

國立中央大學

資訊工程研究所
碩士論文

多軸飛行器的障礙物偵測與避障系統

Obstacle detection and collision avoidance for multicoptors

研究生：盧柏瑋

指導教授：曾定章 教授

中華民國一百零六年七月



國立中央大學圖書館 碩博士論文電子檔授權書

(104 年 5 月最新修正版)

本授權書授權本人撰寫之碩/博士學位論文全文電子檔(不包含紙本、詳備註 1 說明)，在「國立中央大學圖書館博碩士論文系統」。(以下請擇一勾選)

☒ **同意** (立即開放)

☐ **同意** (請於西元 _____ 年 _____ 月 _____ 日開放)

☐ **不同意**，原因是：_____

在國家圖書館「臺灣博碩士論文知識加值系統」

☒ **同意** (立即開放)

☐ **同意** (請於西元 _____ 年 _____ 月 _____ 日開放)

☐ **不同意**，原因是：_____

以非專屬、無償授權國立中央大學、台灣聯合大學系統圖書館與國家圖書館，基於推動「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、微縮、光碟及其它各種方法將上列論文收錄、重製、與利用，並得將數位化之上列論文與論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

研究生簽名：_____ 盧柏瑋 _____ 學號：_____ 104522014 _____

論文名稱：_____ 多軸飛行器的障礙物偵測與避障系統 _____

指導教授姓名：_____ 曾定章 _____

系所：_____ 資訊工程 _____ 所 ☐ 博士班 ☒ 碩士班

填單日期：_____ 106/08/21 _____

備註：

1. 本授權書之授權範圍僅限**電子檔**，紙本論文部分依著作權法第 15 條第 3 款之規定，採推定原則即預設同意圖書館得公開上架閱覽，如您有申請專利或投稿等考量，不同意紙本上架陳列，須另行加填申請書，詳細說明與紙本申請書下載請至本館數位博碩士論文網頁。
2. 本授權書請填寫並**親筆**簽名後，裝訂於各紙本論文封面後之次頁（全文電子檔內之授權書簽名，可用電腦打字代替）。
3. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列論文，應遵守著作權法規定。

國立中央大學碩士班研究生

論文指導教授推薦書

資訊工程學系/研究所 盧柏瑋 研究生所提之論文
多軸飛行器的障礙物偵測與避障系統 係由本人指
導撰述，同意提付審查。

指導教授



(簽章)

106年6月26日

國立中央大學碩士班研究生
論文口試委員審定書

資訊工程學系/研究所 盧柏瑋 研究生所提之論文
多軸飛行器的障礙物偵測與避障系統 經本委員會
審議，認定符合碩士資格標準。

學位考試委員會召集人
委 員

林金良 譚
王 家 慶
曾 慶 章

中 華 民 國 年 月 日

摘要

目前無人機常被用於農業、商業、軍事、娛樂、救災、送貨、國防等領域的事務上。無人機的控制方式有利用遙控、導引或自我飛行等等的方式。無人機在執行任務時，若遇障礙物時，沒有良好的處理機制，可能容易發生事故。因此本研究結合多重感測器進行姿態解算，利用同步定位與地圖構建技術 (*SLAM*) 做障礙物偵測系統，與方向區間柱圖法 (*VFH*) 和障礙物偵測結果來實作飛行器避障系統，並將產生的飛行訊號大小與飛行方向，來控制飛行器以達成穩定飛行的目的。

本論文利用九軸姿態模組得到三軸加速度、三軸角速度與三軸磁場值，計算出目前機體座標系的俯仰角、滾動角、偏航角，再將 *RC* 接收器接收到的使用者操作遙控器的信號做數值穩定控制，得到數值的誤差量，根據角度誤差量大小決定要修正量以供姿態控制時使用。

在障礙物偵測上，我們使用 *SLAM* 方法並結合超音波感測器。首先利用超音波感測器偵測到的障礙物距離數值來做區域地圖網格構建，並利用航向參考演算法找出飛行器的移動方向與距離，最後利用全域地圖網格構建飛行器的方位。得到飛行器位置與障礙物的環境地圖後做 *VFH* 的避障分析，再將飛行器的偵測環境先劃分區塊，計算地圖裡每個網格距離障礙物的可能值，找出每個區塊的障礙物密度，分析成一維平滑極性機率密度圖後，判斷飛行器是否有避障的飛行方向，若沒有即讓飛行器懸停，若有飛行器即完成姿態避障。

最後介紹我們的實驗設備與實驗結果。透過姿態解算實驗，飛行器在其他角度時皆可依據 *IMU* 姿態的變化，利用角度誤差 *PID* 控制使飛行器得到合適地飛行角度，*Arduino* 回傳 *Pixhawk* 相對應地 *PWM* 訊號，使飛行器回歸平穩狀態。透過障礙物偵測與避障實驗，雖飛行器在電腦端執行演算法效率佳，但因 *Arduino* 微控制板處理器硬體的限制，使得障礙物偵測與避障處理效果並沒有達到完全即時性。

Abstract

Just in these few years, unmanned aerial vehicles (*UAVs*) have been massively applied in agriculture, commerce, military, entertainment, disaster relief, delivery, defense, *etc.* In general, *UAVs* can be controlled by controllers, following a target, or autonomous flying. Most *UAVs* have not equipped an obstacle detector to avoid the possible collision. Thus many *UAV* accidents have occurred. In this study, we combine multiple sensors for posture calculation, use synchronous positioning and map construction technology (*SLAM*) to detect obstacles for *UAV*'s collision avoidance.

In this study, a three-axis acceleration, three-axis angular velocity, and three-axis magnetic field are used to obtain the nine-axis attitude. The pitch angle, roll angle, and yaw angle of the *UAV* body are then calculated, and the remote control signal is received to reduce the amount of error.

In the obstacle detection, we use *SLAM* method combining ultrasonic sensors to construct the regional map grid, and use the heading reference algorithm to find the moving direction of the *UAV*. At last, the global map grid with *UAV* position and obstacle locations is constructed. In the collision avoidance, a one-dimensional *VFH* representation is transformed from the global map grid to find possible space for forward flying.

Through the obstacle detection and obstacle avoidance experiment, although the *UAV* is simulated in compute system well, the Arduino micro-controller processor is low-end, it shows lower computational performance; thus, the obstacle detection and avoidance are not demonstrated so good as the simulation.

誌謝

本論文承蒙指導教授 曾定章教授兩年來的諄諄教誨，以及不辭辛勞地指導並協助論文的校正與定稿，使本論文可以順利付梓，在此致上最誠摯的感謝。同時也感謝口試委員 林銀議教授、 王家慶副教授的建議與指導，使得本論文更加完整。

再次感謝 曾定章教授提供良好的研究環境，研究資源不虞匱乏，使我們能夠專心於研究工作，順利完成學業。不僅在學業上使我受益匪淺，也讓我在為人處事方面得到許多的收穫。

在這兩年的日子裡，感謝實驗室徐震濤學長、林昱岐學長以及張宸銘學長的帶領和指導；感謝學長姐、同學及學弟妹們無論是在研究或是生活方面都互相扶持和鼓勵，特別是書恆、美伶、舜暉、仲仁和世偉等學長姐，奕明、明訢、卿瑋和亭玚等同學，以及鎧楠、宛真、鈞惟、寧諄等學弟妹陪我渡過了兩年研究所的時光，不管是渡過難熬的書報週，還是被追著論文進度卻實驗苦無進展的時候，或休閒時刻大家的歡笑時分，都使我心裡甚是感激。很多過不去的時候還好有女友的扶持，總是鼓勵著我，心裡難受時，更是安慰著我使我能更堅強地走下去；感謝天父上帝一路保守著我，賜予軟弱的我智慧與能力，帶領我使我不致缺乏；謝謝我永遠的後盾家人們，一路陪伴著我像個避風港般，使我能重拾力量，感謝很多朋友與長輩們的支持與鼓勵，從你們的身上我學到了很多。

目錄

摘要	ii
Abstract	iii
目錄	v
圖目錄	vi
表目錄	ix
第一章 緒論	1
1.1. 研究動機	1
1.2. 硬體架構	1
1.3. 系統架構	2
1.4. 論文架構	7
第二章 相關研究	8
2.1. 飛行器姿態解算與姿態控制	8
2.2. 障礙物偵測與避障	13
第三章 姿態解算與障礙物偵測	28
3.1. 姿態解算	28
3.2. 障礙物偵測演算法	37
第四章 避障分析與姿態控制	43
4.1. 避障演算法	44
4.2. 飛行器姿態控制	49
第五章 實驗	51
5.1. 實驗設備介紹	51
5.2. 實驗結果與展示	54
5.3. 實驗結果分析	59
第六章 結論及未來展望	61
6.1. 結論	61
6.2. 未來展望	62
參考文獻	63

圖目錄

圖 1.1.	硬體架構圖。	1
圖 1.2.	簡略架構圖。	2
圖 1.3.	完整流程圖。	2
圖 1.4.	姿態解算流程圖。	3
圖 1.5.	PID 控制系統。	4
圖 1.6.	障礙物偵測流程圖。	4
圖 1.7.	避障分析流程圖。	5
圖 1.8.	飛行器姿態控制流程圖。	6
圖 2.1.	呈現四軸飛行器的機體座標系和外部座標，螺旋槳的配置與馬達轉速可組合出 (a)翻滾 (b)俯仰 (c)偏航不同的飛行器移動方式。	8
圖 2.2.	四軸飛行器測試平台。	9
圖 2.3.	姿態控制架構圖。	10
圖 2.4.	四軸飛行器硬體架構。	10
圖 2.5.	在雜訊下不同感測器間的資料整合。	11
圖 2.6.	四軸飛行器飛行控制 (a)代表上下移動的俯瞰圖 (b)飛行器往前移動，pitch 增加 (c)順時針旋轉，yaw 增加 (d)四軸側面圖，roll 增加。	11
圖 2.7.	卡爾曼濾波器過程。	12
圖 2.8.	姿態和高度控制的 PID 控制器。	12
圖 2.9.	飛行避障控制策略。	13
圖 2.10.	完整的四軸飛行器。	14
圖 2.11.	以前後左右的障礙物距離來決定其分數。	15
圖 2.12.	避障飛行系統概念。	16
圖 2.13.	障礙物偵測的感測器佈置情形。	17
圖 2.14.	距離區域。	17
圖 2.15.	碰撞迴避模組的狀態機。	18
圖 2.16.	全系統概念。	20
圖 2.17.	障礙物偵測模組的感測與狀態感知。	20
圖 2.18.	階層式的權重濾波設計。	20

圖 2.19. 紅外線及超音波感測器的佈置。	21
圖 2.20. 光達 (LIDAR) 障礙物偵測。	21
圖 2.21. 人工潛在飛行場來引導飛行器避免碰撞。	22
圖 2.22. 將無人機分割成數個區塊 (藍色)，並計算每一區塊的力量。加諸在無人機上，並計算每一區塊的力量加諸在無人機上。	23
圖 2.23. 採用學習動態模式來預測一個側傾指令與加諸在飛行器上的力量對於偏離飛行軌跡的影響。	23
圖 2.24. 多重感測器。	24
圖 2.25. 連續影像中的 SURF 特徵點。	25
圖 2.26. 估計障礙物距離的數學模式。	25
圖 2.27. 從區域地圖產生的深度地圖。	26
圖 2.28. 動態窗口和 A*二步驟方法。	27
圖 3.1. RC 接收器接受信號後，傳送 PPM 訊號給 PPM Encoder 後將訊號截取至 Arduino 作處理。	34
圖 3.2. 遙控器接收圖。	34
圖 3.3. 串級 PID 控制器。	35
圖 3.4. 超音波感測器配置圖。	37
圖 3.5. 飛行器對應於全域地圖位置。	38
圖 3.6. 超音波測距決定障礙機率數值圖。	39
圖 3.7. 區域地圖配置機率圖。	39
圖 3.8. 飛行器對應於全域地圖位置。	41
圖 3.9. yaw angle = 30 度的全域地圖環境。	42
圖 3.10. yaw angle = -40 度的全域地圖環境。	42
圖 4.1. 方向區間柱圖法流程。	44
圖 4.2. 全域地圖中切割扇型區塊內的障礙物可能性數值圖。	46
圖 4.3. 每區塊的平滑障礙物密度示意圖。	47
圖 4.4. 一維極性機率密度圖。	47
圖 4.5. 一維極性機率密度閾值圖。	48
圖 4.6. 飛行器移動方向圖，綠色代表障礙物環境。	49
圖 4.7. 飛行器控制方式。	50

圖 4.8.	飛行器角度換算對照圖。	50
圖 5.1.	實驗障礙物偵測與避障的六軸飛行器。	52
圖 5.2.	遙控器 Futaba T10。	53
圖 5.3.	Pixhawk 飛行控制板。	53
圖 5.4.	姿態解算實驗。	54
圖 5.5.	障礙物環境與飛行器起飛前狀態。	55
圖 5.6.	飛行器偵測到障礙物環境時避障反應前。	56
圖 5.7.	飛行器避開樹後往左側避障。	56
圖 5.8.	飛行器避開電線桿後往左側避障。	56
圖 5.9.	飛行器執行完避障後安全降落。	57
圖 5.10.	Serial port 顯示姿態控制完輸出的 <i>PWM</i> 訊號。	57
圖 5.11.	飛行器起飛前綁上重物的實驗環境。	58
圖 5.12.	飛行器起飛後的穩定飛行控制。	58
圖 5.13.	飛行器起飛後的自調整平穩狀態。	59
圖 5.14.	在震動下濾波效果顯示圖。	59
圖 5.15.	在姿態解算整合角度前磁場感測器朝向數值，與姿態解算後橘色的 穩定航向數值。	60

表目錄

表 3.1. 常用姿態解算演算法比較。.....	29
表 3.2. 歐拉角與遙控器訊號對應。.....	34
表 5.1. 實驗設備簡表。.....	51
表 5.2. 障礙物偵測與避障演算法的執行時間。.....	60

第一章 緒論

在本章中，我們將說明本論文的研究動機、硬體架構圖、系統架構、與論文架構。

1.1 研究動機

無人機常被用於農業、商業、軍事、娛樂、救災、送貨、國防等領域的事務上。無人機的控制方式有利用遙控、導引或自我飛行等等的方式。無人機在執行任務時，若遇障礙物時，可能沒有良好的處理機制。因此本研究結合多重感測器進行資料整合，利用同步定位與地圖構建技術做障礙物偵測系統，與方向區間柱圖法和障礙物偵測結果來實作飛行器避障系統，並將產生的飛行訊號大小與飛行方向，來控制飛行器以達成穩定飛行的目的。

1.2 硬體架構

飛行器硬體包含使用者控制部分，一個慣性測量單元 (IMU)，11 個超音波測距感測器，一個 GPS 模組，一個 Arduino Mega 2560 微控制板和 Pixhawk 飛控板，每個硬體傳送的訊號與流程，如圖 1.1 所示。

