

Research on the Control and Optimization of Autonomous Vehicles Based on ROS, Pixhawk, and PLC

Department of Mechanical Engineering, Kun Shan University

ABSTRACT

This study retrofitted a four-wheel beach buggy and built a three-tier closed-loop control stack that joins ROS perception, a Pixhawk autopilot, and a PLC drive layer. Mission Planner first supplies fixed waypoints; Pixhawk tracks them; ROS fuses LiDAR, IMU, and dual-antenna RTK-GPS data to spot obstacles and, when needed, issues RC-override commands; the PLC converts these into PWM or voltage signals for the wheel-hub and steering servos. Field trials with RTK-Fix show a mean waypoint error of 1.5 m and an RMSE of 1.8 m; tuning ATC_STR_RAT_P cuts the worst error to 0.5 m. Over 90 % of avoidance events complete their decision cycle in under 0.2 s and succeed. Compared with a Pixhawk-only baseline, the integrated stack raises path overlap on tight curves by more than forty percent and clearly suppresses steering overshoot. Even when satellites are partly blocked, overall error stays within 2 m thanks to multi-sensor fusion and PLC compensation. The ROS–Pixhawk–PLC scheme therefore offers accurate, responsive, and robust navigation for campus patrol, crop care, and park-logistics vehicles. Its modular design also simplifies porting to other ground platforms and provides a practical template for low-speed autonomous mobility in semi-structured outdoor spaces.

Keywords:

ROS, Pixhawk, PLC, autonomous vehicle, system integration, control strategy optimization, sensor fusion

目錄

中文摘要.....	i
英文摘要.....	ii
致謝.....	iii
目錄.....	iv
圖目錄.....	vii
表目錄.....	x
第一章、緒論.....	1
1.1 研究背景與動機.....	1
1.2 文獻回顧.....	2
1.2.1 ROS 系統在自駕領域之應用.....	3
1.2.2 Pixhawk 與 ArduPilot 控制架構.....	4
1.2.3 LiDAR 感測與避障導航策略.....	5
1.2.4 路經優劣分析方式.....	6
第二章、研究內容與方法.....	8
2.1 系統架構概述.....	8
2.2 ROS 架構與模組化控制.....	9
2.2.1 ROS 開發工具軟體簡介.....	10
2.3 ArduPilot 與 MAVROS 控制整合原理.....	13
2.3.1 ArduPilot 開發工具軟體簡介	14
2.3.2 MAVROS 控制整合原理.....	16
2.4 可程式邏輯控制器（PLC）控制應用.....	18
2.5 路徑誤差計算方法.....	20

2.5.1 使用 Shapely 之空間誤差量測方法.....	21
2.5.2 誤差圖表分析說明	22
第三章 硬體選用與理論基礎	25
3.1 實驗沙灘車載台	25
3.1.1 沙灘車載台速度測試	27
3.2 硬體架構與元件選用說明	29
3.3 系統整合方法	37
3.3.1 ROS 與 LiDAR 資料處理流程.....	38
3.3.2 Pixhawk 與 ROS 整合機制.....	38
3.3.3 PLC 與 Pixhawk 對接控制邏輯設計.....	39
第四章 模擬分析	42
4.1 系統環境建置與初步測試	42
4.1.1 Quanergy M8 LiDAR 安裝與 RViz 顯示驗證.....	42
4.1.2 Pixhawk 6C 與 ROS 系統通訊驗證	43
4.2 Gazebo 模擬環境建置與初步驗證	45
4.2.1 r1_rover 載具模擬平台建立	46
4.2.2 Gazebo 內載具光達感測與障礙物偵測	47
4.3 自動避障導航程式設計與 Gazebo 模擬測試	48
4.3.1 自動避障導航演算法設計與實現.....	49
4.3.2 Gazebo 環境下載具自主避障測試與分析	51
4.4 Quanergy M8 LiDAR 實體感測與 Gazebo 模擬融合測試.....	52
4.4.1 Quanergy M8 LiDAR 與 Gazebo 模擬環境融合機制建構.....	52
4.4.2 實體光達掃描障礙物後，Gazebo 模擬轉向行為測試	53
4.4.3 Pixhawk 6C 控制模式自動切換功能測試（AUTO 模式轉 Manual 模式）	54
第五章 實驗規劃與執行	56

5.1 測試環境與參數設定	57
5.2 任務路徑設定與行駛軌跡記錄	59
5.3 系統穩定性與準確性分析	61
5.3.1 GPS 系統準確性分析結果	63
5.4 系統優化分析	66
5.4.1 載具路徑偏差原因	66
5.5 轉向參數調整優化整體任務	70
5.5.1 任務數據紀錄分析	71
5.5.2 系統性之路徑誤差分析	74
5.6 訊號不良環境下之導航誤差分析與評估	81
5.7 載具避障測試	85
5.7.1 左/右轉避障測試	85
5.7.2 間隙穿越測試避障測試	89
5.7.3 校園實地測試	91
第六章 結論與建議	95
6.1 研究成果歸納	95
6.2 與現有研究差異及性能優勢	96
6.3 未來研究建議與展望	96
參考資料	98

圖目錄

圖 2.1 基本架構.....	8
圖 2.2 ROS 系統架構圖.....	9
圖 2.3 GAZEBO 示意圖	11
圖 2.4 Rviz 示意圖.....	12
圖 2.5 MISSION PLANNER 示意圖.....	14
圖 2.6 行駛紀錄示意圖.....	15
圖 2.7 模式切換流程示意.....	17
圖 2.8 訊號中介轉換機制流程示意	18
圖 2.9 基本 PLC 流程圖示意	19
圖 2.10 路徑投影示意圖.....	21
圖 2.11 任務點距離誤差分析示意.....	22
圖 2.12 路徑重合度分析示意圖.....	23
圖 2.13 路徑相似度分析示意.....	24
圖 3.1 沙灘車側視圖.....	25
圖 3.2 沙灘車俯視圖.....	26
圖 3.3 測試路線圖.....	27
圖 3.4 檔位/速度圖示意	28
圖 3.5 統硬體整合架構圖.....	29
圖 3.6 QUANERGY M8 LiDAR	30
圖 3.7 PIXHAWK 6C.....	31
圖 3.8 RTK-F9P 模組	32
圖 3.9 ARDUINO MEGA 2560.....	33
圖 3.10 DAC 模組.....	34
圖 3.11 HOLYBRO RADIO TELEMETRY SET(915).....	35
圖 3.12 台達 DVP-SX2 系列.....	36
圖 3.13 系統整合示意圖.....	37
圖 3.14 LiDAR 資料處理流程圖.....	38
圖 3.15 ARDUINO PWM 訊號轉換數值監控畫面.....	40
圖 3.16 PLC 階梯圖對應 ARDUINO DAC 輸出電壓之控制狀態.....	41

圖 4.1 QUANERGY M8 LiDAR 點雲資料呈現.....	43
圖 4.2 PIXAWK 6C 與 ROS 建立溝通.....	44
圖 4.3 MAVROS 發佈的飛控資訊.....	44
圖 4.4 R1_ROVER 模擬載具	45
圖 4.5 GAZEBO 搭配 RViz 視·覽工具.....	46
圖 4.6 光達與模擬載具 R1_ROVER.....	47
圖 4.7 POINTCLOUD2 3D 點雲	48
圖 4.8 將 LiDAR 過濾出所需部分.....	49
圖 4.9 避障邏輯圖.....	50
圖 4.10 (A)偵測障礙物; (B)偵測到障礙物後向點雲較少處轉彎.....	51
圖 4.11(A)偵測障礙物; (B)穿越複雜縫隙環境.....	52
圖 4.12 模擬環境融合示意圖.....	53
圖 4.13 M8 LiDAR 掃描點雲避障(紅框表示).....	54
圖 4.14 無點雲時轉換模式.....	55
圖 5.1 環境干擾場域示意圖.....	56
圖 5.2 (A)設備照 ; (B)測試場域.....	57
圖 5.3 測試路線圖.....	59
圖 5.4 RTK Fix 模式.....	60
圖 5.5 (A)多次任務執行之路徑紀錄; (B)將路徑經緯度數據透過 EXCEL 視覺化.....	61
圖 5.6 GPS 準確性分析示意圖	65
圖 5.7 人控駕駛情境示意圖.....	66
圖 5.8 自動駕駛情境示意圖.....	67
圖 5.9 轉向邏輯偏差示意圖.....	67
圖 5.10 任務點距離誤差分析.....	68
圖 5.11 路徑重合度分析.....	69
圖 5.12 未調整參數之 PIDA 與 PIDS 結果	71
圖 5.13 ATC_STR_RAT_P 0.1	72
圖 5.14 ATC_STR_RAT_P 0.5	72
圖 5.15 ATC_STR_RAT_P 1.0	73
圖 5.16 ATC_STR_RAT_P 1.5	73
圖 5.17 任務路徑圖	74
圖 5.18 ATC_STR_RAT_P: 0.1	75

圖 5.20 ATC_STR_RAT_P: 1.0	75
圖 5.22 ATC_STR_RAT_P: 2.0	76
圖 5.24 ATC_STR_RAT_P: 3.0	76
圖 5.26 ATC_STR_RAT_P: 4.0	77
圖 5.28 ATC_STR_RAT_P: 5.0	77
圖 5.29 ATC_STR_RAT_P/路徑重合度圖	79
圖 5.30 未調整參數(左)與調整參數(右)對比圖.....	80
圖 5.31 訊號不良場域路徑.....	81
圖 5.32 (A)不良環境測試結果 1 (B) 不良環境測試結果 2	82
圖 5.33 較差 GPS 訊號數據圖.....	83
圖 5.34 較好 GPS 訊號數據圖.....	84
圖 5.35 載具左側閃避過程.....	86
圖 5.36 (A) 載具未偵測障礙物(AUTO); (B) 載具右側偵測到障礙物(TURN LEFT)	86
圖 5.37 載具右側閃避過程.....	87
圖 5.38 (A) 載具未偵測障礙物(AUTO); (B) 載具右側偵測到障礙物(TURN RIGHT)	87
圖 5.39 間隙穿越過程	89
圖 5.40 實地測試路線圖	91
圖 5.41 (A)測試場域距離; (B)實際路徑軌跡圖	92
圖 5.42 校園測試任務點距離誤差分析.....	93
圖 5.43 校園測試路徑重合度分析.....	94

表目錄

表 2-1 主要 ROS TOPICS、頻率與場域	10
表 2-2 模式比較表.....	16
表 3-1 沙灘車硬體規格表.....	26
表 3-2 載具速度表.....	28
表 3-3 硬體規格與功能對照表.....	29
表 3-4 主控電腦規格表	36
表 5-1 外部環境 RTK 模式啟用表.....	57
表 5-2 任務點距離誤差數值表.....	68
表 5-3 路徑重合度誤差數值表.....	69
表 5-4 任務點距離誤差分析表.....	78
表 5-5 路徑重合度分析表	79
表 5-6 左右避障測試表.....	88
表 5-7 縫隙穿越測試表	90

第一章、緒論

1.1 研究背景與動機

近年來，隨著人工智慧、物聯網、大數據與自動化技術的快速進步，自動駕駛技術逐漸成為全球重要的發展趨勢之一。全球許多國家與企業紛紛投入大量資源，期望透過自動駕駛技術提高交通安全、降低交通事故發生率，並有效減少人力成本及提升運輸效率。此外，自駕技術的普及亦有助於解決人口高齡化所帶來的行動不便問題，並能應用於環境探測、災難救援、精密農業及軍事等領域，具有極大的經濟價值與社會效益。

儘管現有的自動駕駛技術已在汽車、無人機、無人船舶等領域取得長足的進步，但在面對複雜、崎嶇且非結構化的戶外環境（如沙灘、山區、農田等地形）時，仍存在諸多挑戰。特別是環境感知技術的精準度與可靠性、即時避障與導航策略的有效性，以及硬體控制的穩定性與即時反應能力，皆需進一步提升以滿足實際應用需求。因此，如何開發一套能適應戶外複雜環境且具備高度穩定性的自動駕駛系統，成為當前自動駕駛研究中極需解決的重要課題。

基於上述背景，本研究致力於探討如何結合ROS (Robot Operating System)、Pixhawk 飛控系統與 PLC (Programmable Logic Controller)，打造出能在戶外複雜地形中穩定可靠運行之智慧自駕系統。透過 ROS 之高度模組化與擴展性，Pixhawk 之成熟穩定飛控架構，以及 PLC 之高即時性與穩定硬體控制能力，期望建立一套高效能且實務價值高的自動導航平台，以滿足戶外探索、環境監測、農業自動化及災難救援等應用需求，並為後續研究與產業界應用提供重要參考。

1.2 文獻回顧

為了深入探討自駕載具系統整合面臨的技術挑戰及其相應解決策略，本節將對當前相關領域的研究進行回顧與分析，並對重點文獻進行系統化整理與歸納。目前自駕載具的研究主要聚焦於系統架構整合、導航感測技術以及精準路徑追蹤等領域。為凸顯本研究的創新性與價值，本節將分為四個部分逐步闡述。首先，就系統架構整合而言，ROS（Robot Operating System）正逐步成為自駕技術應用中的核心工具。憑藉其模組化設計、高擴展性能與跨平台兼容性，ROS 為感知、控制與決策功能提供了一個高效的平台。然而，在實務操作中，ROS 仍面臨通訊延遲及異構硬體整合的挑戰。因此，第一部分將全面回顧 ROS 系統在自駕應用中的現狀，並分析其發展局限與實用挑戰，詳見 1.2.1 節。其次，在導航控制技術方面，Pixhawk 飛控系統及 ArduPilot 控制架構以其穩定的技術性能和廣泛應用深受業界重視，尤其在執行戶外環境導航任務方面表現出顯著優勢。因此，本節第二部分將探討 Pixhawk 與 ArduPilot 在導航控制研究中的現有成果，並評估其對本研究之應用價值，詳見 1.2.2 節。接著，關於環境感知與避障導航技術，LiDAR（光探測與測距）因其高精度感測能力和即時性，已成為自駕載具中不可或缺的關鍵技術。然而，如何將 LiDAR 感測技術與實用避障策略有機結合仍是一大挑戰。因此，本節第三部分將聚焦於 LiDAR 技術在戶外導航與避障領域的應用進展，詳見 1.2.3 節。最後，針對路徑追蹤性能的實證評估，過往研究中常引用多種量化指標，例如球面距離（Haversine distance）、路徑重合度，以及曲線相似性度量，來綜合評估載具的空間及時序追蹤能力。然而，如何選擇適當的評估指標以全面且客觀地反映系統效能仍是一項值得深入研究的課題。因此，本節第四部分將解析常見路徑追蹤評估方法及其實務應用情境，詳見 1.2.4 節。通過上述四個主題的文獻回顧，本研究將明確辨識目前自駕載具技術的局限與機遇，從而為後續構建創新系統整合架構與解決策略提供堅實的理論基礎與實務參考。

1.2.1 ROS 系統在自駕領域之應用

Robot Operating System (ROS) 為一開源機器人作業系統，具備模組化設計、完整工具鏈與跨平台整合能力，已廣泛應用於自駕領域之感知、決策與控制。其「發布-訂閱」通訊模型能在多節點間高效交換資料，支援複雜自駕車系統的感測整合與決策運算。文獻 [1] 提出資料流 / 執行流程感知的 ROS 效能分析框架，指出節點通訊延遲常成系統瓶頸；文獻 [2] 透過 ROS 整合 LiDAR、IMU、GPS 與 RViz、Gazebo，展示 ROS 在異質感測與控制裝置間的高擴展性。Kato 等人將 ROS 1 打造成開源自駕叢集 Autoware.AI，並於嵌入式平台驗證其部署效能 [3]；Post 與 Bianco 在農田鬆軟地形實作 GNSS / LiDAR 融合導航，證實 ROS 於非結構環境的可靠度 [4]；Quinn 等人結合 ROS 1 與 Pixhawk，建立訊號橋接與實車試驗流程，提供「ROS + Pixhawk」混合架構的實務範例 [5]。Marín 等人提出完整的 ROS-based 智慧車輛軟體平台，展示感知、路徑規劃與控制模組的彈性整合 [6]；Neto 等人於 FASTEN 專案中利用靜態分析工具量化 ROS 1 程式碼缺陷改善幅度 [7]；Santos 等人發表 HAROS 框架，以模型驅動方式抽取 ROS 1 計算圖並執行形式驗證，提升可靠性 [8]；Tardioli 等人開發 Pound 節點，透過多核心與優先級機制顯著降低 ROS 1 在無線多機網路中的延遲與抖動 [9]；Prescinotti Vivan等人則在 CARLA 模擬環境評估多種 ROS-based 感知與控制策略，提出可快速遷移至實車的低成本流程 [10]。綜合文獻可知，ROS 系統已建立完整開發生態並擁有多元實車與模擬案例，但節點通訊延遲、靜態品質保證與跨域硬體整合仍為系統性能與韌性的關鍵課題；本研究遂結合 ROS 感知決策、Pixhawk 導航與 PLC驅動之三階層閉環架構，以戶外沙灘車平台驗證其導航精度與系統可靠度，回應上述文獻所揭示之挑戰與需求。

1.2.2 Pixhawk 與 ArduPilot 控制架構

Pixhawk 與 ArduPilot 開源生態累積了多種實車與模擬案例，進一步印證其硬體彈性與軟體整合價值。早期 Meier 團隊釋出的 PIXHAWK open-source computer vision framework 說明飛控板將多 IMU 與視覺模組整合於單一平台，並透過 MAVLink 回傳姿態與影像時戳，為後續視覺導航與板端估測奠定基線 [11]。在地面載具方面，Hasan 針對森林監測情境評估 ArduPilot-controlled rover 的航跡精度，證實 Rover 韌體在 GNSS 與輪速計融合下能維持 0.6 m 內 RMS 誤差，提供本研究沙灘車參數調校的量化參考 [12]。對於離線驗證流程，Xiao 開發的 XTDrone 框架將 PX4-SITL、Gazebo 與 ROS1 封裝為模組化平台，可快速切換機型並匯出 SDF，讓研究者先在鎖步環境測試控制迴圈再移植至實機，顯著縮短調參週期 [13]。室內定位場域也有以 Pixhawk 為核心的實務做法。ArduPilot 官方示範專案利用 Raspberry Pi 3 於 ROS 中辨識 ArUco 板，將估得位姿透過 MAVROS 注入位置控制器，完成無 GPS 室內自主飛行，說明 Pixhawk 可接受外部定位源並維持 50 Hz 內迴授 [14]。ArduPilot 開發文件進一步整理 ROS1 與 Rover/Copter 模式的 topic 對映與示例 launch 檔，為 MAVROS + ArduPilot 整合提供權威配置指引 [15]。在模擬層，社群討論帖展示如何以 RotorS xacro 與 PX4-SITL 建立客製化機型並透過 MAVROS 確認 EKF 時序，提供自建 Gazebo 車體模型的實作細節 [16]。此外，PX4 社群案例說明將 ROS Kinetic、Gazebo 7 及 PX4-SITL 組合，利用 mavros_posix_sitl.launch 進行外部定位控制，展示 Pixhawk 系列飛控在 SITL 階段即可驗證 off-board 指令邏輯 [17]，Pixhawk 結合 ArduPilot 韌體後能以 MAVLink 與 ROS 節點雙向互通，形成「板端即時控制 + 主機決策層」之兩層式架構。陳謙信的碩士論文將 Jetson TX2 置於頂層執行航點決策與 YOLO 影像辨識，透過 MAVROS 以 UART 介面向 Pixhawk 下達 off-board 指令，並先以 Gazebo SITL 測試後移植至六旋翼實機，驗證雙向資料流與姿態回饋皆可於

50 Hz 內穩定運行，提供板-機分工的完整實證流程 [18]。此外，Xiao 等人開源的 XTDrone 平台整合 PX4-SITL、Gazebo 與 ROS1，採鎖步同步以確保感測時間戳一致，並已示範視覺 SLAM、隊形飛行等案例；其模組化設計允許研究者在虛擬環境迭代控制 / 估測演算法，再無縫部署至 Pixhawk 實機，為本研究採用 SITL-HIL 連續驗證鏈提供即時性與可移植性的佐證 [19]。

綜合文獻，Pixhawk 結合 MAVROS 與 SITL-HIL 鏈可在真實與模擬環境保持一致的指令介面。地面與空中平台的多樣實證亦顯示，只要透過標準 MAVLink 橋接，就能讓 ROS 節點專注於避障決策與路徑管理，而將高速姿態與動力控制交由 ArduPilot 板端即時迴圈處理，正符合本研究沙灘車平台的分層設計理念。

1.2.3 LiDAR 感測與避障導航策略

自駕載具的「多控制架構」整合已逐步從單一感測或單一控制迴路，走向將 LiDAR、Radar、視覺、慣測、GNSS 與多層控制器協同的全方位設計。張豪文採 U-Net 於 LiDAR 點雲進行即時語義分割，行人與車輛精度比 SqueezeSeg 提升約二成，證實深度模型能在板端即時感知中提供高解析度語意輸出 [20]。彭晨益則結合 MotionNet 運動分割與 YOLOv8 目標框定，於 Oxford Radar RobotCar 小型移動物件偵測準確度提升超過 20 %，展示雷達動靜態融合在惡劣天候的優勢 [21]。張脩寬以多目標蟻群優化調整模糊規則，讓雙輪平台在動態場景維持速度與避障距離的最適折衷，說明智慧演算法可直接優化決策層參數 [22]。從感測到決策再到執行，多模態融合與層級化軟體平台是主流。Sensors 期刊的回顧指出，相機、LiDAR、Radar 混合校正後可顯著降低單一感測缺陷帶來的誤檢，並概述各類融合框架的通訊延遲瓶頸 [23]。De la Cruz 提出 ROS-JADE 多層代理式架構，讓物流車不僅能在底層閉環運動控制，也能透過 MAS 協商任務，實證「決策層—運動層」鬆耦合設計的可移植性 [24] Svec 等人比較多種感測組合的導航路徑規劃後

指出，將 EKF 融合輸入於 ROS navigation stack 可把長走廊漂移誤差控制在 0.15 m 之內 [25]。Dudas 於 Gazebo 建立多車協同模擬，將硬體驅動話題映射至虛擬傳輸協定，為雲端大規模車隊測試提供可重現基準 [26]。最後，ROMR 專案釋出成本低於 1500 美元、負載 90 kg 的 ROS-based 開源移動底盤，整合板載 MCU 運動控制與 x86 感知規劃，展示社群硬體對多控制架構快速原型的支持 [27]。

綜合可見，多控制架構的核心趨勢為：前端以多模態深度感知取得高可靠環境模型，中層透過 ROS 或 MAS 等中介件實現感知-決策-控制解耦，底層交由 Pixhawk、PLC 或 MCU 在高頻道驅動迴圈中確保即時性與安全。這類分層設計亦呼應本研究 ROS–Pixhawk–PLC 架構，在感測融合、優化控制及工控安全之間取得平衡，為戶外自駕載具提供可持續擴充的技術基礎。

1.2.4 路經優劣分析方式

在軌跡優劣分析領域研究者常結合球面距離誤差與曲線相似度量以全面衡量自主載具在空間和時序上的追蹤品質。Bennett 等人在低速 GPS 距離估算中應用 Haversine 公式並配合濾波，展示球面距離在減小短程誤差的效果[28]，Nichat 等人將 Haversine 用於地標啟發式 A*搜尋，提高了大範圍路徑估計的效率與準確性[29]，Jiang 等人的時間序列多徑分析方法則為逐點誤差時序特性提取提供思路，可結合 Haversine 誤差序列區分誤差源[30]。單點誤差雖直觀但難呈現整段路徑形變因此曲線相似度量逐漸受到重視：Han 等驗證 Fréchetdistance 在接管軌跡評估中的時空敏感性[31]，Tao 等系統比較 DTW、Fréchet、LCSS、EDR 等度量特點為不同應用場景選型提供指導[32]，Wan 等則提出基於 DTW 的模糊軌跡關聯方法在異步多源匹配中表現出高魯棒性[33]。在控制評估層面 Rokonuzzaman 等人回顧各類路徑跟蹤控制方法對誤差的影響為誤差量測提供背景[34]Liu 等的無模型自適應控制展示了在測量干擾條件下的誤差抑制策略[35]。對於覆蓋任務可借鑑 PourArab

等在3D混合覆蓋路徑規劃思路通過Path-OverlapRatio等指標在田間作業中即時評估覆蓋質量並觸發重新規劃以優化行進距離與能耗[36]。



第二章、研究內容與方法

本章旨在說明本研究中自駕載具控制系統的整體設計與整合方法。系統以 ROS 為整合核心，搭配Quanergy M8 LiDAR 進行感測，透過Pixhawk 6C 與 ArduPilot 實現導航控制，並結合 PLC 作為實體控制器，完成一套具備自主避障能力的智慧載具控制系統。以下將分別說明系統架構、整合方式與硬體選用。

2.1 系統架構概述

本研究所建構之自駕載具控制系統，分為感測層、決策層與執行層。感測層負責擷取環境資訊如圖 2.1 所示，主要使用 Quanergy M8 LiDAR 獲取 3D 點雲資料；決策層以 ROS 系統進行資料處理與避障邏輯判斷；執行層由 Pixhawk 飛控模組與 PLC 控制器組成，分別對應導航模式與實體載具的驅動控制。系統中各層間透過 TCP/IP、MAVLink 與 ROS 自定義主題進行通訊與指令傳遞，構成一個可擴充與高度模組化的自主控制平台。

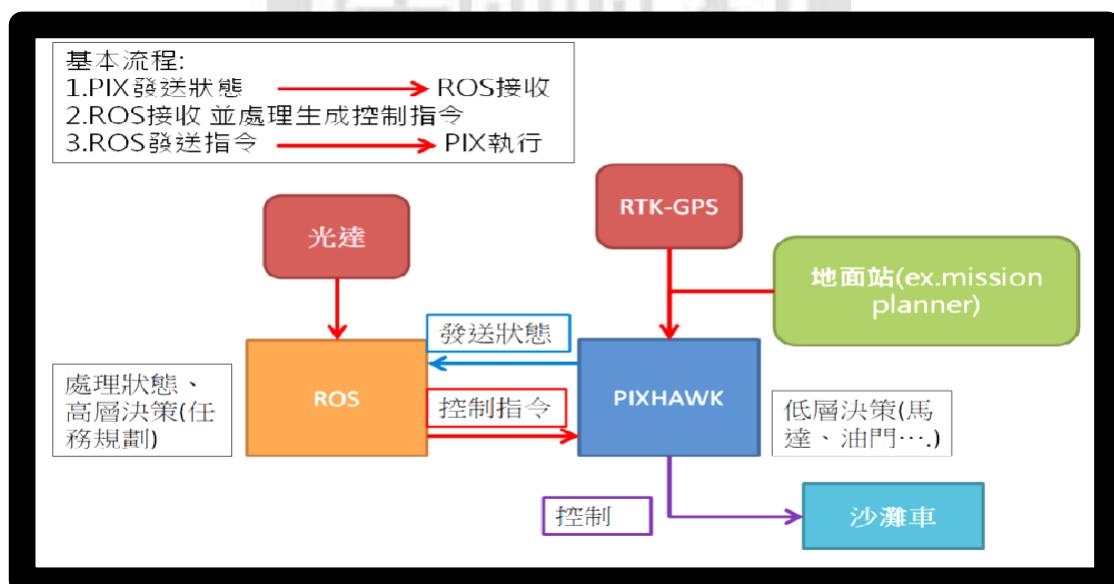


圖 2.1 基本架構

2.2 ROS 架構與模組化控制

Robot Operating System (ROS) 是一套開源的機器人軟體架構，提供模組化、分散式的開發環境。本系統採用 ROS Noetic 版本作為整合平台，透過 Publisher-Subscriber 通訊架構與多節點模組設計，使各功能模組（感測、避障、控制、資料回饋）能夠獨立運作並彼此協調如圖 2.2 所示

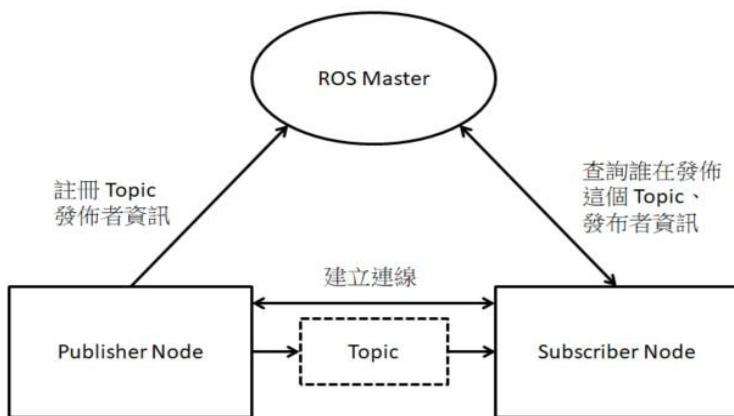


圖 2.2 ROS 系統架構圖

系統在 Gazebo 模擬 與 實車測試 採用不同的控制 Topic:

模擬：控制節點以 /cmd_vel (geometry_msgs/Twist, 20 Hz) 直接驅動 r1_rover。

實車：改以 /mavros/rc/override (mavros_msgs/OverrideRCIn, 20 Hz) 傳送 PWM，因此實體流程中不會出現 /cmd_vel。其餘感測 Topic (例如 /pointcloud2、/imu/data) 在兩種環境下皆保持相同頻率與訊息型別詳見表 2.1。

表 2-1 主要 ROS Topics、頻率與場域

Topic 名稱	Hz	ROS 訊息型別	接收節點	使用場域
/pointcloud2	10	/ PointCloud2	obstacle_detector	Sim & Real
/imu/data	50	sensor_msgs/Imu	state_estimator	Sim & Real
/cmd_vel	20	geometry_msgs/Twist	gazebo_ros_control	Sim only
/mavros/rc/override	20	mavros → pixhawk	mavros	Real only
/mavros/state	1	mavros_msgs/State	diagnostics_node	Sim & Real

在本研究中，Quanergy M8 LiDAR 以 ROS Topic 形式發佈點雲資料至感測節點，由避障邏輯模組接收後進行空間分析與行為判斷，最終透過控制節點發布控制指令至 MAVROS 與 PLC。此種模組化架構提升了系統開發靈活性與除錯效率，也利於後續功能擴充或模組更替。

2.2.1 ROS 開發工具軟體簡介

本研究採用 Robot Operating System (ROS) 作為核心整合平台，主要考量其具備高度模組化、活躍的開發社群與完整的感測器與控制器支援能力，能有效整合 Pixhawk、LiDAR、PLC 與多項運算模組。在實驗流程中，為協助系統模擬、資料驗證與除錯，本研究搭配使用 ROS 原生支援的 Gazebo 與 RViz 兩套工具。

Gazebo 是一款開源的三維動態模擬軟體，與 ROS 有良好整合能力，可模擬實體車體、環境障礙與感測器輸出。本研究選用 Gazebo 的主要原因在於其能快速建立可重複的模擬環境，便於開發初期針對避障與控制演算法進行虛擬測試，同時降低對硬體設備的依賴與損壞風險。其物理引擎與感測器模組（如 LiDAR、IMU）可模擬接近真實情境之資料輸出，便於預先驗證導航行為如圖 2.3 所示。

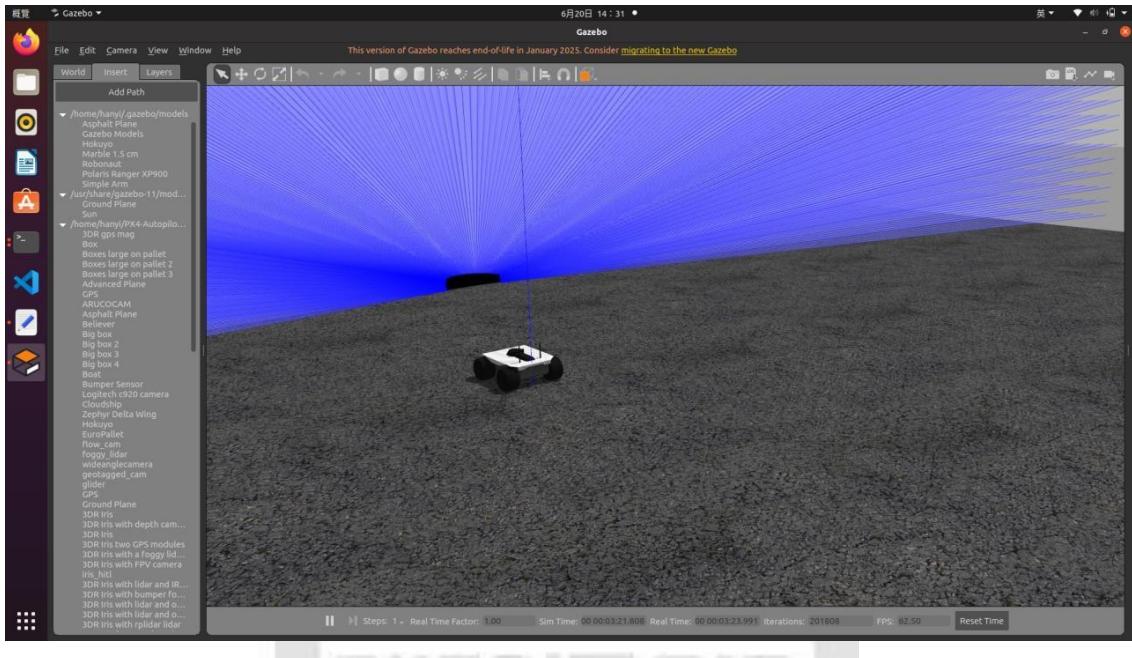


圖 2.3 Gazebo 示意圖

RViz 則為 ROS 提供的三維視覺化工具，專為感測器與控制資料的即時觀察與除錯所設計。本研究使用 RViz 作為 LiDAR 點雲視覺化、路徑觀察與 TF 坐標檢查等用途。選用 RViz 的原因為其能即時呈現 ROS 中各主題（topics）的資料流，並支援多種視覺層疊顯示，能顯著提升除錯效率與系統透明度，特別是在驗證感測器輸出與控制命令對應行為上具高度助益如圖 2.4 所示。

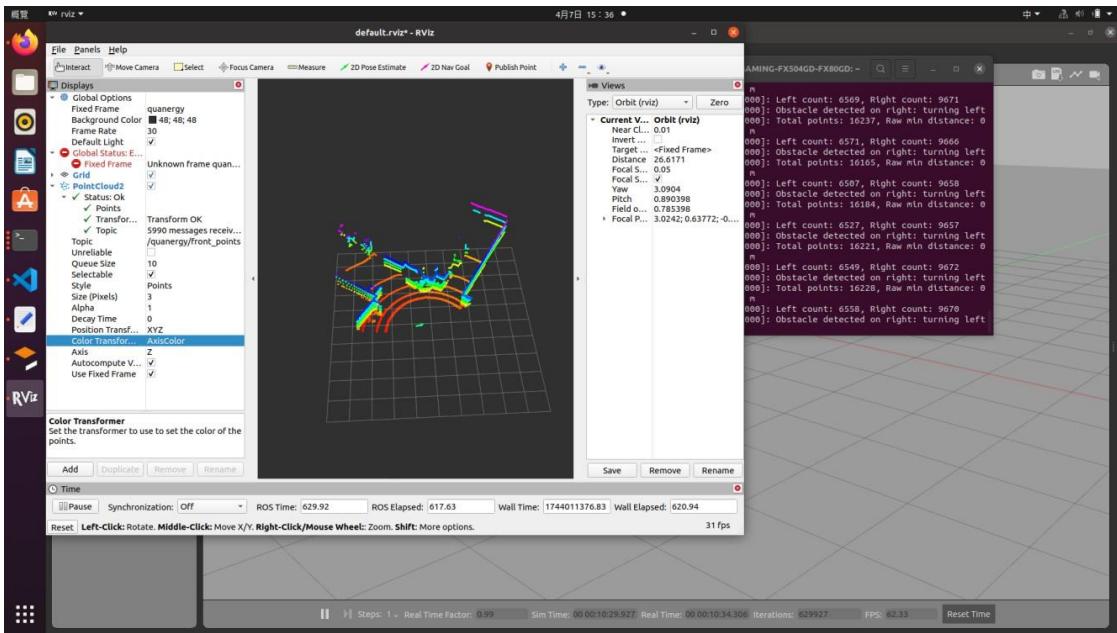


圖 2.4 Rviz 示意圖

綜上所述，Gazebo 與 RViz 的結合使用，使本研究在模擬、除錯與系統調參階段皆具備高度可視性與可控性，為後續實體整合與演算法驗證提供穩定的實驗基礎。

2.3 ArduPilot 與 MAVROS 控制整合原理

在自駕載具系統中，控制層的整合扮演關鍵角色，尤其當需同時處理感測資料、導航策略與硬體輸出時，單一架構往往無法兼顧即時性與模組彈性。因此，本研究採用 ArduPilot 作為底層控制核心，搭配 ROS 與 MAVROS 架構作為上層感知與決策平台，實現跨層級控制模組之整合。

ArduPilot 提供穩定且完整的控制邏輯，可直接驅動伺服馬達並支援多樣控制模式，而 ROS 則擅長於感測資料處理、行為規劃與模擬分析。透過 MAVROS 模組，兩者得以建立通訊橋接，使 ROS 節點得以存取 Pixhawk 傳來之即時感測資料，並發送控制命令影響載具行為，形成一套具即時反應與高度可調性的整合控制架構。

本節將分別說明 ArduPilot 開發工具與 MAVROS 在控制指令傳遞上的整合方式，以說明本研究中如何建構出一套跨平台且實用的自駕控制系統。



2.3.1 ArduPilot 開發工具軟體簡介

ArduPilot 為一套開源的飛行與地面載具控制軟體平台，支援多旋翼、固定翼、地面車輛與水上載具等多種型態的自主載具控制系統。其穩定性高、社群活躍，並具備豐富的擴充模組與參數設定功能，廣泛應用於研究、開發與教育領域。

在 ArduPilot 系統中，Mission Planner 是最常見且功能完整的地面站軟體之一，如圖 2.5 (Ground Control Station, GCS) 所示，用於與飛控模組（如 Pixhawk）連線進行任務設定、參數調整、狀態監控與資料記錄等工作。本研究即採用 Mission Planner 作為 Pixhawk 6C 的參數管理與任務規劃平台。

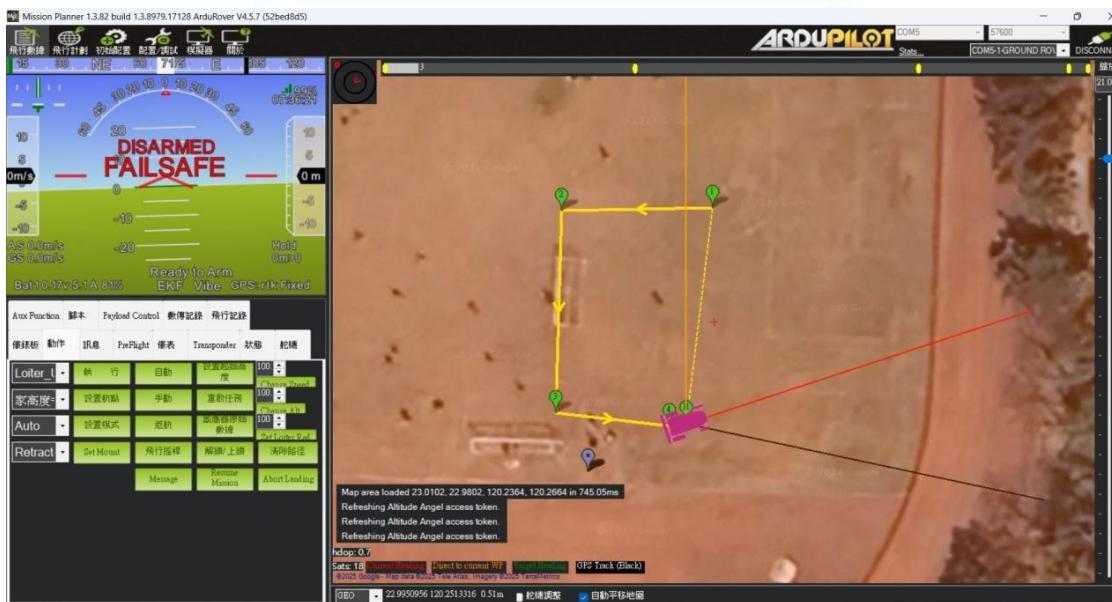


圖 2.5 Mission Planner 示意圖

Mission Planner 具備圖形化介面，能以地圖方式直觀規劃航點任務 (Waypoint Missions)，並可即時監控飛控系統回傳的資料，包括 GPS 位置、姿態角度 (Roll, Pitch, Yaw)、電壓、速度、感測器狀態等。該軟體亦支援完整的參數編輯器，可直接存取並修改 ArduPilot 韌體中的各項參數設定，如導航控制增益、感測器偏移校正與安全限制設定等。

在本研究中，Mission Planner 除了用於初期的硬體測試與飛控韌體參數設定外，亦應用於記錄載具行駛過程的飛行日誌 (Log) 資料，以進行路徑誤差分析與控制

回饋驗證。其所提供的日誌匯出與圖形比對功能，有助於後續數據分析與系統參數調整的進行如圖 2.6 所示。



圖 2.6 行駛紀錄示意圖

整體而言，Mission Planner 作為 ArduPilot 架構下的核心開發與監控工具，具備高度穩定性與完整功能，對於開發者在參數管理、任務驗證與實驗數據紀錄等方面均提供強大支援，是本研究不可或缺的重要工具之一。

2.3.2 MAVROS 控制整合原理

ArduPilot 是一套開源飛控系統，支援多平台載具控制（包括飛行器、車輛與船舶），具備多種控制模式與豐富的感測整合能力。本系統使用 Pixhawk 6C 作為控制核心，並搭載 ArduRover 韌體，以 MANUAL 與 AUTO 模式見表 2.2 實現基礎導航與手動控制切換。

表 2-2 模式比較表

分類項目	MANUAL 模式	AUTO 模式
控制邏輯	由操作者直接經由遙控器或 RC Override 控制各通道 PWM 輸出，無自動導航功能。	由飛控根據預先設定之任務航點自動執行導航與控制。
訊號來源	控制指令來自外部使用者（如遙控器或 MAVROS 發送的 Override 訊號）。	控制指令由飛控內部任務規劃模組依據任務清單自動產生。
ROS 通訊	僅透過 /mavros/rc/override 話題發送控制指令。	-
控制彈性	操作彈性高，能即時對突發狀況做出反應，適合避障與即時控制。	行為受限於任務航點，對即時狀況反應能力較低。
外部依賴	需穩定外部輸入（如 ROS 或遙控器）才能維持運作。	可獨立運行任務，即使在無外部輸入下亦可完成導航。
應用場景	實驗中用於即時避障控制，可根據 LiDAR 感測結果快速做出修正。	實驗中作為任務主導模式，執行預先規劃的航點導航行為。
資料追蹤性	僅記錄 Override 指令與即時反應，無任務紀錄功能。	可透過日誌記錄任務執行進度與航跡資訊。

系統透過 MAVROS 作為 ROS 與 ArduPilot 飛控系統之間的通訊橋接，建立 MAVLink 協議通訊連結。MAVROS 允許 ROS 節點透過話題（如 /mavros/rc override）將控制命令轉換為 MAVLink 訊息，並同步回傳飛控狀態（如 /mavros/state）、姿態資訊（如 /mavros/imu/data）與導航數據，作為系統判斷與控制之依據。

用 ArduPilot 的 AUTO 模式 執行 waypoint 任務，並於執行過程中由 ROS 節點根據 LiDAR 資訊判斷是否需進行避障。當偵測到障礙物時，系統將透過 /mavros/rc override 話題持續發送 PWM 覆寫訊號，暫時取代原任務指令進行方向或速度修正如圖 2.7 所示。此方法可避免因短暫障礙導致模式跳轉或任務中斷，且一旦無障礙物後，系統自動停止發送 override 訊號，即可無縫恢復 AUTO 模式任務執行。

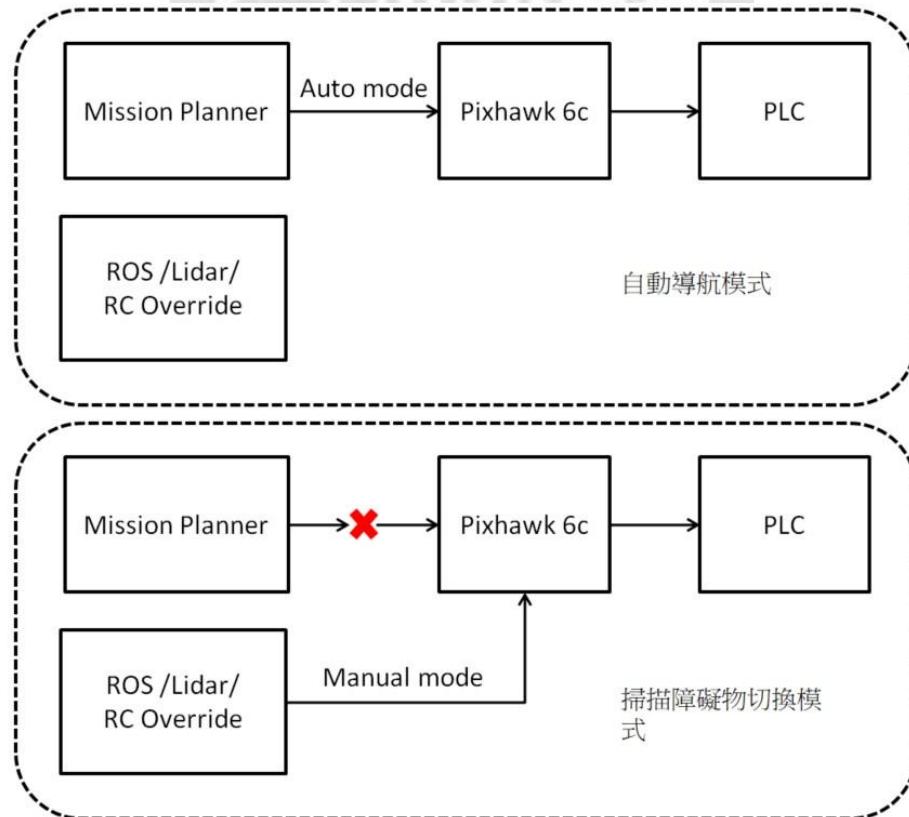


圖 2.7 模式切換流程示意

此設計相較於直接切換至 MANUAL 模式更具彈性，並可保留飛控原有的航點規劃功能，實現任務導航與即時避障之並行運作。

RC Override 機制允許外部系統（如 ROS）直接以數值方式指定各通道的 PWM 輸出值，例如通道 1 控制轉向、通道 3 控制前進油門。此方式能繞過導航層的封包解析機制，直接觸發 Pixhawk 的 PWM 輸出，反應快且具高控制自由度，特別適用於需即時避障或臨時控制接管的場景。

2.4 可程式邏輯控制器 (PLC) 控制應用

PLC (Programmable Logic Controller) 廣泛應用於工業自動化領域，具備高穩定性、即時反應與耐環境干擾的特性，適合作為低層驅動控制器。本系統使用 PLC 負責接收 ROS 及 Mission Planner 運算後之最終控制命令，並執行馬達驅動與方向轉向等實體控制任務。

由於 Pixhawk 所輸出之 PWM 訊號無法直接供 PLC 解析，本研究設計一中介轉換機制：將 PWM 訊號傳送至 Arduino Mega 2560 進行脈寬擷取與運算，再透過 DAC (數位類比轉換器) 將處理後訊號轉為電壓輸出，輸入至 PLC 之類比輸入端。PLC 依據輸入電壓範圍，執行對應邏輯，例如控制轉向角度範圍、判斷左右轉臨界點或啟動驅動馬達如圖 2.8 所示。



圖 2.8 訊號中介轉換機制流程示意

PLC 內部控制邏輯採用梯形圖撰寫，主要控制元件包括轉矩讀值（如 D508）、絕對值運算（D250）、轉向中間值（D408）與左右轉閾值（D300 與 D302）等指令組合，最終根據判斷邏輯輸出對應 I/O 信號至機構元件。

PLC 控制邏輯中，方向控制主要依據 D508 所儲存之類比電壓轉換值。首先判斷其是否小於零，若為負值代表需反轉方向輸出；反之，則取其絕對值並與方向機中間值 D408（預設為 K400）比較。若 $D508 > D408$ ，代表偏右，系統則輸出右轉訊號並清除左轉；若 $D508 < D408$ ，則輸出左轉訊號並清除右轉；若兩者相近則無動作。上述邏輯如圖 2.9 所示。

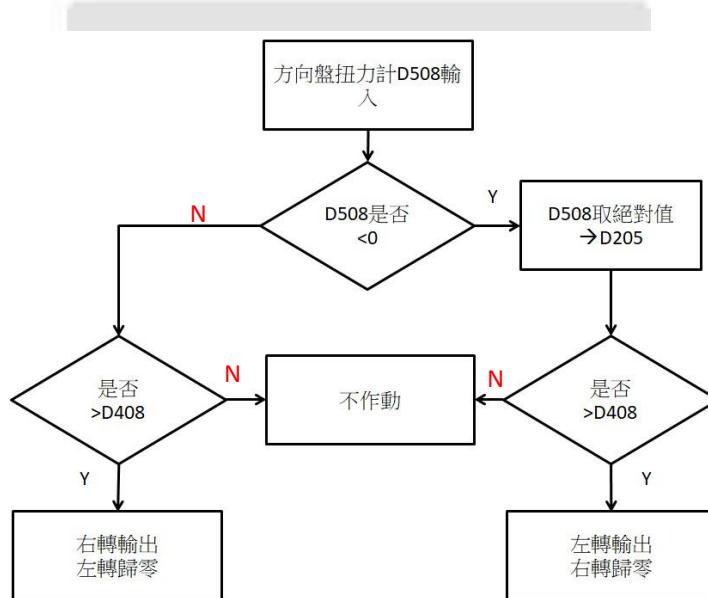


圖 2.9 基本 PLC 流程圖示意

本研究之 PLC 整合設計，使高階控制命令能順利轉換為工業級硬體可解析的控制訊號，並維持整體控制迴路之穩定性與反應速度，適合應用於戶外沙灘車等需強調環境適應性與穩定輸出的載具平台。

2.5 路徑誤差計算方法

為有效評估自駕載具於任務執行過程中，實際運行軌跡與原定導航路徑之間的偏差情形，本研究採用 Haversine 演算法計算任務點與實際軌跡點間的球面距離。儘管測試範圍僅有數十公尺，地球曲率所產生的影響相對微小，但考量計算精度與方法一致性，仍選擇此具高通用性與可靠性的球面距離計算方法。

Haversine 演算法假設地球為一理想球體（半徑約 6371 公里），透過球面三角函數推導兩組經緯座標點間之最短距離。其數學公式定義如下：

$$d = 2 r \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\varphi}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right)$$

其中 r 為表地球半徑（約 6371 公里）， φ_1 , φ_2 為兩點之緯度， $\Delta\varphi$ 為緯度之差值， $\Delta\lambda$ 為精度之差值。

在具體實作層面，本研究透過 Python 環境內 Geopy 套件提供之 geodesic() 函式，逐點計算每個任務點與實際軌跡點間的球面距離，並搭配 NumPy 函式庫進行距離資料的儲存與後續統計分析。透過此方法獲取每個導航點相對於實際軌跡之偏差，最終統計並呈現包括最大誤差、均方根誤差（RMSE）等量化指標，以客觀評估自駕載具的導航精度表現，作為後續系統性能最佳化之依據。

2.5.1 使用 Shapely 之空間誤差量測方法

為了量化自動駕駛載具實際行駛路徑與任務路徑（CMD）之間的偏差情形，本文採用 Shapely 幾何運算套件進行空間誤差分析。首先，將 CMD 所對應的經緯度座標序列建立為一條折線路徑，該折線由多個節點組成，並視為一段在二維空間中具備連續性的幾何線段。接著，針對每一個 GPS 紀錄點，透過 Shapely 提供的 `.project()` 方法，計算其在 CMD 折線上的幾何投影位置，此投影點為 GPS 點與折線距離最短之點，可能位於任一線段中，並不侷限於 CMD 節點本身。之後，利用 `.interpolate()` 方法，根據投影距離回推出折線上對應的實際投影座標如圖 2.10 所示。

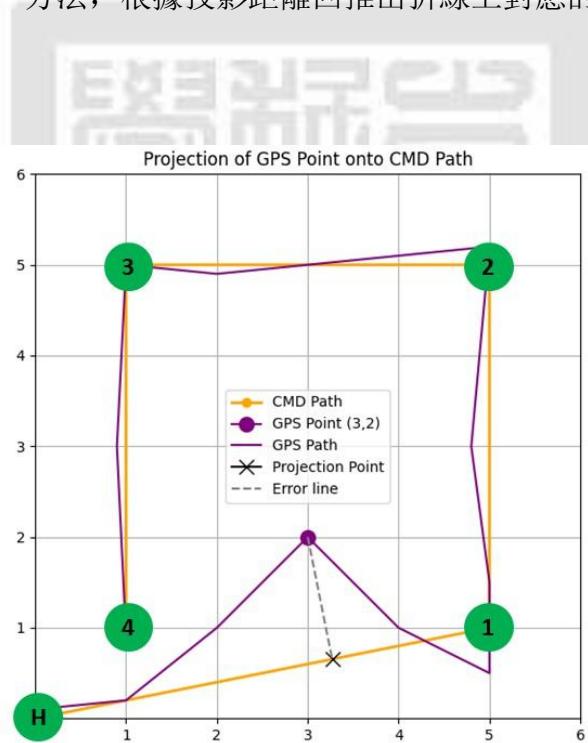


圖 2.10 路徑投影示意圖

最終再使用 Geopy 套件之 `geodesic()` 函式，計算 GPS 點與其投影點之間的距離，該距離即為該點在空間上的偏差（誤差）。藉由對所有 GPS 點進行此操作，可得出全程的最大誤差、平均誤差、均方根誤差（RMSE）等統計指標，作為路徑準確性之評估依據。

2.5.2 誤差圖表分析說明

為量化自駕載具於任務執行期間之路徑追蹤精度，本文透過兩種誤差分析方法進行系統化評估。具體方法與圖表說明如下：

1. 任務點距離誤差分析

本方法以每個 CMD 任務點作為參考基準，從載具實際行駛的 GPS 軌跡中，尋找距離該任務點最近的軌跡點並計算其間的誤差距離如圖 2.11 所示。圖中之橫軸（X 軸）為任務點編號，其中編號 1 表示起始點（H），編號 2 表示任務點 1，依序類推；縱軸（Y 軸）為誤差距離，單位為公尺。此外，圖中紅色虛線表示設定之容許誤差門檻（2 公尺）。以此圖為例，任務點 2 的最大偏離達 2.4 公尺，已超過設定之誤差門檻，因此為不合格；相較之下，任務點 4 的最大偏離為 1.4 公尺，位於容許範圍內，屬合格範圍內的定位誤差。

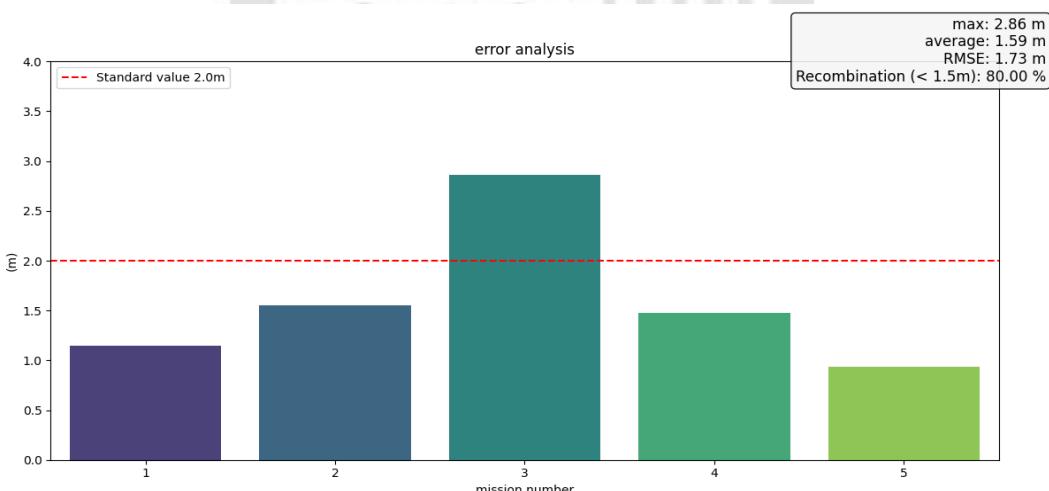


圖 2.11 任務點距離誤差分析示意

此方法的優點在於能直觀、快速地辨識出載具對於特定任務點的準確性，進而對於控制器參數與定位系統校正提供明確的調整方向。

2. 路徑重合度分析

路徑重合度分析則為另一種更全面的誤差檢驗方式，此方法將整條實際行駛的 GPS 路徑投影至 CMD 路徑折線上，並計算每個 GPS 軌跡點與對應 CMD 路徑間的最短距離，以此量化整趟任務的整體偏差狀況如圖 2.12 所示。圖中橫軸（X 軸）代表整趟任務的 GPS 軌跡資料點序列，縱軸（Y 軸）為每個軌跡點之誤差距離，單位同樣以公尺呈現。紅色虛線則為設定的容許誤差門檻（2 公尺）。依圖例可知，整趟任務有約 59% 的路徑點落在容許誤差範圍內，即表示該次任務約有四成的行駛路徑超出所允許的誤差範圍，需進一步改善。

