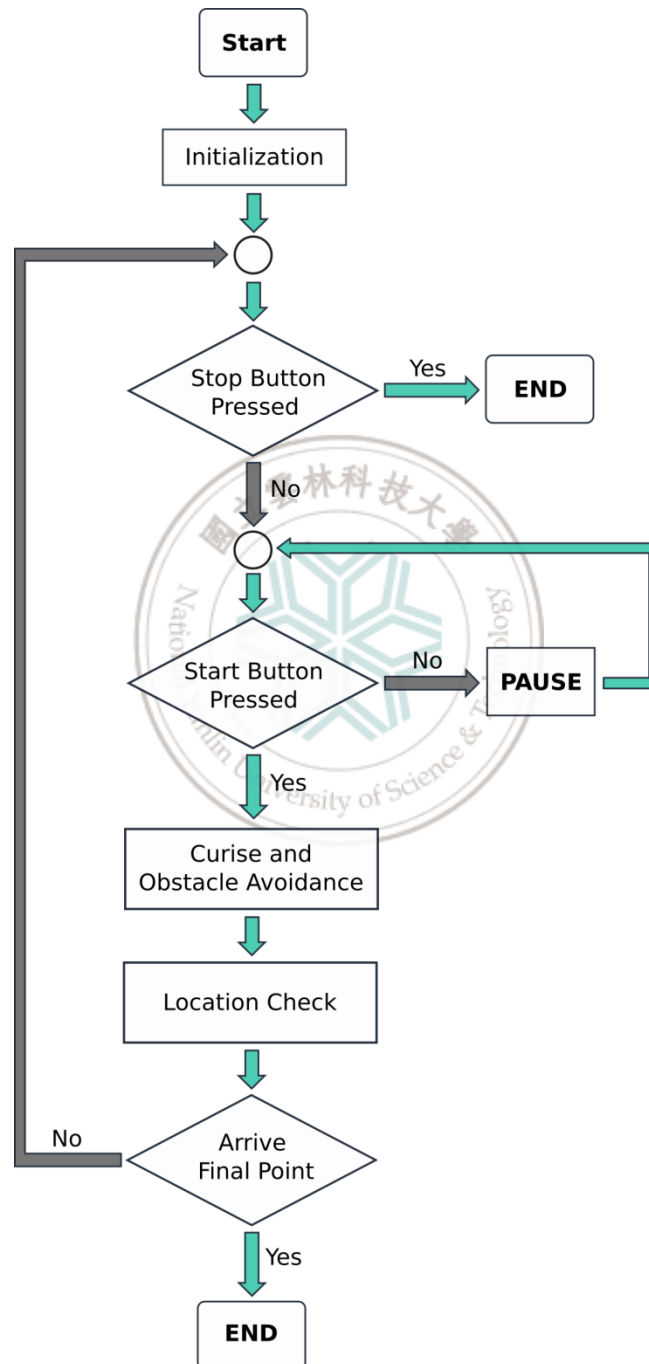


圖 3-21：程式流程圖

第4章 實驗討論與結果

本實驗討論針對機器人開發過程所遇到的問題來做討論與解決，其中包括參數的使用與特徵相同時該如何解決等問題。

而實驗結果除迴轉位置外，依照特徵位置分為三部份，分別為走廊位置、樓梯位置和轉角位置，每部分將使用光學雷達之掃描結果呈現機器人巡航路徑，而機器人在環境中使用半徑 ω_s 的圓來表示，並標記機器人的方向，來比較機器人尺寸與環境之間的關係。

4.1. 實驗討論

4.1.1. 自走車位於走廊位置

在 VFH+ 建立一極座標直方圖 P_i 時，使用公式 3-1 將參數 $a = 1500, b = 1$ ，使機器人巡航至第二段走廊時造成嚴重擺盪，原因為偵測範圍太短，使機器人巡航於第二段走廊時有過多的候選方向。解決方法為將參數 a 使用最大的走廊寬度 2700，將偵測範圍增大，增大至所有走廊位置之寬度進入此偵測範圍內。