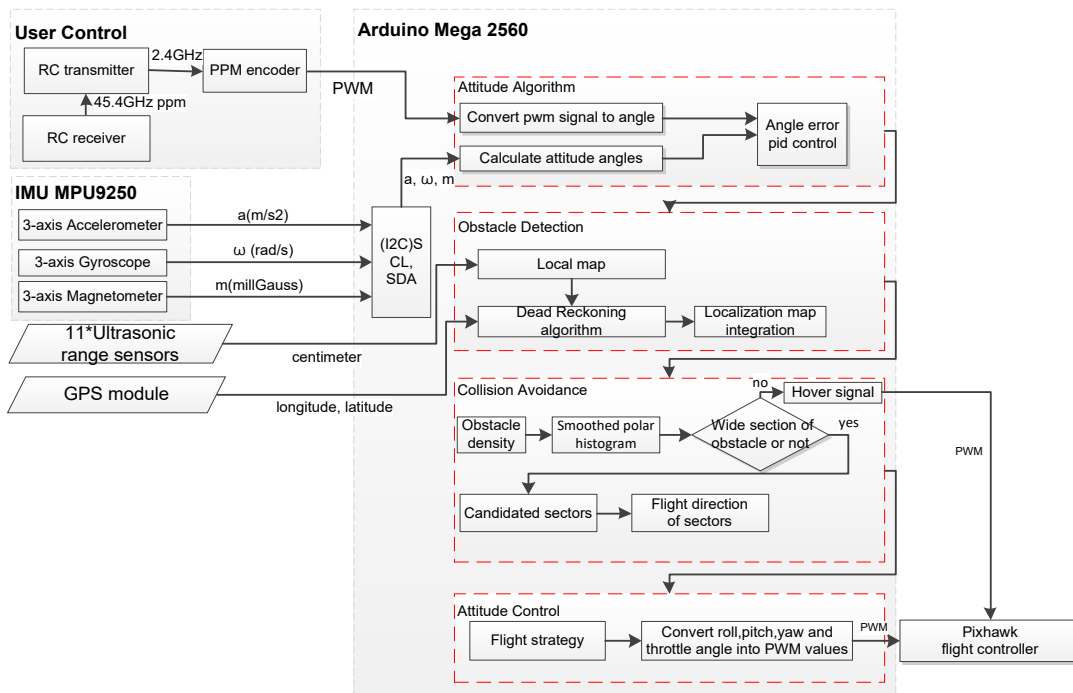


圖 1.1. 硬體架構圖。

1.3 系統架構

本系統包含姿態解算與障礙物偵測及避障系統和飛行器姿態控制四個部分，簡略流程如圖 1.2 所示；完整流程如圖 1.3 所示。

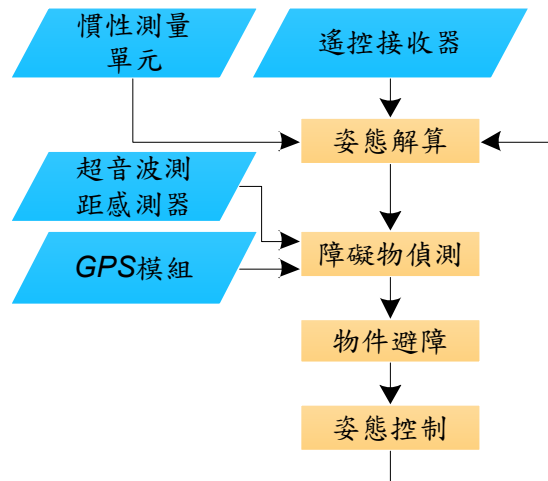


圖 1.2. 簡略流程。

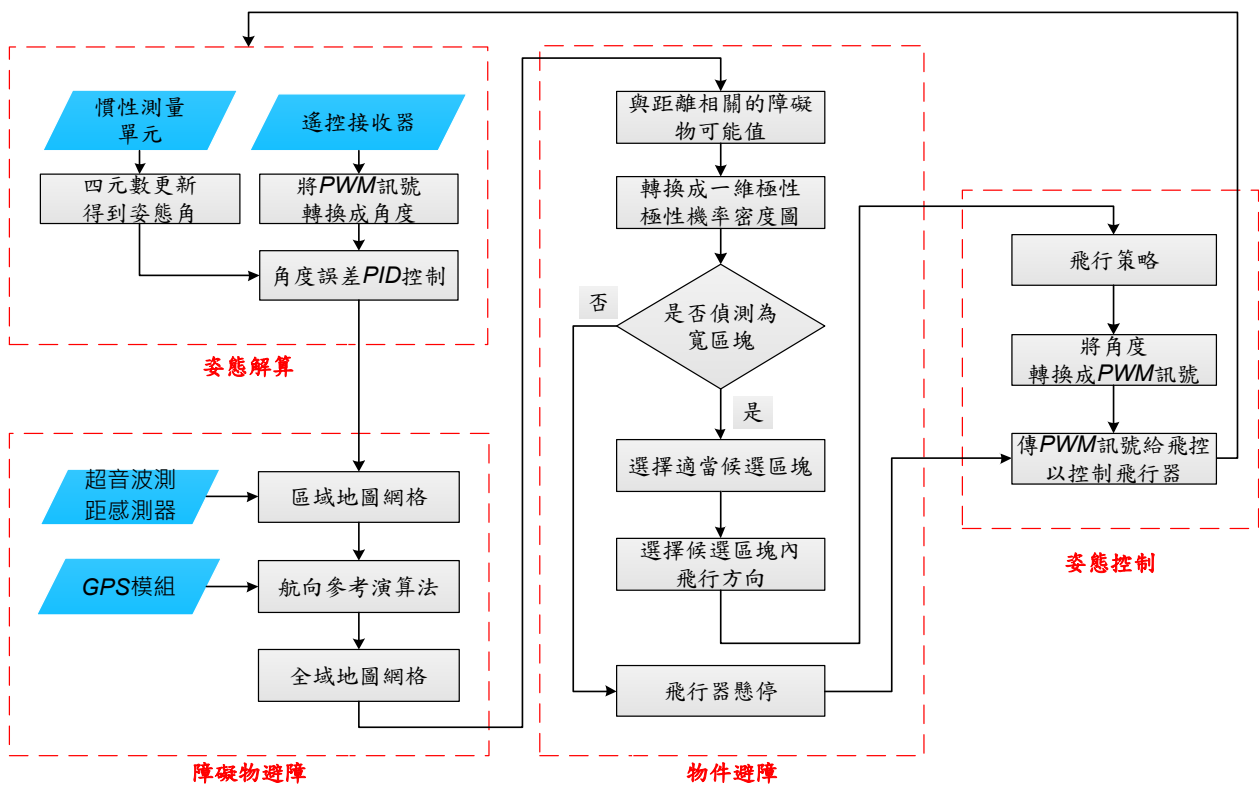


圖 1.3. 完整流程。

- i. 飛行器姿態解算流程如圖 1.4 所示。
 - ii. 障礙物偵測系統流程如圖 1.5 所示。
 - iii. 飛行器避障分析流程如圖 1.6 所示。
 - iv. 飛行器姿態控制流程如圖 1.7 所示。
- A. 飛行器姿態解算。利用數據融合方法能使九軸姿態感測模組 (9Dof) 其包含陀螺儀、加速度計與磁力計得到較可信的數值，並使數值做複雜的四元數計算 (Quaternion) 來求解姿態角，再將使用者控制的遙控接收器信號，轉化為對應的角度，最後使用 *PID* 控制系統計算角度誤差後，利用數值回授的機制，將數值能控制在較穩定的範圍內，其流程如圖 1.4 所示。

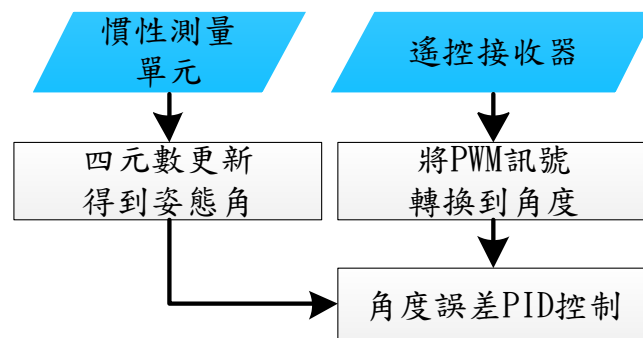


圖 1.4. 姿態解算流程。

- a. 數據融合：根據其他陀螺儀、加速度計和磁力計的數值將其做姿態解算，並利用 *PID* 控制流程，將每個數值做調整並利用四元數計算和正規化，並利用數值去做歐拉角 (Euler angle) 的計算，歐拉角即是我們要得到的飛行器姿態角，分別為俯仰角 (roll)、翻滾角 (pitch) 和偏航角 (yaw)。
- b. *PWM* 訊號轉換角度：利用飛行器遙控器的 *PWM* 訊號最小值與最大值，決定其每個信號的平均值，將其定義為俯仰角、翻滾角和偏擺角 0 度時的反應，角度以 0 度為基準 ± 90 度，以加信號為角度增加，以減信號為角度減小。
- c. *PID* 控制系統：*PID* 控制器 (比例-積分-微分控制器)，*PID* 控制器的比例單元 *P*、積分單元 *I* 和微分單元 *D* 分別對應目前誤差、過去累計誤

差及未來誤差，可以根據歷史數據和差別的出現率來調整輸入值，使系統更加準確而穩定，如圖 1.5 所示。

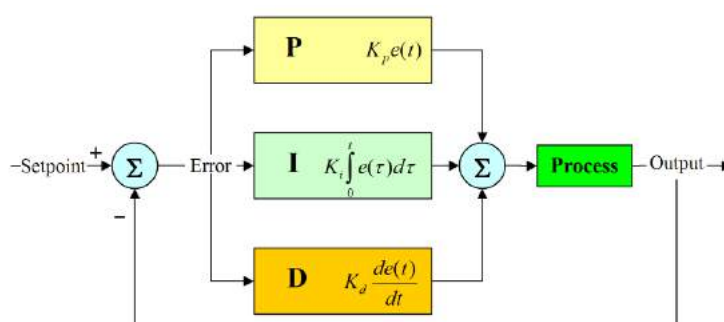


圖 1.5. *PID* 控制系統。

- B. 障礙物偵測系統。利用同步定位與地圖構建方法 (Simultaneous localization and mapping, *SLAM*) 將超音波感測器偵測到的數值來做區域地圖網格構建，並利用航向參考演算法找出飛行器的移動方向與位置，最後利用全域地圖網格構建將飛行器定位，其流程如圖 1.6 所示。

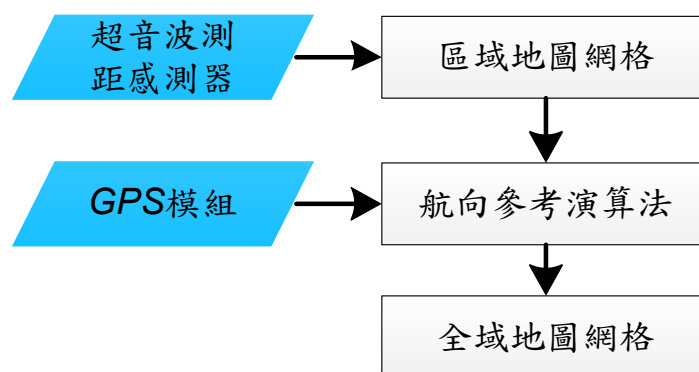


圖 1.6. 障礙物偵測流程。

- a. 區域地圖網格：本研究所定義的區域地圖網格為 8×10 的大小，利用 11 個超音波感測器偵測的距離數值，將其依照每個網格對應現實距離 $80\text{cm} \times 80\text{cm}$ 的大小，並依據角度範圍內若有偵測到物體會將地圖網格內覆蓋區域填上障礙物可能存在的機率數值。

- b. 航向參考演算法：利用區域地圖為相對座標，全域地圖為絕對座標，根據全球定位系統 (global positioning system, GPS) 模組獲取的經緯度資訊，計算飛行器移動時的距離，再依據偏航角獲得飛行器的航向。
- c. 全域地圖網格：本論文所定義的全域地圖網格為 24×24 的大小，每個網格對應現實距離為 $80\text{cm} \times 80\text{cm}$ 的大小，將飛行器移動時所偵測的區域地圖更新到全域地圖裡。

C. 飛行器避障分析。本研究避障方法採用方向區間柱圖法 (vector field histogram, VFH)，將全域地圖做每 15 度的扇形區塊劃分，一共 24 個扇形區塊，將每區塊涵蓋的網格裡的障礙物機率數值與到飛行器中心點的距離做運算後，得到每個格子與距離相關的障礙物可能值，將區塊包含的格子算出的與距離相關的障礙物可能值做加總，再取前後各五個數值做平均得到一維極性機率直方圖，並判斷是否有連續的區塊小於門檻值，若有即稱為寬區塊並依據與障礙物的相對關係找出候選區塊，再把上一時刻的飛行方向考慮進來得到最後要控制飛行器的方向，避障分析流程如圖 1.7 所示。

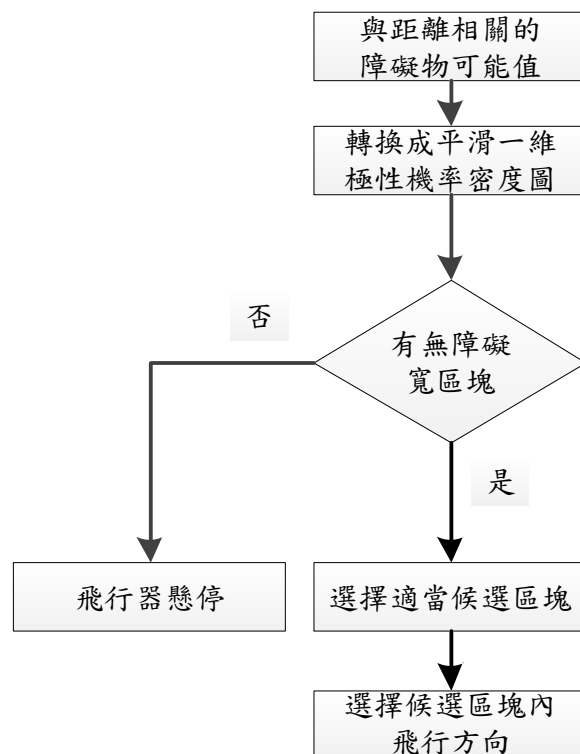


圖 1.7. 避障分析流程。

- a. 與距離相關的障礙物可能值：利用全域地圖網格裡的數值為障礙物機率值與每個網格對應到飛行器中心的距離計算出與距離相關的障礙物可能值。
- b. 平滑一維機率密度圖：將全域地圖轉換成飛行器瞬間位置的一維極性機率密度圖，將全域地圖以每隔 15 度做一個區塊的劃分，共劃分成 24 個區塊，將區塊內涵蓋的網格做與距離相關的障礙物可能值加總，再取前後各五個數值做平均得到平滑化的結果，並產生出一維平滑機率密度圖。
- c. 選擇候選區塊：根據平滑的一維機率密度圖數值，若是有超過五個低於門檻值的區塊即定義為寬區塊。
- d. 選擇候選區塊內飛行方向：以上一時刻的飛行方向為基準，與寬區塊裡最靠近上一時刻的飛行方向的扇型區域且低於門檻值的方向做結合，決定飛行器最後的飛行方向。