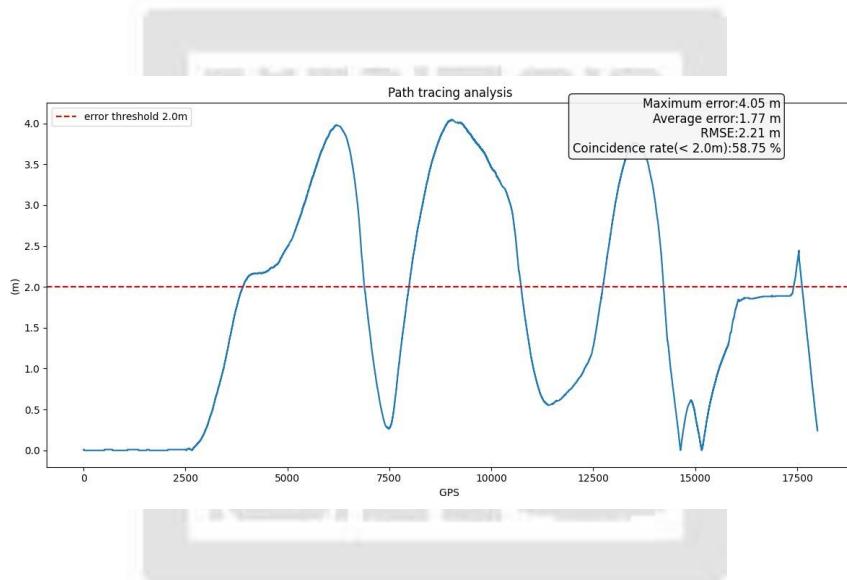


圖 2.12 路徑重合度分析示意圖

相較於第一個方法，此分析法著重於整體路徑追蹤性能的評估，而非單點誤差，能夠更有效地凸顯載具在整趟任務期間整體的追蹤穩定性與偏差特徵，對系統整體性能改善提供全面性的參考依據。

3. 路徑相似度分析

為進一步驗證 RTK-GPS 定位系統之穩定性，以及確認路徑誤差是否主要來自控制或機構因素，本研究結合上述兩種方式，透過將不同次數的任務軌跡互相找尋最近物差點並進行交互比對，觀察其相似性與差異程度。如圖 2.13 所示，圖中橫軸（X 軸）表示經序列化的 GPS 軌跡點編號，縱軸（Y 軸）為兩條路徑間的誤差距離，單位為公尺；紅色虛線代表設定的容許誤差門檻（1.5 公尺）。此外，圖表右側提供了最大誤差、平均誤差、均方根誤差（RMSE）及命中率（誤差低於 1.5 公尺之比例）等統計資訊。

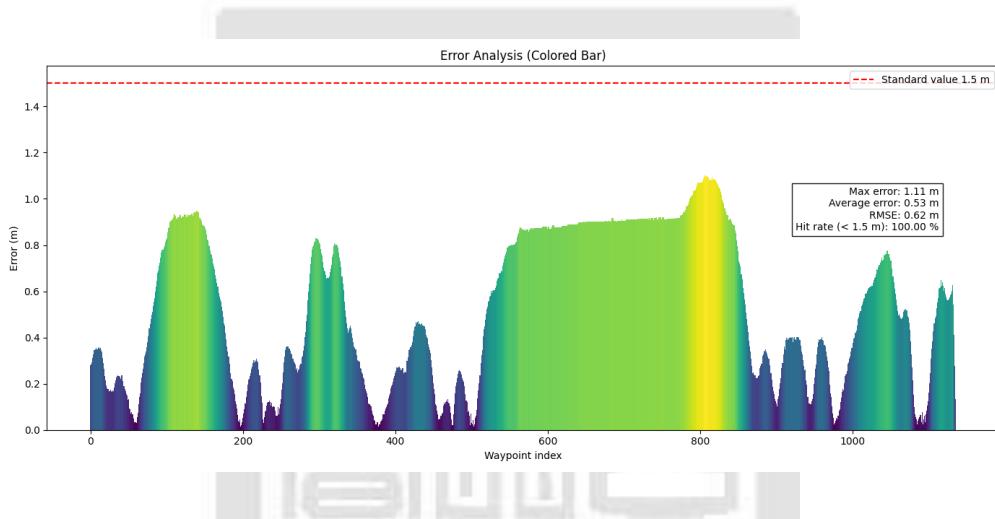


圖 2.13 路徑相似度分析示意

透過此分析方法可直觀地確認，儘管路徑經過多次重複任務執行，仍具有極高的重合度，整體誤差控制在容許範圍內（命中率達 100%）。這表示 RTK-GPS 定位系統本身的誤差影響非常有限，能有效排除 GPS 訊號本身作為路徑差異主要來源之可能性。因此，若任務執行中仍觀察到較大的路徑偏差，則應進一步針對控制參數、載具機構穩定性或其他非 GPS 因素進行調整與改進。

綜合而言，透過路徑相似度分析，不僅能有效驗證 GPS 系統本身之穩定性，更可協助明確後續載具控制策略的改良方向。

第三章 硬體選用與理論基礎

本章旨在說明本研究系統整合設計背後所依據的技術基礎與控制原理。系統以 ROS 為整合核心，透過 MAVROS 與 ArduPilot 飛控進行通訊，搭配 Quanergy M8 LiDAR 與 PLC 控制模組，實現自駕載具之感測、判斷與驅動控制流程。以下將分別針對 ROS 系統架構、ArduPilot 與 MAVROS 整合原理、PLC 控制邏輯與 PWM 訊號轉換機制進行說明。

3.1 實驗沙灘車載台

本研究之實驗載台為一輛電動驅動之小型沙灘車，如圖 3.1 所示。車體主結構由高強度鋼管焊接組成，具有輕量化與高耐用性特點，並配備適合戶外崎嶇地形之大型越野輪胎，適用於複雜戶外環境進行各項測試任務。



圖 3.1 沙灘車側視圖

沙灘車之驅動系統採用台達電子 AC 伺服驅動器（型號：ASD-A2-0421-U）搭配對應的伺服馬達，具備閉迴路控制、高反應速度及精確的速度調整能力。電力來源則使用 48V 大容量鋰電池組，提供足夠的續航能力與穩定輸出電壓，以確保車輛長時間穩定運作。

此外，本載台原本之駕駛座椅已拆卸並進行改裝，成為 ROS 控制系統與主控電腦的安裝平台，以利相關電子設備之穩定放置與線路配置如圖 3.2 所示。



圖 3.2 沙灘車俯視圖

透過以上硬體配置，本載台具備優異的動態性能及環境適應性，適合作為戶外自主導航、路徑規劃與避障功能測試之實驗平台。提供相關規格見表 3-1。

表 3-1 沙灘車硬體規格表

項目	規格說明
車體結構	高強度鋼管焊接結構
輪胎規格	越野輪胎
驅動方式	台達電子 AC 伺服馬達（750W-A2）與驅動器（ASD-A2-0421-U）
轉向機構	台達電子 AC 伺服馬達（400W-A2）與電阻尺（KPM22-5 mm）
電池容量	48V 鋰電池組

3.1.1 沙灘車載台速度測試

本載具具備 9 段速度檔位，而執行任務時之行駛速度將直接影響導航路徑的穩定性與任務完成品質，因此有必要建立一套標準化的速度模式作為後續測試之依據。

研究採用如圖 3.3 所示之測試方式，於場地中央規劃一條簡單直線路徑作為任務目標，並透過 Pixhawk 自動導航功能執行該路線，以觀察不同速度檔位下之行駛表現與完成時間。測試過程中，載具於每一檔位下皆重複進行多次任務執行，並記錄實際完成時間與路徑偏差等相關數據，作為後續控制參數優化與性能評估之依據。



圖 3.3 測試路線圖

測試過程中，針對本載具所具備之 9 段速度檔位，分別進行三次任務執行並取其平均值，以估算各檔位之實際平均行駛速度。測試結果如表 3-2 所示，可觀察到自檔位 3 以後，速度增幅趨於平緩，顯示高檔位與實際行駛速度之間已無顯著差異。故綜合考量效率與穩定性，後續戶外性能測試將統一採用檔位 9 進行，以確保速度表現達最大化，相關示意見圖 3.4 紅圈標註處。

表 3-2 載具速度表

速度(s)	V(m/s)	V(km/m)
1	0.22	0.79
2	0.24	0.85
3	0.54	1.94
4	0.71	2.56
5	0.76	2.74
6	0.78	2.79
7	0.80	2.87
8	0.82	2.97
9	0.82	2.96

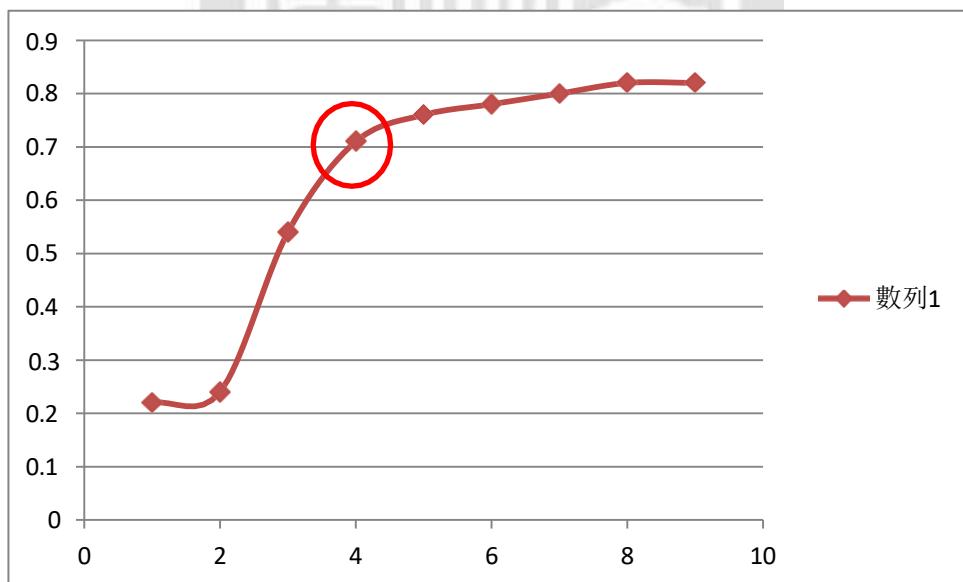


圖 3.4 檔位/速度圖示意

3.2 硬體架構與元件選用說明

系統以 ROS 為整合核心，透過 Mavros 與 ArduPilot 之 Pixhawk 6C 飛控模組建立通訊連接；感測資料由 Quanergy M8 LiDAR 傳入 ROS 系統進行處理後，由 Pixhawk 6C 輸出 PWM 訊號至 Arduino mega 2560，並經 DAC 模組轉換為電壓訊號輸入 PLC，完成整體控制迴路，如圖 3.5 示。詳細規能對照表見表 3-3

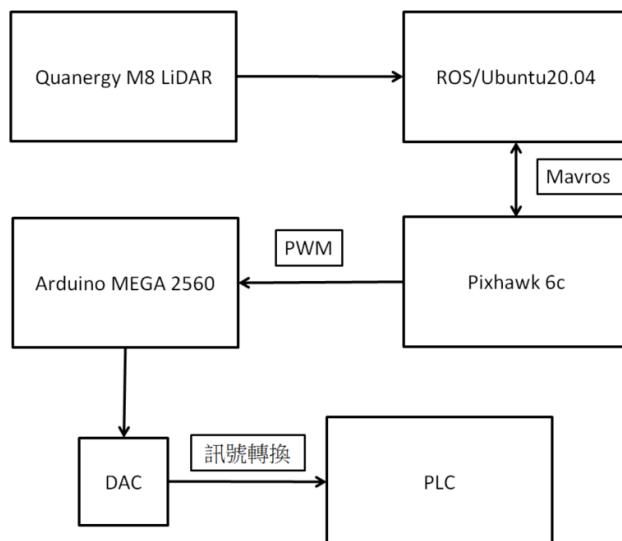


圖 3.5 統硬體整合架構圖

表 3-3 硬體規格與功能對照表

硬體名稱	型號 / 廠牌	通訊介面 / 協定	備註說明
飛控系統	Pixhawk 6C (ArduRover)	MAVLink (UART)	運行 ArduRover 韌體
感測器	Quanergy M8	UDP (PointCloud2)	安裝於車體前方
媒介轉換	Arduino Mega 2560	PWM 輸入 / I ² C 輸出	連接 Pixhawk 與 DAC
DAC 模組	DFRobot GP8403	I ² C	輸出給 PLC 的類比電壓
工業控制器	PLC (型號可補)	類比電壓輸入 (0~5V)	Ladder 程式撰寫
無線遙控模組	Holybro Radio Telemetry	MAVLink	使用 USB 連線至地面站

本系統所採用之主要硬體模組如下：

(1) Quanergy M8 LiDAR： 本研究採用美國 Quanergy 公司開發之 M8 三維光達 (3D LiDAR) 作為前向環境感測元件，外觀如圖 3.6 示。該裝置具備 8 條雷射發射線，能夠以 360 度水平旋轉方式進行掃描，掃描頻率可達 20 Hz(系統預設 10Hz)，雷射發射率為 53,828 Hz，具有高解析度、高密度、即時性的空間感知能力。M8 採用飛行時間 (Time-of-Flight, TOF) 測距原理，具備多重回波 (multiple returns) 功能，可同時輸出每個點的水平角度、垂直角度、距離與反射強度，進而構成稠密點雲資訊，特別適用於複雜戶外環境下進行障礙物辨識與輪廓建構。本系統中，M8 LiDAR 透過乙太網路連接至主控電腦，並結合 Quanergy 官方提供之 ROS 驅動程式實現資料整合。點雲資料發佈至 /pointcloud2 主題後，由自訂 ROS 節點進行即時處理與分析。該感測器於高反射干擾或強烈陽光照射等環境條件下，仍能維持穩定輸出，為本研究之避障決策模組提供高可靠度之前方環境感知依據。



圖 3.6 Quanergy M8 LiDAR

(2) Pixhawk 6C：該模組為美國 Holybro 公司推出之高性能飛行控制器，專為無人載具與自駕系統應用設計，支援多種開源自動駕駛軟體，包括 PX4 與 ArduPilot。外觀見圖 3.7。該模組內建高精度六軸與三軸 IMU（Inertial Measurement Unit）組件、氣壓計（Barometer）、磁力計與溫度感測器，可實現穩定的姿態估測與高度感知能力。Pixhawk 6C 提供多組 PWM 輸出通道、UART、CAN、I²C、SPI 等通訊介面，可廣泛連接 GPS、電調（ESC）、遙控接收器、光達、攝影機等周邊裝置。此外，控制器內建 STM32H743 主控晶片，具備 480 MHz 高速處理能力，適用於複雜任務控制與多感測器資料處理需求。在通訊協定方面，Pixhawk 6C 支援 MAVLink（Micro Air Vehicle Link）協定，並可透過 MAVROS 將其與 ROS（Robot Operating System）進行整合，使機器人系統可直接存取飛控狀態、位置資訊與姿態估測等數據。透過此整合方式，可實現 ROS 節點對飛控進行任務派送（如導航點設定）、狀態監控與控制回饋，亦有助於整合更多自訂感測模組與高階決策系統。



圖 3.7 Pixhawk 6C

(3) RTK F9P GPS：搭載高精度 RTK 功能之多頻 GNSS 模組（如 u-blox F9P）可同時接收 GPS、GLONASS、Galileo、BeiDou 等多種衛星訊號，並具備即時動態定位解算（RTK Fix）能力，在基地台差分訊號輔助下，能提供最高可達公分級的絕對定位精度。此模組可透過 UART 或 USB 介面與 Pixhawk 或主控電腦進行通訊，並輸出標準 NMEA(National Marine Electronics Association) 格式之定位資料。當 GNSS 模組接入 Pixhawk 控制器後，ArduPilot 將其資訊納入 EKF (Extended Kalman Filter) 融合架構中進行姿態與位置估測，進而提供穩定且高精度的導航依據。若 GNSS 模組改接入 ROS 主控電腦，亦可透過 MAVROS 或自行開發之接收節點接收定位資訊並進行處理，應用於自動導航、軌跡紀錄或狀態監控等任務，外觀如圖 3.8 所示。



圖 3.8RTK-F9P 模組

(4) Arduino Mega 2560：在本研究中扮演 PWM 訊號擷取與數位轉類比轉換的中介角色，其具備多達 54 組數位輸入輸出（I/O）腳位與多組計時器模組，適合用於處理高頻訊號與時間相關的控制任務。於本系統中，Arduino 接收由 Pixhawk 6C 所輸出的 RC Override PWM 訊號，並利用內部計時器計算 PWM 脈波寬度，轉換為對應的整數數值。此數值再依照事先定義之比例關係對應至電壓範圍（0~5V），並透過 I²C 通訊傳輸至 DAC 模組，進一步輸出為類比電壓供 PLC 使用。此轉換過程不僅解決 Pixhawk PWM 輸出與 PLC 類比電壓輸入之介面不相容問題，亦確保整體轉換延遲低、資料穩定，能即時反映 ROS 所計算之導航決策指令（如圖 3.9 所示）。



圖 3.9 Arduino Mega 2560

(5) DAC 模組：本系統選用 DFRobot 所生產之 GP8403 DAC 模組作為數位轉類比（Digital-to-Analog Conversion）元件，外觀如圖 3.10。其具備雙通道 12-bit 分辨率輸出能力，可透過 I²C 通訊介面與 Arduino Mega 2560 進行資料交換。經由 PWM 訊號處理後所得之整數數值，經由 Arduino 計算對應比例後傳送至 DAC 模組，再由 DAC 輸出對應之電壓範圍（0V 至 5V）。此類比電壓將作為 PLC 類比輸入模組之控制參數，用以控制伺服馬達的驅動強度與方向判斷等功能。該模組具備穩定輸出與低雜訊特性，可有效對應即時變化之控制命令，並確保輸出訊號準確一致，不會對下游 PLC 造成邏輯誤判或失控風險。



圖 3.10 DAC 模組

(6) Holybro 無線電遙測模組: 本研究採用 Holybro 所推出之 SiK 無線電遙測模組 (Telemetry Radio V3)，其支援 MAVLink 協定並具備高穩定性與中長距離的無線通訊能力。該模組主要用途為建立 Pixhawk 飛控模組與地面站之間的橋接通訊通道，允許使用者透過 QGroundControl (QGC)或 Mission Planner 進行參數設定、任務上傳與即時監控外觀如圖 3.11 所示。



圖 3.11 Holybro 無線電遙測模組(915)

(7) PLC 控制器：作為車體驅動系統的核心控制元件，負責執行轉向電機與驅動馬達等實體控制任務。其具備高度穩定性與可程式邏輯控制能力，並支援多組數位與類比 I/O 通道，能穩定應對戶外惡劣環境下的即時控制需求，適合應用於重視可靠性與耐環境性的自駕平台如圖 3.12 所示。

資料來源：

<https://www.deltaww.com/zh-tw/products/PLC-Programmable-Logic-Controllers/49>



圖 3.12 台達 DVP-SX2 系列

(8) 主控電腦平台：為 ASUS ROG Strix，安裝 Ubuntu 作業系統與 ROS Noetic，負責各模組資料處理與控制指令生成，為整體系統運算核心詳見表 3-4。

表 3-4 主控電腦規格表

處理器	Intel Core i7-8750H (第 8 代, 6 核心)
記憶體	8 GB DDR4
儲存裝置	250 GB SSD
顯示卡	NVIDIA GeForce GTX 1050 Ti (4GB GDDR5)
作業系統	Ubuntu 20.04 LTS
用途	為 ROS 系統開發、Lidar 點雲處理與 PX4 / MAVROS 模擬測試主要平台

3.3 系統整合方法

延續 2.1 所述系統架構，本節將說明各層級間的具體整合方式與資料流程。感測層使用 Quanergy M8 3D LiDAR 蒐集環境點雲資訊，並透過 ROS 進行資料處理與障礙辨識；決策層以 ROS 為基礎平台，負責避障邏輯判斷與控制命令生成；執行層則由 Pixhawk 6C 與 PLC 組成，分別處理導航控制與馬達驅動輸出。整體架構如下：LiDAR → ROS 感測與避障節點 → 控制指令 → Pixhawk/PLC 雙向執行如圖 3.13 所示。

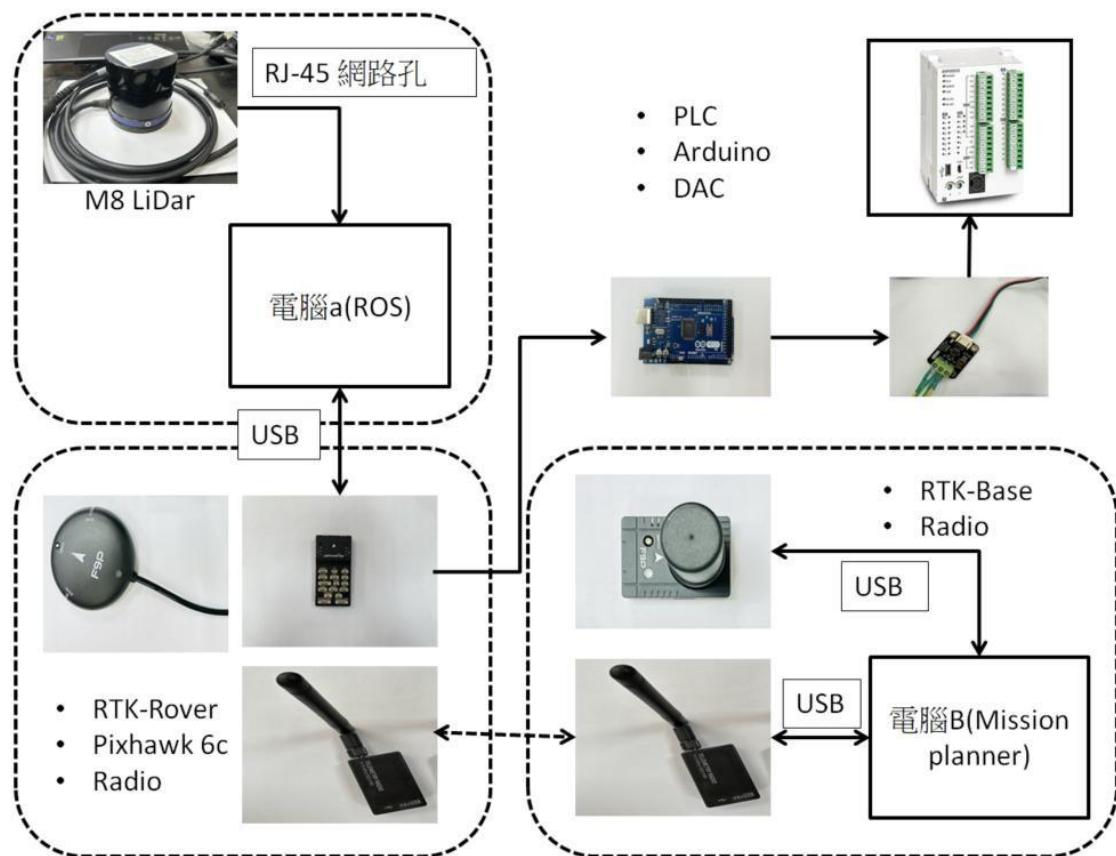


圖 3.13 系統整合示意圖

3.3.1 ROS 與 LiDAR 資料處理流程

Quanergy M8 透過網路與主機連接後，ROS 系統可透過官方提供的驅動程式將資料發佈至/`pointcloud2` 主題。接著使用 PCL 函式庫進行資料裁切、障礙物區塊判斷，提取出車輛前方與左右側的高密度障礙資訊如圖 3.14 所示。該資料再輸入至避障判斷模組，決定轉向或減速邏輯。

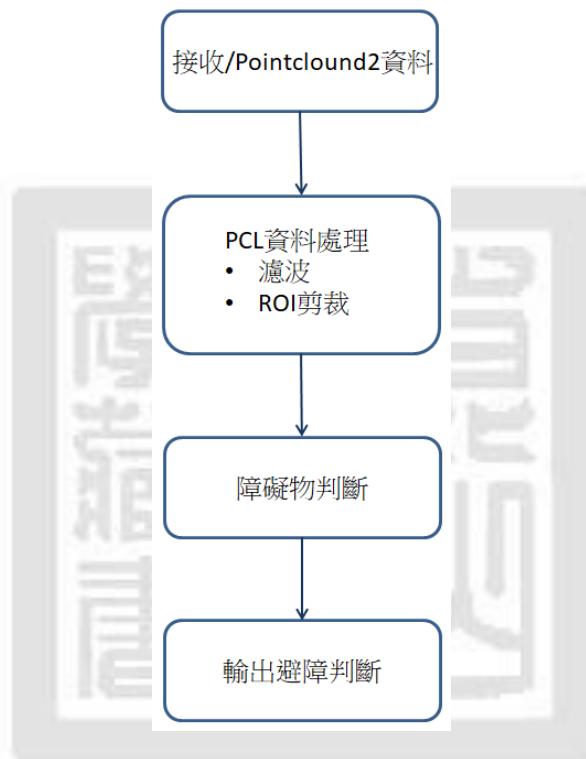


圖 3.14 LiDAR 資料處理流程圖

3.3.2 Pixhawk 與 ROS 整合機制

系統選用 Pixhawk 6C 作為飛控控制核心，並搭載 ArduPilot 韌體。透過 MAVROS 與其建立 MAVLink 通訊通道，可讀取飛行模式 (/mavros/state)、姿態資訊與速度命令等資料。考量 GUIDED 模式需額外提供角速度資料，且可能影響 RC Override 功能穩定性，因此本研究將飛控模式設為 MANUAL，並透過 RC Override 訊號控制左右輪 PWM 輸出，達成轉向與行進控制。

3.3.3 PLC 與 Pixhawk 對接控制邏輯設計

本研究設計一訊號轉換模組，將 ROS 產生的導航控制命令，以 Pixhawk 6C 透過 RC Override 方式發送 PWM 訊號至 Arduino Mega 2560。Arduino 摷取 PWM 訊號並即時轉換為數位值後，再以 DAC 模組轉成相應的 0–5 V 電壓訊號，供 PLC 類比輸入模組讀取。為清楚呈現此訊號轉換流程，本研究針對「全輸出、未輸出、左轉、右轉」四種控制情境分別記錄 Arduino 之 PWM 轉換數值（圖 3.15），



```

48 //----- 47 | val2 = map(duration_1, 1080, 1878, 0, 255);
49 dac.setDacOutVoltage(val1, 0); 48 | //-----
50 dac.setDacOutVoltage(val2, 1); 50 | dac.setDacOutVoltage(val1, 0);
51 delay(1); 51 | dac.setDacOutVoltage(val2, 1);
52 dac.store(); 52 | delay(1);
53 //----- 53 | dac.store();
54 //----- 54 | //-----
55 //analogWrite(min_val); 55 | //analogWrite(max_val);

```

輸出 序列埠監控窗 X 輸出 序列埠監控窗 X

訊息 (按 Enter 發送訊息發送到 COM8 上的 Arduino Mega or Mega 2560) 訊息 (按 Enter 發送訊息發送到 COM8 上的 Arduino Mega or Mega 2560)

```

5056 5063 1893 1887 5056 5056 1887 1887 5056 5056

```

(a)

(b)

```

48 //----- 46 | val2 = map(duration_1, 1080, 1878, 0, 255);
49 dac.setDacOutVoltage(val1, 0); 47 | }
50 dac.setDacOutVoltage(val2, 1); 48 | //-----
51 delay(1); 50 | dac.setDacOutVoltage(val1, 0);
52 dac.store(); 51 | dac.setDacOutVoltage(val2, 1);
53 //----- 52 | delay(1);
54 //----- 53 | dac.store();
55 //analogWrite(min_val); 54 | //-----
55 //analogWrite(max_val); 55 | //analogWrite(min_val);

```

輸出 序列埠監控窗 X 輸出 序列埠監控窗 X

訊息 (按 Enter 發送訊息發送到 COM8 上的 Arduino Mega or Mega 2560) 訊息 (按 Enter 發送訊息發送到 COM8 上的 Arduino Mega or Mega 2560)

```

5053 44 1897 1894 5056 44 1887 1893 5056 44 5056

```

(b)

(d)

圖 3.15 Arduino PWM 訊號轉換數值監控畫面：(a) 全輸出，兩組 PWM 值皆達最高；(b) 未輸出，兩組 PWM 值皆為最小值；(c) 左轉，PWM 值呈現左高右低趨勢；(d) 右轉，PWM 值呈現右高左低趨勢。