4.1.2. 自走車位於轉角位置

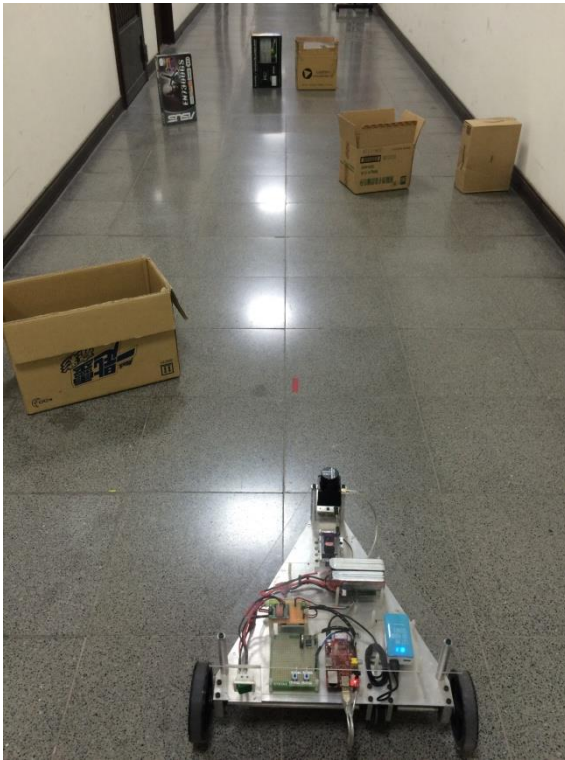
由於當機器人巡航至轉角位置時，某些時候會因為機器人轉向角位置有些許差異，導致機器人誤判為迴轉位置，解決方法為在控制法則裡使用兩個旗標，避開機器人在轉角位置時的誤判。

第一個旗標設定位置為機器人進入轉角位置時的(a)位置，第二個旗標為走廊位置，若第一個旗標位置尚未開啟，第二個旗標也不會開啟，就算機器人偵測位置為迴轉位置(錯誤的迴轉位置)也不會進行迴轉，當兩旗標開啟時，表示已經過轉角位置且進入第三段走廊位置，接著偵測是否進入迴轉位置，是的話則迴轉至回程路線。