D. 飛行器姿態控制。依照飛行避障演算法得出的飛行方向後，控制飛行器的翻滾角與偏航角，再將角度換算回 PWM 訊號，飛行器姿態控制流程，如圖 1.8 所示。

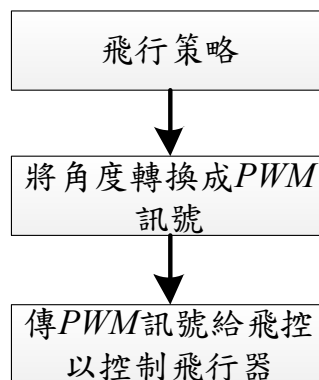


圖 1.8. 飛行器姿態控制流程。

- a. 飛行策略：將避障演算法得到的飛行器方向換算成控制飛行器動作的訊號，用滾動角與偏航角做相對應的姿態控制使飛行過程是趨向自然

的動作。

- b. 將角度轉換成 *PWM* 訊號：將飛行策略決定後的角度訊號，轉化為俯仰角、滾動角、偏航角的相對應角度，再將所有的角度換算為飛行器遙控器最小值與最大值的相對應轉換。

1.4 論文架構

本論文共有六章，第二章介紹多軸飛行器的姿態解算與姿態控制的相關研究，和障礙物偵測與避障系統的相關研究，。第三章介紹本論文使用的飛行器姿態解算方法與障礙物偵測演算法，主要使用 *SLAM* 方法建立障礙物偵測環境，將九軸姿態模組計算出飛行器姿態，和超音波感測器與 *GPS* 模組，第四章介紹我們使用的 *VFH* 方法做飛行器的避障分析，並將分析結果產生飛行器相關姿態控制，第五章是實驗設備介紹與結果和分析，最後，第六章是結論及未來展望。

第二章 相關研究

在第二章相關研究中，我們將探討飛行器求解姿態的方法與控制的飛行策略，與障礙物偵測演算法與避障方法的相關研究。

2.1 飛行器姿態解算與姿態控制

飛行器主要的姿態解算與控制為飛行控制版的核心，其感測器的彼此整合運算與去除雜訊更是許多研究學者討論的議題，簡易的姿態解算過程是基於慣性測量單元 (Inertial measurement unit, *IMU*) 系統，一般的 *IMU* 內會裝有三軸的陀螺儀和三軸加速度計，僅提供物體在三維空間中的角速度和加速度，並不具有提供準確可靠的物體姿態數據功能，而現今飛行器大多採姿態航向參考系統 (Attitude and heading reference system, *AHRS*) 系統，包含了嵌入式的姿態數據解算單元與提供翻滾和俯仰和航向訊息，這類系統會為飛行器提供準確可靠的姿態與航行訊息，比 *IMU* 系統多了磁力計數據和 *GPS* 數據可進行姿態解算，其技術為採用多傳感器數據融合技術與卡爾曼濾波器來進行姿態估計。

Hoffmann *et al.* [1] 呈現一個完整的四軸無人機機電設計流程，包含機械設計與模組化四軸和制動器動力學和姿態穩定控制，如圖 2.1 所示。

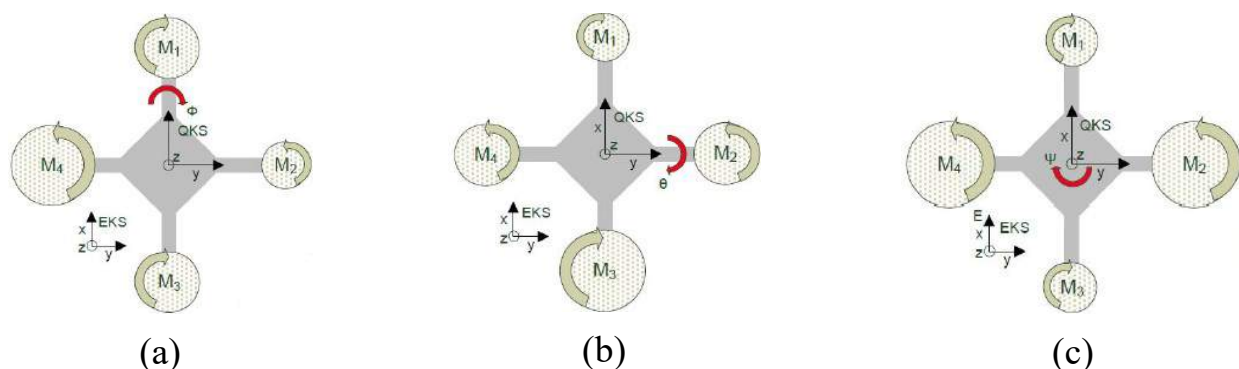


圖 2.1. 呈現四軸飛行器的機體座標系和外部座標，螺旋槳的配置與馬達轉速可組合出 (a) 翻滾 (roll) 動作 (b) 俯仰 (pitch) 動作 (c) 偏航 (yaw) 動作不同的飛行器移動方式。

機電設計包含感測器與制動器的選擇，以及如何安裝在機身內和信號處理與姿態穩定，四軸動力學衍生自 euler-lagrange 形式或牛頓 euler 方程式，這些行程為此論文模型的基礎。除此之外，完整機械模型捕捉的轉子 (rotor) 與馬達的運動學和感測器使真實的模擬狀態估計和避迴路行為。藉由客制設計低成本的慣性測量單元其有陀螺儀和三軸加速度計來觀察無人機運動，原始感測器信號會被濾波與利用卡爾曼濾波來融合數值以估計姿態和位置，而姿態穩定控制器是基於傳統的 *PID* 控制器和一個積分反推控制器設計而成。

穩定的姿態估計可以由低成本的微機電系統 (microelectromechanical systems, *MEMS*) 加速度計和陀螺儀信號配合卡爾曼濾波的數據融合達成，做了穩定器 (gimbal) 遷入四軸飛行器實驗，如圖 2.2 所示。

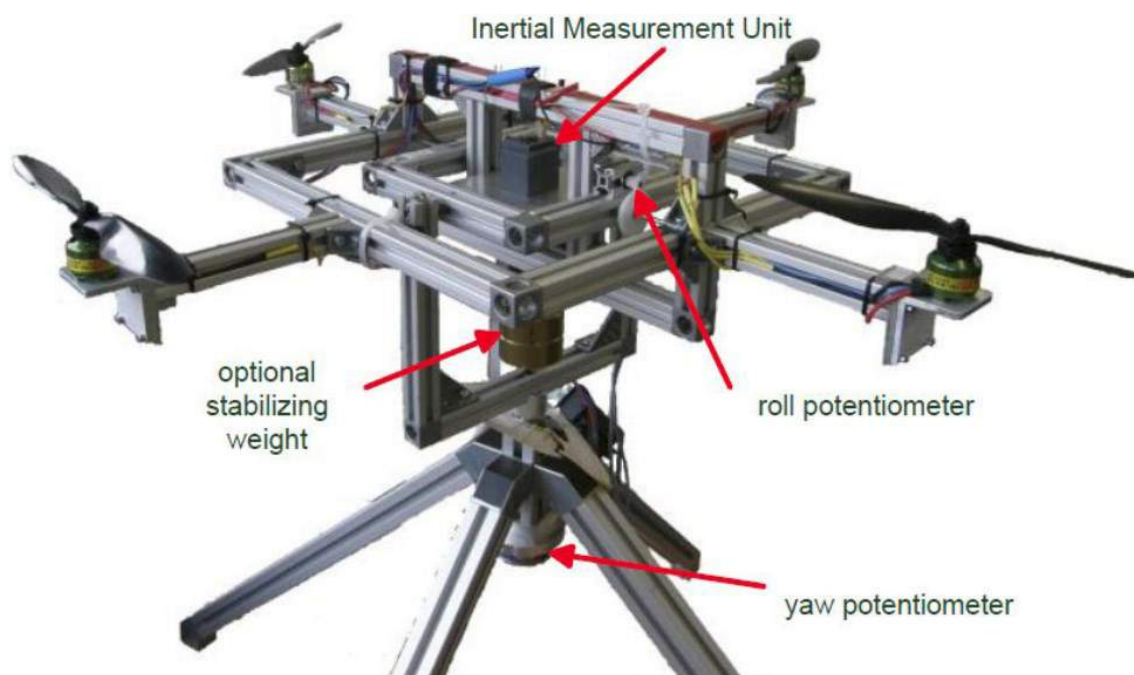


圖 2.2. 四軸飛行器測試平台。

利用 gimbal 以測試在姿態穩定上參考相對信號追蹤和干擾去除與穩定性的量化分析和比較 *PID* 和積分反推 (integral backstepping) 控制器設計，整體控制架構由前饋高度補償器和反饋姿態和高度控制器組成，如圖 2.3 所示。

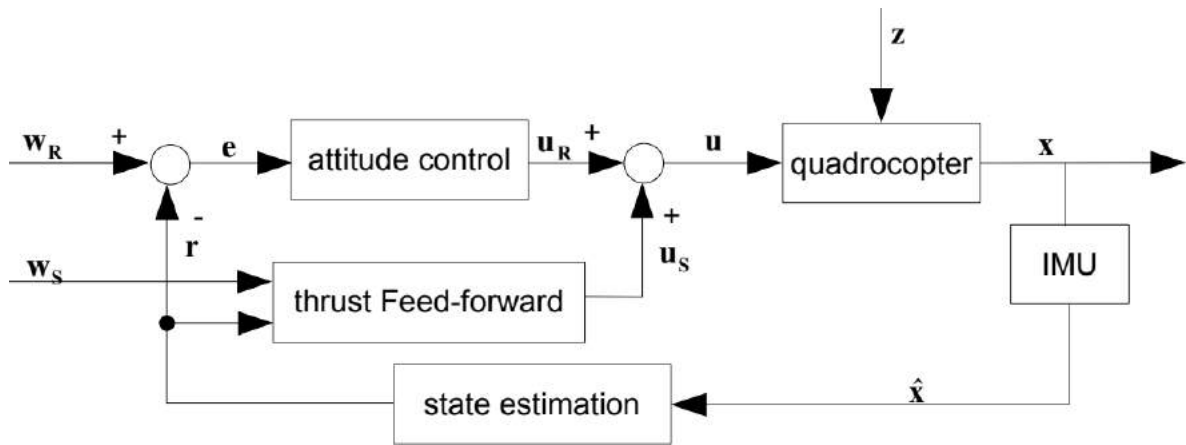


圖 2.3. 姿態控制架構圖。

Chan *et al.* [2] 提出可在低成本下利用輕量的 *MEMS* 感測器作為姿態估計的方案，低成本的硬體架構如圖 2.4 所示。四軸飛行器搭載姿態航向參考系統 (*AHRS*)，系統可整合不同的感測器資料作姿態穩定化使其減少誤差，論文利用 16 位元的嵌入式微控制器和資料整合方法，減少陀螺儀誤差和測量時震動與雜訊帶來的影響，如圖 2.5 所示。

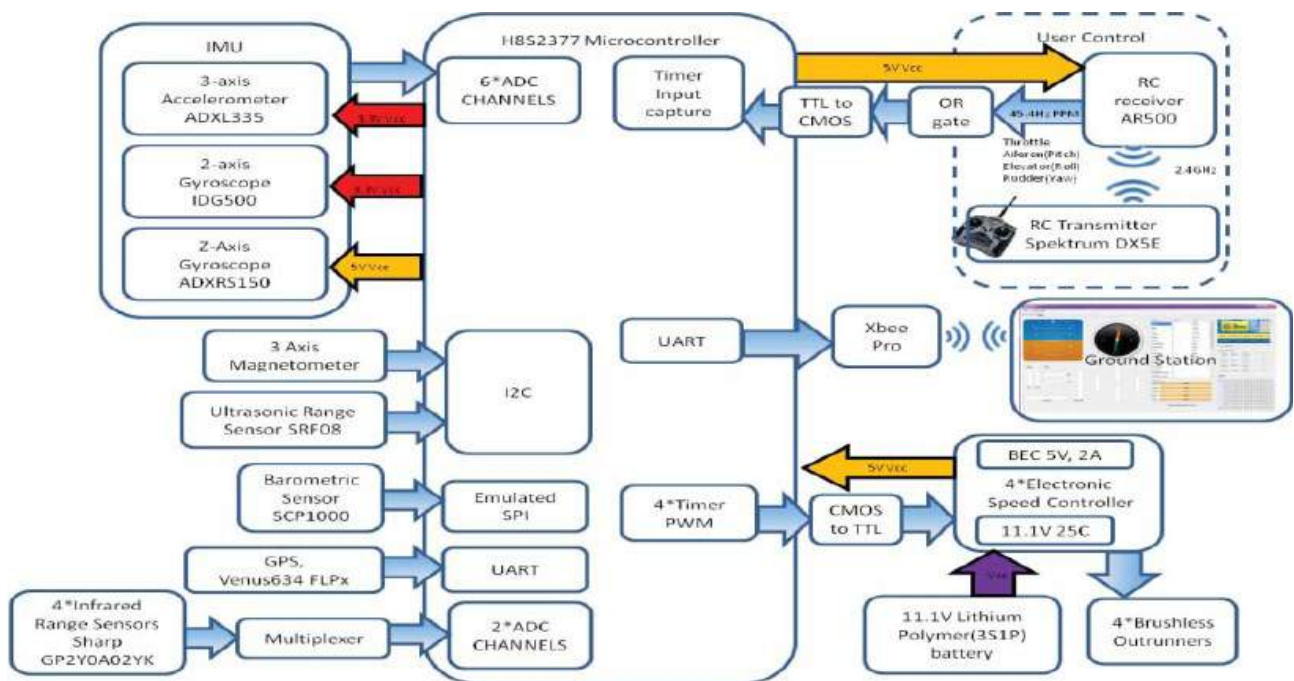


圖 2.4. 四軸飛行器硬體架構。

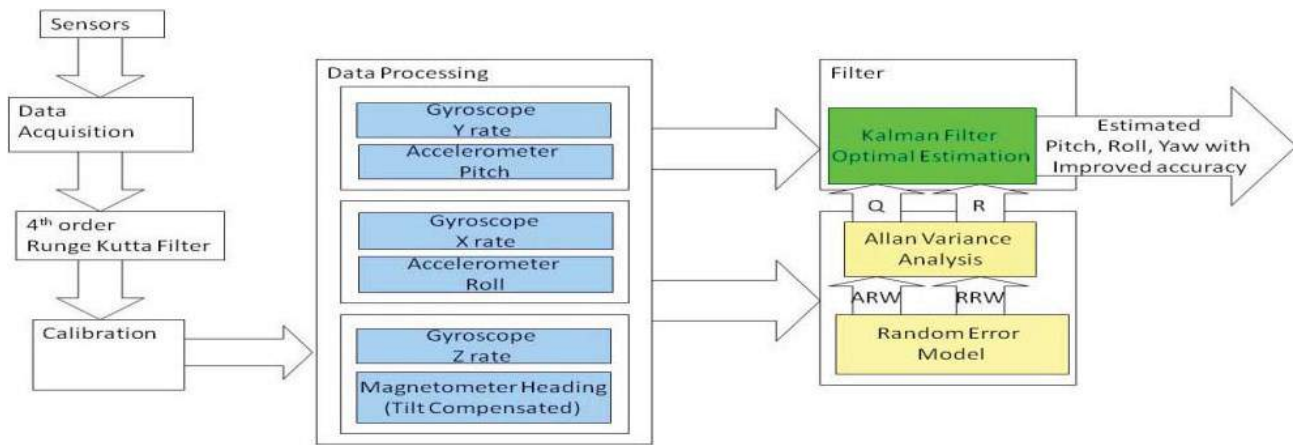


圖 2.5. 在雜訊下不同感測器間的資料整合。

四軸飛行器可穩定盤旋在室內並可即時反應干擾物，並即時準確的飛行控制，如圖 2.6 所示。姿態穩定技術是實作卡爾曼濾波器，如圖 2.7 所示，並結合控制器，如圖 2.8 所示。