以及 PLC 階梯圖之接收電壓與相應的控制邏輯反應（圖 3.16），從而驗證高層 ROS 指令經此轉換模組後能準確落實於 PLC 之實際控制端。

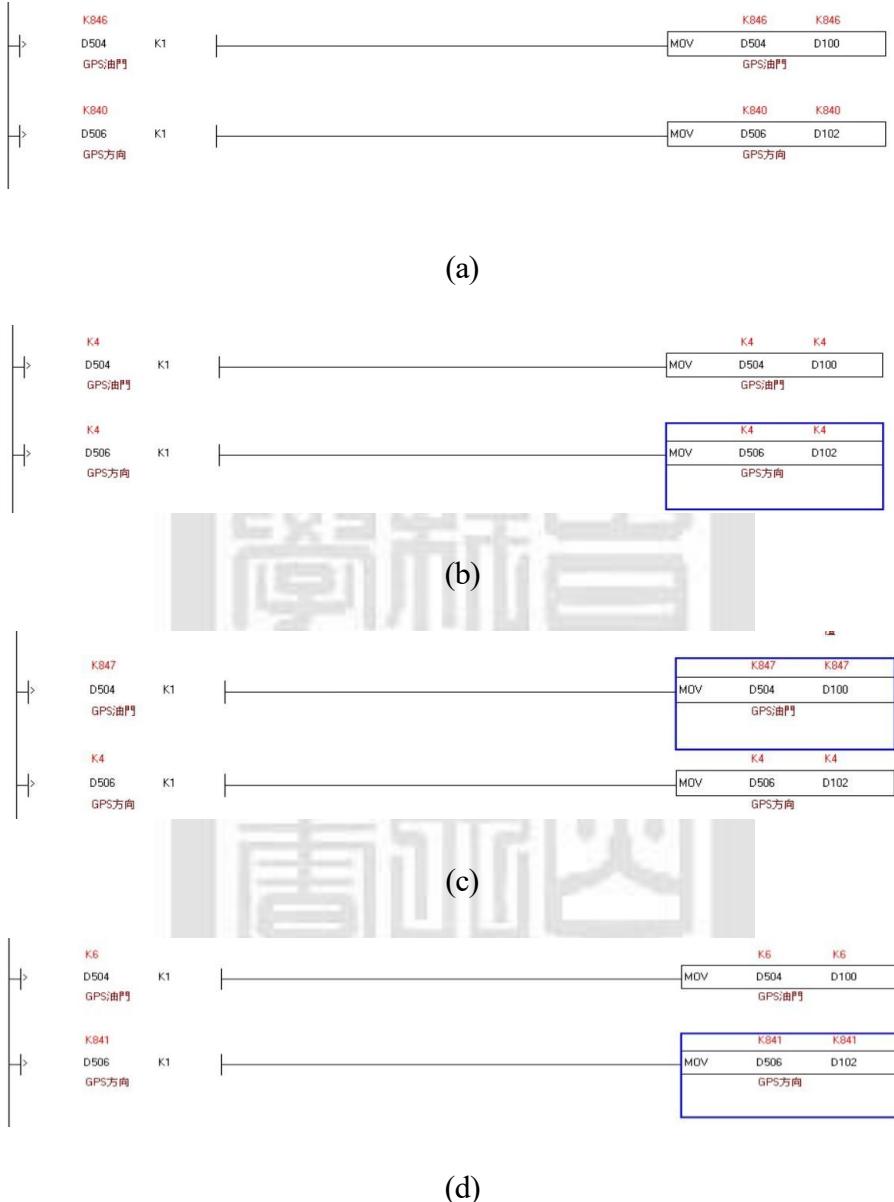


圖 3.16 PLC 階梯圖對應 Arduino DAC 輸出電壓之控制狀態：(a) 全輸出，馬達與轉向同時動作；(b) 未輸出，馬達與轉向皆靜止；(c) 左轉，僅啟動左向轉向邏輯；(d) 右轉，僅啟動右向轉向邏輯。

第四章 模擬分析

4.1 系統環境建置與初步測試

為了確保 ROS、Pixhawk 及 PLC 等多個控制架構整合的可行性，本研究先行在個人電腦上以 Ubuntu 作業系統分別進行各子系統的局部模擬與初步驗證，待各子系統的運行穩定且測試結果符合預期後，再逐步進行跨系統整合，以降低後續實體測試可能面臨的複雜性與風險。本節首先針對實驗中使用的 Quanergy M8 LiDAR 進行驅動安裝與參數設定，透過 ROS 內建的 RViz 工具來檢視光達所輸出的掃描點雲資料，以驗證感測器與 ROS 環境間的資料傳輸是否正常。

並於 Gazebo 模擬器中建立並配置 Rover_r1 無人載具模型，透過設定合適的物理與感測參數，確認載具在虛擬環境內的基本運動特性以及感測器資料回饋的準確性與穩定性。最後，本測試亦針對 Pixhawk 6C 飛控系統進行軟硬體設定，透過 MAVROS 套件連結 Pixhawk 與 ROS 系統，初步測試通訊與指收回饋的穩定性，確認飛控系統可正常接受來自 ROS 的命令並提供回傳資訊，以作為後續跨系統整合與自主導航功能實現的基礎。

4.1.1 Quanergy M8 LiDAR 安裝與 RViz 顯示驗證

本研究採用 Quanergy 公司所生產之 M8 系列 LiDAR 作為主要感測器，用以實現載具的即時障礙物偵測與自主避障功能。Quanergy M8 LiDAR 具備 360°全周掃描能力，最大掃描頻率可達 20 Hz，可提供高精度且即時的三維點雲資訊，適合應用於自主導航載具的環境感知任務。

首先在 Ubuntu 環境中安裝 Quanergy 官方提供的 ROS 驅動程式。由於 Quanergy 公司已將 M8 LiDAR 的軟體開發套件（SDK）開源於 GitHub，方便使用者進行系統整合與後續開發，因此本研究透過該平台取得官方最新版本的 SDK 與相關套件，並加以建置與編譯。接著，透過 TCP/IP 協議在 LiDAR 與 ROS 系統之間建立資料傳輸通道，以確保資料傳輸的穩定性與即時性。

完成軟體安裝與通訊設定後，透過 ROS 內建之視覺化工具 RViz 進行驗證測試，成功顯示並確認 LiDAR 所產生的三維掃描點雲資料如圖 4.1 所示，並在 RViz 介面內即時呈現實驗環境內障礙物的空間分布與輪廓特徵。經觀察，Quanergy M8 LiDAR 的資料具備良好的穩定性與準確性，充分滿足後續系統整合、Gazebo 模擬及實際載具測試階段對環境感測與即時避障性能之需求。

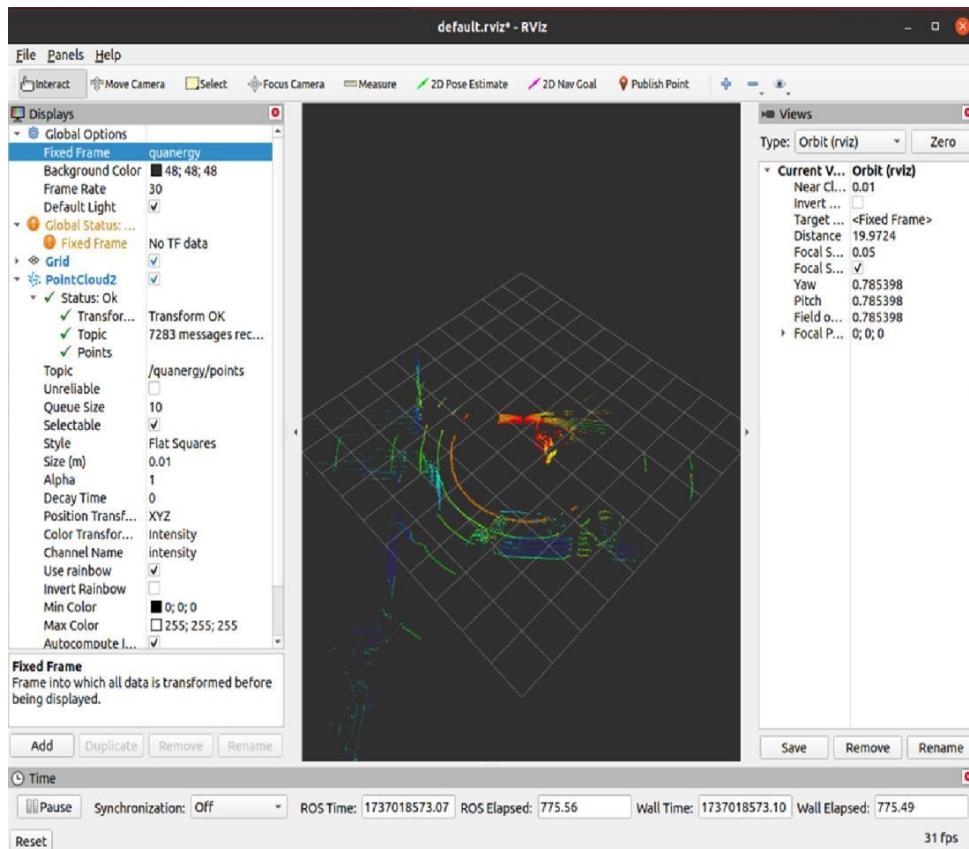
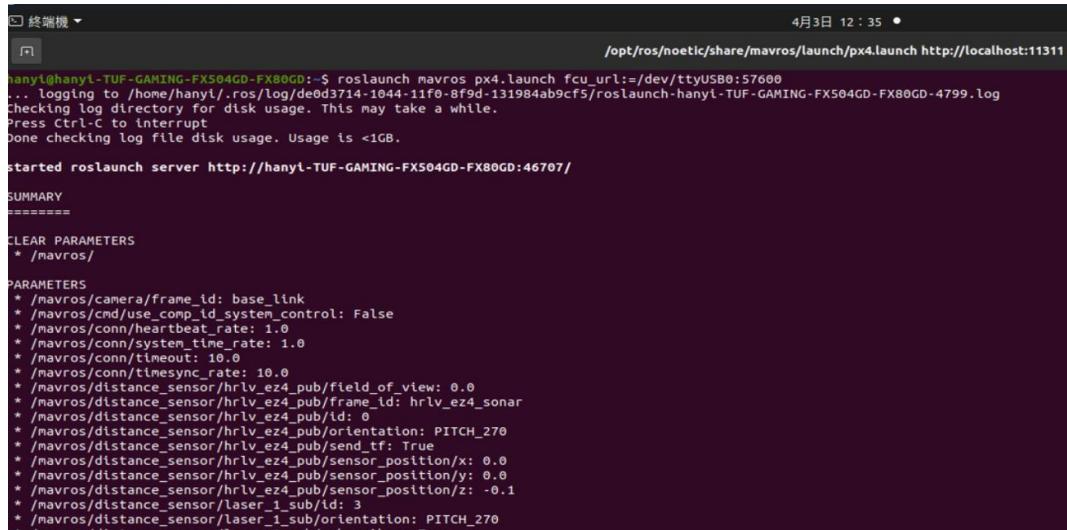


圖 4.1 Quanergy M8 LiDAR 點雲資料呈現

4.1.2 Pixhawk 6C 與 ROS 系統通訊驗證

為實現自駕載具的控制指令下達與回傳監控，Pixhawk 6C 飛控系統與 ROS 之間的通訊透過 MAVROS 模組進行整合。MAVROS 作為 ROS 與飛控之間的橋接節點，負責轉譯 ROS 訊息為 MAVLink 協議格式，使雙方能順利交換導航指令、狀態資訊與感測資料。系統啟動時，透過指令 `rosrun mavros px4.launch fcu_url:=/dev/ttyUSB0:115200` 成功連線後如圖 4.2 所示，即可觀察到 MAVROS 節點正確載入並初始化，包括心跳訊號（heartbeat）、系統時間同步（timesync）、感

測器參數設定等多項關鍵資訊。終端機輸出顯示各項參數已被正確掛載，並持續更新，證明 Pixhawk 與 ROS 之間的雙向通訊運作正常。

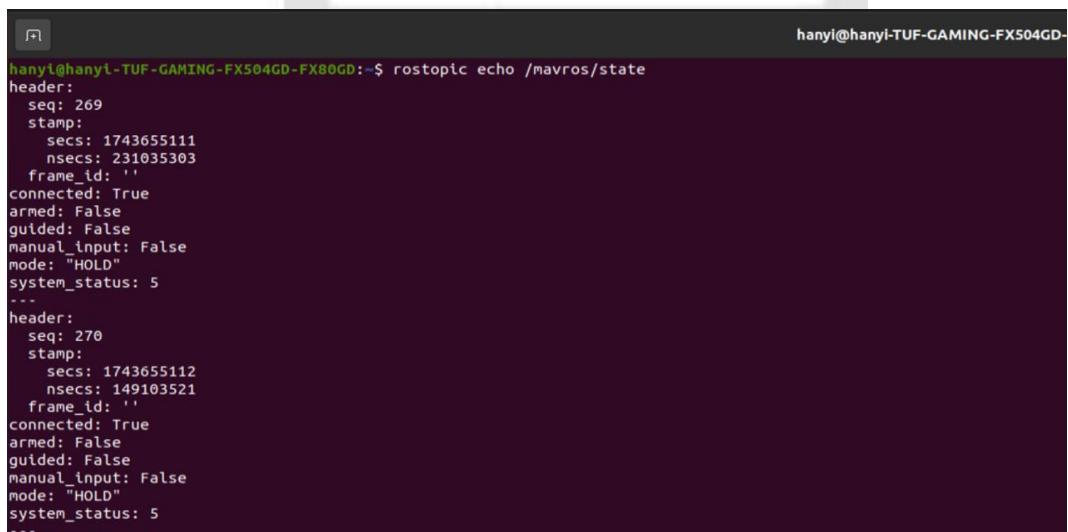


```
hanyi@hanyi-TUF-GAMING-FX504GD-FX80GD:~$ roslaunch mavros px4.launch fcu_url:=/dev/ttyUSB0:57600
... logging to /home/hanyi/.ros/log/de0d3714-1044-11f0-8f9d-131984ab9cf5/mavros.launch-hanyi-TUF-GAMING-FX504GD-FX80GD-4799.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://hanyi-TUF-GAMING-FX504GD-FX80GD:46707/
SUMMARY
=====
CLEAR PARAMETERS
 * /mavros/
PARAMETERS
 * /mavros/camera/frame_id: base_link
 * /mavros/cmd/use_comp_id_system_control: False
 * /mavros/conn/heartbeat_rate: 1.0
 * /mavros/conn/system_time_rate: 1.0
 * /mavros/conn/timeout: 10.0
 * /mavros/conn/timesync_rate: 10.0
 * /mavros/distance_sensor/hrlv_ez4_pub/field_of_view: 0.0
 * /mavros/distance_sensor/hrlv_ez4_pub/frame_id: hrv_ez4_sonar
 * /mavros/distance_sensor/hrlv_ez4_pub/id: 0
 * /mavros/distance_sensor/hrlv_ez4_pub/orientation: PITCH_270
 * /mavros/distance_sensor/hrlv_ez4_pub/send_tf: True
 * /mavros/distance_sensor/hrlv_ez4_pub/sensor_position/x: 0.0
 * /mavros/distance_sensor/hrlv_ez4_pub/sensor_position/y: 0.0
 * /mavros/distance_sensor/hrlv_ez4_pub/sensor_position/z: -0.1
 * /mavros/distance_sensor/laser_1_sub/id: 3
 * /mavros/distance_sensor/laser_1_sub/orientation: PITCH_270
 * /mavros/distance_sensor/laser_1_sub/subscriber: True
```

圖 4.2 Pixhawk 6c 與 ROS 建立溝通

此外，透過 rqt_graph 或 rostopic echo 等工具，也能即時監控 MAVROS 發佈的飛控資訊與回傳資料，包括飛行模式、IMU 訊息、GPS 定位、姿態角度等圖 4.3 所示，整體驗證結果顯示，Pixhawk 6C 可穩定地與 ROS 進行連線與資料互動，系統整合狀況良好。



```
hanyi@hanyi-TUF-GAMING-FX504GD-FX80GD:~$ rostopic echo /mavros/state
header:
  seq: 269
  stamp:
    secs: 1743655111
    nsecs: 231035303
  frame_id: ''
connected: True
armed: False
guided: False
manual_input: False
mode: "HOLD"
system_status: 5
---
header:
  seq: 270
  stamp:
    secs: 1743655112
    nsecs: 149103521
  frame_id: ''
connected: True
armed: False
guided: False
manual_input: False
mode: "HOLD"
system_status: 5
---
```

圖 4.3 MAVROS 發佈的飛控資訊

4.2 Gazebo 模擬環境建置與初步驗證

為了在實體測試前進行安全且可控的系統模擬，使用 ROS 環境中建置 Gazebo 模擬平台，作為開發與測試自駕載具導航與避障功能的虛擬場域。Gazebo 具備高擬真度的物理模擬引擎，能有效模擬不同地形、障礙物配置與感測器行為，並支援與 ROS 系統即時資料交換，利於進行虛實整合測試。

模擬平台中所使用的載具模型為 r1_rover 如圖 4.4 所示，具備左右輪驅動結構與基本 lidar 感測器插件，可模擬地面載具之基本運動行為。透過整合 ROS 控制節點與 Gazebo 的 Plugin，實現對虛擬載具的速度控制、姿態讀取及位置回報等功能，建立完整的控制回路。

資料來源: https://docs.px4.io/main/en/frames_rover/aion_r1.html



圖 4.4 r1_rover 模擬載具

初步驗證階段，先於 Gazebo 中設置簡單環境與障礙物配置，搭配 RViz 視覺化工具即時觀察模擬載具的運動狀態與感測器資料是否與預期一致如圖 4.5 所示。測試內容包含直線行駛、轉彎運動與靜態避障反應等情境，驗證模擬平台的正確性與穩定性。測試結果顯示，Rover_r1 載具模型可穩定運作於模擬環境中，感測器資料更新即時且與實體配置相符，為後續整合 Quanergy M8 LiDAR 與 Pixhawk 飛控系統的模擬避障測試提供良好基礎。

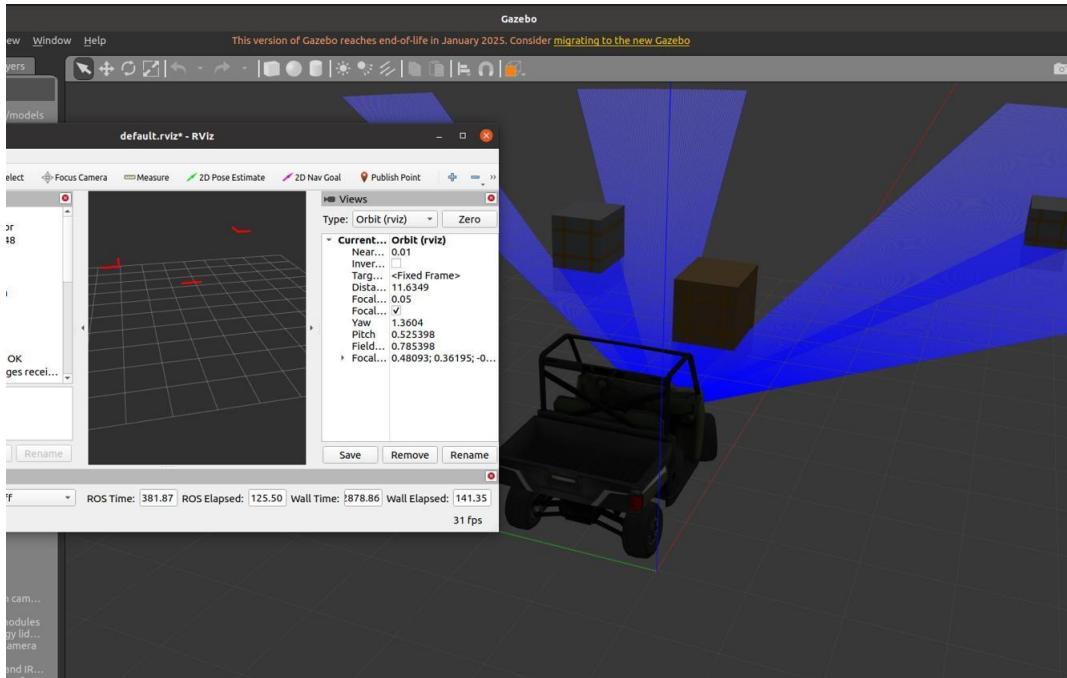


圖 4.5 Gazebo 搭配 RViz 視覺化工具

4.2.1 r1_rover 載具模擬平台建立

在本模擬環境中，選用 Gazebo 與 ROS 相容的 r1_rover 載具模型作為測試平台，該模型具備簡易的輪式移動機構，能模擬地面載具的基本運動行為。r1_rover 模型預設搭載基本控制介面如圖 4.6 所示，包括左右輪速度控制（差速驅動）與 IMU 等感測器模擬，方便與 ROS 控制節點整合。

本階段以現有模型進行初步建置與驗證，並未特別調整質量、摩擦力或懸吊等物理參數。主要目的是確認模型可於 Gazebo 環境中穩定啟動與運行，並能正確接收來自 ROS 的速度指令，完成前進與轉向等基本動作。透過/cmd_vel 主題發佈速度命令，觀察載具行為，確認其運動反應穩定、無異常跳動，為後續整合光達感測與導航避障功能奠定基礎。

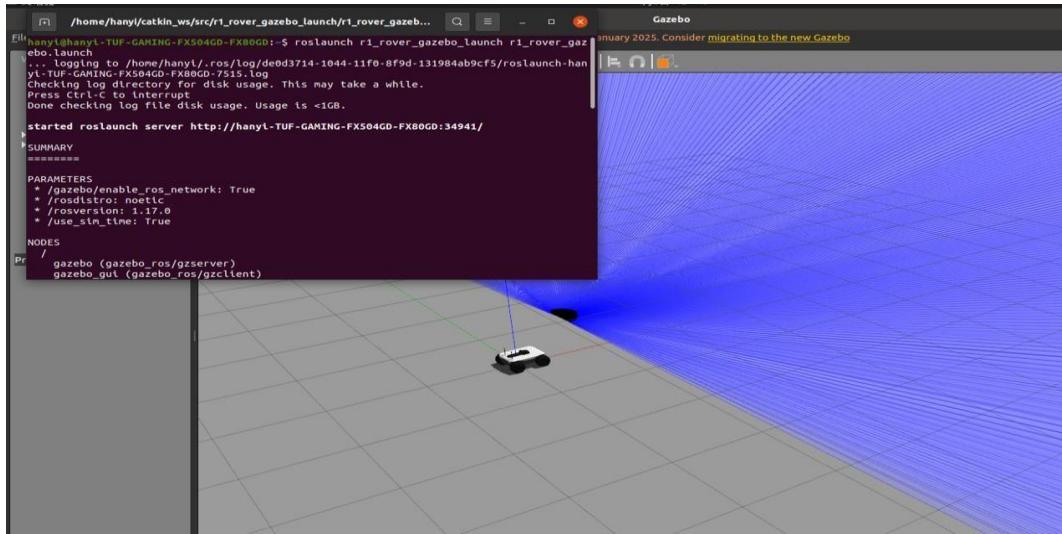


圖 4.6 光達與模擬載具 r1_rover

4.2.2 Gazebo 內載具光達感測與障礙物偵測

為模擬真實環境中使用 Quanergy M8 3D LiDAR 進行障礙物感測的情境，於 Gazebo 模擬平台中將 3D LiDAR 插件掛載至 r1_rover 載具模型上，並配置具體障礙物場景以進行掃描測試。所使用的 3D LiDAR 插件可模擬環繞式的點雲輸出，具有多線掃描特性，能產生與實體 Quanergy M8 相似的高密度空間點雲資訊。

在模擬場景中設置多個靜態障礙物（如紙箱與立方體），當載具移動接近障礙物時，模擬光達會即時回傳掃描結果，並以點雲形式發佈至 ROS 中的 /pointcloud2 主題。透過 RViz 進行視覺化，可觀察到 3D 點雲清晰地呈現周遭環境中的障礙物輪廓與位置如圖 4.7 所示。

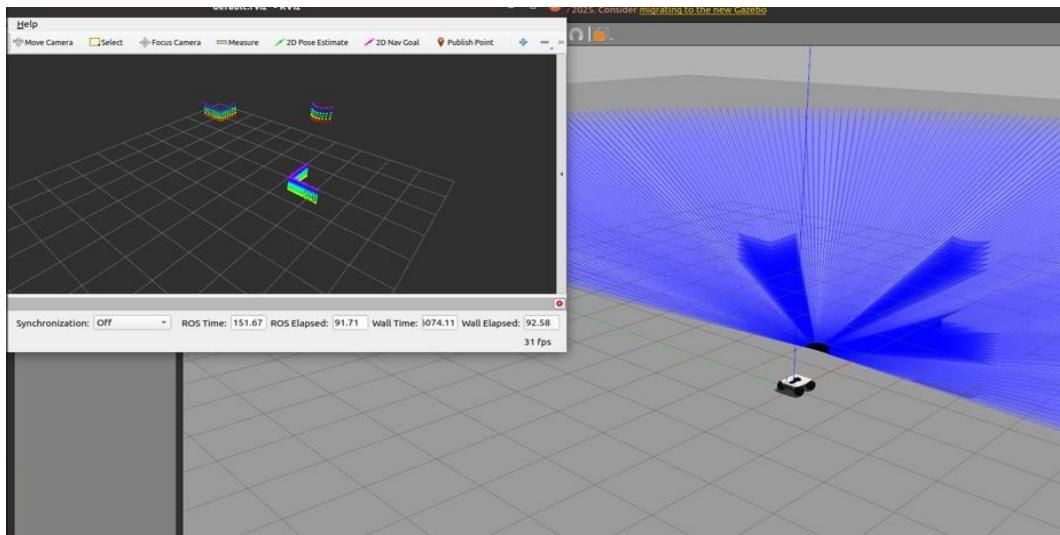


圖 4.7 pointcloud2 3D 點雲

經過多次模擬測試，確認光達資料在載具移動過程中能穩定更新，掃描結果準確反映障礙物分布，點雲密度與覆蓋範圍亦符合預期。此模擬結果顯示 Gazebo 所生成的 3D 感測資料已足以支援後續避障策略設計與導航決策邏輯之開發，亦為實體系統整合與驗證階段提供可信的前期依據。

4.3 自動避障導航程式設計與 Gazebo 模擬測試

為驗證自主避障功能於載具整合架構中的實際可行性，開發一套基於 ROS 環境之避障導航控制程式，並在 Gazebo 模擬環境中進行多組場景測試。控制程式以感測器資料為依據，判斷載具與障礙物的相對位置，進而調整前進路徑以避開障礙，最終達成導航穩定性與反應即時性的需求。本避障演算法目前以靜態障礙物為主，尚未針對動態障礙物進行反應測試，本節將說明避障導航邏輯設計方法與程式結構，並展示於模擬環境中進行之實測結果。

4.3.1 自動避障導航演算法設計與實現

自動避障控制程式架構，主要由三個模組組成：感測器資料處理模組、障礙物偵測與判斷模組、以及避障決策與控制輸出模組。程式以 ROS 節點形式實現，運行於 ROS 主系統中，並訂閱 3D LiDAR 所發佈之 /pointcloud2 主題資料。首先將點雲資料透過 PCL (Point Cloud Library) 進行前處理圖如 4.8 所示，包括 ROI 範圍選取與地面去除等步驟，以提升處理效率並濾除無效資訊。接著進行空間障礙物偵測，將障礙物區域分割為左右前方區塊，藉由判斷每個方向的點雲密度與距離閾值，計算可通過區域。

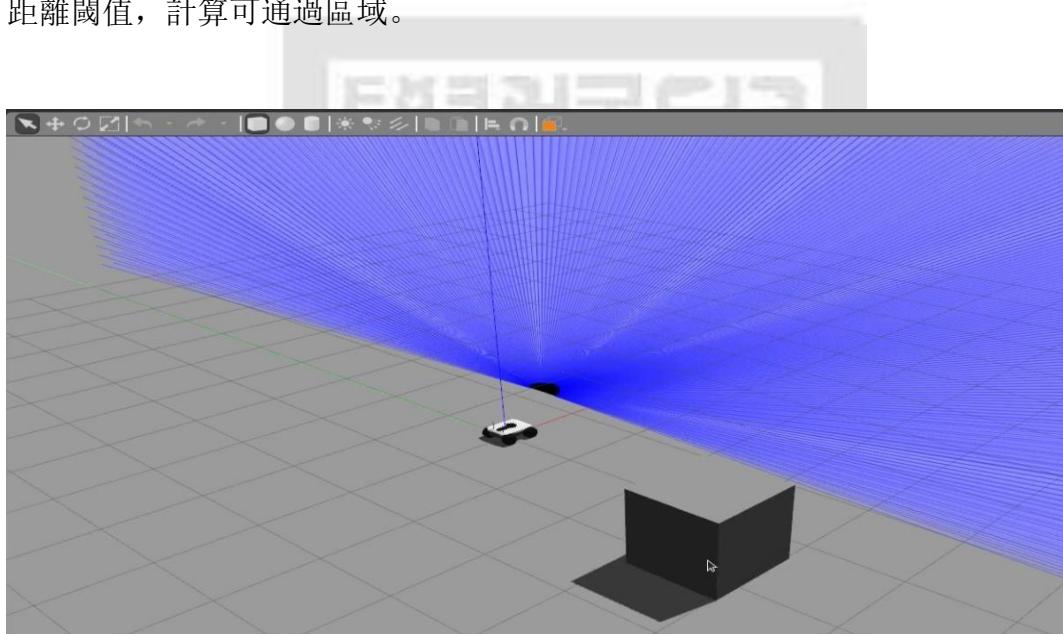


圖 4.8 將 lidar 過濾出所需部分

當偵測到前方障礙物距離低於安全閾值時，系統將根據左右區域的障礙物分布情況，選擇相對安全的方向進行轉向圖 4.9 所示，同時適度降低行進速度以確保反應時間。最終以 ROS 發佈 /cmd_vel 控制訊息，驅動 Gazebo 中的 r1_rover 模型執行實際避障動作。