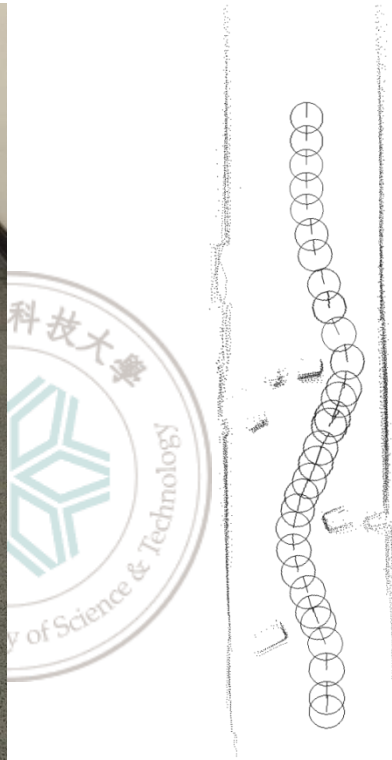
4.2. 實驗結果

4.2.1. 自走車位於走廊位置之結果

由第一段走廊做為走廊位置實驗結果呈現，對照圖 4-1(a) 與 圖 4-1 (b) 可以清楚看見機器人避開走廊上之障礙物後再回到走廊中央繼續巡航。



(a)走廊位置實驗環境



(b)走廊位置雷達量測結果

圖 4-1：走廊位置實驗結果

4.2.2. 自走車位於樓梯位置之結果

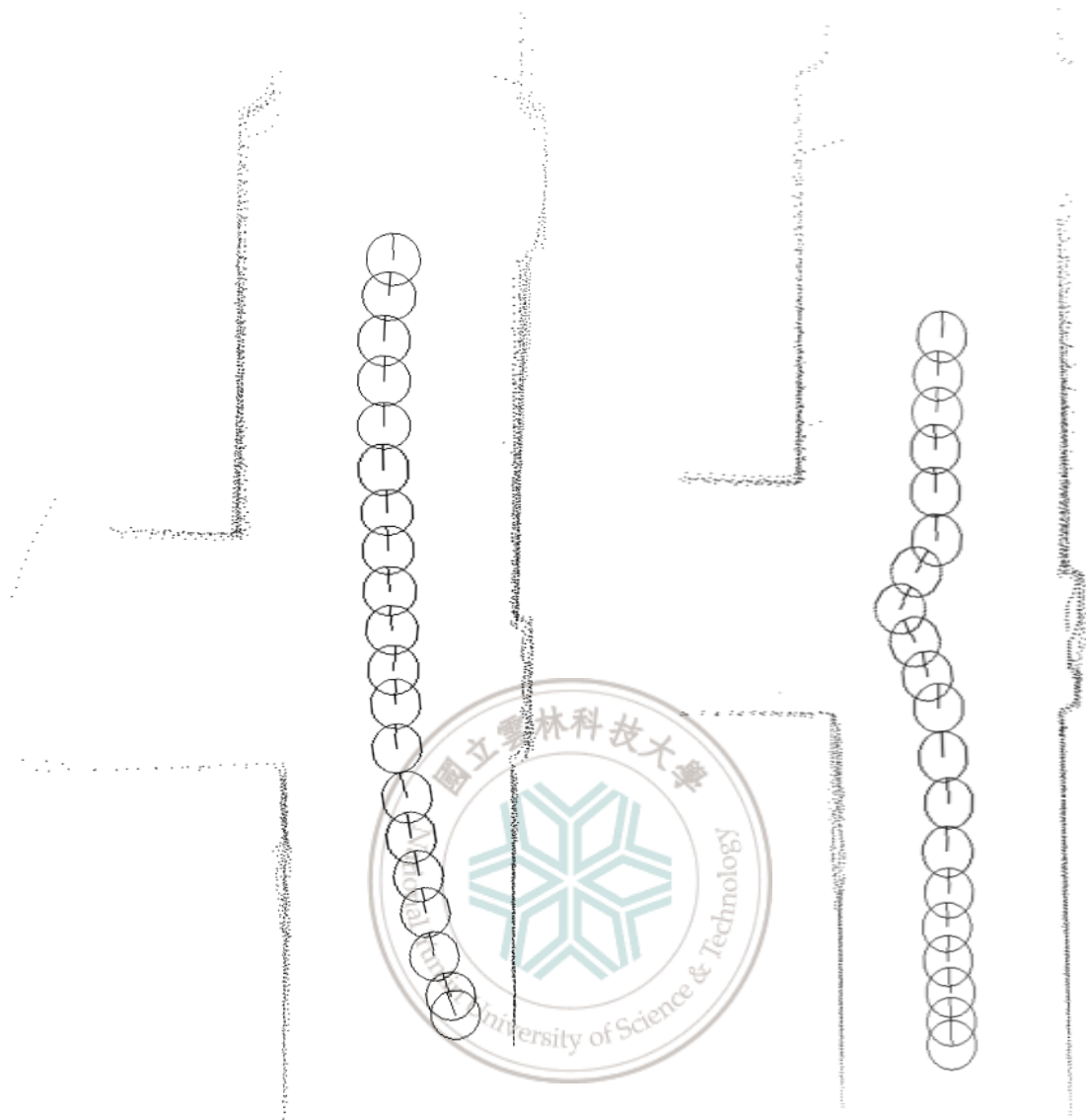
樓梯位置結果由樓梯位於左側與右側表示，以及樓梯位置補償前與補償後之比較。

4.2.2.1. 樓梯位置位於自走車左側

機器人將由第一段走廊末端開始巡航，並經過樓梯位置後巡航至第二段走廊前端，如圖 4-2 所示。在尚未加入補償值時機器人巡行於樓梯位置時會出現嚴重的擺動情形，如圖 4-3(a)所示，而加入補償值後使機器人巡航於樓梯位置時擺盪情況降低，如圖 4-3 (b)所示。