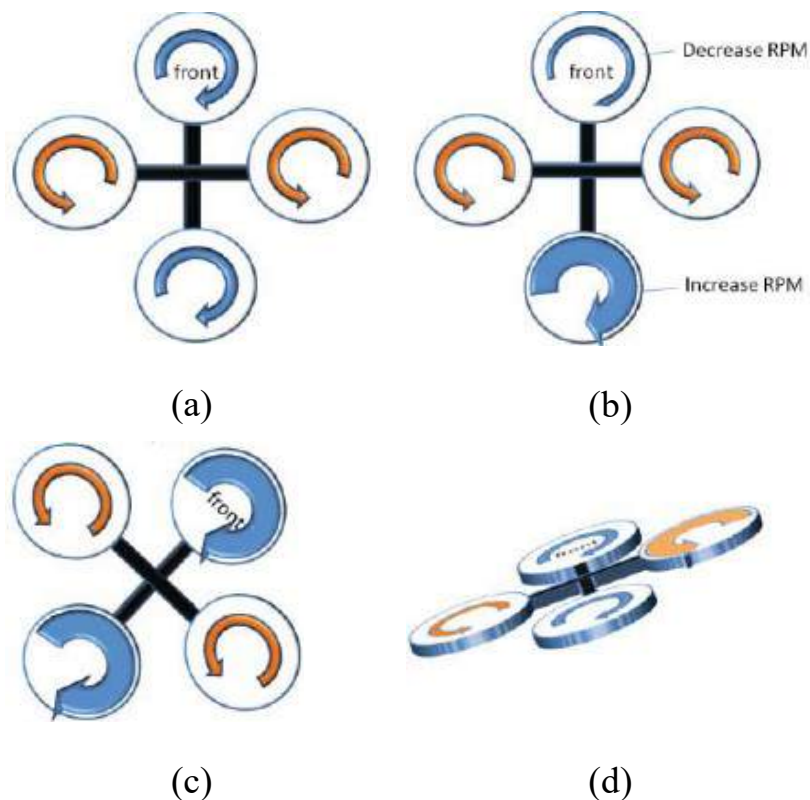


圖 2.6. 四軸飛行器的飛行控制。(a) 代表上下移動的俯瞰圖。(b) 往前移動，pitch 增加。(c) 順時針旋轉，yaw 增加。(d) 四軸的側面圖，roll 增加。

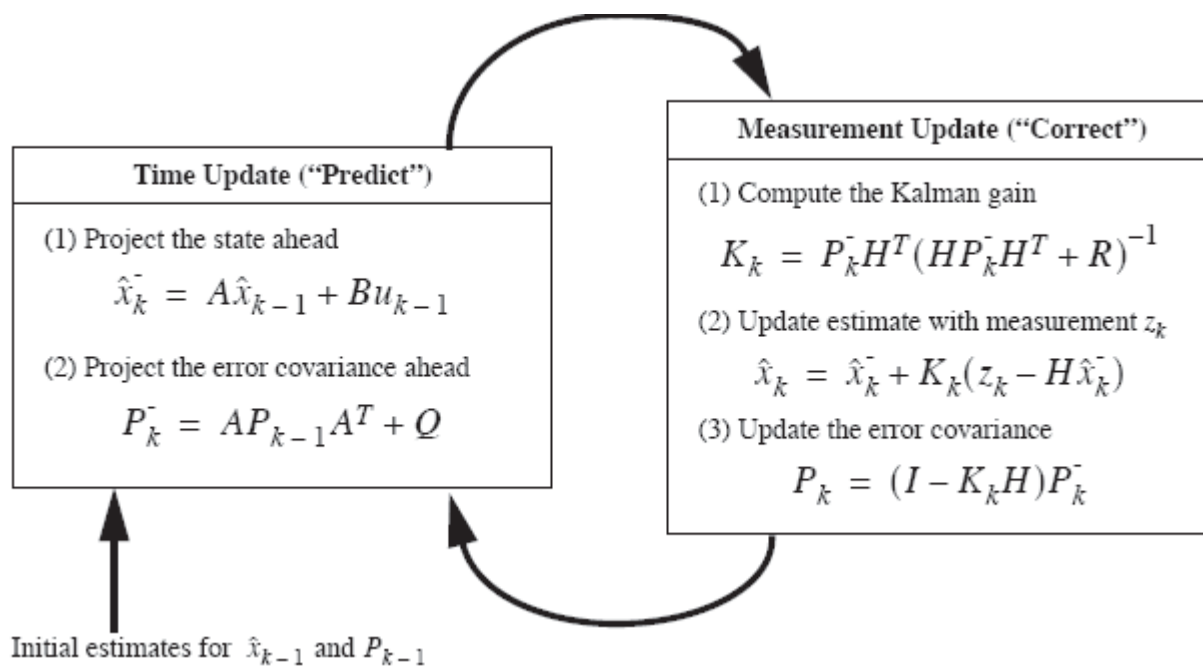


圖 2.7. 卡爾曼濾波器過程。

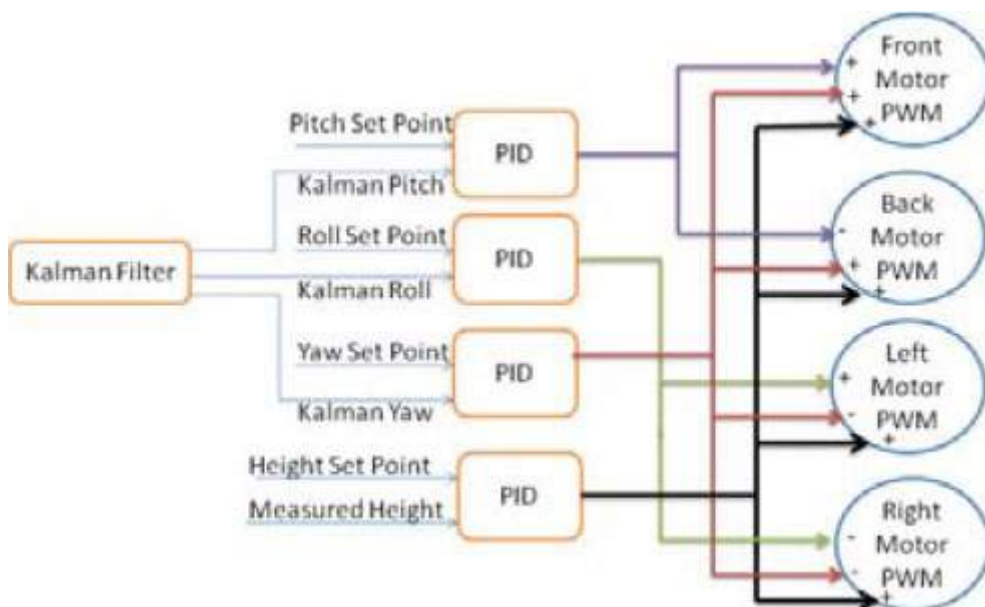


圖 2.8. 姿態和高度控制的 *PID* 控制器。

2.2 障礙物偵測與避障

避障是無人機 (UAV) 在室內導航所面臨最具挑戰性的問題，Luo *et al.* [3] 發展出一個根據掃描雷射測距儀 (scanning laser range finder, SLRF) 的室內避障策略 (strategy)，可即時產生高階飛行指令與飛行路徑供無人多軸飛行器飛行控制使用。這個避障策略包括三個飛行模式，包括前飛 (forward flight)、闢道探索 (gate exploration)、與懸停轉彎 (hovering turn) 模式，如圖 2.9 所示。

在飛行過程中，無人機遇到障礙物時，能根據 SLRF 的測量結果，從一個飛行模式切換到另一模式。在實驗中模擬與實際範例都被提出來驗證這個研究策略的有效性。測試結果證實這個策略方法的可行性及有效性。

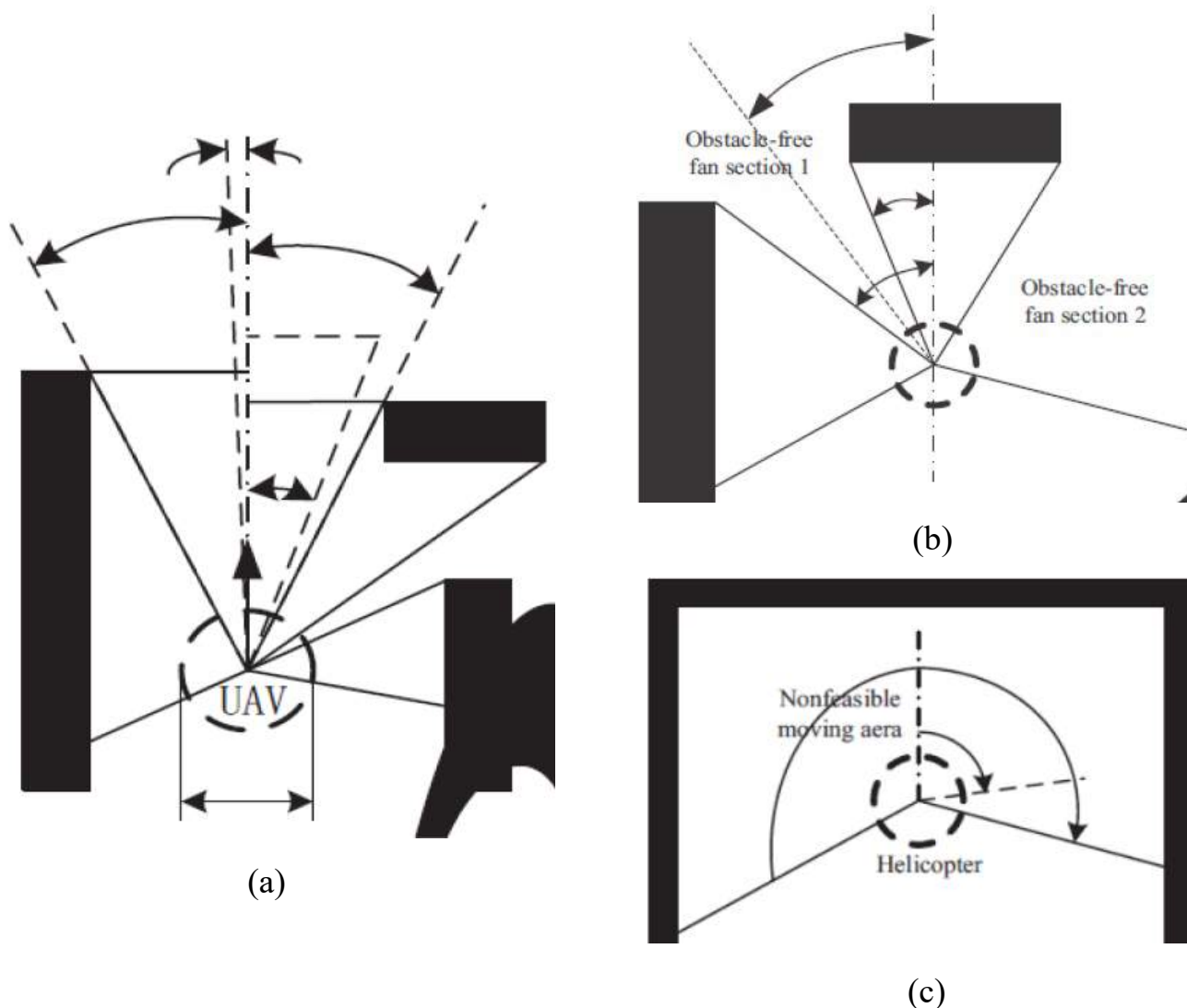


圖 2.9. 飛行避障控制策略。(a)前飛模式 (b)闢道探索模式 (c)懸停轉彎模式。

Wagster *et al.* [4] 報告在加州卡爾波利波莫納 (Cal Poly Pomona) 有關於大學生開發無人駕駛飛行器 (UAV) 之避障能力的結果。他們採用一個四軸飛行器做為研究平台；並使用快速隨機探索樹 (Rapidly exploring random trees, *RRT*) 演算法於避障的能力 (capability) 評估。展示使用較便宜的現有商用感測器與處理器設計和實現飛行器於避障演算法的結果。

避障感測器是六個安裝於訂製支架上的聲納感測器，可讓四軸飛行器能夠確定上下前後左右的障礙物距離，如圖 2.10 所示。

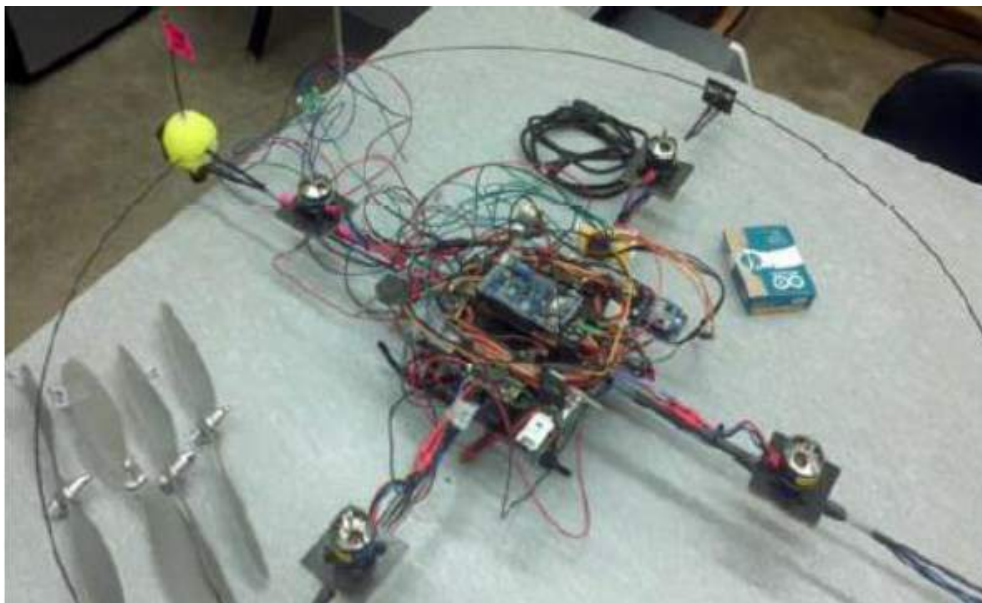


圖 2.10 . 完整的四軸飛行器。

整個系統的避障與定位 (mapping) 計算細分為三個層次，以減小碰撞風險，並確保擷取的避障資訊是有效的。

第一層次的避障是採用前方感測器來確定飛行器前方是否有障礙物，並確認障礙物的距離。設定一個與前方障礙物的最小碰撞距離，若飛行器與障礙物的距離小於該最小碰撞距離，則啟動一個俯仰角 (pitch angle)，使得飛行器減速並遠離障礙物，返回到一個安全的距離。