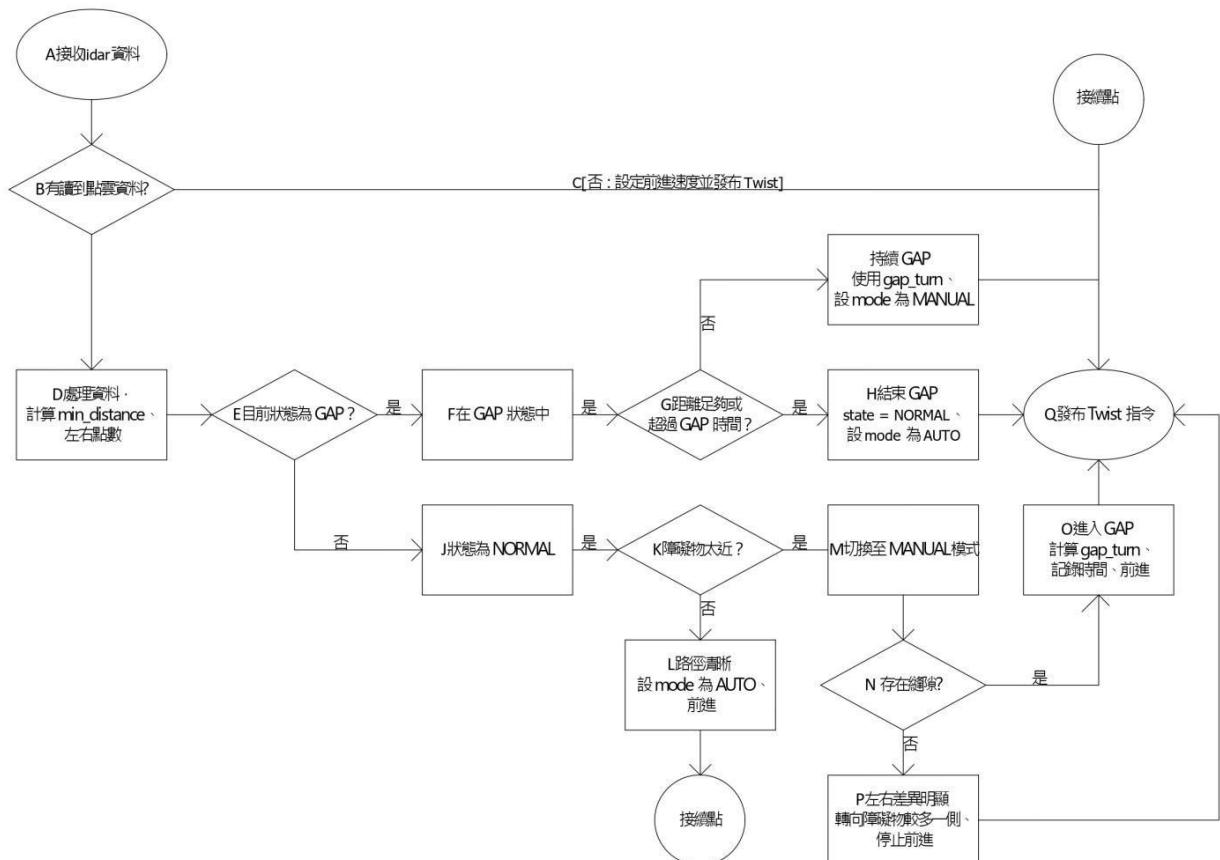


圖 4.9 避障邏輯圖

4.3.2 Gazebo 環境下載具自主避障測試與分析

完成避障演算法實作後，於 Gazebo 模擬場景中設置多組障礙物排列方式，模擬不同環境情境下載具之避障反應能力與穩定性。測試環境包括直線前進遭遇單一障礙、左右側同時有障礙、以及多彎曲通道等複雜地形。

在測試過程中，透過 RViz 觀察實時點雲資料與避障指令的發佈狀態，並記錄載具運動軌跡與反應時間。結果顯示，載具在感測器資料更新頻率充足、控制回路穩定的情況下，能即時進行障礙物辨識並有效做出方向避讓選擇如圖 4.10 紅色箭頭所示，避免碰撞發生。

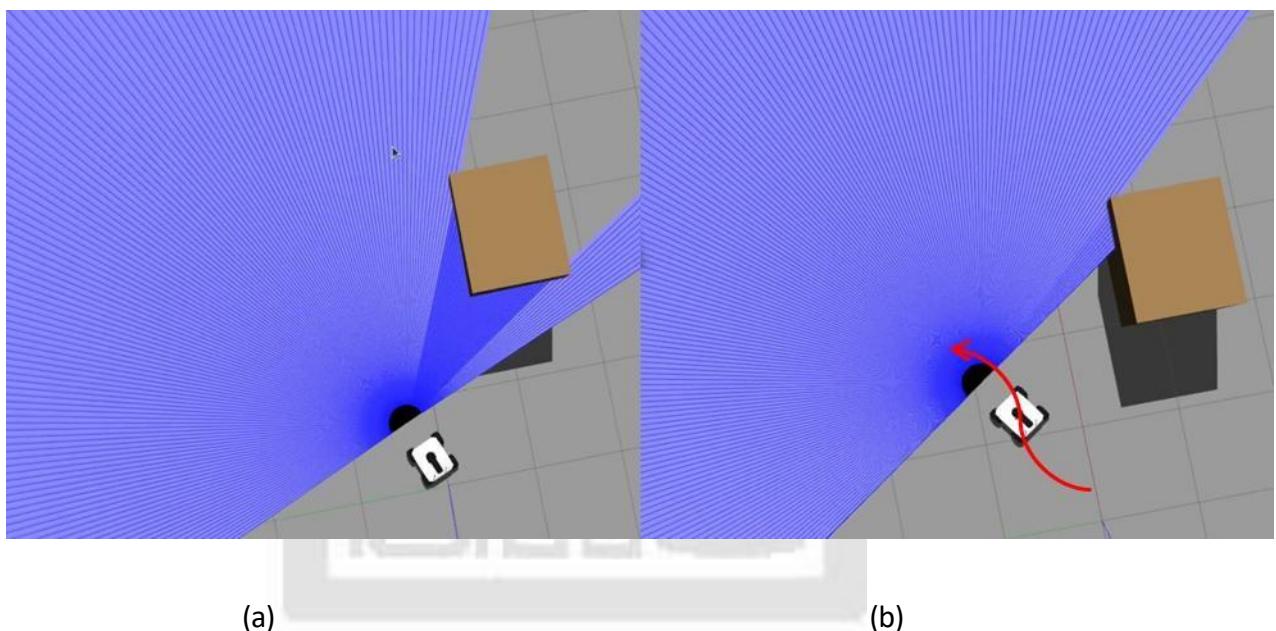


圖 4.10 (a)偵測障礙物；(b)偵測到障礙物後向點雲較少處轉彎

此外，模擬結果亦顯示，避障控制策略在多種環境下具良好穩定性如圖 4.11 紅色箭頭所示，對動態感測回饋與控制指令具高度敏感度，可作為後續實體測試的控制邏輯基礎。整體驗證系統具備即時感測、快速並穩定執行的避障導航能力，並具備擴充整合至實體 PLC 控制平台之潛力。

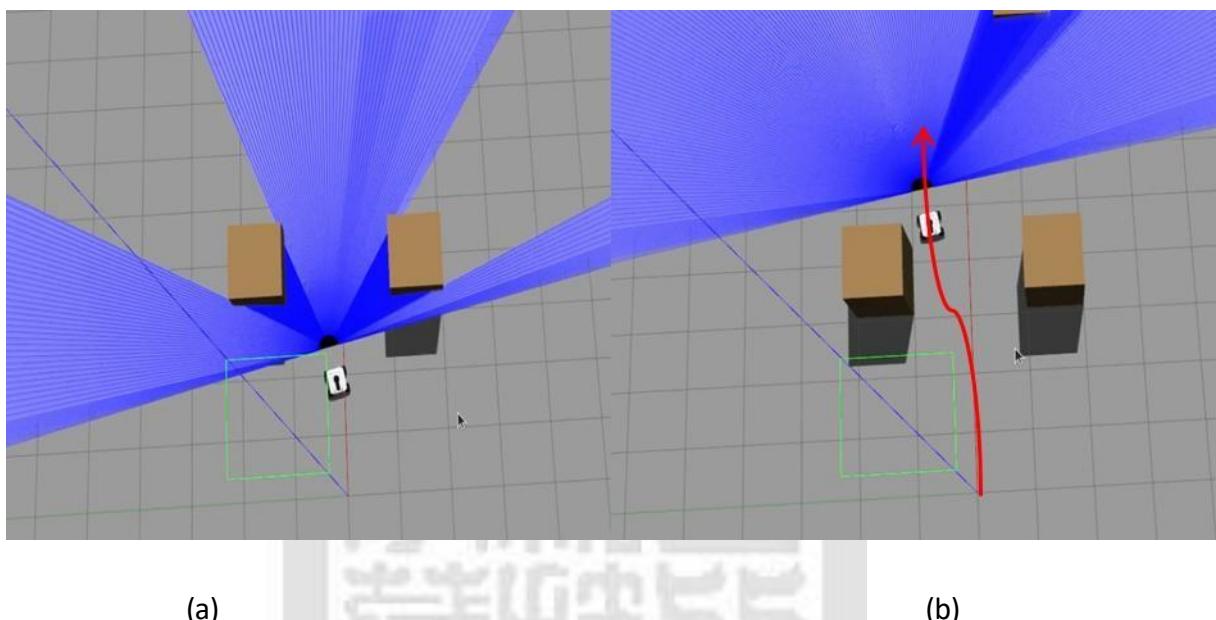


圖 4.11(a)偵測障礙物；(b)穿越複雜縫隙環境

4.4 Quanergy M8 LiDAR 實體感測與 Gazebo 模擬融合測試

為驗證實體感測器與模擬環境的整合可行性，本章進行 Quanergy M8 3D LiDAR 掃描資料與 Gazebo 模擬環境的資料同步與行為觸發測試。透過將來自實體 LiDAR 的即時點雲資料導入 ROS 環境，使 Gazebo 中的模擬載具能根據實體世界中的感測資料做出相對應的動作反應。此步驟不僅驗證模擬系統的感知與行為對映能力，也為後續進行實體載具控制建立基礎。

4.4.1 Quanergy M8 LiDAR 與 Gazebo 模擬環境融合機制建構

首先將 Quanergy M8 LiDAR 實體掃描所輸出的點雲資料，透過 ROS 導入系統中，並確認資料可穩定發佈於 /pointcloud2 主題。由於 Gazebo 模擬中的虛擬 LiDAR 通常由 Gazebo 內建插件產生，因此本實驗中關閉模擬光達插件，改由實體資料驅動決策模組，使載具行為完全由真實感測器所觸發如圖 4.12 所示。

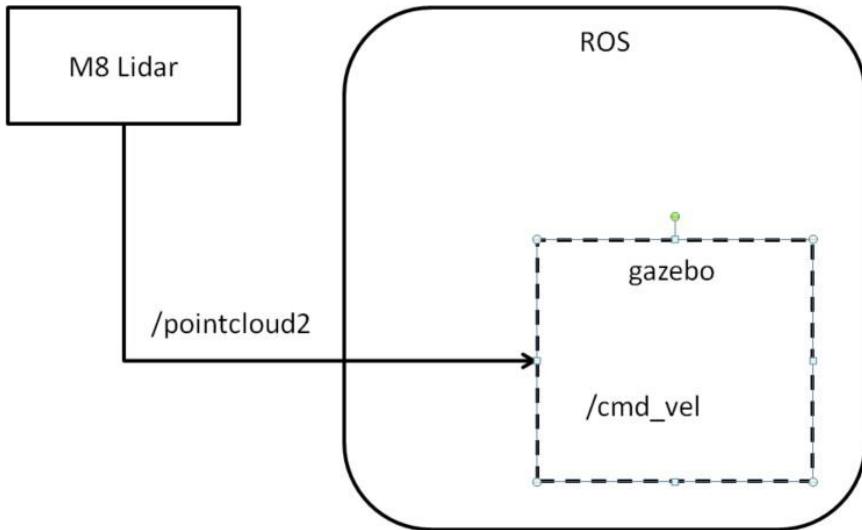


圖 4.12 模擬環境融合示意圖

融合機制上，實體 LiDAR 與模擬載具之間並無直接物理連結，而是透過 ROS 資料橋接，使得感測器輸入與控制輸出仍可維持在模擬閉環中運作。LiDAR 點雲資料先經過前處理與障礙物辨識，接著由避障控制腳本決定轉向策略，並將結果發佈至 Gazebo 模擬載具的 /cmd_vel 控制主題。

4.4.2 實體光達掃描障礙物後，Gazebo 模擬轉向行為測試

在實驗場域中放置可辨識障礙物如圖 4.13 所示，使用實體 Quanergy M8 對其進行掃描，點雲資料即時傳入 ROS 環境。模擬載具在 Gazebo 環境中行進時，當感測器資料判定前方出現障礙物時，避障邏輯會即時做出避讓判斷，並更新模擬載具運動方向。

從 RViz 視覺化結果與 Gazebo 載具運動軌跡觀察，可清楚見到模擬載具行進中因感測到障礙而轉向、停下或繞行的行為，證實實體感測器可有效驅動模擬環境中的邏輯反應。此種虛實混合測試方式能大幅降低實體載具尚未整合前的測試風險，並提升整體系統建置效率。

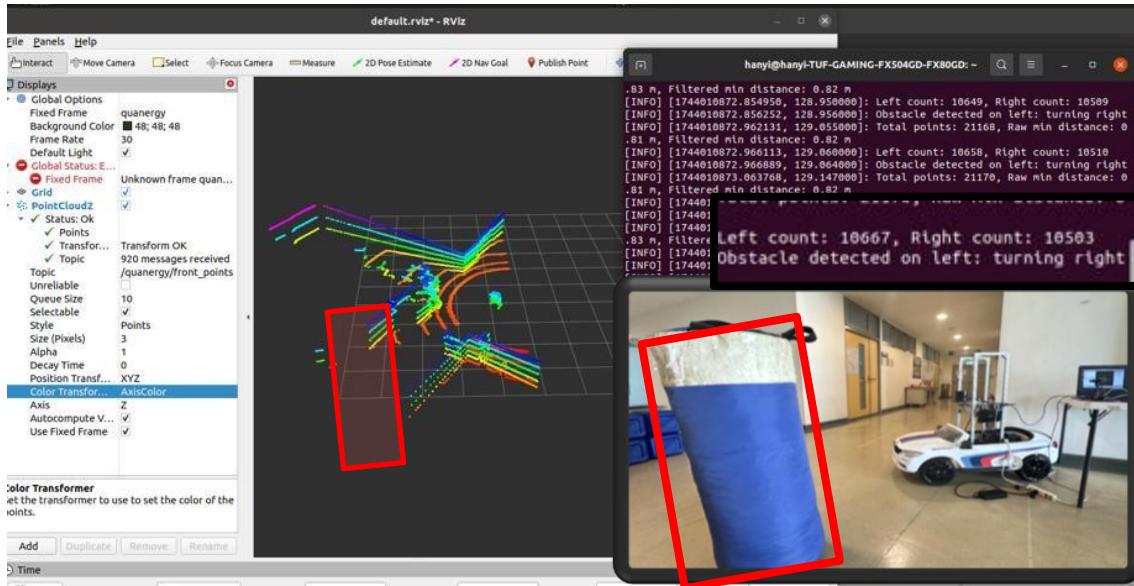


圖 4.13 M8 Lidar 掃描點雲避障(紅框表示)

4.4.3 Pixhawk 6C 控制模式自動切換功能測試 (AUTO 模式轉 Manual 模式)

為進一步模擬整合過程中飛控系統的狀態切換需求，進行 Pixhawk 6C 於 MAVROS 環境下的飛行模式切換測試。當避障邏輯偵測到需要進行路徑調整時，系統將觸發飛控從 AUTO 模式切換至 GUIDED 模式，允許 ROS 以程式方式直接發佈控制指令至 Pixhawk。

在此測試中，載具於 AUTO 模式下根據預設航點前進，當遭遇障礙物而進入避障判斷階段時，程式即自動切換至 Manual 模式，並透過 rostopic echo/mavros/state 觀察模式是否變換如圖 4.14 紅框所示。模式切換過程中可觀察到 Pixhawk 回傳之飛行模式狀態已成功更新，ROS 與飛控間的通訊與控制接續順利，且避障反應迅速，無明顯延遲。

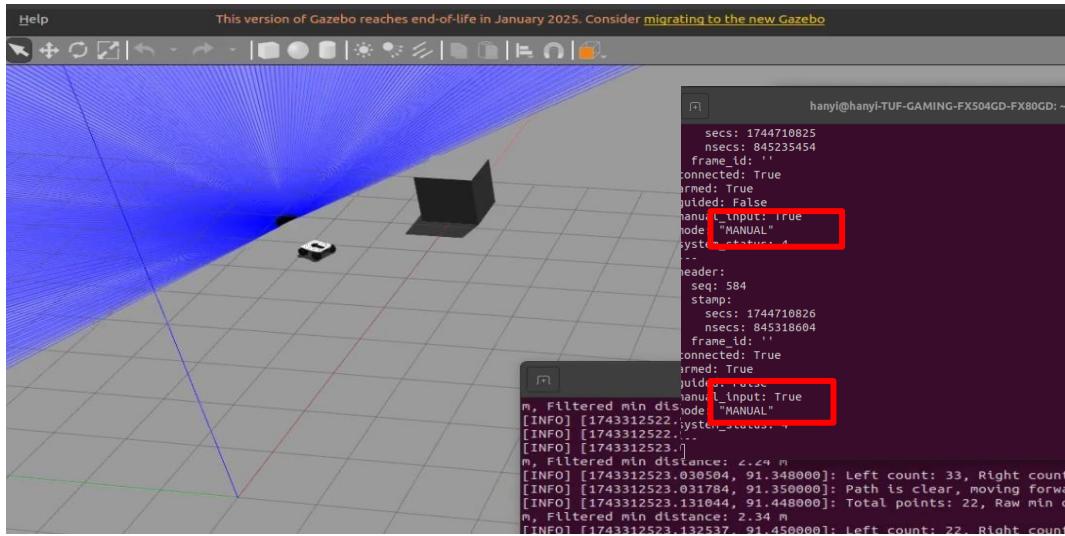


圖 4.14 無點雲時轉換模式

此項實驗結果驗證系統可實現即時感測 → 判斷 → 模式切換 → 動作執行的完整流程，為後續實驗打下穩固基礎。

實作過程中亦發現，在未適當設定飛控參數時，MAVROS 所發送之 RC Override 指令即使在 MANUAL 模式下也無法被 Pixhawk 正常接收，導致控制訊號無效。經查證後確認，需將參數 SYSID_MYGCS 設定為 1，方可讓來自 MAVROS 的 RC Override 指令具備足夠權限被 Pixhawk 接收與執行。此外，在 AUTO 模式下系統預設會忽略 RC Override 指令，因此若需在導航過程中由 ROS 端干預飛控行為，必須透過持續發送 override 訊號來保持 AUTO 模式的持續，否則將自動退回 HOLD 模式。此現象已透過實測與系統日誌驗證確認，為本系統整合穩定性的重要調整依據之一。

第五章 實驗規劃與執行

本研究以電動驅動之小型沙灘車作為實驗平台，於戶外環境中執行自動駕駛任務，並同步進行控制與優化參數的調整與驗證，以實現本論文所探討之控制策略與效能優化目標。

在初步測試中，我們觀察到 RTK-GPS 在某些環境條件下會出現定位精度下降的現象。如在圖 5.1 中黃色箭頭所示之位置，受限於建築物密集環繞，常導致衛星訊號遭遮蔽或產生多重路徑效應（Multipath），進而影響定位準確性並降低導航穩定性。

為避免上述環境因素對試驗結果造成干擾，並確保控制邏輯與導航效能評估的公正性，後續實驗將場域遷移至開闊、無高層建築遮蔽的空地，作為穩定且具可比性的測試環境。



圖 5.1 環境干擾場域示意圖

5.1 測試環境與參數設定

本研究之實體驗證階段，選擇於校園內一處開放空地進行實驗，如圖 5.2 所示。

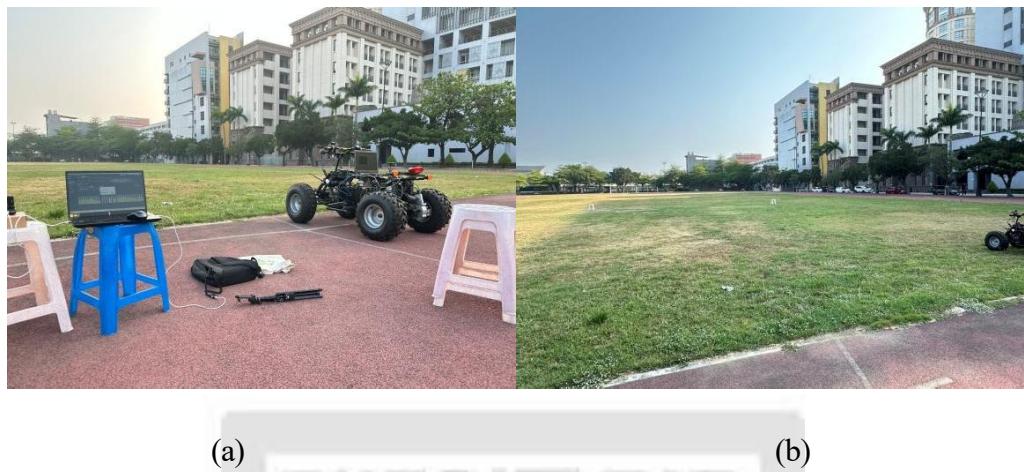


圖 5.2 (a)設備照；(b)測試場域

該測試場域地勢平坦，四周建築物密度低，能有效避免多重路徑效應（Multipath）對 RTK-GPS 精度造成的干擾，並提供穩定的 GNSS 訊號接收條件，適合作為自主導航系統之測試場域。為排除遮蔽造成的誤差，本測試特別避開高樓林立與複雜環境，以確保定位與感測資料的準確性，此外，外部環境啟用 RTK-fix 要求可詳見表 5-1。

表 5-1 外部環境 RTK 模式啟用表

可見/鎖定衛星數	satcount	≥ 18
水平方位精度稀釋因子	hdop	≤ 0.9
RTK 解算的衛星總數	rtk_num_sats	≥ 5
RTK 差分修正訊息的延遲時間	rtk_age_ms	$< 200 \text{ ms}$
RTK 固定 (Fixed)	status	6

RTK-GPS 系統採用 H-RTK F9P 作為基準站與流動站模組。基準站架設於無遮蔽的開闊處，並架高固定，透過無線電遙測模組（Radio Telemetry）將差分訊號即時傳送至車載端流動站，以達成 RTK Fix 級別之定位精度。車載端則將定位資訊經由 UART 接口送入 Pixhawk 6C 作為導航輸入依據。

車體配備方面，Quanergy M8 三維光達安裝於車體前方，提供高密度點雲資訊作為障礙物偵測依據。Pixhawk 6C 飛控模組則為整體控制核心，負責接受來自 ROS 的導航指令，並依據任務需求切換飛行控制模式。本測試中，Pixhawk 預設運行於 AUTO 模式，進行多點導航任務；當偵測到前方有障礙物時，ROS 系統會發送 RC Override 指令，將 Pixhawk 控制模式切換至 MANUAL 模式進行避障操作，待障礙排除後再自動回復至 AUTO 模式以繼續執行導航任務。

為實現 PWM 訊號轉換並與 PLC 控制器對接，系統中導入 Arduino Mega 2560 作為中介處理器，負責擷取 Pixhawk 所輸出之 PWM 脈波寬度，並經內部演算後送入 DAC 模組，轉換為對應電壓訊號。最終由 PLC 類比輸入模組讀取此電壓，據以控制驅動馬達與方向伺服馬達，以完成整體自駕控制鏈路的實體驗證。

實驗過程中所有重要參數（如 RTK GPS 訊號狀態、經緯度、任務資訊等）皆透過 Pixhawc 6c 即時紀錄，以作為後續誤差分析與性能評估之依據。此測試規劃不僅符合實務操作需求，亦可作為後續系統優化與擴充之依據。

5.2 任務路徑設定與行駛軌跡記錄

本研究於校園內的操場上進行自動導航任務實驗，測試載具依預設航點自動行駛之能力與穩定性。實驗場域規劃一矩形封閉路徑作為任務區段，包含四個航點，總長約 50 公尺，具備連續直行與直角轉彎路徑，如圖 5.3 所示。另外圖中黃色箭頭為預設導航路徑，紫色方塊代表車體位置，藍色標記為 RTK-GPS 基地台所在位置。



圖 5.3 測試路線圖

實驗任務以原點作為起訖點，形成閉合巡迴路線，利於分析路徑閉合誤差與控制穩定性。該幾何配置可有效測試系統之轉向反應與直線追蹤能力，並進行後續誤

差比對。任務路徑經由 Mission Planner 軟體進行設計並上傳至 Pixhawk 6C 模組。導航執行時使用 AUTO 模式，由 Pixhawk 控制車體依順序自動通過各航點。考量現場可能出現的突發情況，亦預留可切換至 MANUAL 模式之備援方案，以維持任務之完整執行。為達成高精度定位，本次實驗採用 RTK F9P 定位模組，並設置 RTK-Base 基地台作為固定參考點，基地台安置於場域一側開放地面上，遠離建築遮蔽與反射物體，以減少多重路徑效應（Multipath）對定位精度之影響。車體搭載的 RTK-Rover 移動站與基地台配對後可實現 RTK Fix 模式，將定位誤差降低至公分等級如圖 5.4 紅圈處所示。



圖 5.4 RTK Fix 模式

導航過程中，Pixhawk 6C 控制器自動紀錄任務執行期間的飛行數據（Dataflash log），包含 GPS 座標、航點通過情況、模式切換與控制輸出等。後續將透過 Mission Planner 將該紀錄資料匯出，進行軌跡分析與偏差比對。

5.3 系統穩定性與準確性分析

本節旨在針對本研究所建置之自駕載具系統進行穩定性與 GPS 定位準確度的驗證。由於 GPS 定位精度為導航任務成功與否的關鍵因素之一，因此本實驗透過重複執行相同的任務路徑，記錄並分析載具每次行駛軌跡的重現性，藉此排除因 GPS 訊號不穩所導致之任務執行誤差。

實驗中，載具以固定的初始位置與相同的預定路徑進行多次自動導航測試如圖 5.5 所示，透過 Pixhawk 6C 飛控模組內建的資料記錄功能，完整記錄每次導航任務過程中的 GPS 位置座標資料。

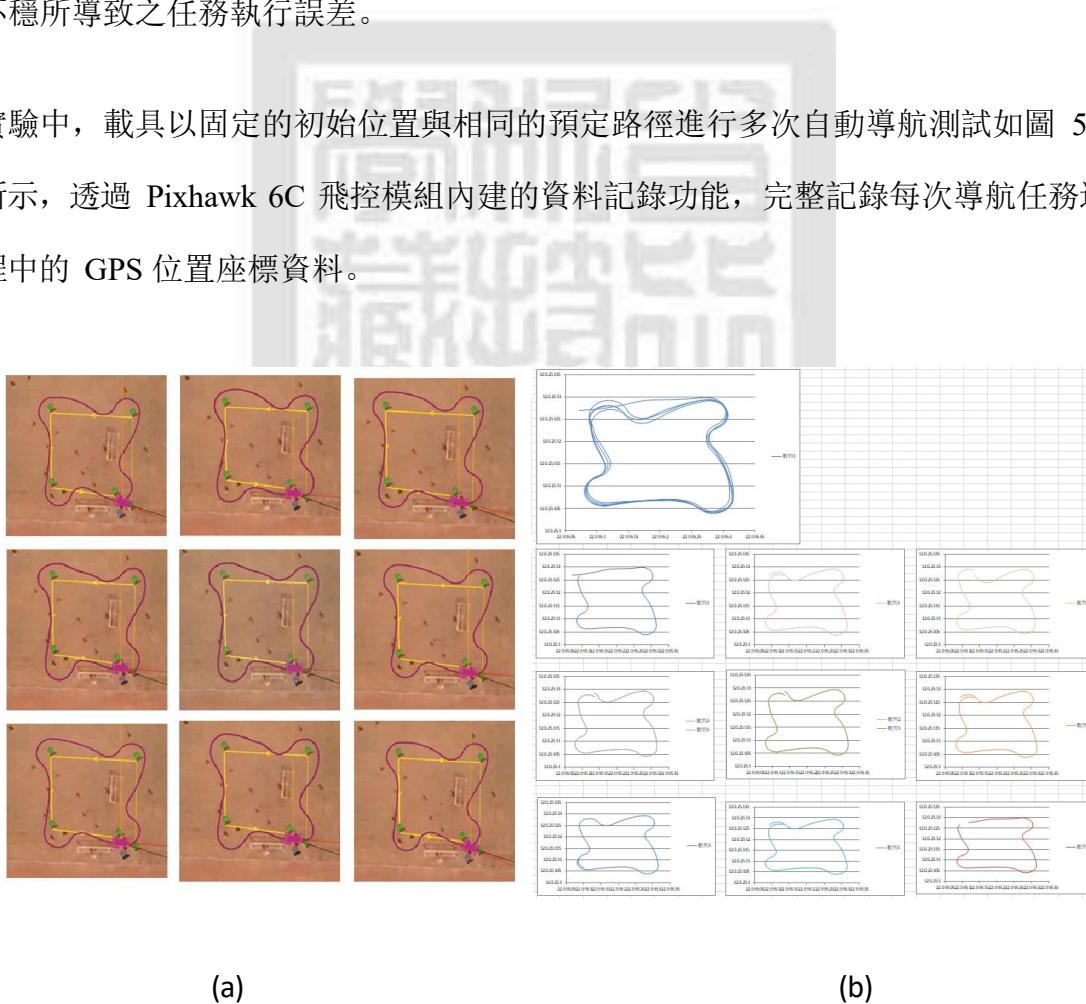


圖 5.5 (a)多次任務執行之路徑紀錄；(b)將路徑經緯度數據透過 Excel 視覺化

此記錄資料再以Haversine演算法逐點計算每一次導航測試所產生的實際軌跡與理論設定軌跡間的偏差，並透過多次實驗數據之平均誤差、最大誤差及均方根誤差（RMSE）進行綜合分析。若經實驗結果顯示每次導航任務的重複路徑軌跡具有高度一致性，則可證明系統使用的 RTK-GPS 定位具有足夠穩定性與可靠性，此時導航任務所觀察到的任何偏差便可歸因於載具控制層面的機械結構、伺服控制反應時間或是控制演算法本身，後續即可聚焦於這些層面進行優化與改善。