圖 4-2：樓梯位於自走車左側實驗環境



(a)使用樓梯位置補償

(b) 未使用樓梯位置補償

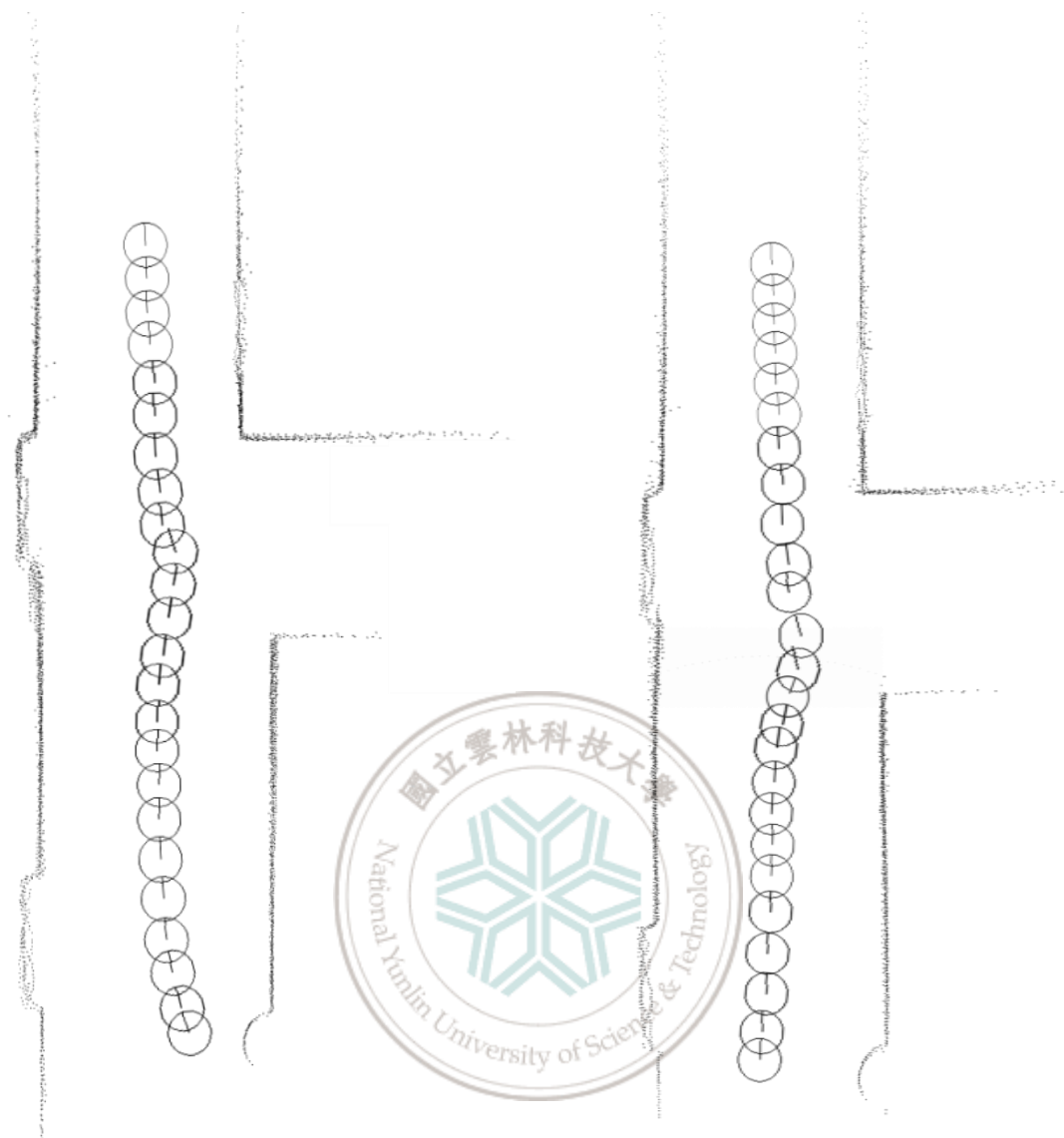
圖 4-3：樓梯位置補償實驗結果

4.2.2.2. 樓梯位置位於自走車右側

機器人將由第二段走廊前端開始巡航，並經過樓梯位置後巡航至第一段走廊末端，如圖 4-4 所示。由於第二段走廊與第一段走廊寬度不相同，使機器人巡航於樓梯位置也相對有偏差，巡航至樓梯位置時也會出現擺動情形，如圖 4-5(a)所示，而加入補償值後使機器人巡航於樓梯位置時擺盪情況降低，如圖 4-5 (b)所示。