第二層次的避障是使用左右兩側的感測器，讓飛行器使用推進演算法 (push algorithm) 以維持在兩個障礙物中間。該演算法使用左右距離資料以控制飛行器的左右傾斜角 (roll angle)，避免飛行器靠近障礙物。

第三層次的避障是使用快速隨機探索樹演算法，它不僅可迴避障礙物，而且還可規劃在其周圍的飛行路徑。在實作階段，四軸飛行器依據可移動路徑的點數來決定其飛行路徑。四軸飛行器被限制小幅度的前後左右移動量，如圖 2.11 所示。

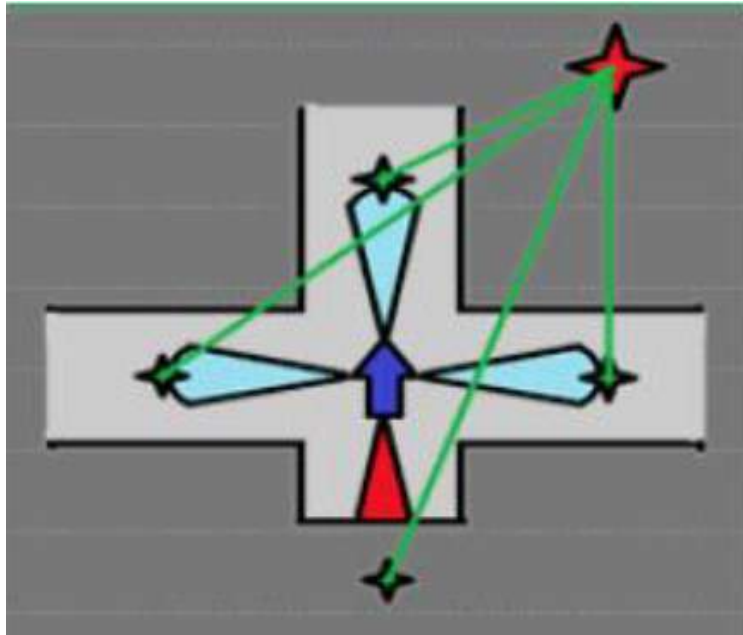


圖 2.11. 以前後左右的障礙物距離來決定其分數。

Gageik *et al.* [5] 開發低成本的超音波感測器 (ultrasonic sensors) 及簡單的資料融合做四軸飛行器的障礙物偵測與飛行避障研究。

戶外的自主飛行無人機可用 *GPS* 做定位及導航，但在室內則必須採用其他設備或方法；例如，可使用相機系統做障礙物偵測、防撞、及定位。但缺點是需要繁複的運算。另外也可用 3D 相機系統；例如，微軟的 Kinect 相機或其他雷射掃描儀。但是任何光學感測器對於光線變化或透明物都不太穩定。因此煙、蒸汽、及氣體會讓光學偵測器偵測失敗。相反的，超音波感測器不會受上述狀況的影響。但是只使用超音波感測器可能仍不足以支持全自主 (fully autonomous) 的飛行。

四軸飛行器的障礙物偵測與飛行避障研究的方法分成兩個主要模組：一個用於障礙物偵測，另一個用於防撞。兩個模組是獨立做出來的。因此，對於更改或添加不同類型的感測器，這個概念仍然有效及適用。

圖 2.12 是避障飛行系統的整體概念。經過濾的超音波數據與慣性測量單元 (inertial measurement unit, *IMU*) 的資料融合 (data fusion)，做為障礙物偵測模組依據。而碰撞迴避模組則以障礙物偵測的結果為依據以控制飛行器飛行，從電腦或遙控器 (radio control) 傳輸用啟用及停用模組功能的運行如同傳送命令控制飛行器飛行一樣。

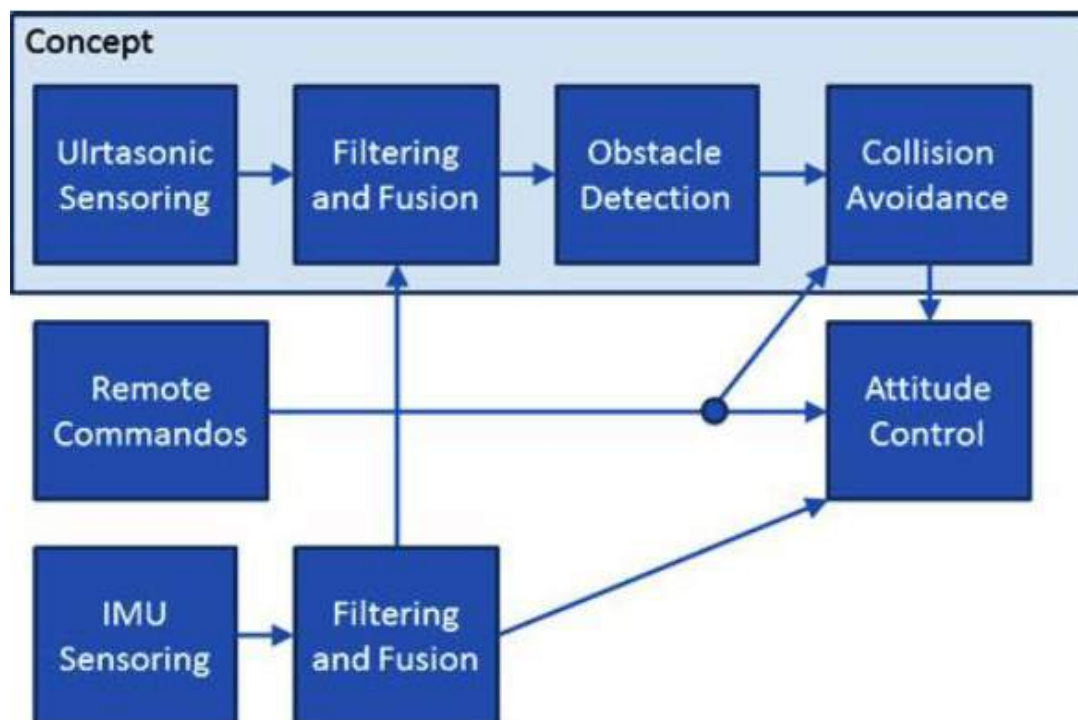


圖 2.12. 避障飛行系統概念。

在障礙物偵測模組中，多個超音波感測器可用於增加偵測的解析度及感測數據的可靠性。由於超音波感測器具有寬廣的偵測角度，但偵測的解析度很低；因此相鄰超音波的感測範圍有部份重疊，則使用感測器的數量增加，偵測解析度也可以提高。圖 2.13 顯示使用 12 超音波感測器的 360 度偵測架構；其中虛線顯示感測器 S3 與感測器 S5 的重疊情形。

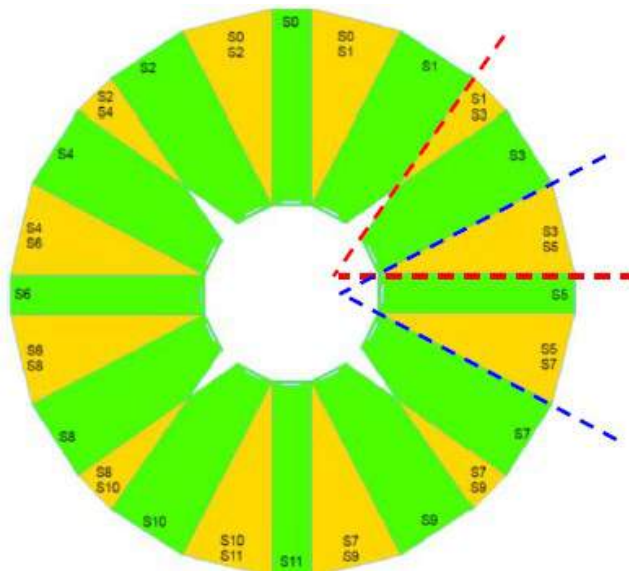


圖 2.13. 障礙物偵測的感測器佈置情形。

碰撞迴避模組根據與四軸飛行器的距離，將四軸飛行器周邊的區域分成為三個子區域：遠或安全區(綠色)、靠近區域(黃色)、及危險區域(紅)，如圖 2.14 所示。

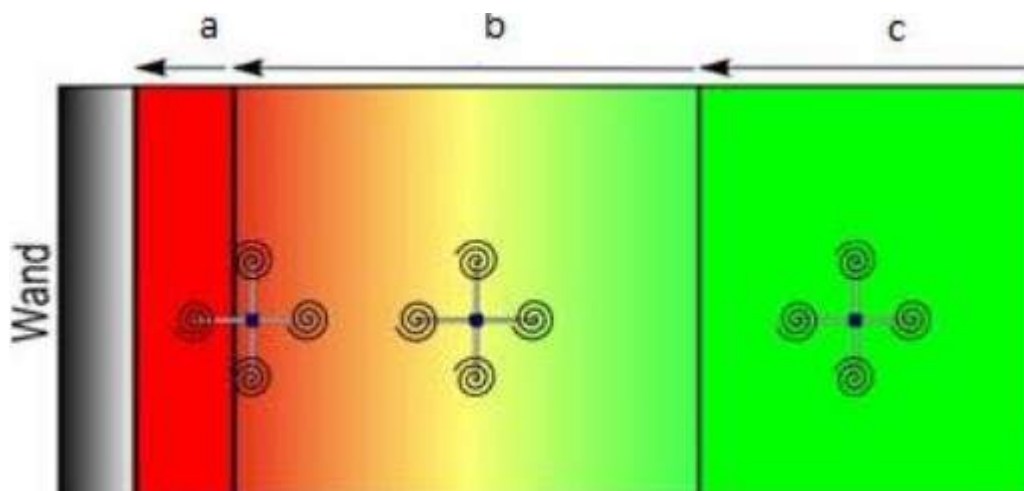


圖 2.14. 距離區域(distance zones)。

碰撞迴避模組的控制模式可以用狀態機來描述，如圖 2.15 所示。一開始，如果主動防撞功能是關閉的，則四軸飛行器是處於狀態 0。如果主動模式啟動了，則四軸飛行器可根據與障礙物的距離，改變 1, 2, 或 3 狀態。如果附近沒有障礙

物 (物距 $> a+b$)，則啟動狀態 1 (安全區)。如果有障礙物，在靠近區域 ($a >$ 物距 $> a+b$)，則啟動狀態 2 (靠近區域)，並且啟動前後俯仰或左右側傾動作，以降低飛行器的速度。在危險區域時 ($a <$ 物距)，則啟動狀態 3，而且飛行器與障礙物的距離是改由 PID 控制器，避免飛行器進一步接近障礙物。這樣的狀態機控制機制在每個方向都是必要的。

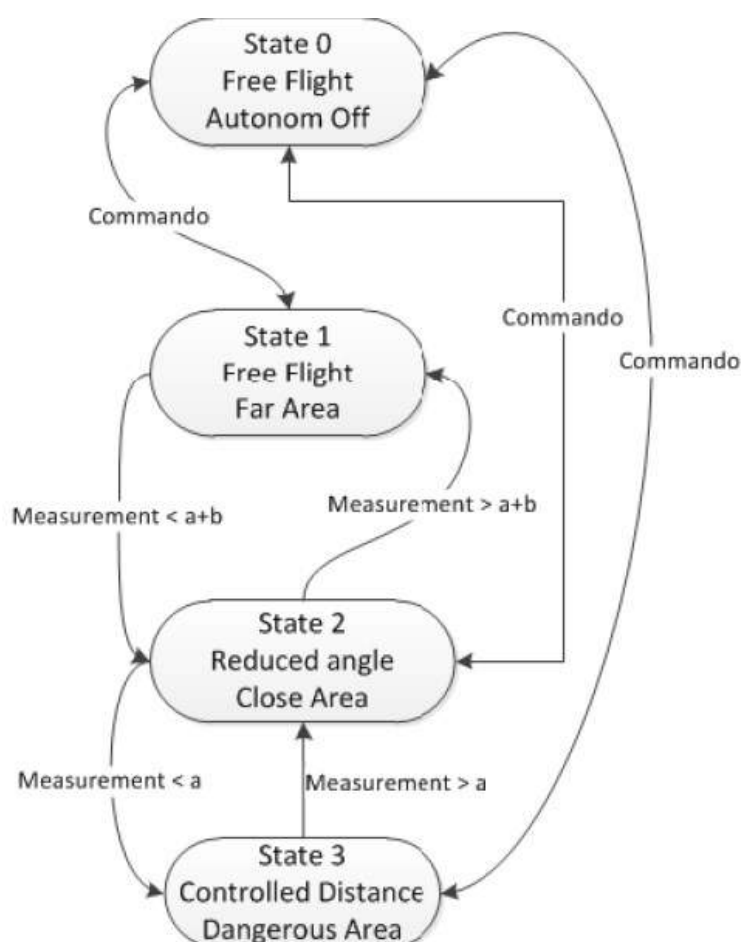


圖 2.15. 碰撞迴避模組的狀態機。

Gageik *et al.* [6] 提出一個創新及簡單的四軸飛行器之障礙物偵測與避免碰撞的解決方案。障礙物偵測所使用的感測器都是低價格的超音波與紅外線測距儀；這些感測器的價格比昂貴的感測器像是雷射掃描儀便宜許多，當然相對也有較多的雜訊。對於這樣的系統，需要考量設計、實作、及信號處理與控制演算法的參數最佳化的問題，這也就是本論文的主題。為了改進資料融合的效果，慣性

與光學感測器被用來獲得對比用的距離資料。其結果顯示，一架無人機能夠有效的控制距離以避免碰撞，但是這個系統比簡單的解決方案更複雜與強大。雖然如此，本解決方案仍然是簡單的，且低計算負擔。因此，避免碰撞不需要記憶體和耗時的同步定位與地圖標記等功能。

一般避障的方法可分為兩類：第一類包含以控制反向飛行的方式避免碰撞。這種方式的重大缺點是牽涉到改變無人機該次任務的路徑。第二類方法可以被描述為以 *SLAM* 為基礎的解決方案。這類方法是以地圖、定位、及導航來避免碰撞。在地圖內，以複雜的 *SLAM* 演算法來定位 (positioning) 與對應 (mapping)。與第一類方法比較，第二類方法並無任務上的限制，但防撞要求 *SLAM* 演算法，這個演算法需要大量的記憶體與計算能力。本研究所提的方法可以分類為介於上兩類之間，避免這些方法的缺點。

除了上面所提的偵測設備外，超音波與紅外感測器可被用來當作互補 (complementary) 的偵測設備。紅外線感測器就像所有光學感測器一樣，在光線不良的環境下；例如，煙或霧環境，容易失敗；也不能偵測透明的障礙物。相反的，超音波感測器沒有這種缺點。但是超音波感測器無法偵測有吸音表面的物體；例如，衣服或窗簾材質。因此超音波感測器不適合用來偵測人體；紅外線感測器沒有這個問題。

與現有的相關飛行器比較，這些飛行器沒有使用冗餘的感測器來偵測障礙物；本研究所提的方法結合了兩種互補的技術，且具有可接受的價格與可靠性。這樣的組合非常適合搜索與救援 (rescue, SAR) 任務的應用，其中無人機可用於地下車庫發生火災後，於煙霧瀰漫的環境中尋找受傷人員。