反之，若實驗結果呈現明顯且不規律之軌跡誤差分布，則代表 RTK-GPS 定位系統存在穩定性或多路徑效應（Multipath）干擾問題，此時後續研究則需著重 GPS 系統改善，或進一步研究搭配其他感測器進行輔助定位與修正之策略。

透過上述系統性分析，本研究能更有效地區分載具控制系統與 GPS 定位系統各自對導航任務造成的影響，明確指引後續載具系統優化與發展方向。

5.3.1 GPS 系統準確性分析結果

為驗證本研究實驗所採用之 RTK-GPS 定位系統是否具備足夠穩定性，特別針對相同任務路徑進行數次重複實驗，以檢視不同次數執行下的導航軌跡是否一致。若 GPS 資料具備高穩定性，則理論上於同一任務條件下，所記錄之行駛路徑應高度重疊，並呈現極低之位置誤差變異，反之則可能存在定位精度波動。

在本次驗證中，所有任務均使用相同起點、相同航點設定與相同導航控制參數進行。每次任務執行結束後，系統透過 Pixhawk 6C 飛控模組所記錄之飛行紀錄檔（Dataflash log），擷取對應之 GPS 座標資料，並進行交叉比對分析。比對方式採用每筆軌跡與平均軌跡間進行點對點誤差計算，並採用 Haversine 演算法求得球面距離。

各組分析之具體數據摘錄如下：

最大誤差 (Max error) :

經本研究多次測試最大誤差範圍介於 1.10 至 1.50 公尺之間，結果顯示多次路徑執行過程中，均未有明顯的導航偏離狀況，維持了高度的一致性。

平均誤差 (Average error) :

各次路徑相互比對之平均誤差值介於 0.21 至 0.74 公尺，展現相當的穩定性顯示該導航系統於本次任務中能有效達成精確定位與穩定導航之需求。

均方根誤差 (RMSE) :

RMSE 值分布於 0.29 至 0.88 公尺之間，顯示整體導航誤差控制在可接受範圍內。其中最小值 0.29 公尺反映系統極高之精度與穩定性，即便最大值達 0.88 公尺，亦未顯著偏離，證實導航系統於各狀況下皆具備良好之穩定性與可靠性。

路徑重合率 (Hit rate) :

所有十組數據之容許誤差命中率皆達到 100 %，表示所有路徑誤差均落在預設之容許誤差 1.5 公尺內，系統整體的表現高度穩定且符合預期要求。

根據實際數據的綜合分析，實驗結果明確證明本研究所採用之 RTK-GPS 定位系統具備高度的定位穩定性與一致性參照圖 5.6。各組重複測試結果可見所有路徑皆落於紅色虛線(容許誤差)之下，表示整體 GPS 之誤差極小且穩定（參照 2.5.2），確實可作為自駕載具導航系統精度評估之客觀依據。



圖 5.6 GPS 準確性分析示意圖

5.4 系統優化分析

本節旨在於排除 GPS 誤差因素後，進一步探討影響載具路徑偏差的潛在原因，並透過參數調整以提升實際行駛軌跡與預定任務路徑（CMD）的重合程度。分析過程中，首先比對各次任務路徑與 CMD 之誤差數據，確認系統性偏差存在後，再於 Mission Planner 中調整轉向控制參數，以期優化整體導航效能。

5.4.1 載具路徑偏差原因

本研究所使用之載具為由傳統人駕小型沙灘車改裝而成，並保留其後輪驅動 / 前輪轉向之機構設計。在人為駕駛情境中，駕駛者通常會根據下個目標點的方向提早進行轉向操作，使車輛在轉彎過程中順勢通過目標點並平滑進入下一段路徑如圖 5.7 所示。

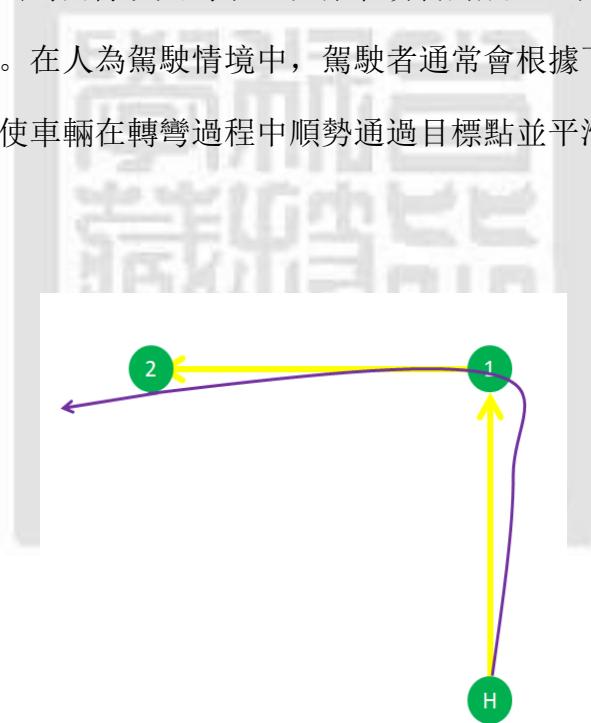


圖 5.7 人控駕駛情境示意圖

然而，在自動導航模式下，導航系統須等待載具完全抵達當前目標點後，才能觸發前往下一目標點的控制指令如圖 5.8 所示。

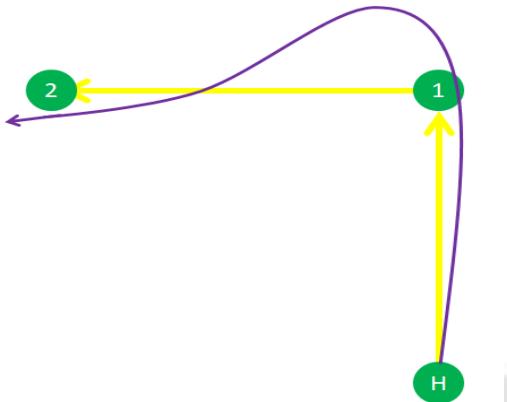


圖 5.8 自動駕駛情境示意圖

此種“先到點再轉向”的控制邏輯，無法模擬人類的預判操作，容易導致於目標點後出現明顯的轉向延遲現象，形成所謂“過衝（overshoot）”之偏差（如圖紅圈處所示）。此偏差在轉彎半徑較小或連續彎道任務中尤為明顯如圖 5.9 所示，且會顯著影響整體路徑重合度。

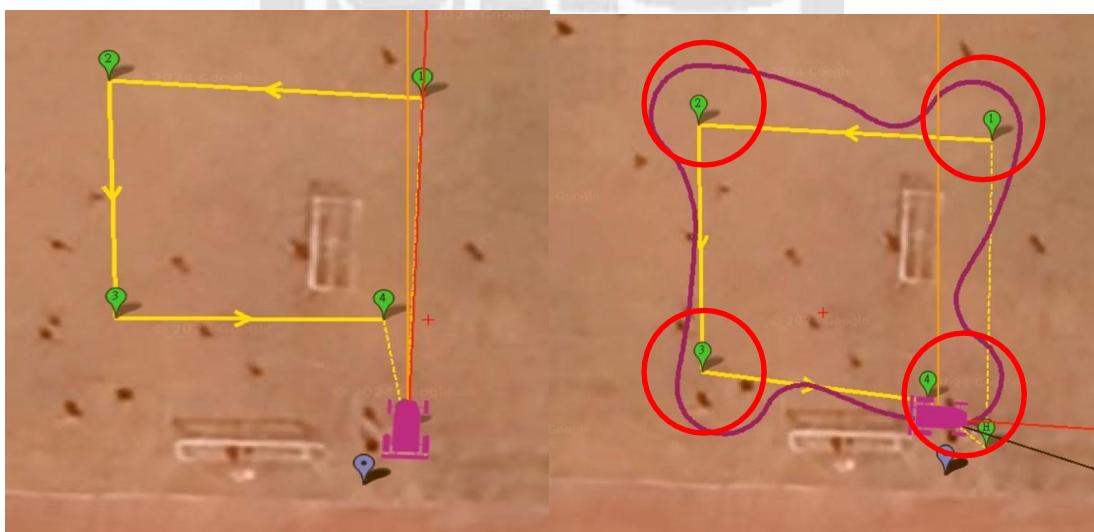


圖 5.9 轉向邏輯偏差示意圖

使用任務點距離誤差分析可觀察到，在需要轉彎的地方如任務點 1、2、3(即圖表 X 軸的 2、3、4)因為轉向邏輯的原因擁有較高的誤差率圖 5.10 所示，詳細數值可見表 5-1。

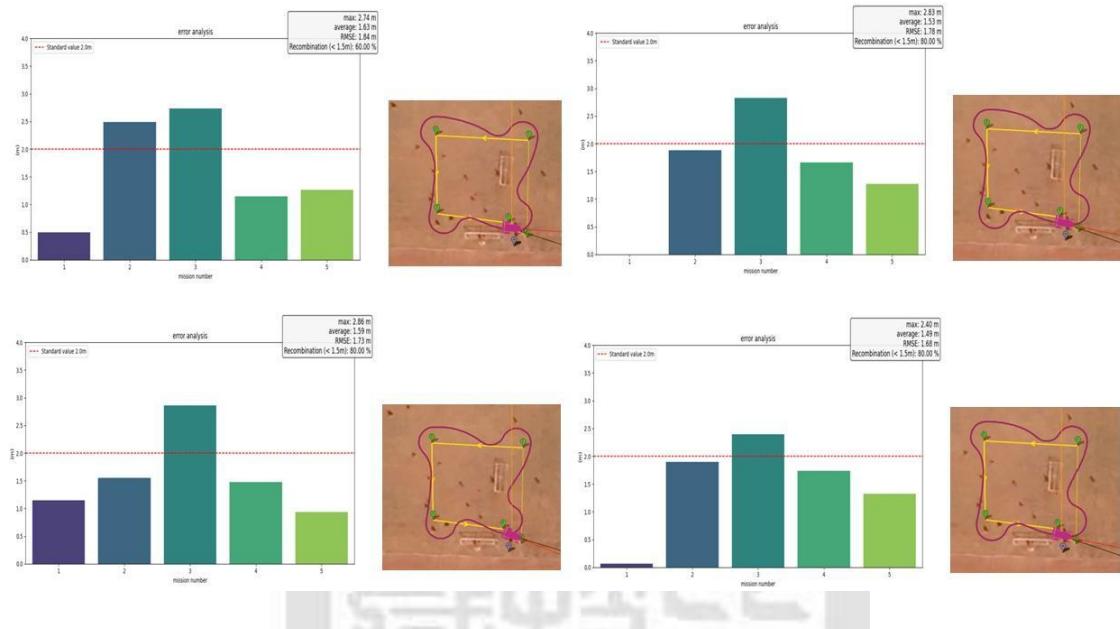


圖 5.10 任務點距離誤差分析

表 5-2 任務點距離誤差數值表

	最大誤差(m)	平均誤差(m)	RMSE	點合格率(%)
Test 1	2.74	1.53	1.82	60
Test 2	2.83	1.53	1.78	40
Test 3	2.86	1.59	1.73	60
Test 4	2.4	1.49	1.68	40

使用路徑重合度分析結果如下圖 5.11，詳細數值可見表 5-2。

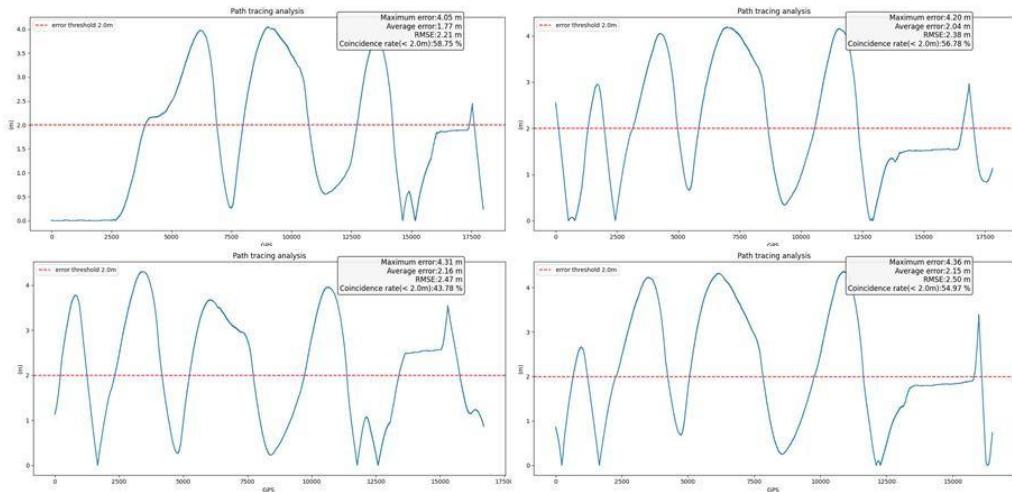


圖 5.11 路徑重合度分析

表 5-3 路徑重合度誤差數值表

	最大誤差(m)	平均誤差(m)	RMSE	路徑重合率(%)
Test 1	4.05	1.77	2.21	58.75
Test 2	4.20	2.04	2.38	56.78
Test 3	4.31	2.16	2.47	43.78
Test 4	4.36	2.15	2.50	54.97

由上述實驗數據可觀察到，儘管部分任務之路徑重合率表現未盡理想，然整體誤差分布趨勢卻高度一致，顯示在相同任務條件下，RTK-GPS 之定位精度具穩定性，其對路徑偏差之影響相對有限。進一步分析發現，誤差較大之區段多集中於轉彎位置，顯示路徑偏差主要來自於載具於轉向階段的控制反應，而非定位誤差所致。因此，未來系統優化應聚焦於轉彎行為之控制邏輯與靈敏度設定，藉由調整轉向參數提升動態響應能力，以改善彎道處的路徑偏差，進而提升整體導航精度與路徑一致性。

5.5 轉向參數調整優化整體任務

本節旨在針對前節實驗所歸納之結果進行深入探討。根據分析結果顯示，路徑偏差主要集中於載具執行轉向階段，尤以進出彎道時的誤差最為顯著，顯示控制器對於方向變換之響應能力仍有待提升。此現象可歸因於轉向參數設定不當，導致回應過慢或過度修正，使得實際路徑與預期軌跡產生偏移。

因此，本節將聚焦於自動導航控制器中轉向相關參數之調整，並設計一系列控制實驗以逐步驗證其影響，評估不同參數組合對轉向精度、路徑重合度之影響。期望透過系統性優化達成更穩定且精準之轉向控制。此外，本節亦將結合任務路徑誤差圖（如誤差色條圖、RMSE 分佈圖）與 PID 控制響應圖進行交叉比對，藉此評估調整成效是否能有效降低轉向過衝、延遲與震盪等現象。實驗結果將作為後續參數定案與戶外導航性能驗證之依據，並為未來擴充控制器至多樣地形與應用場景提供調整準則與實證依據。

5.5.1 任務數據紀錄分析

本實驗利用 Mission Planner 所提供的數據分析功能，透過圖像化方式對自動導航過程中的控制表現進行評估。為判斷轉向控制參數的優劣，研究中選取兩組關鍵指標作為判斷依據，分別為 PIDA (PID Angle) 與 PIDS (PID Steering)。PIDA 反映車輛實際航向與目標航向之間的誤差調整能力，可用來觀察方向控制的即時性與準確性；PIDS 則代表控制器最終輸出至伺服系統的命令量，用以判斷控制反應是否穩定且足夠補償轉向需求。透過這兩組數據的可視化分析，可快速辨識各組 PID 參數在控制響應上的差異，並結合後續定量指標，找出最適合的轉向控制參數組合。其中最主要的調控參數為 ATC_STR_RAT_P，作用為車輛轉向角速度控制器的比例增益，在未調整的情況下，該參數初始設定為 0.2。如下圖所示，代表轉向角度期望值的 PIDA.Tar (紅線) 理應與代表實際轉向反應的 PIDA.Act (綠線) 呈現趨勢一致的變化。然而，實際結果顯示 PIDA.Act 並未如預期般響應，於紅圈處可明顯觀察到兩者之間的偏差。此外，作為轉向控制依據的 PIDS.Tar (藍線) 與實際輸出 PIDS.Act (黃線) 理論上亦應同步變化，但圖中黃線呈現幾近水平、缺乏動態變化的趨勢，進一步顯示其轉向輸出極差，詳見圖 5.12。

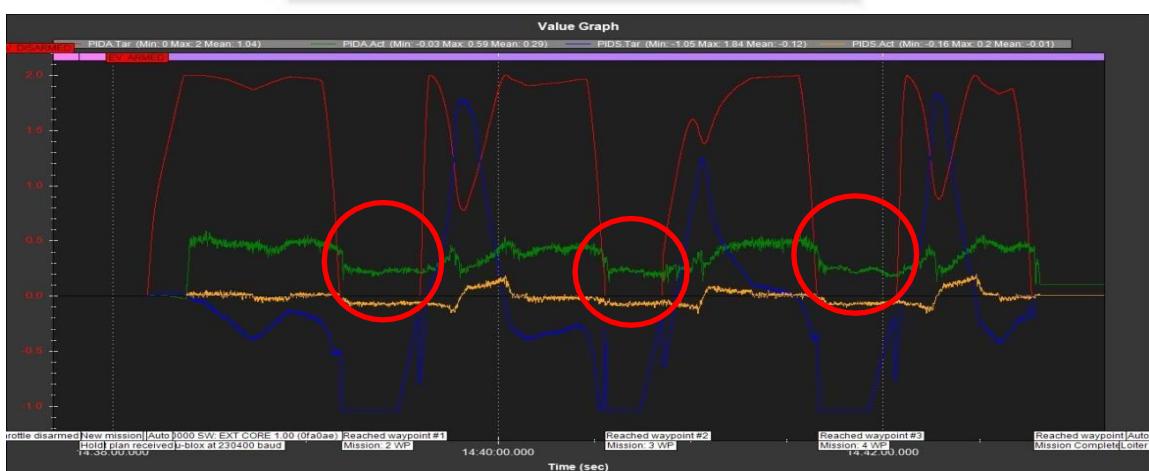


圖 5.12 未調整參數之 PIDA 與 PIDS 結果

接下來將展示不同 ATC_STR_RAT_P 參數設定下的實驗結果，分別為 0.1（圖 5.13）、0.5（圖 5.14）、1.0（圖 5.15）、1.5（圖 5.16）。由圖中可觀察到，當 ATC_STR_RAT_P 調整至 1.0 以上時，PIDA 與 PIDS 的反應趨於穩定，其中 PIDA.Tar 與 PIDA.Act 呈現相近的上升與下降幅度，顯示轉向角度之期望值與實際反應趨於一致。同樣地，PIDS.Tar 與 PIDS.Act 亦展現出更同步的變化趨勢，顯示轉向控制的輸出精度有顯著提升。

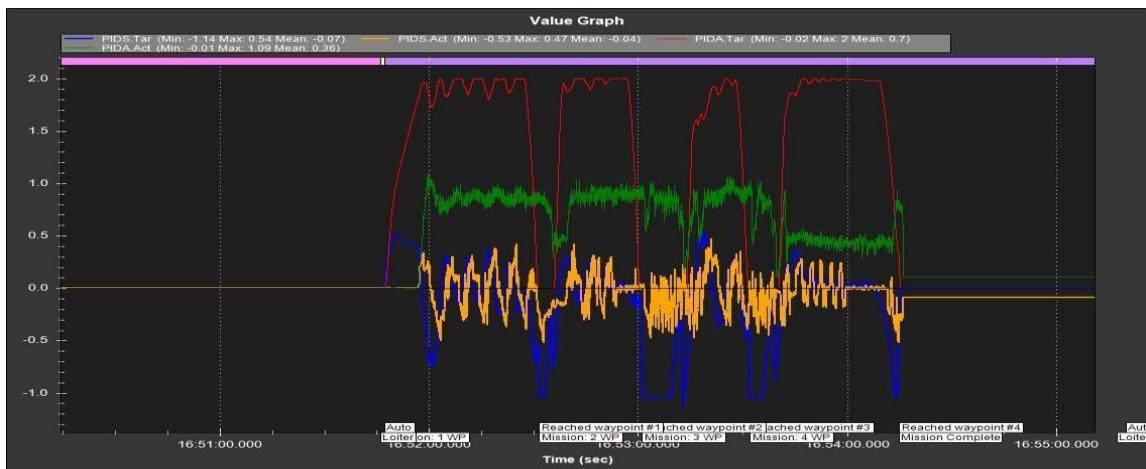


圖 5.13 ATC_STR_RAT_P 0.1

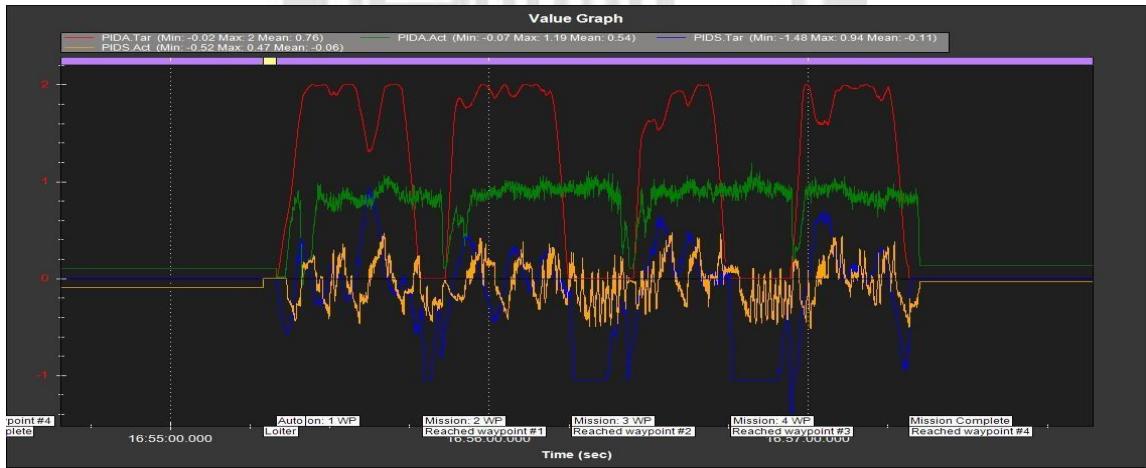


圖 5.14 ATC_STR_RAT_P 0.5



圖 5.15 ATC_STR_RAT_P 1.0

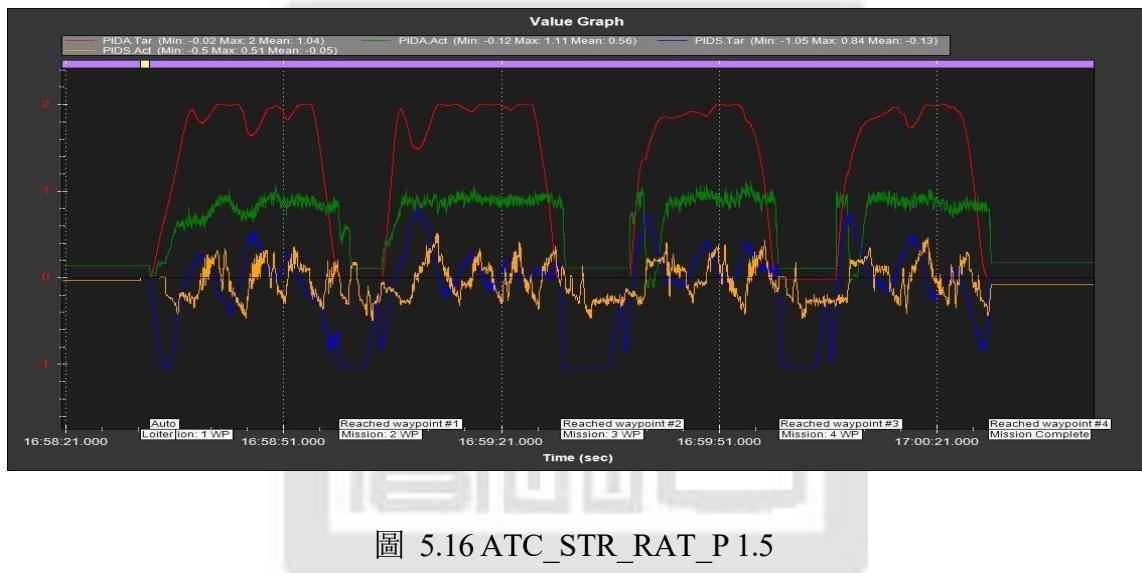


圖 5.16 ATC_STR_RAT_P 1.5

由實驗數據可得知，提升轉向靈敏度參數（ATC_STR_RAT_P）能有效改善轉向過程中所產生的誤差。整體結果顯示，較高的靈敏度設定有助於提升轉向反應的即時性與精確性，進而優化路徑追蹤表現。後續將透過系統性之路徑誤差分析，進一步評估各參數組合的實際效能，以尋求最適之轉向控制設定。

5.5.2 系統性之路徑誤差分析

承接前節分析結果可知，提升轉向靈敏度參數（ATC_STR_RAT_P）可有效改善載具於轉向過程中所產生之偏差現象。為進一步驗證此結果並找出最適參數值，本文設計一組系統化實驗，於固定速度模式下，調整不同轉向靈敏度參數，並重複執行相同任務以進行成效評估。

實驗過程中，所有測試均於一致環境條件與相同任務軌跡下進行，該任務路徑如圖 5.17 所示。任務完成後，將各組結果分別進行「任務點距離誤差分析」與「路徑重合度評估」，以多層面觀察不同參數設定下之導航表現與誤差分布情形。



圖 5.17 任務路徑圖

本次測試共調整 11 組 ATC_STR_RAT_P 之參數數值，分別為：0.1、0.5、1.0、1.5、2.0、2.5、3.0、3.5、4.0、4.5 與 5.0，其對應分析結果詳列於表 5-3、5-4，並輔以圖 5.18 至圖 5.28 之視覺化圖表輔助說明，以利全面評估各參數對轉向性能之影響，並據此歸納出最適轉向靈敏度參數範圍。