圖 4-4：樓梯位於自走車右側實驗環境



(a)使用樓梯位置補償

(b) 未使用樓梯位置補償

圖 4-5：樓梯位置補償實驗結果

4.2.3. 自走車位於轉角位置之結果

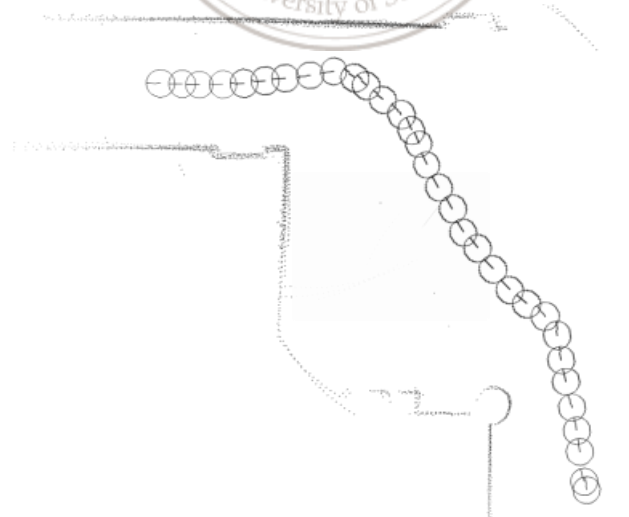
轉角位置結果由左轉轉角位置與右轉轉角位置表示。

4.2.3.1. 自走車左轉轉角位置

由第二段走廊尾端開始巡航轉角位置至第三段走廊前端，如下圖 4-6 所示，圖 4-6 (a) 為機器人位於第二段走廊末端位置，圖 4-6 (b) 則為機器人完成轉角位置巡航圖。