四軸飛行器及低空間飛行優化的成果仍可應用於其他類型的無人機。此飛行器研究成果是 AQopterI8 計畫的一部份，這一計畫是要發展室內自主飛行四軸飛行器的應用；例如，協助消防隊員於火災搜索與救援任務的應用。

圖 2.16 是從一個高階的觀點描述這一系統分割成障礙物偵測 (OD) 與避免碰撞 (CA) 兩個模組的概念。障礙物偵測模組包括感測 (sensing) 與狀態感知 (situation awareness, SA)，如圖 2.17 所示。感測是讀取、調節、與過濾紅外線及超音波的偵測資料、光流、與慣性測量單元 (IMU) 的感測資料，計算出距離 d ，

如圖 2.18 所示。紅外線及超音波感測器的佈置如圖 2.19 所示。

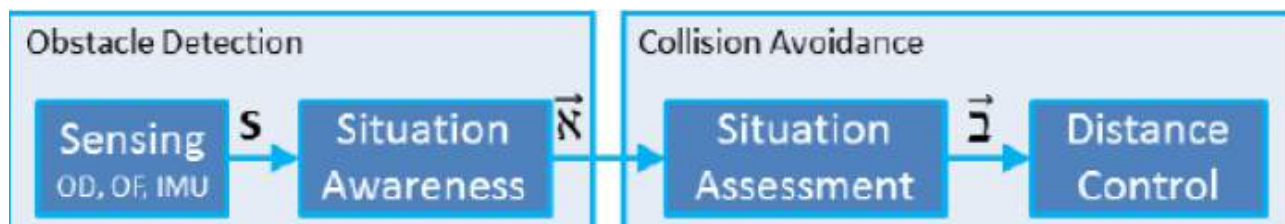


圖 2.16. 全系統概念。

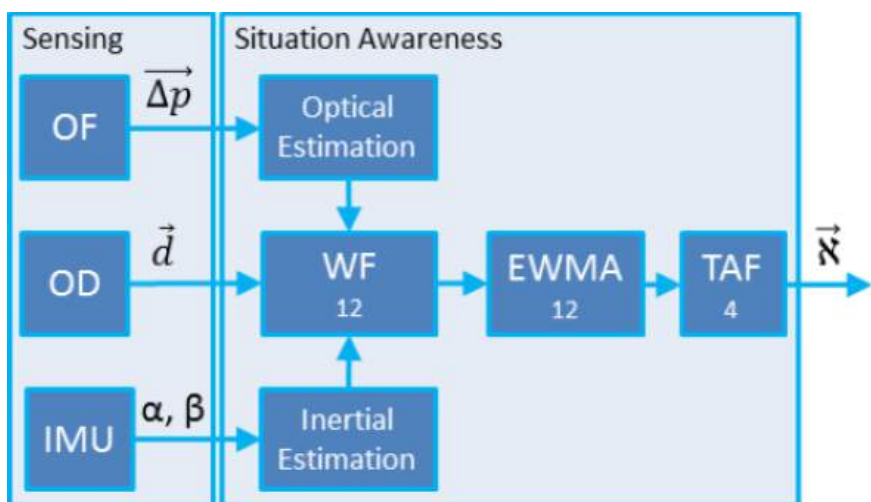


圖 2.17. 障礙物偵測模組的感測與狀態感知。

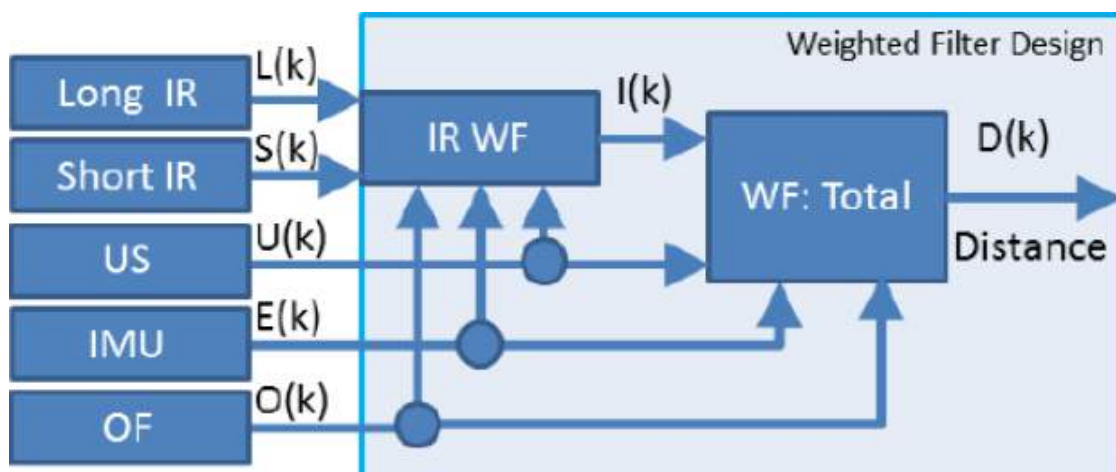


圖 2.18. 階層式的權重濾波設計。

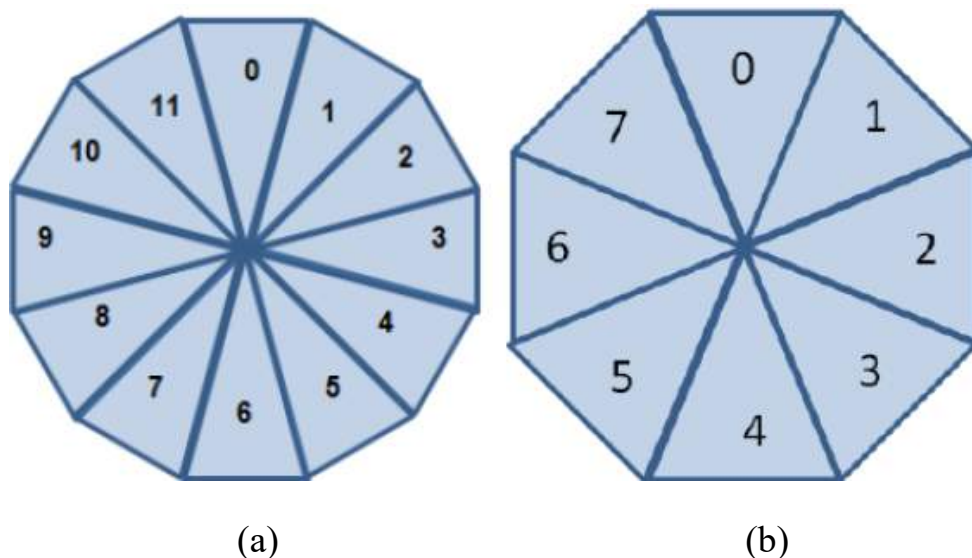


圖 2.19. 紅外線及超音波感測器的佈置。共使用了 12 個超音波測距儀及 8 個紅外線測距儀圍繞著飛行器。(a) 超音波測距儀的佈置。(b) 紅外線測距儀的佈置。

Ferrick *et al.* [7] 為有限計算能力的低價格消費性四軸飛行器發展新的演算法。使用雷射測距儀 (*LIDAR*) 配合 *OpenCV* 影像處理函式庫的計算，在簡單規劃的網格地圖上，執行避障飛行。這些演算法會產生沿著牆壁尋找沒有障礙物的自由路徑，如圖 2.20 所示。這樣的飛行系統可讓未經訓練和不熟練的使用者，在室內環境中遙控一個低成本的飛行平台；而這個平台可用市售的零件來組裝而成。

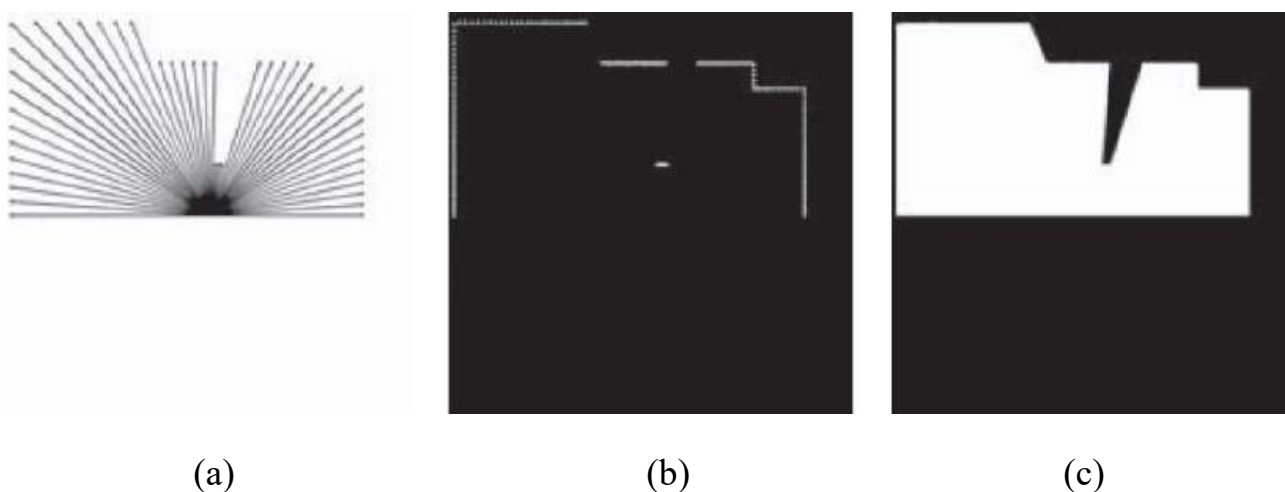


圖 2.20. 光達 (*LIDAR*) 障礙物偵測。(a) 光達 (*LIDAR*) 原始資料。(b) 偵測到的障礙面。(c) 分析出無障礙的空間。

在有靜態與動態障礙物時，有可靠的避障功能是無人機導航的關鍵。地面上的移動機器人通常都配備 2D 或 3D 的雷射測距儀，可偵測一個二維的工作空間，且具備充分的精確度與高效率特性。微型無人機運行在 3D 環境中，由於載重的限制，無法使用最先進的快速 3D 感測器。因此微型無人機只能夠偵測無人機附近的小型障礙物；距離近，因此障礙物偵測要能夠快速，以避免碰撞。

Nieuwenhuisen *et al.* [8, 9] 根據人工潛在飛行場 (artificial potential fields, APF) 如圖 2.21 所示，作者提出避免碰撞系統，系統可以讓無人機採用學習動態模式(learned motion model) 來預測未來飛行軌跡中之障礙物的影響。

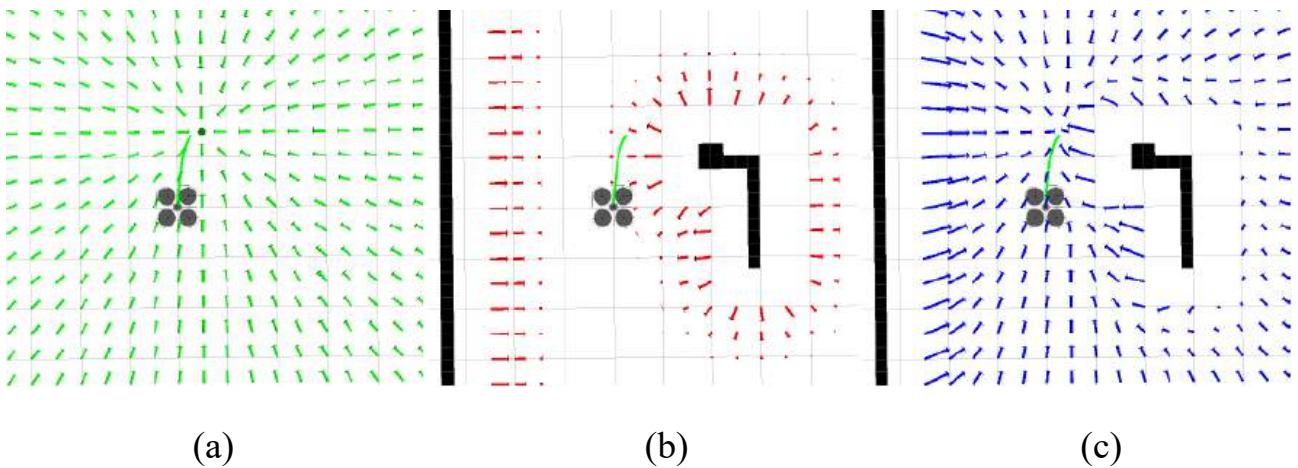


圖 2.21. 綠色粗線是飛行器的軌跡，該會受到加權總吸引力（綠色線場）及反向力（紅色線場）的影響。最後引申出的影響。最後引申出人工潛在飛行場來引導飛行器避免碰撞。(a) 加權總吸引力場。(b) 反向力場。(c) 人工潛在飛行場。

人工潛在飛行場是由吸引力場及反向力場 (repulsive forces) 所構成的；其中吸引力場是由飛行目標點創作出來的，而反向力場是由飛行器與障礙物之間的距離計算出來的，如圖 2.22 及圖 2.23 所示。