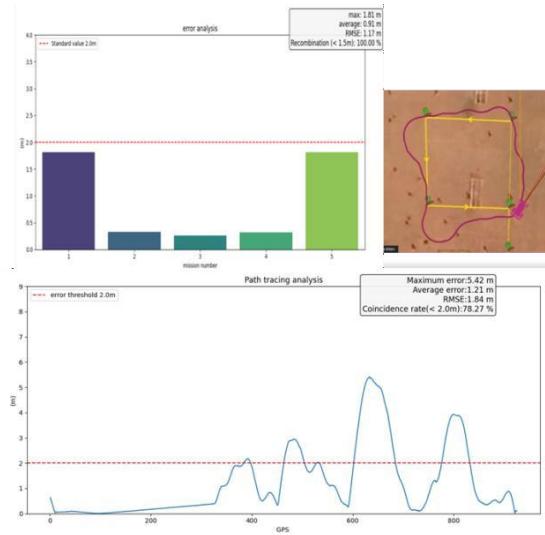


圖 5.18 ATC_STR_RAT_P: 0.1

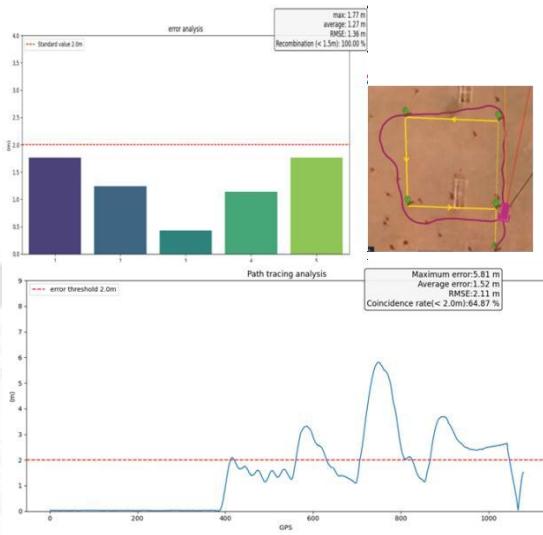


圖 5.19 ATC_STR_RAT_P: 0.5

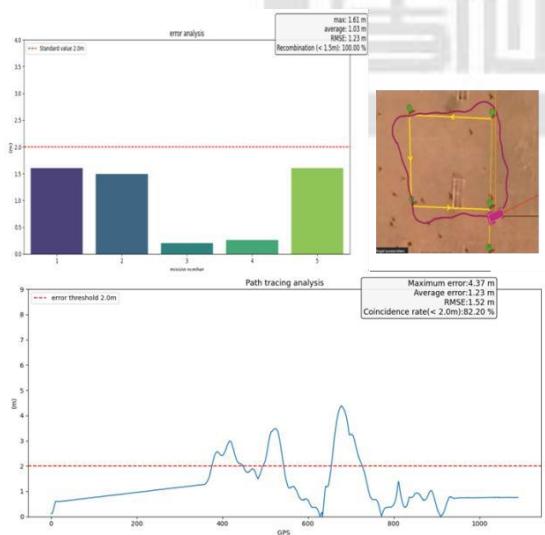


圖 5.20 ATC_STR_RAT_P: 1.0

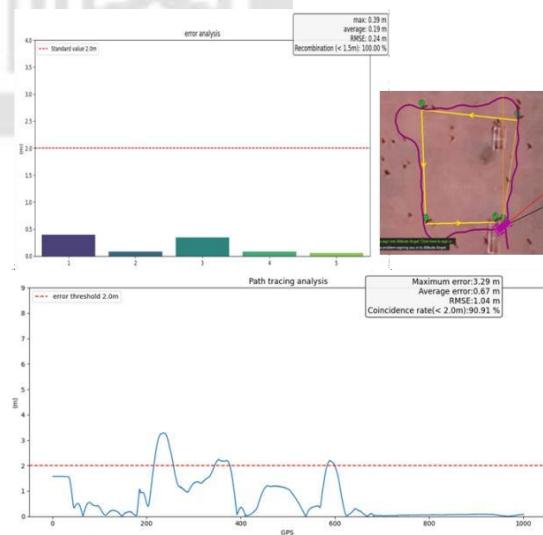


圖 5.21 ATC_STR_RAT_P: 1.5

以下是 ATC_STR_RAT_P 之參數數值 2.0~3.5。

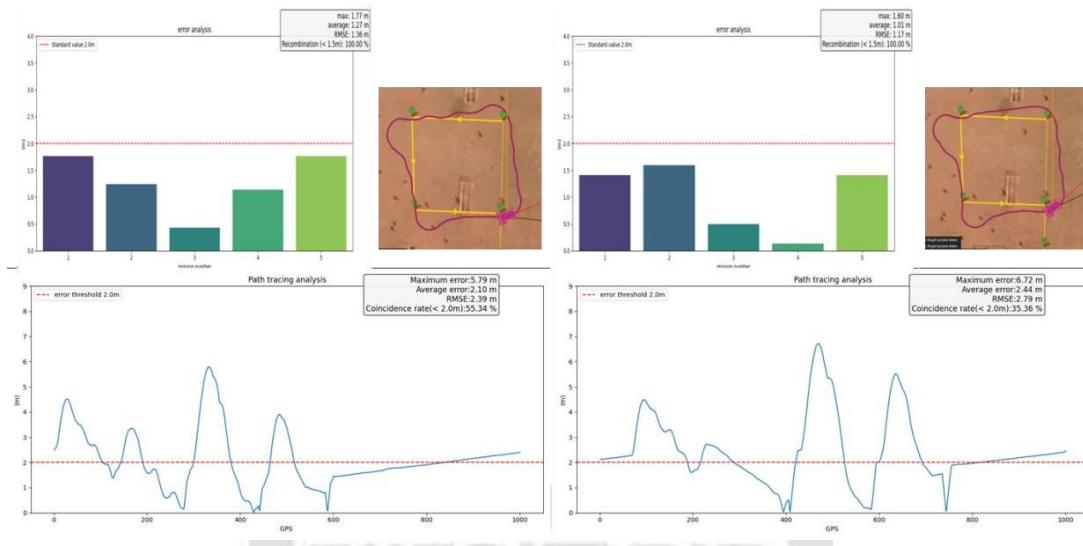


圖 5.22 ATC_STR_RAT_P: 2.0

圖 5.23 ATC_STR_RAT_P: 2.5

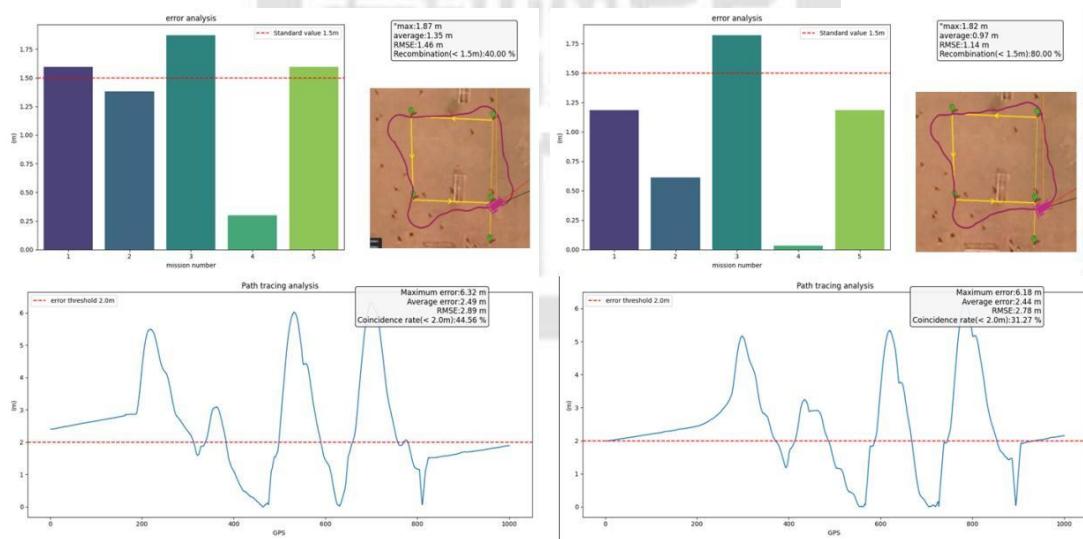


圖 5.24 ATC_STR_RAT_P: 3.0

圖 5.25 ATC_STR_RAT_P: 3.5

以下是 ATC_STR_RAT_P 之參數數值 4.0~5.0。

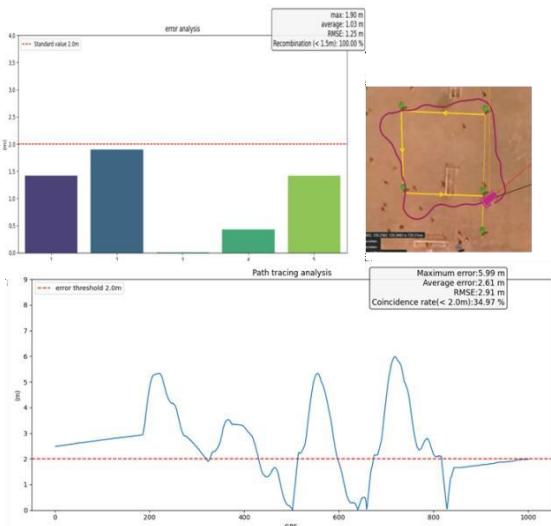


圖 5.26 ATC_STR_RAT_P: 4.0

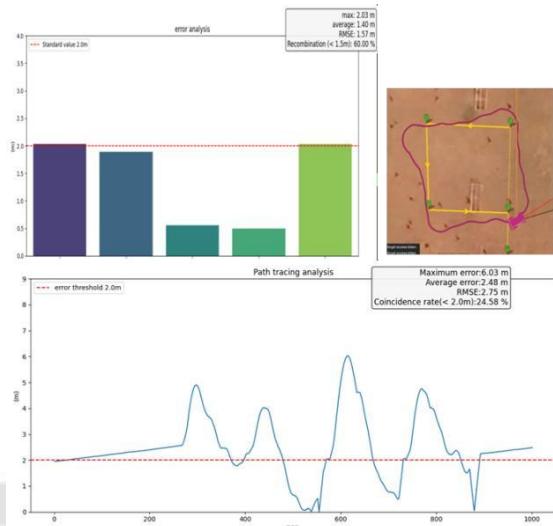


圖 5.27 ATC_STR_RAT_P: 4.5

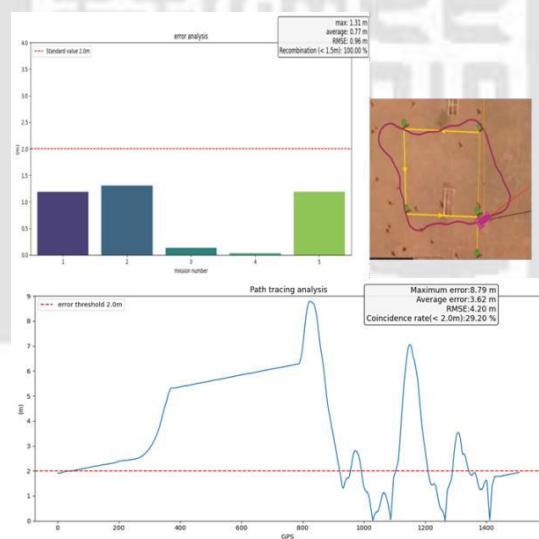


圖 5.28 ATC_STR_RAT_P: 5.0

綜合圖 5.18 至圖 5.28 所示之路徑追蹤圖形，配合表 5-3（任務點距離誤差分析）數據結果，可明確觀察到當 ATC_STR_RAT_P 設定為 1.5 時，於各項評估指標上皆表現最為優異。在任務點距離誤差方面（表 5-3），P 值為 1.5 時其最大誤差僅為 0.47 公尺，平均誤差為 0.23 公尺，均方根誤差（RMSE）為 0.26 公尺，且重合率達 100%，顯示任務點精準度與一致性皆具高度可靠性。

表 5-4 任務點距離誤差分析表

	max	average	RMSE	Reoc
0.1	3.52	0.96	1.61	80%
0.5	1.24	0.88	0.94	100%
1.0	1.50	0.88	1.06	100%
1.5	0.47	0.23	0.26	100%
2.0	1.06	1.13	1.26	60%
2.5	1.37	0.79	0.95	100%
3.0	1.87	1.35	1.46	40%
3.5	1.82	0.97	1.14	80%
4.0	1.90	1.17	1.39	60%
4.5	1.89	1.01	1.26	60%
5.0	1.99	1.13	1.28	80%

另外，透過表 5-4(路徑重合度分析)亦可進一步驗證該參數設置的優勢，當 P 值為 1.5 時，其重合率達到 90.91%，為所有測試參數中表現最佳的組合。相對而言，當參數設定過低時，控制器響應過緩，導致路徑偏移程度增加；而當參數設定過高（如 3.5 以上）時，則產生過度修正的現象，使得控制輸出不穩，反而造成轉向過度與路徑震盪，使重合度顯著下降詳見圖 5.29 之視覺化指標。

表 5-5 路徑重合度分析表

	max	average	RMSE	Reoc
0.1	5.24	1.21	1.84	78.27%
0.5	5.83	1.48	2.08	65.52%
1.0	4.37	1.23	1.52	82.20%
1.5	3.29	0.67	1.04	90.91%
2.0	5.80	2.07	2.37	55.54%
2.5	6.72	2.44	7.79	35.36%
3.0	6.32	2.49	2.89	44.56%
3.5	6.18	2.44	2.78	31.27%
4.0	5.99	2.61	2.91	34.97%
4.5	6.03	2.48	2.75	24.58%
5.0	8.79	3.62	4.2	29.20%

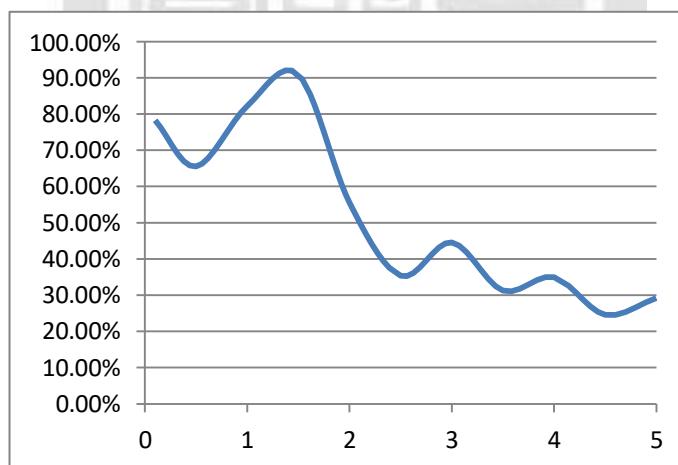


圖 5.29 ATC_STR_RAT_P/路徑重合度圖

綜上所述，ATC_STR_RAT_P 設定為 1.5 時，不僅能有效提升任務點的追蹤精度，同時在整體路徑重合表現上也達到穩定且準確的控制效果，為本研究中轉向靈敏度參數之最佳設置值，最終測試結果見圖 5.30。

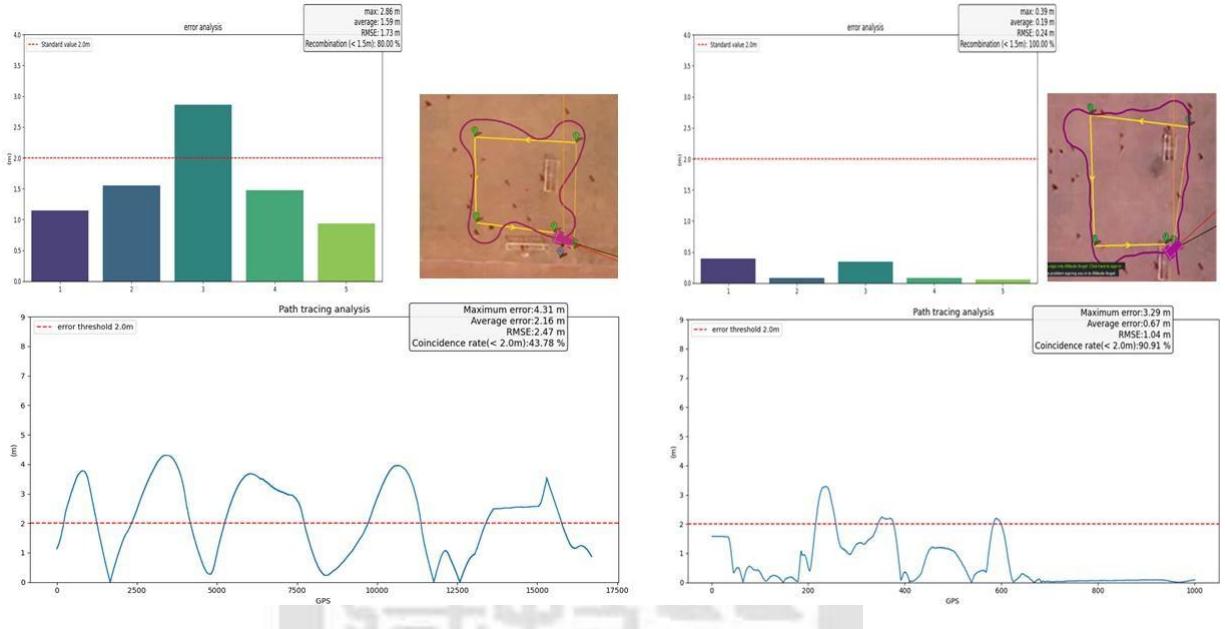


圖 5.30 未調整參數(左)與調整參數(右)對比圖

5.6 訊號不良環境下之導航誤差分析與評估

在實際應用情境中，自駕載具經常需於訊號受干擾或遮蔽之場域執行任務，例如林蔭道路、建築密集區或高牆圍繞區域。此類環境常導致 GPS 訊號品質不穩，進而引發定位誤差放大、航跡偏移等問題，對導航穩定性與任務準確性造成潛在威脅。為深入探討訊號遮蔽環境對本研究自駕系統之影響，本節設計特定「訊號不良場域」進行實地測試，並透過飛行紀錄資料觀察 GPS 狀態變化。隨後，分別以「任務點距離誤差分析」與「路徑重合度分析」兩種方法量化導航精度表現，藉以比較不同遮蔽程度下系統的誤差特性與表現差異。

本節測試場域選擇在高樓林立處，此處有較嚴重的多重路徑效應（Multipath），這樣就能有效地觀察在「訊號不良場域」載具的運動情形，設定的路徑如圖 5.31 所示。



圖 5.31 訊號不良場域路徑

經由測試數據可觀察到，在「訊號不良場域」中，載具執行任務時存在明顯的不穩定性。從任務點距離誤差分析結果可見，其最大誤差高達約 3.5 公尺，平均誤差亦達 1.5 公尺左右，RMSE 則落在 1.8 至 1.9 公尺之間。這些數據反映出，在 GPS 訊號受遮蔽或多重路徑效應影響下，自駕載具的定位準確性顯著降低。進一步的路徑重合度分析也顯示，在此類環境中，整體重合度普遍下降至 50 - 60% 左右，表現遠低於開放場域。雖然誤差多數集中於轉彎處，顯示轉向階段為誤差主因之一，但訊號品質不佳仍會造成全局性的路徑漂移，對自駕導航準確性造成不容忽視的影響，如圖 3.32 所示。

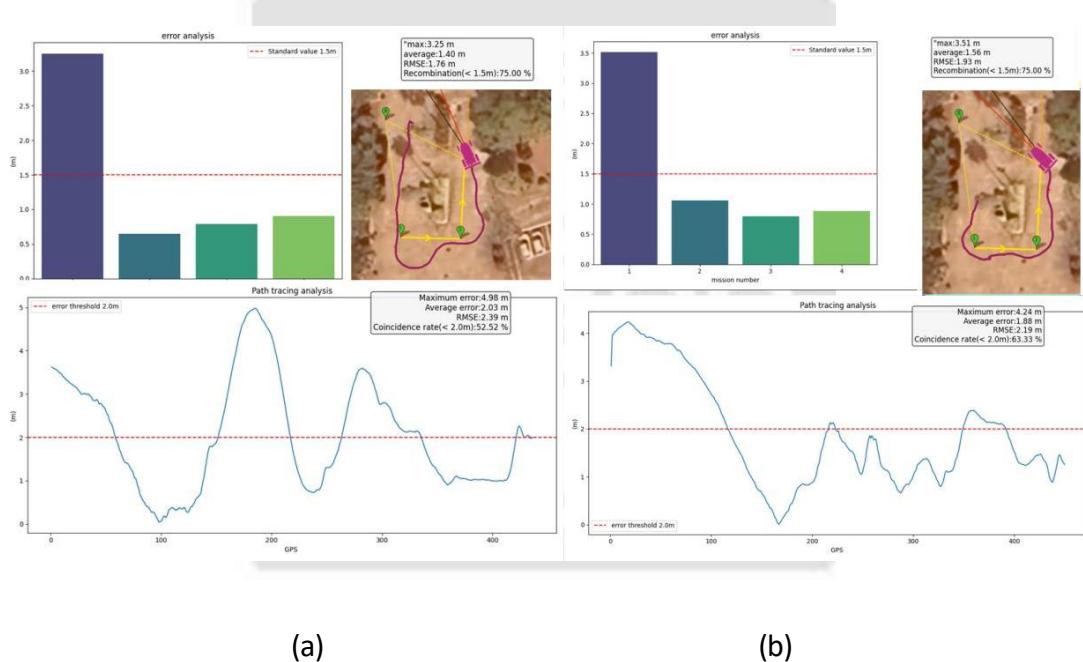


圖 5.32 (a)不良環境測試結果 1 (b) 不良環境測試結果 2

為進一步釐清訊號品質對定位表現的影響，本研究亦針對飛行紀錄中的 GPS 水平精度參數 HAcc (Horizontal Accuracy) 進行分析。結果顯示，在訊號不佳的任務中，HAcc 值常波動於 0.7~1.4 公尺之間，偶有飆升至 2 公尺以上的情形圖 5.33，與高訊號品質任務中穩定維持於 0.01~0.6 公尺的表現相比，差異明顯圖 5.34 所示。這進一步驗證了在建築物密集或遮蔽環境中，訊號品質下降會直接反映在定位誤差與控制穩定性上。

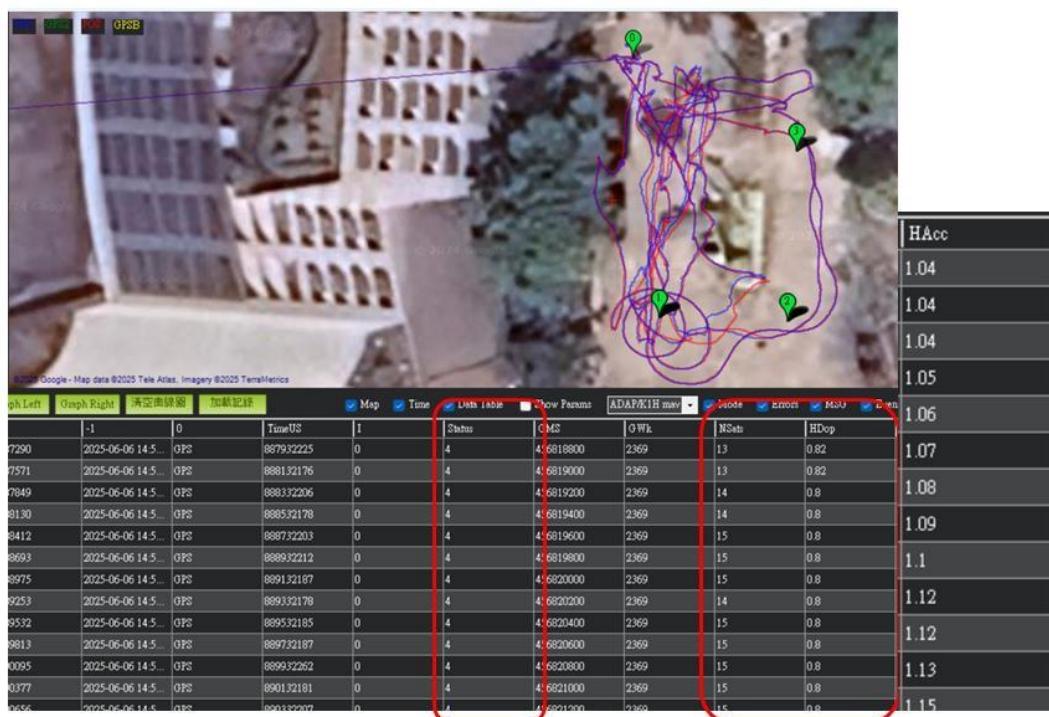


圖 5.33 較差 GPS 訊號數據圖



圖 5.34 較好 GPS 訊號數據圖

然而，值得肯定的是，儘管在嚴苛環境下仍出現定位飄移與重合度降低等現象，但整體路徑仍能維持在合理誤差範圍內，未出現失控或嚴重偏離任務軌跡的情形，顯示本系統具備初步自駕應用潛力。若能搭配 RTK 定位、LiDAR 融合等輔助系統，有望進一步提升導航穩定性與可靠度，拓展其在複雜環境中的應用可能性。

5.7 載具避障測試

當完成最適轉向參數之調整後，本節將進一步探討載具於實際環境中執行避障操作的可行性，並驗證系統於任務執行過程中是否能穩定且即時地切換控制模式，達成正確的避障行為。本階段測試亦將作為系統整體穩定性與應用效能的重要依據。為全面驗證避障能力，本實驗設計三種典型情境進行測試，分別為：左轉避障測試、右轉避障測試及間隙穿越測試。其中左轉與右轉測試用以觀察載具在遭遇單側障礙物時是否能正確判斷方向並執行閃避動作；而間隙測試則模擬載具在面對兩側障礙且需精準穿越狹窄空間時的反應能力與控制穩定性。

5.7.1 左/右轉避障測試

本實驗旨在驗證載具於遭遇單側障礙物時之避障能力與模式切換準確性。測試場景設定於戶外平坦路面，並以塑膠椅子分別作為左側與右側障礙物，模擬載具於執行任務期間突然遭遇障礙的情境。

在左轉避障測試中，塑膠椅子置於載具行進方向右側，系統須於感測器偵測到障礙物後自動切換至避障模式，並向左側進行閃避繞行如圖 5.35，繞過障礙後再順利回歸原任務路徑。實驗過程中，系統能即時識別障礙並進行避障操作，模式順利從導航模式切換至避障模式，避障完成後自動恢復至導航模式如圖5.36 紅框處，最終成功抵達目標點。



圖 5.35 載具左側閃避過程

```

hanyi@hanyi-TUF-GAMING-FX504GD-FX80GD:~ [1] 
[INFO] [1744710441.329966]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.426832]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.427844]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.530873]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.531910]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.655661]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.656589]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.753278]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.825332]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.826332]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.924378]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.925362]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.054095]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.057370]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.122666]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.123677]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.223388]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.224529]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.323491]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.324607]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.446853]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710442.449458]: [RC Override] AUTO mode - Neutral override
[INFO] [1744710441.329966]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710825.899841]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710825.983729]: [RC Override] Turn LEFT
[INFO] [1744710825.984667]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.082028]: [RC Override] Turn LEFT
[INFO] [1744710826.082938]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.182024]: [RC Override] Turn LEFT
[INFO] [1744710826.182947]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.284623]: [RC Override] Turn LEFT
[INFO] [1744710826.386547]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.388622]: [RC Override] Turn LEFT
[INFO] [1744710826.391772]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.488520]: [RC Override] Turn LEFT
[INFO] [1744710826.488536]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.580587]: [RC Override] Turn LEFT
[INFO] [1744710826.581641]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.683638]: [RC Override] Turn LEFT
[INFO] [1744710826.684629]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.809163]: [RC Override] Turn LEFT
[INFO] [1744710826.811884]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710826.911398]: [RC Override] Turn LEFT
[INFO] [1744710826.913764]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710827.084952]: [RC Override] Turn LEFT
[INFO] [1744710827.087039]: [RC Override] Throttle: 1093, Steer: 1891
[INFO] [1744710827.087039]: [RC Override] Throttle: 1093, Steer: 1891

secs: 1744710440
nsecs: 826398369
frame_id: ''
connected: True
armed: True
guided: True
manual_input: False
mode: "AUTO"
system_status: 4
...
header:
  seq: 100
  stamp:
    secs: 1744710441
    nsecs: 826566140
    frame_id: ''
connected: True
armed: True
guided: True
manual_input: False
mode: "AUTO"
system_status: 4
secs: 1744710825
nsecs: 845235454
frame_id: ''
connected: True
armed: True
guided: False
manual_input: True
mode: "MANUAL"
system_status: 4
...
header:
  seq: 584
  stamp:
    secs: 1744710826
    nsecs: 845318604
    frame_id: ''
connected: True
armed: True
guided: False
manual_input: True
mode: "MANUAL"
system_status: 4

```

(a)

(b)

圖 5.36 (a) 載具未偵測障礙物(Auto); (b) 載具右側偵測到障礙物(Manual)

同樣地，在右轉避障測試中，塑膠椅子改置於載具左側路徑，系統則執行右側閃避策略。經實測，載具亦能正確辨識障礙方位並選擇適當閃避方向圖 5.37 紅箭頭所示，模式切換準確無誤，完成避障後亦能順利接軌原始路徑完成任務圖 5.38 紅框處所示。