(a)左轉轉角位置實驗環境



(b)轉角位置雷達量測結果

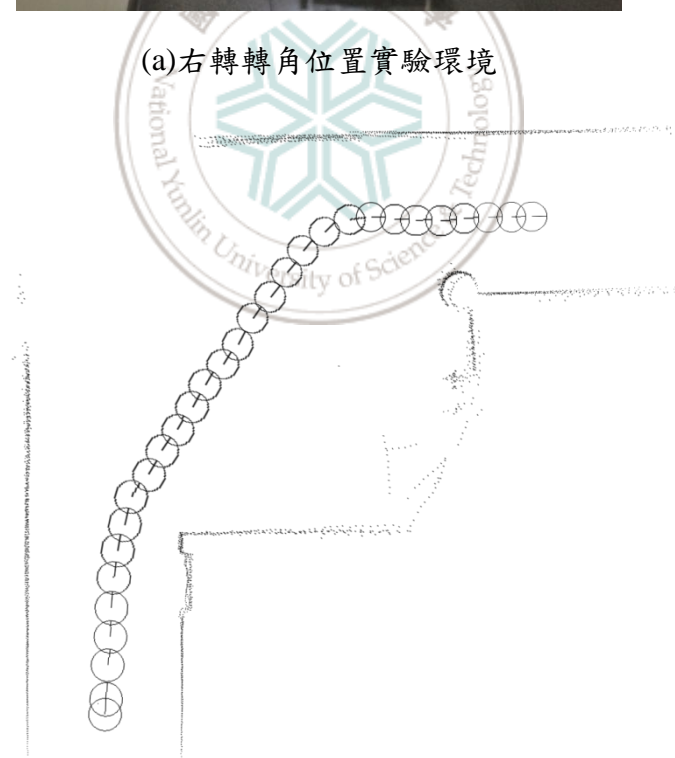
圖 4-6：左轉轉角位置實驗結果

4.2.3.2. 自走車右轉轉角位置

由第二段走廊尾端開始巡航轉角位置至第三段走廊前端，如下圖 4-7 所示，圖 4-7 (a) 為機器人位於第二段走廊末端位置，圖 4-7 (b) 則為機器人完成轉角位置巡航圖。



(a)右轉轉角位置實驗環境



(b) 右轉轉角位置雷達量測結果

圖 4-7：右轉轉角位置實驗結果

第5章 結論與建議

5.1. 結論

第一代 Yun-Trip 使用攝影機與影像處理進行室內巡航，其重量較重且影像處理速度所需時間較長，造成巡航速度降低與耗電量增加。本論文開發的 Yun-Trip II 使用光學雷達（LiDAR）整合室內特徵與避障功能，利用走廊的寬度與特徵，加入路線判斷方式，得知目前巡航位置與迴避障礙物，來完成室內巡航。

硬體方面，由於不需要進行影像處理，本論文使用 Raspberry Pi 嵌入式電腦搭配 GUN/Linux 作業系統取代了原先的工業主機板與 Windows XP 作業系統，利用原先 Yun-Trip 的車體底盤開發 Yun-Trip II 自走車，此嵌入式電腦具有直接輸出 PWM 與數位 I/O 訊號的功能，簡化了周邊設備，因此 Yun-Trip II 降低了車輛重量及體積，讓巡航速度增加，同時也降低電力消耗。

5.2. 建議

本論文的路線規劃演算法只針對特定環境做規劃，若是巡航到死路中，會因為光學雷達掃描範圍受到限制使機器人無法向後方移動，導致巡航停止，因此，可以在光學雷達下方加裝一旋轉平台，偵測後方環境，或是將巡航路線記錄，當機器人巡航至死路時，可追蹤巡航路線並找出是哪個決策造成巡航至死路，將機器人往回巡航至另一個決策方向。

另外可以增強室內定位功能的強健性，由於當機器人巡航至轉角位置時，某些時候會因為機器人轉向角位置有些許差異，導致機器人誤判為迴轉位置，雖然此問題已解決，但機器人對於環境特徵還是有待加強，建議可以使用即時定位與地圖重建 (Simultaneous Localization And Mapping)，來建構室內地圖，增強機器人室內定位功能的強健性，使環境定位更加完整。

參考文獻

- [1] 顏大翔，2008，智慧型自主式車輛機器視覺系統之研究，碩士論文，雲林科技大學，雲林
- [2] 黃平雲，2012，智慧型自主式車輛的環境適應之改善研究，碩士論文，雲林科技大學，雲林
- [3] 詹鈞博，2013，自主式車輛於障礙物偵測與迴避之研究，碩士論文，雲林科技大學，雲林
- [4] 劉圻武，2014，姿態整合量測系統、全球定位系統與光學雷達之無人自走車導航研究，碩士論文，雲林科技大學，雲林
- [5] Ulrich, I. and J. Borenstein, 1998, VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. *IEEE Intl. Conf. on Robotics and Automation*, 1527-1577
- [6] Simmons, R., 1996 The Curvature-Velocity Method for Local Obstacle Avoidance. *IEEE Intl. Conf. on Robotics and Automation*.
- [7] Borenstein, J. and Y koren, 1991, The Vector Field histogram – Fast Obstacle Avoidance for Mobile Robots. *IEEE Intl. Conf. on Robotics and Automation*, 7(3), 278-288
- [8] 陳依璟，2012，自走車之路徑規劃與位置追蹤，碩士論文，國立中央大學，桃園