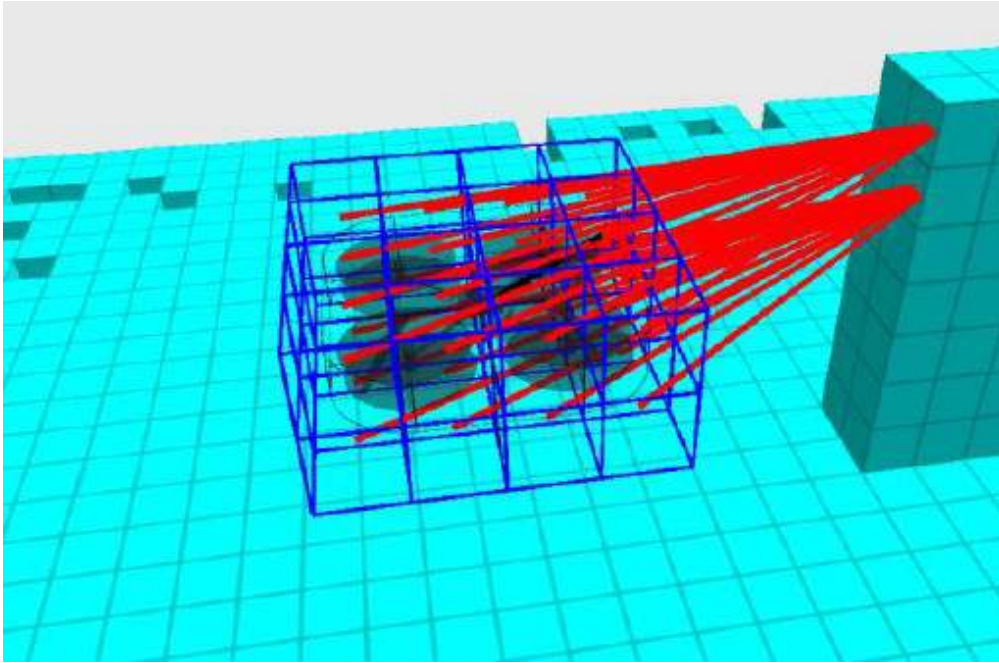


圖 2.22. 將無人機分割成數個區塊 (藍色)，並計算每一區塊的力量。加諸在無人機上，並計算每一區塊的力量加諸在無人機上。

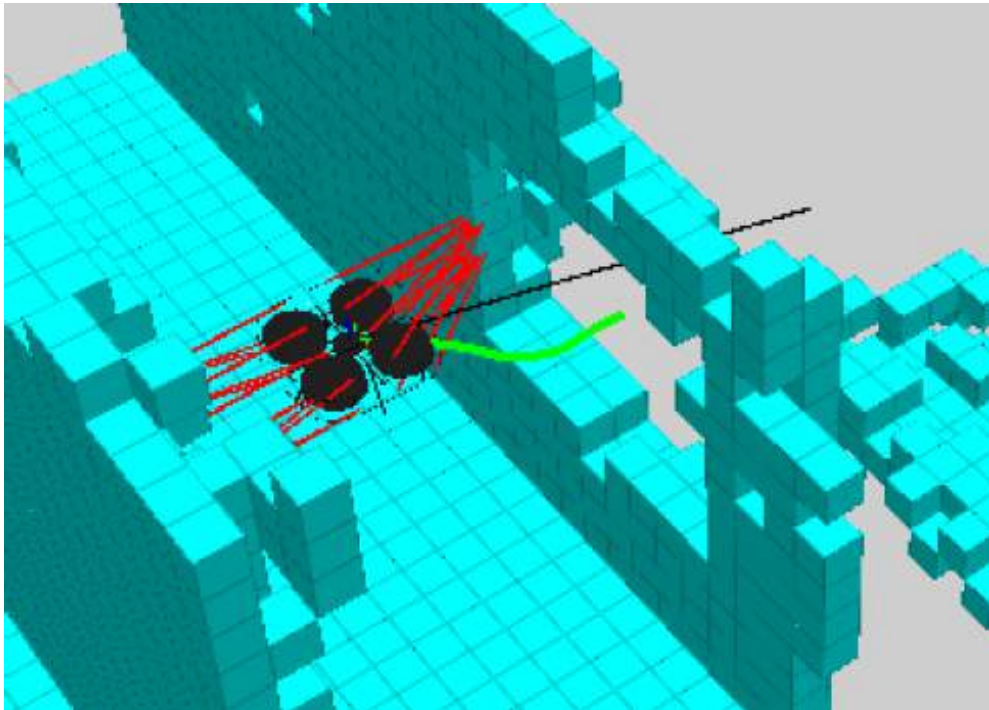


圖 2.23. 採用學習動態模式來預測一個側傾 (roll) 指令對於偏離飛行軌跡 (綠色) 的影響。目前加諸在飛行器上的力量是用紅色線表示。

在真正的自主系統中（無需外接感測器和控制），飛行器上的感測器與處理能力是限制了增加自主性和複雜性的主要因素。Holz *et al.* [9] 提出了一個硬體設置與處理管道，讓一個完全自主的無人機可在各方向上都能偵測到障礙物。針對飛行環境中之不同障礙物的特性，可以採用不同的感測模組，如圖 2.24 所示。

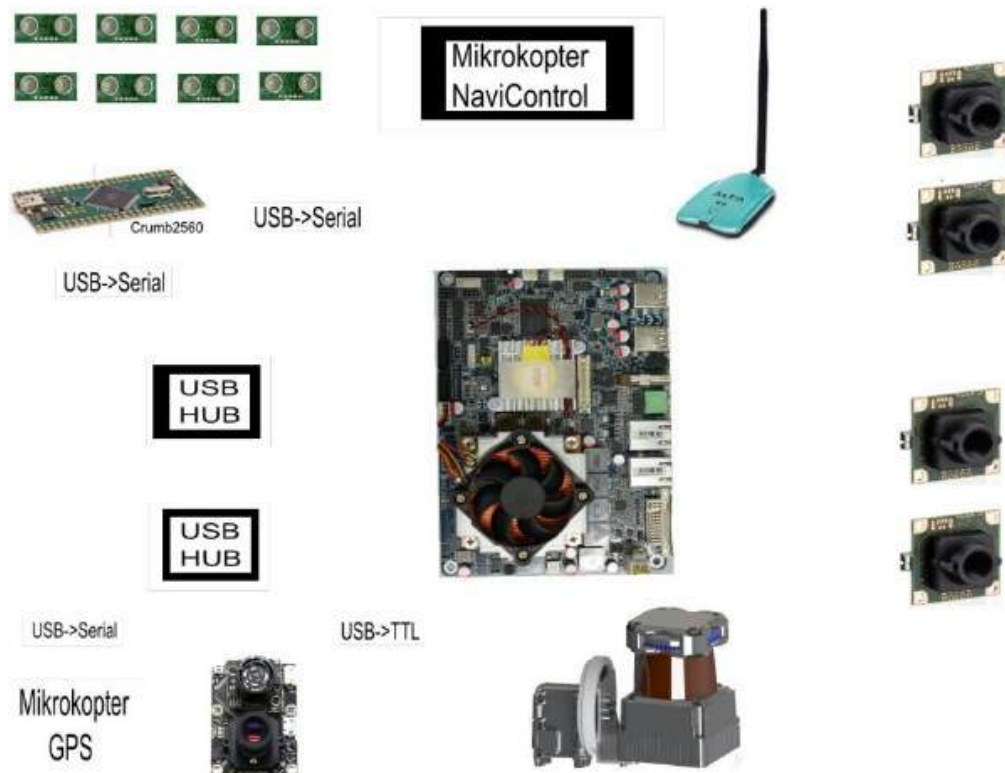


圖 2.24. 多重感測器，有 2 個立體相機 (stereo-camera pairs)、1 個 3D 雷射掃描測距儀、8 個超音波測距儀、1 個光流相機、及 1 個 GPS 感測器。

Mori and Scherer [10] 探討了一些飛行器前方之障礙物偵測的方法。他們是以障礙物在連續影像中的大小變化來偵測障礙物。他們首先使用 *SURF* 方法擷取影像中的特徵點，再以模板比對 (template matching) 比較連續影像中的特徵點以偵測障礙物並估計飛行器與障礙物的距離，如圖 2.25 所示。

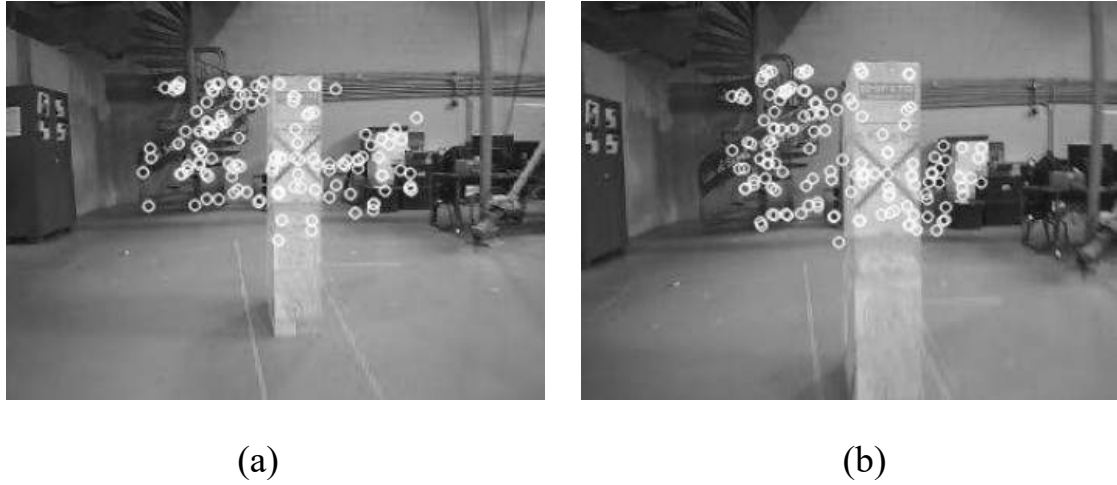


圖 2.25. 連續影像中的 *SURF* 特徵點。(a) 前一張影像。(b) 目前影像。

Saha *et al.* [11] 則是為低價格且無 *GPS* 的無人飛行器發展一個以單眼視覺做即時的障礙物偵測與避障。作者們首先提出一個數學模式來估計無人機到一障礙物的距離，接著將此距離應用於避障的演算中。

在三度空間中，首先以擴充卡曼濾波器 (extended Kalman filter, *EKF*) 及慣性度量單元 (inertial measurement unit, *IMU*) 的資料，估計無人機上的相機位置；再將影像中的障礙物特徵點以反投影 (back projecting) 的方式投影到三度空間中，估計無人機到障礙物距離的數學模式如圖 2.26 所示。

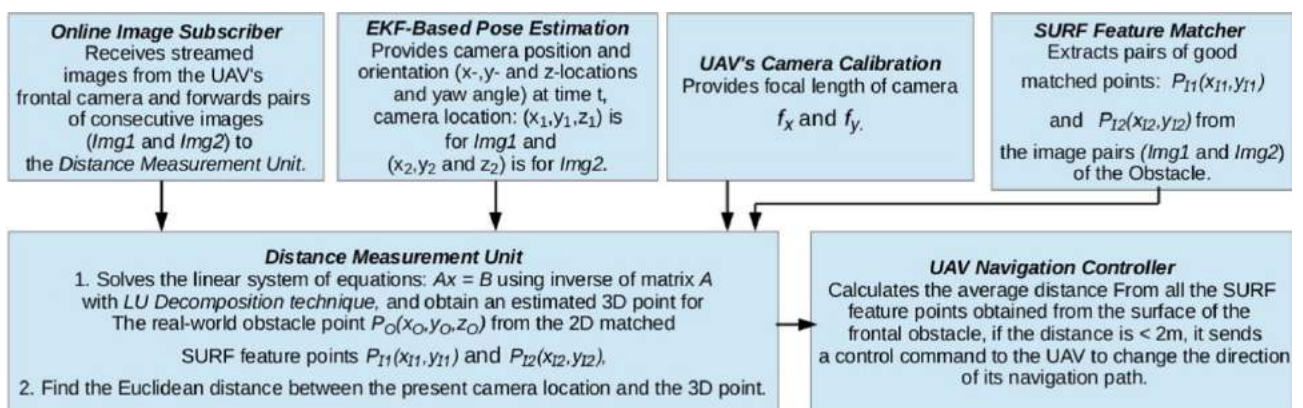


圖 2.26. 估計障礙物距離的數學模式。

Berger *et al.* [12] 提出一種新的避障反應技術，可被用在四軸飛行器以及不須精確定位技術，以飛行器中央呈現區域地圖與用球面座標來表現點雲，這個區域地圖會由一個深度感測器和慣性測量單元和註冊算法來更新，如圖 2.27 所示。

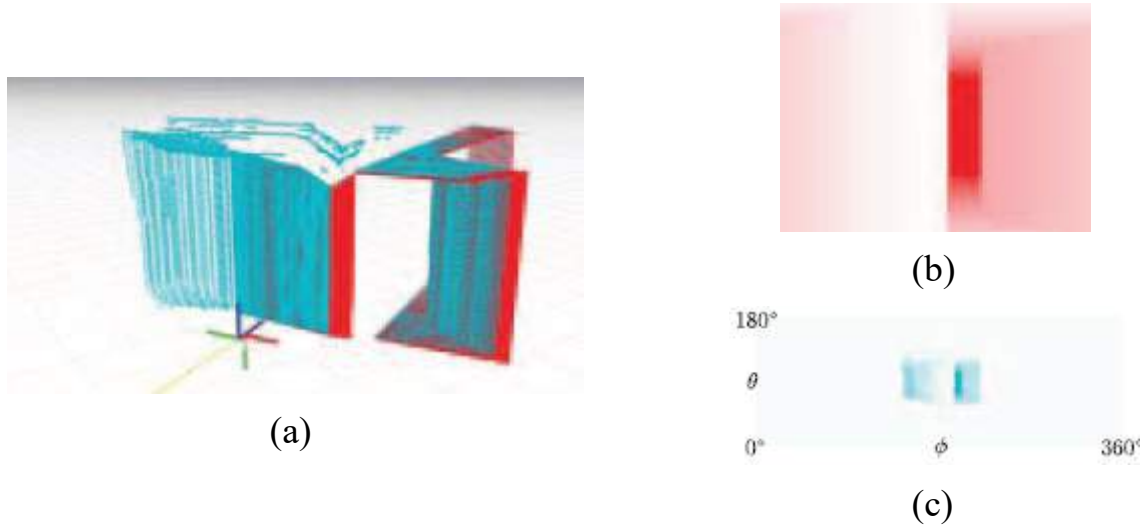


圖 2.27. 從區域地圖產生的藍色點雲為深度地圖，紅色點雲為感測器讀值。(a) 3D 視角。(b) 感測器讀值。(c) 球體深度地圖。

為了解決自主飛行器的避障和導航的問題，需要路徑規劃和反饋演算法。路徑規劃技術可計算出環境中的全局最佳路徑像是 A^* 演算法。將 A^* 演算法合併進全局地圖後，路徑規劃演算法可以計算出最佳解。因此發展出階層式 A^* 演算法用來解決大規模的全局路徑規劃演算法，雖犧牲了保證最佳解，但換取了顯著計算花費的時間。

作者提出一個動態視窗方法去計算速度向量基於當下的區域地圖，作者提出利用 OctoMap 結構去計算兩次 A^* 演算法， A^* 演算法可提供一個路徑來轉換速度向量，兩個方法都只利用區域資訊，兩個演算法在一個現實環境中用模擬器評估控制和感測器，結果可以被證實在一個真實平台中運行演算法，如圖 2.28 所示。



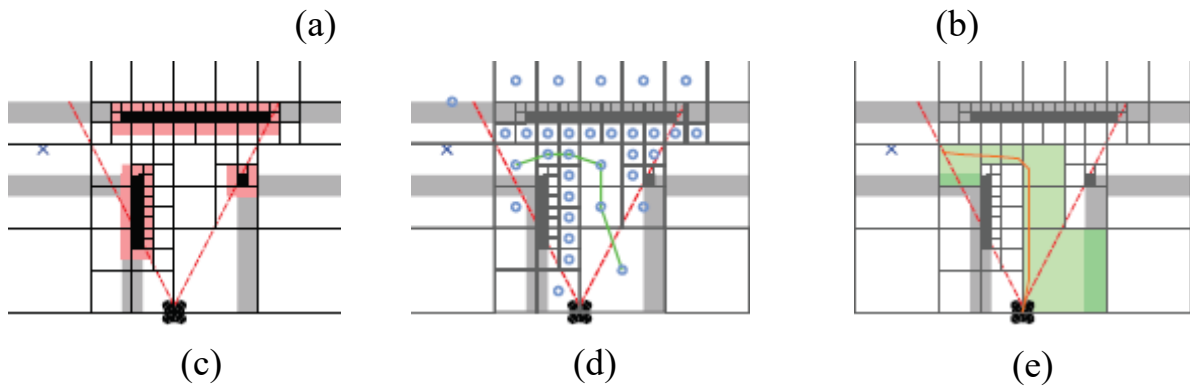


圖 2.28. 動態窗口和 A^* 二步驟方法，圖表顯示了 2D 中的概念但算法卻是在 3D 環境中工作，藍交叉點代表目標點，灰線代表障礙物，黑線代表感覺到的障礙物，黑點代表四軸飛行器。(a) 飛行器視野中的障礙物，紅色線代表飛行器視角方向。(b) DWA 演算法產生之路徑，綠色線代表可允許的速度。(c) OctoMap 架構產生的分區與不可飛行區域。(d) 在 OctoMap 終端點裡找出 A^* 演算法步驟產生的路徑，藍色圓圈代表 OctoMap 允許的節點，綠色折線為 A^* 算法的結果。(e) A^* 步驟二的演算法有高準確率，但會受到 A^* 步驟一的演算法終端點的選擇限制。表示 A^* 第二步驟計算出的區域，紅色折線代表最後的路徑。