圖 5.37 載具右側閃避過程

```
[INFO] [1744710884_2558858]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_357616]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_358624]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_4569569]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_551061]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_557709]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_558892]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_655106]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_656270]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_782602]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_812753]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_837420]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_976470]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_977528]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_981454]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_981541]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_183516]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_2852402]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_288746]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_390130]: [RC Override] AUTO mode - Neutral override  
[INFO] [1744710884_393545]: [RC Override] AUTO mode - Neutral override  
  
[INFO] [1744710846_339191]: [RC Override] Throttle: 1891, Steer: 1500  
[INFO] [1744710846_418253]: [RC Override] Throttle: 1891, Steer: 1500  
[INFO] [1744710846_518121]: [RC Override] Throttle: 1891, Steer: 1500  
[INFO] [1744710846_518121]: Turn RIGHT  
[INFO] [1744710846_519174]: [RC Override] Throttle: 1093, Steer: 1093  
[INFO] [1744710846_620674]: Turn RIGHT  
[INFO] [1744710846_621769]: [RC Override] Throttle: 1093, Steer: 1093  
[INFO] [1744710846_621769]: Turn LEFT  
[INFO] [1744710846_725202]: [RC Override] Throttle: 1093, Steer: 1891  
[INFO] [1744710846_816891]: Turn LEFT  
[INFO] [1744710846_949315]: Turn RIGHT  
[INFO] [1744710846_950209]: [RC Override] Throttle: 1093, Steer: 1891  
[INFO] [1744710846_950209]: Turn RIGHT  
[INFO] [1744710847_016111]: [RC Override] Throttle: 1093, Steer: 1891  
  
[INFO] [1744710847_114290]: TURN RIGHT  
[INFO] [1744710847_115334]: [RC Override] Throttle: 1093, Steer: 1891  
[INFO] [1744710847_215567]: [RC Override] Throttle: 1093, Steer: 1891  
[INFO] [1744710847_312461]: [RC Override] Throttle: 1093, Steer: 1891  
[INFO] [1744710847_313582]: Turn RIGHT  
[INFO] [1744710847_412096]: [RC Override] Throttle: 1093, Steer: 1093  
[INFO] [1744710847_413878]: [RC Override] Throttle: 1093, Steer: 1093
```

```
hanyi@hanyi-TUF-GAMING-FX504GD-FX80GD:~
```

```
secs: 1744710843  
nsecs: 845352766  
frame_id:  
connected: True  
armed: True  
guided: True  
manual_input: False  
mode: "AUTO"  
system_status: 4  
  
header:  
  seq: 003  
  stamp:  
    secs: 1744710844  
    nsecs: 845186419  
  frame_id:  
connected: True  
armed: True  
guided: True  
manual_input: False  
mode: "AUTO"  
system_status: 4  
  
secs: 1744710485  
nsecs: 829216192  
frame_id:  
connected: True  
armed: True  
guided: False  
manual_input: True  
mode: "MANUAL"  
system_status: 4  
  
header:  
  seq: 208  
  stamp:  
    secs: 1744710486  
    nsecs: 829439916  
  frame_id: ''  
connected: True  
armed: True  
guided: False  
manual_input: True  
mode: "MANUAL"  
system_status: 4  
***
```

(a) (b)

圖 5.38 (a) 載具未偵測障礙物(Auto); (b) 載具右側偵測到障礙物(Turn Right)

綜上結果顯示，本系統於左右兩側避障操作中皆能準確辨識障礙並正確完成模式切換及控制響應，具備良好之環境適應性與任務連貫性，驗證所設計之避障機制具實用性與穩定性。測試總結果如表 5-5 所示，其中第 7 次測試出現任務失敗情形，經分析為長時間運行 ROS 系統導致 log 暫存資料累積過多，進而引發系統訊號延遲。經清除 `~/.ros/log` 資料夾後重新執行，問題即獲得改善，後續測試皆無再出現相同異常。

表 5-6 左右避障測試表

次數	右避障是否成功	左避障是否成功
1	成功	成功
2	成功	成功
3	成功	成功
4	成功	成功
5	成功	成功
6	成功	成功
7	失敗	失敗
8	成功	成功
9	成功	成功
10	成功	成功

5.7.2 間隙穿越測試避障測試

承接前節之測試，本節進一步進行載具於狹窄間隙中的避障能力驗證。在實驗中，我們把避障間距定為 1.5 公尺，原因如下：系統偵測避障點的閾值是 2 公尺，當障礙物間距超過這個值時，載具只會把最先偵測到的物體當避障參考，可能就會忽略原本可以通過的縫隙。如果避障間距設太大，就難以辨識並穿越這些狹窄空間，相反若設定太小，又可能因車身尺寸和轉向半徑的限制而增加碰撞風險。因此綜合考量本研究所使用之載具車長約為 1 公尺，為確保穿越過程中具備足夠的轉向與修正空間，遂將間隙寬度設定為 1.5 公尺，作為本次實驗之標準。實驗過程中，於測試場地擺放兩張間距為 1.5 公尺之塑膠椅模擬障礙物，並觀察載具於自動導航模式下是否能順利識別通行區域、完成避障決策，並成功穿越間隙，以驗證系統於狹窄環境中的應用可行性與穩定性。實驗結果顯示，載具於偵測到障礙物後，行為表現與模擬結果一致，皆會優先選擇障礙間距較寬處繞行，並以 S 型曲線順利穿越障礙物間隙，如圖 5.39 紅箭頭所示。此結果證實，本系統具備在複雜環境中正確辨識通行空間並執行動態避障之能力，展現出良好的導航靈活性與可靠性。



圖 5.39 間隙穿越過程

測試總體結果如表 5-6 所示，於 10 次間隙穿越測試中，僅第 7 次出現失敗情形，前節已說明其主因為系統暫存過多導致之訊號延遲，並已於後續排除。整體而言，本系統之避障邏輯展現出良好的準確性與穩定性，具備實際應用之可靠性。

表 5-7 縫隙穿越測試表

次數	縫隙穿越是否成功
1	成功
2	成功
3	成功
4	成功
5	成功
6	成功
7	失敗
8	成功
9	成功
10	成功

5.7.3 校園實地測試

本節最終將於校園環境中規劃一條完整實測路線，並啟用所有系統功能（包含避障機制與 RTK 高精度定位）以執行自動導航任務，驗證系統於實際場域中的整合效能與穩定性。測試路線如圖 5.40 所示，載具將從校門口出發，於紅圈處完成左轉動作後沿校園主要車行道路直行，途中經過兩處小幅度轉彎（標註於藍圈處），最終抵達指定終點位置。此測試設計貼近實際應用場景，旨在全面評估系統之導航能力與環境適應性。

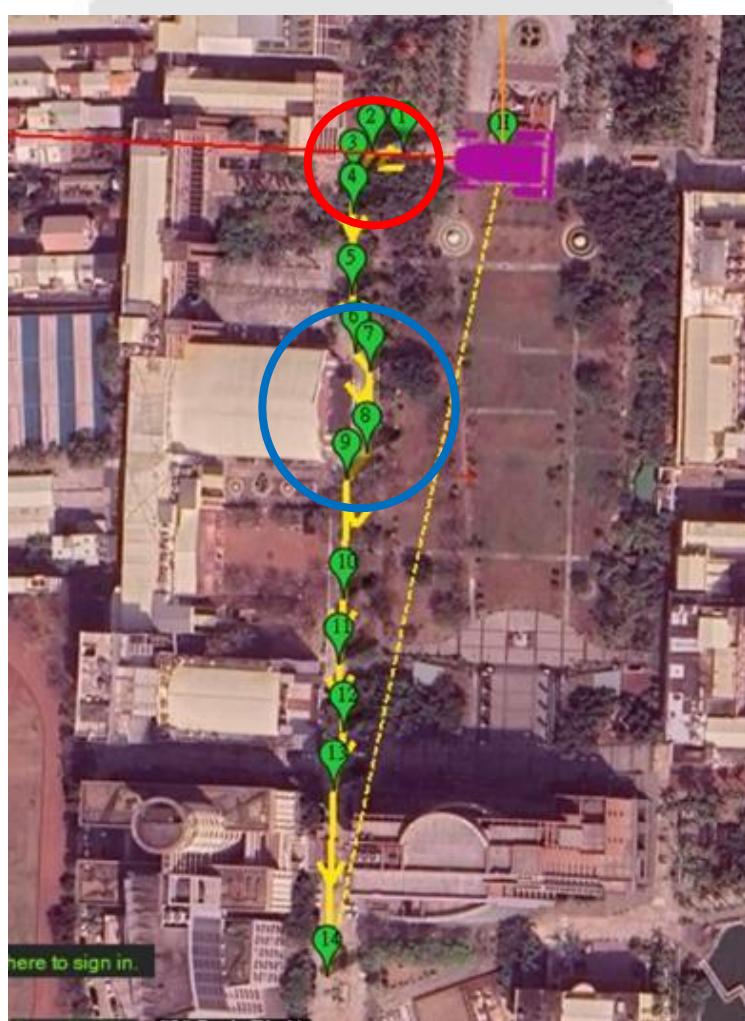


圖 5.40 實地測試路線圖

測試路徑如圖 5.41 所示，載具於約 125 公尺的導航範圍內展現出良好的穩定性，整體行進軌跡平穩，無明顯過衝（overshoot）或偏航現象。該結果有效驗證本系統於實際環境下整體功能整合之可行性與執行效能，顯示其具備應用於中短距離自動導航任務的潛力。

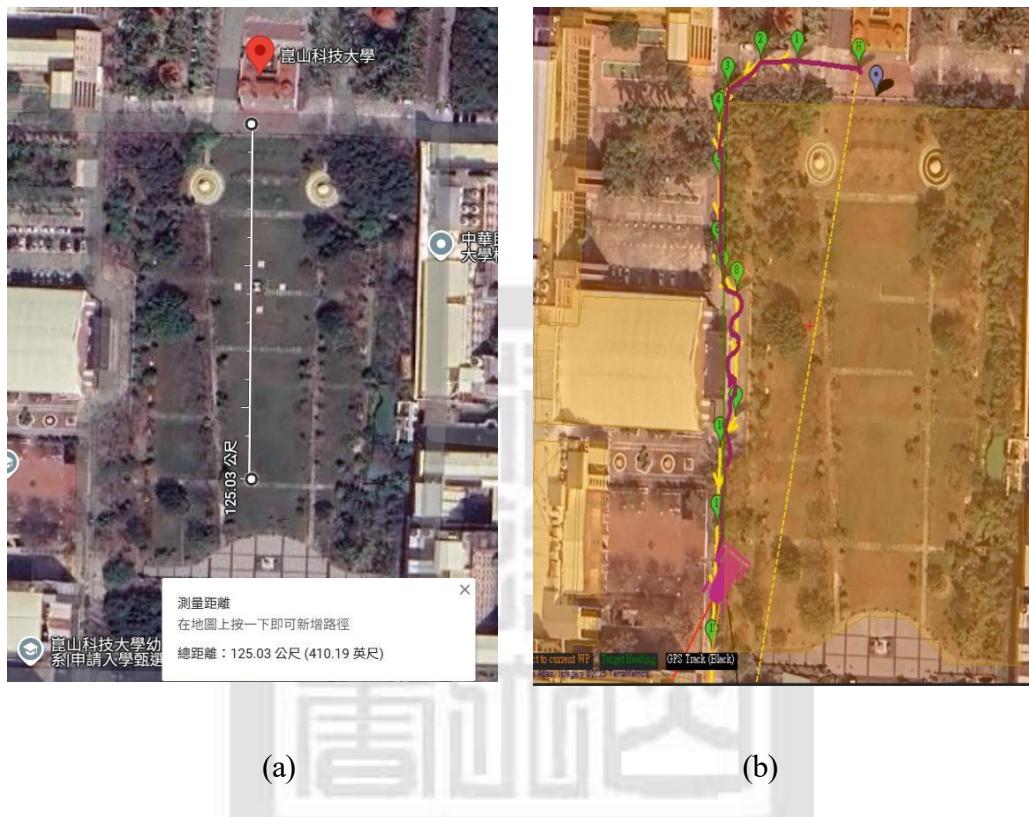


圖 5.41 (a)測試場域距離；(b)實際路徑軌跡圖

圖 5.42 為針對本次校園實地導航測試所進行之任務點距離誤差分析，圖中以紅色虛線標示出誤差容許上限（1.5 公尺），用以視覺化標準容許範圍。從 13 個任務點中可觀察到，整體最大誤差為 4.00 公尺，平均誤差為 1.26 公尺，RMSE 為 1.68 公尺，其中有 61.54% 的樣本誤差落在 1.5 公尺以內。由此可見，儘管部分任務過程出現偏離現象，但整體導航表現仍維持在可接受範圍，具備穩定基礎。

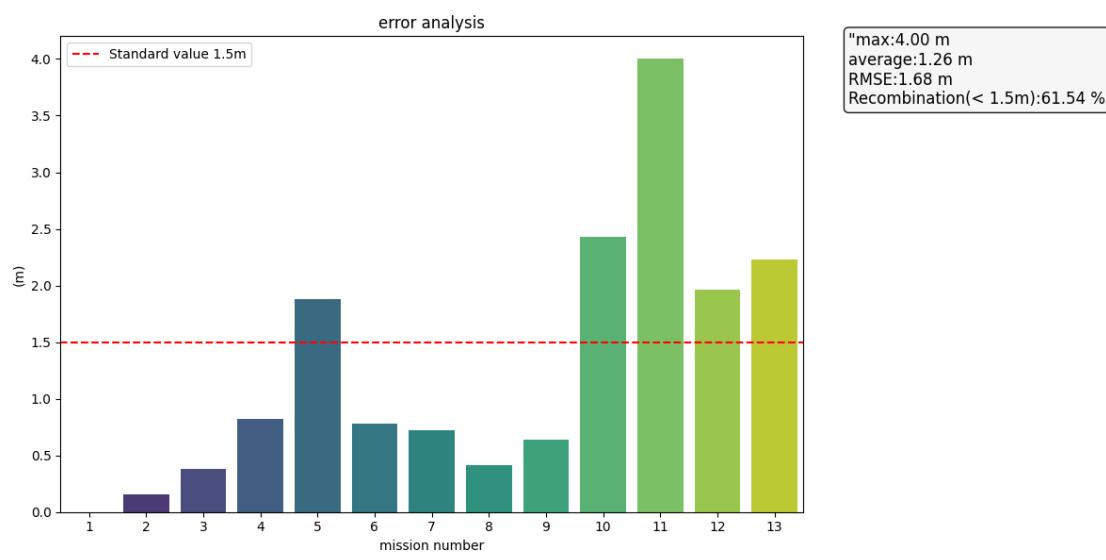


圖 5.42 校園測試任務點距離誤差分析

進一步探討誤差分佈於任務執行過程之具體表現，圖 5.43 呈現該次導航任務的完整路徑誤差曲線。由圖可知，多數時間段之誤差皆低於紅色虛線標示的 2 公尺閾值，顯示大部分導航路徑誤差穩定可控。本次測試最大誤差為 3.98 公尺，平均誤差為 1.18 公尺，RMSE 為 1.44 公尺，路徑重合率（誤差小於 2 公尺）達 80.64%，此測試驗證本系統在訊號良好情境下之實用性與穩定性。

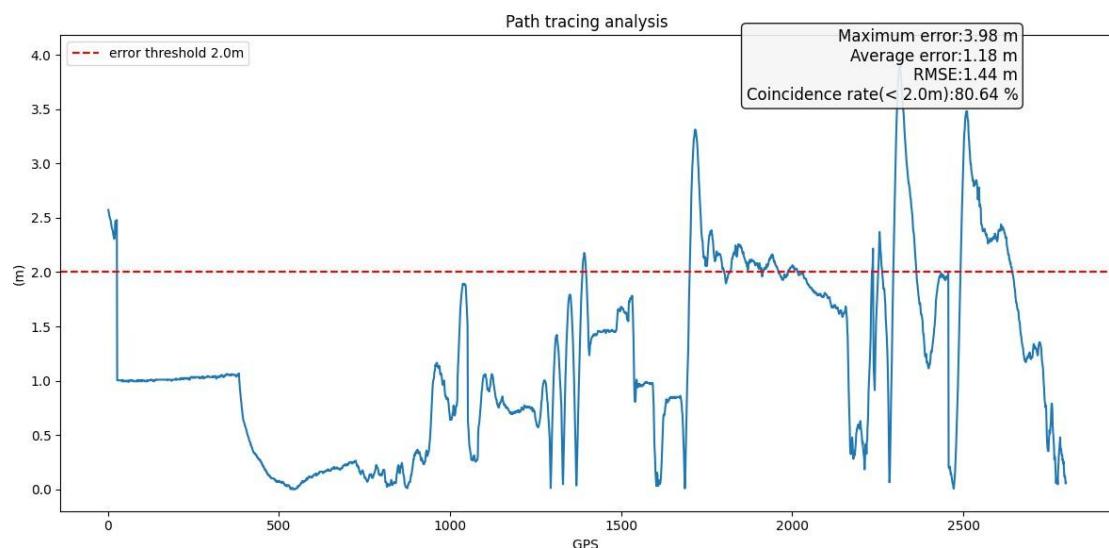


圖 5.43 校園測試路徑重合度分析

第六章 結論與建議

本研究成功提出並驗證了一套整合 ROS、Pixhawk 與 PLC 系統的創新自駕載具控制架構。透過戶外實地實驗（以沙灘車為平台），證實該系統於複雜地形環境下具備良好的定位精度、即時避障能力及路徑追蹤穩定性。透過 Gazebo 模擬、RTK-GPS 定位、LiDAR 感測及 PLC 之 PWM/DAC 多層訊號轉換等實驗流程，進一步確認本研究所提「多控制架構整合」在實務應用中的可行性與穩定性。

6.1 研究成果歸納

本研究提出的 ROS - Pixhawk - PLC 架構，成功克服了傳統系統在跨硬體整合上的限制。ROS 負責模組化的感知資訊整合與即時決策，Pixhawk 提供穩定高效的導航控制，PLC 則擔當即時精準的硬體控制角色。實驗結果證明，該架構能顯著提升自駕載具的性能與穩定性。

經過戶外實地測試，本系統採用 RTK-GPS 定位並結合Haversine 演算法進行分析，整體定位誤差平均僅約 1.5 公尺，RMSE 為 1.8 公尺；通過進一步的系統參數調整（例如將轉向參數ATC_STR_RAT_P 調至 1.5），最大誤差可有效降低至 0.47 公尺，平均誤差則縮減至 0.23 公尺，RMSE 更下降至 0.26 公尺。在彎道與直線路段中，路徑重合度均能維持在優良範圍內，完全滿足如戶外探索、農業巡覽及環境探測等高要求應用場景的需求。

針對即時避障性能，本系統利用 LiDAR 感測器結合 ROS 控制架構，在多次模擬真實場景的戶外實測中（如左轉、右轉及狹窄通道穿越），顯示出極高的反應速度及正確的避障方向選擇能力。經系統日誌處理流程優化後，也未再因任務中斷問題影響運行，展現出了該避障邏輯穩定且具實務應用價值的可靠性。

6.2 與現有研究差異及性能優勢

相較於近期相關研究成果（如楊釋詠，2024），本研究在創新性與優勢方面展現以下幾點：首先，硬體控制穩定性獲得顯著提升。本系統採用工業級 PLC 作為即時控制介面，結合 Pixhawk 飛控系統，有效提高硬體控制的可靠性與耐用性，使其在戶外嚴苛環境中表現出更高的穩定性；其次，本系統使用RTK-GPS 定位技術，與傳統以 AMCL 為主的室內外切換定位方法相比，更適合用於戶外長距離導航，能顯著降低外部環境因素的影響；此外，本研究提出的整合架構具有極高的實務應用潛力，特別適用於農業自動化、環境監測及救援任務等複雜戶外應用場景。

根據公開的自駕載具規格資料，目前戶外自駕載具的定位精度普遍要求維持在 ± 2 m 以內（95%信心水準），而 RTK GNSS 在理想情況下甚至可達到厘米級精度。此外，工業級 AGV 的路徑重合度一般期望達到 90%以上。相比之下，本研究的戶外定位平均誤差約為 1.5 m，經調整後 RMSE 更僅為 0.26 m，已符合絕大多數產業界對戶外導航定位精度的需求。然而，在路徑重合度方面（約 60%-70%），仍有進一步改進的空間。未來可通過融合更高精度的感測器與多傳感器定位技術，不斷提升系統性能，力求達到甚至超越產業標準水準。

6.3 未來研究建議與展望

相較於最近相關研究成果（如楊釋詠，2024），本研究的創新點與優勢主要體現在以下幾個方面：首先，在硬體控制部分，本研究建議未來系統可進一步整合 AMCL、視覺 SLAM 等多種定位方案，並融合 IMU 和雙天線 RTK，搭配 EKF 或 UKF 濾波器進行感測訊號的融合處理，以降低 GNSS 訊號不穩定造成的影響，強化其在多樣環境中的適應能力。此外，建議針對內外場域切換控制策略進行深入探討，以實現更高精度和穩定性的導航表現。同時，可升級感測系統至 3D LiDAR 並採用 DroneCAN 或 CANFD 通訊協議，直接驅動伺服馬達，結合狀態

機與 PLC Watchdog 機制，以提升系統的整體可靠性與穩健性。

在智能化層面，可考慮未來導入深度學習或強化學習技術（如 YOLOv8、PointPillars）來優化障礙物偵測效能，並藉助模型預測控制（MPC）實現路徑規劃與速度控制的最佳化設計，從而進一步提升即時避障與決策的效率。此外，建議拓展系統應用至多載具協同作業場景，採用多智能體通訊架構（如 ROS2 或 JADE），以實現載具間的即時協調與更靈活的團隊合作能力。

在實際應用推廣方面，初期可將系統部署於校園巡檢、農田植保或海灘清潔等封閉或半封閉場域，並逐步實現與 PLC 與 SCADA 系統的整合，構建自駕載具、監控系統及雲端平台三位一體的智慧服務模式，符合國內低速自駕載具的相關法規要求，促進技術在產業中的落地應用。

綜上，本研究提出的 ROS–Pixhawk–PLC 自駕載具控制系統通過戶外實驗驗證，已充分展現其在定位精準度、系統整合性與即時控制能力上的優越性與穩定性。未來若能持續強化感測融合技術、能耗管理以及車隊協同操作能力，其性能則可進一步提升，同時能於更多產業應用場景中展示自駕載具智慧化應用的巨大潛力與價值。

參考資料

- [1] 王紹華, 《一個基於 ROS 的自動駕駛系統效能分析框架》, 國立成功大學資訊工程學系博士論文, 臺南, 2024 年初版。
- [2] 楊釋詠、江智偉, 《電動自駕載具軟體組件研究分析與應用》, 崑山科技大學機械工程系碩士論文, 臺南, 2024 年 8 月。
- [3] S. Kato, S. Tokunaga, Y. Maruyama, *et al.*, “Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems,” *Proc. Ninth ACM/IEEE Int. Conf. on Cyber-Physical Systems (ICCPs 2018)*, Porto, 2018.
- [4] M. A. Post, A. Bianco, X. T. Yan, “Autonomous Navigation with ROS for a Mobile Robot in Agricultural Fields,” *Proc. 14th Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO 2017)*, pp. 79-87, Glasgow, 2017.
- [5] M. W. Parker, B. Quinn, J. Bates, *et al.*, “Exploring Cold Regions Autonomous Operations,” *Journal of Terramechanics*, vol. 96, pp. 159-165, 2021.
- [6] P. Marín-Plaza, A. Hussein, D. Martin, A. de la Escalera, “Complete ROS-based Architecture for Intelligent Vehicles,” *Proc. IEEE Intelligent Vehicles Symp. (IV 2018)*, Changshu, 2018.
- [7] T. Neto, R. Arrais, A. Sousa, A. Santos, G. Veiga, “Applying Software Static Analysis to ROS: The Case Study of the FASTEN European Project,” *Proc. IEEE/IFAC Int. Conf. on Cyber-Physical & Human-Systems*, 2020.
- [8] A. Santos, A. Cunha, N. Macedo, “ROS Software Quality Assessment through Static Code Analysis and Property-Based Testing,” ROS-Industrial Conference 2018, Stuttgart, 2018.
- [9] D. Tardioli, R. Parasuraman, P. Ögren, “Pound: A ROS Node for Reducing Delay and Jitter in Wireless Multi-Robot Networks,” *arXiv preprint arXiv:1707.07540*, 2017.

- [10] G. Prescinotti Vivan, N. Goberville, Z. Asher, N. Brown, J. Rojas, “No Cost Autonomous Vehicle Advancements in CARLA through ROS,” SAE Technical Paper 2021-01-0106, SAE International, 2021.
- [11] L. Meier、P. Tanskanen 、F. Fraundorfer、M. Pollefeys , The PIXHAWK Open-Source Computer Vision Framework for MAVs, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* , Vol. XXXVIII-1/C22, Zurich, Switzerland, 2011.
- [12] A. S. Wardoyo、I. P. Sary、I. T. Maulana, “An Analysis of the Performance of Autonomous Navigation on an Ardupilot-Controlled Rover,” *Ultima Computing: Jurnal Sistem Komputer*, 14 (2): 82-89, Indonesia, 2022.
- [13] K. Xiao、S. Tan、G. Wang、X. An、X. Wang、X. Wang, XTDrone: A Customizable Multi-Rotor UAVs Simulation Platform , *Proceedings of the IEEE International Conference on Robotics and Automation Sciences (ICRAS 2020)*, pp. 1-6, Xiamen, 2020.
- [14] https://ardupilot.org/dev/docs/ros-aruco-detection.html?utm_source=chatgpt.com
- [15] https://ardupilot.org/dev/docs/ros.html?utm_source=chatgpt.com
- [16] https://discuss.px4.io/t/sitl-setup-with-xacro/6437?utm_source=chatgpt.com
- [17] https://discuss.px4.io/t/px4-sitl-with-gazebo-custom-vehicle-model/12066?utm_source=chatgpt.com
- [18] 陳謙信, 基於 ROS 之整合 Pixhawk 飛控板與 Jetson TX2 模組之無人機自 主飛行控制系統設計, 淡江大學航空太空工程學系碩士論文, 新北, 2022 初版。
- [19] K. Xiao、S. Tan、G. Wang、X. An、X. Wang、X. Wang, XTDrone: A Customizable Multi-Rotor UAVs Simulation Platform (開源手冊 v1.2) , GitHub, <https://github.com/robin-shaun/XTDrone>, 2021。:contentReference[oaicite:10]{index=10}
- [20] 張豪文, U-net 深度學習於道路行人、車、機踏車之 LiDAR 三維點雲即時 語義分割, 國立高雄大學資訊工程學系碩士論文, 高雄, 2020 初版。
- [21] 彭晨益, 融合運動分割及目標偵測提升應用於掃描式成像型雷達之移動目標 偵測・國立陽明交通大學電控工程研究所碩士論文, 新竹, 2024 初版。

- [22] 張脩寬，基於多目標蟻群優化之模糊系統應用於導航避障雙輪車輛，國立暨南國際大學電機工程學系碩士論文，南投，2024 初版。
- [23] D. J. Yeong, G. Velasco-Hernandez, J. Barry, J. Walsh, *Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review*, Sensors, 21 (6): 2140, MDPI, Basel, Switzerland, 2021.
- [24] Jon Martin, Oskar Casquero, Brais Fortes, Marga Marcos, *A Generic Multi-Layer Architecture Based on ROS-JADE Integration for Autonomous Transport Vehicles*, Sensors, 19 (1): 69, MDPI, Basel, Switzerland, 2019.
- [25] Vygantas Ušinskis, Michał Nowicki, Andrius Dzedzickis, Vytautas Bučinskas, *Sensor-Fusion Based Navigation for Autonomous Mobile Robot*, Sensors, 25 (4): 1248, MDPI, Basel, Switzerland, 2025.
- [26] Stanislav Alexovič、Milan Lacko、Ján Bačík, Simulation of Multiple Autonomous Mobile Robots Using a Gazebo Simulator, 收錄於 *Advances in Manufacturing, Production Management and Process Control* (Springer 出版 , ISBN 978-3-031-21435-6) , 頁 433-444, 2023 初版。
- [27] Linus Nwankwo, Clemens Fritze, Konrad Bartsch, Elmar Rueckert, *ROMR: A ROS-based Open-Source Mobile Robot*, HardwareX, 14: e00426, Elsevier, 2023.
- [28] Benjamin Bennett, Accurate Distance Calculation Using GPS While Performing Low Speed Activity, 碩士論文, Department of Computer and Information Science, University of Oregon, Eugene, U.S.A., 2018.
- [29] Mangesh K. Nichat, “Landmark Based Shortest Path Detection by Using A* Algorithm and Haversine Formula”, International Journal of Innovative Research in Computer and Communication Engineering, 1 (2): 298-303, 2013.
- [30] 江偉銘，應用時間序列分析於 GPS 多路徑效應之研究，國立中央大學土木工程學系碩士論文，桃園，2008 初版。
- [31] Doo Won Han、Jundi Liu、X. Jessie Yang 等, “Toward Integrated Takeover Performance Measurement: Validation of Fréchet Distance as a Takeover Performance Metric”, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 68 (1): 866-872, 2024.

[32] Yaguang Tao、Alan Both、Rodrigo I. Silveira 等, “A Comparative Analysis of Trajectory Similarity Measures”, *GIScience & Remote Sensing*, 58 (5): 643-669, 2021。

[33] Siqi Wan、Huaqiao Mu、Ke Han 等, “A Fuzzy Track-to-Track Association Algorithm with Dynamic Time Warping for Trajectory-Level Vehicle Detection”, *International Journal of Transportation Science and Technology*, 17: 95-108, 2025。

[34] Mohammad Rokonuzzaman Navid、Mohajer Navid、Saeid Nahavandi、Shady Mohamed, Review and Performance Evaluation of Path Tracking Controllers of Autonomous Vehicles, *IET Intelligent Transport Systems*, 15: 646-670, 2021。

[35] 劉世達、林廣、紀洪海、王莉, R-DMFAC 無模型自適應控制於自主車輛路徑追蹤之應用, *公路與交通科研與發展*, 18 (3): 68-75, 北京, 2024。

[36] Danial Pour Arab、Matthias Spisser、Caroline Essert, “3D Hybrid Path Planning for Optimized Coverage of Agricultural Fields: A Novel Approach for Wheeled Robots”, *Journal of Field Robotics*, 42: 455-473, 2025。