第三章 姿態解算與障礙物偵測

在本章中，我們將先介紹飛行器是如何進行姿態解算求解出飛行器姿態，並將飛行器的遙控接收器收到的信號刻度轉換為角度的刻度，再將飛行器上的九軸姿態模組與遙控器轉換後的角度做角度誤差 PID 控制，得到相對穩定的飛行器姿態。

第二節我們將介紹如何進行障礙物偵測。我們利用論文 [13-15] 做同步定位與地圖構建 (Simultaneous localization and mapping, $SLAM$)，在每個時刻利用機上超音波感測器偵測的距離數值，將距離數值更新至佔據網格裡產生區域網格地圖，再利用航向推算演算法 (Dead Reckoning Algorithm) 將最後狀態的位置與目前時刻的位置，去估算飛行器的移動速度、加速度與方向；最後利用每時刻的區域網格地圖和航向推算演算法和慣性測量單元的資料將全域地圖做更新以供避障演算法使用。本章的第一節將介紹姿態解算演算法，其中第一節的第一項介紹姿態算法，第二項介紹 PWM 訊號轉換成角度，第三項介紹角度誤差 PID 控制，第二節將介紹障礙物偵測演算法，其中第二節的第一項介紹區域網格地圖構建，第二項介紹航向推算演算法；第三項介紹建置全域地圖構建。

3.1 姿態解算

在本節中，將會介紹我們使用的姿態解算的方法，先透過九軸姿態模組其包含加速度計、陀螺儀與磁力計三個感測器，分別得到三軸加速度、三軸角速度與三軸磁場值，利用 Mahony *et al.* [16] 互補濾波演算法計算出目前機體坐標系的俯仰角、滾動角、偏航角，再將 RC 接收器接收到的使用者操作遙控器的信號做數值穩定控制 (PID Control)，得到的數值是誤差量，根據角度誤差量大小決定要多少修正量以供姿態控制時使用。

3.1.1 姿態算法

物體在三維空間的旋轉，可以有三個自由度 (Degree of Freedom, *DoF*) roll, pitch, yaw，而表示物體在三維空間旋轉的方法有很多，例如：歐拉角、旋轉矩陣、旋轉向量、四元數...等，以下將簡單說明，並簡單說明歐拉旋轉定理。

姿態有多種數學表示方式，常見的是四元數，歐拉角，矩陣和軸角，都可用來表示三維空間裡的旋轉。他們各自有其自身的優點，在不同的領域使用不同的表示方式。在多軸飛行器中使用到了歐拉角法和方向餘弦法和四元數法，演算法比較如表 3.1 所示。

表 3.1. 常用姿態解算演算法比較。

演算法	優點	缺點	補充
歐拉角法	1. 直接利用歐拉角微分方程解算出 pitch, roll, yaw。 2. 較易於理解且概念直觀。	三角函數運算多，計算量大。	1. 適應於姿態角變化不大的情況。 2. 不適用載體全姿態旋轉。
方向餘弦法	1. 姿態矩陣微分方程求解時避免了歐拉角法中出現的奇異點問題。 2. 可以全姿態運行。	參數量過多，計算量大，即時性不佳。	很少在工程中使用。
四元數法	1. 四元數微分方程只需求解四個參數，計算量小。 2. 演算法簡單，易於操作。	漂移比較嚴重。	可以在求解四元數中修正漂移，應用較廣泛。

- i. 歐拉角法 (Euler Angle)：歐拉 (Euler) 在 1776 年提出來的，其原理是動坐標系相對參考坐標系之間的位置關係可以用一組歐拉角來描述。以歐拉角為姿態表示，其變數只有三個而且互相獨立。若以前述三個旋轉角度，依前後次序旋轉航向角 (yaw)、俯仰角 (pitch) 及側滾角 (roll) 則可獲得附體座標與導航座標的轉換矩陣 C_n^b 。

$$C_n^b = R(x, \phi)R(y, \theta)R(z, \psi) = \begin{pmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \cos \psi \sin \theta \sin \phi - \cos \phi \sin \psi & \cos \phi \cos \psi + \sin \theta \sin \phi \sin \psi & \cos \theta \sin \phi \\ \cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi & -\cos \psi \sin \theta + \cos \phi \sin \theta \sin \psi & \cos \theta \cos \phi \end{pmatrix} \quad (3.1)$$

C_n^b 表示為導航座標 (navigation frame) 轉換至附體座標 (body frame) 之轉換矩陣，為了確保歐拉角之唯一性。

用此方法求解得到的姿態矩陣永遠是正交矩陣，當載體俯仰角為 90° 時，將出現奇異點，因此該方法不能進行全姿態解算，其使用存在一定的局限。

ii. 方向餘弦法 (Direction Cosine Matrix, DCM)：一種用向量的方向餘弦來表示姿態矩陣的方法，可用來表示繞定點轉動的兩個坐標系之間的關係。該方法求解姿態矩陣避免了歐拉角法所遇到的奇異點問題。

步驟 1. 先建立旋轉矩陣 R_x, R_y, R_z ，繞 X, Y, Z 軸旋轉角度分別為 θ, ϕ, ψ ，

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{pmatrix}, R_y = \begin{pmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{pmatrix}, R_z = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.2)$$

步驟 2. 設一向量 $P(X, Y, Z)$ ，並依 $Z-Y-X$ 順序對向量旋轉，對 Z 軸旋轉， P' 為 P 對 Z 軸旋轉後的向量軸，

$$P' = R_z \cdot P \quad (3.3)$$

步驟 3. 對 Y 軸旋轉， P'' 為 P' 對 Y 軸，

$$P'' = R_y \cdot P' = R_y \cdot R_z \cdot P \quad (3.4)$$

步驟 4. 對 X 軸旋轉， P' 為 P'' 對 X 軸，

$$P''' = R_x \cdot P'' = R_x \cdot R_y \cdot R_z \cdot P \quad (3.5)$$

步驟 5. 令 $R_{xyz} = R_x \cdot R_y \cdot R_z$ ，則

$$R_{xyz} = \begin{pmatrix} \cos \phi \cos \psi & \cos \phi \sin \psi & -\sin \phi \\ \cos \psi \sin \theta \sin \phi - \cos \theta \sin \psi & \cos \theta \cos \psi + \sin \theta \sin \phi \sin \psi & \cos \phi \sin \theta \\ \cos \theta \cos \psi \sin \phi + \sin \theta \sin \psi & -\cos \psi \sin \theta + \cos \theta \sin \phi \sin \psi & \cos \theta \cos \phi \end{pmatrix} \quad (3.6)$$

iii. 四元數法 (Quaternion)：英國愛爾蘭數學家 WR. Hamilton 在 1843 年發現的數學概念。單位四元數 (unit quaternion) 可以用於表示三維空間裡的旋轉，它與常用的另外兩種表示方式 (歐拉角和方向餘弦矩陣) 是等價的，可以解決歐拉角與方向餘弦矩陣旋轉所產生的萬向鎖 (gimbal lock) 問題，且四元數表示能夠更方便地給出旋轉的轉軸與旋轉角，計算時間也比其餘兩種表示方式少。

四元數由一個實數與三個虛數組成，可以定義為

$$Q = q_0 + q_1 i + q_2 j + q_3 k = \cos\left(\frac{\theta}{2}\right) + n \times \sin\left(\frac{\theta}{2}\right) \quad (3.7)$$

其中 n 軸為轉軸方向， $\frac{\theta}{2}$ 為轉動角度大小，直角坐標系的單位向量 i, j, k 關係為

$$\begin{aligned} i \cdot i = j \cdot j = k \cdot k &= -1 \circ \\ i \cdot j = -j \cdot i = k, j \cdot k &= -k \cdot j = i, k \cdot i = -i \cdot k = j \circ \end{aligned} \quad (3.8)$$

設有兩四元數 Q_A, Q_B ，其乘法關係為

$$\begin{aligned} Q &= Q_A \cdot Q_B = (q_{A0} + q_{A1}i + q_{A2}j + q_{A3}k) = (q_{B0} + q_{B1}i + q_{B2}j + q_{B3}k) \\ &= \begin{pmatrix} q_{A0} \cdot q_{B0} - q_{A1} \cdot q_{B1} - q_{A2} \cdot q_{B2} - q_{A3} \cdot q_{B3} \\ q_{A1} \cdot q_{B0} + q_{A0} \cdot q_{B1} - q_{A3} \cdot q_{B2} + q_{A2} \cdot q_{B3} \\ q_{A2} \cdot q_{B0} + q_{A3} \cdot q_{B1} - q_{A0} \cdot q_{B2} - q_{A1} \cdot q_{B3} \\ q_{A3} \cdot q_{B0} - q_{A2} \cdot q_{B1} - q_{A1} \cdot q_{B2} + q_{A0} \cdot q_{B3} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ i \\ j \\ k \end{pmatrix} \quad (3.9) \end{aligned}$$

旋轉四元數可定義為

$$\begin{aligned} Q_{xyz} &= Q_z \cdot Q_y \cdot Q_x = \left(\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\psi}{2}\right)k \right) \cdot \left(\cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\phi}{2}\right)j \right) \cdot \left(\cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)i \right) \\ \Rightarrow \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} &= \begin{pmatrix} \cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right) \\ \cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right) - \sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) \\ \cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right) - \cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right) \end{pmatrix} \cdot \begin{pmatrix} 1 \\ i \\ j \\ k \end{pmatrix} \quad (3.10) \end{aligned}$$

設一向量 $R = xi + yj + zk$ ，及旋轉後向量 $R' = x'i + y'j + z'k$ ，四元數滿足

$$R' = Q^{-1} \cdot R \cdot Q \quad (3.11)$$

四元數倒數為

$$Q^{-1} = \frac{1}{Q} = \frac{Q^*}{Q \cdot Q^*} = \frac{Q^*}{|Q|} \quad (3.12)$$

當四元數被歸一化後 $|Q| = 1$

$$Q^{-1} = Q^* \quad (3.13)$$

可得到

$$R' = Q^* \cdot R \cdot Q \quad (3.14)$$

將其展開

$$\begin{aligned} x'i + y'j + z'k &= (q_0 - q_1i - q_2j - q_3k) \cdot (xi + yj + zk) \cdot (q_0 + q_1i + q_2j + q_3k) \\ \Rightarrow \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} &= \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \end{aligned} \quad (3.15)$$

令四元數旋轉矩陣為 M_q

$$M_q = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (3.16)$$

將 Q_{xyz} 帶入 M_q 會得到與 R_{xyz} 相同結果，

$$\begin{aligned} M_q &= \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \\ &= \begin{pmatrix} \cos\phi \cos\psi & \cos\phi \sin\psi & -\sin\phi \\ \cos\psi \sin\theta \sin\phi - \cos\theta \sin\psi & \cos\theta \cos\psi + \sin\theta \sin\phi \sin\psi & \cos\phi \sin\theta \\ \cos\theta \cos\psi \sin\phi + \sin\theta \sin\psi & -\cos\psi \sin\theta + \cos\theta \sin\phi \sin\psi & \cos\theta \cos\phi \end{pmatrix} \end{aligned} \quad (3.17)$$

將歐拉角轉成四元數，由 M_q 可得出三個姿態角。

$$\theta = \tan^{-1} \frac{M_{23}}{M_{33}}, \phi = -\tan^{-1} \frac{M_{13}}{M_{11}}, \psi = \tan^{-1} \frac{M_{12}}{M_{11}} \quad (3.18)$$

但其優越性在於計算量小、精度高、可避免奇異性，進行姿態求解時估計出這些誤差是很重要的。與方向餘弦法相比，四元數法的優點在於不僅歪斜誤差等於零；而且刻度誤差的推導很簡單，能得出便於進一步分析的解析表達式，而方向餘弦法只有在特殊的情況下才能分析和檢測到刻度誤差，且不能得出通用的結論。通過從不同角度對歐拉角法、方向餘弦法和四元數法進行對比。結果表明四元數法具有最佳的性能。

3.1.2 *PWM* 訊號轉換成角度

首先介紹兩種調變訊號的方法，第一個為脈波寬度調變 (pulse width modulation, *PWM*) 是將類比訊號轉換為脈波的一種技術，一般轉換後脈波的週期固定，但脈波的工作週期會依類比訊號的大小而改變。第二個為脈波位置調變 (pulse position modulation, *PPM*)，*PPM* 訊號可由 *PWM* 訊號組合而成。

PPM 訊號為本論文的 Pixhawk 飛行控制板與遙控接收器的通訊訊號，將接收到的訊號透過 *PPM* 編碼器將 *PPM* 訊號轉化為多個通道的 *PWM* 訊號，如圖 3.1 所示，yaw, pitch, roll 訊號最小值與最大值範圍定義為 ± 90 度，throttle 訊號以起飛的最小值定義為飛行器控制的最小值，歐拉角對應遙控器訊號的關係如表 3.2 所示，遙控器訊號如圖 3.2 所示，以遙控器左手邊搖桿為例，上下撥動代表飛行器油門值訊號上升與下降，左右撥動代表飛行器偏航值訊號下降與上升，若以遙控器右手邊搖桿為例，上下撥動代表飛行器俯仰值訊號上升與下降，左右撥動代表飛行器滾動值訊號下降與上升。

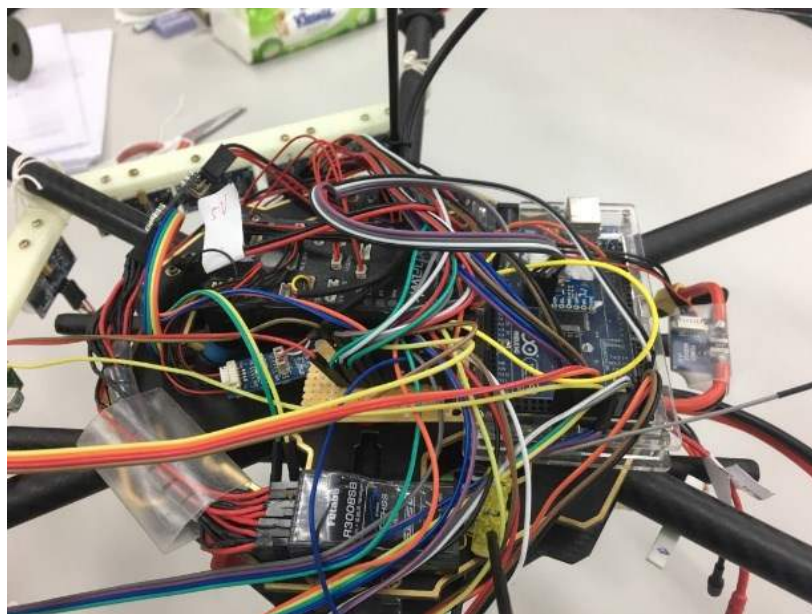


圖 3.1. RC 接收器接受信號傳送 PPM 訊號給 PPM Encoder 後將訊號擷取至 Arduino 作處理。

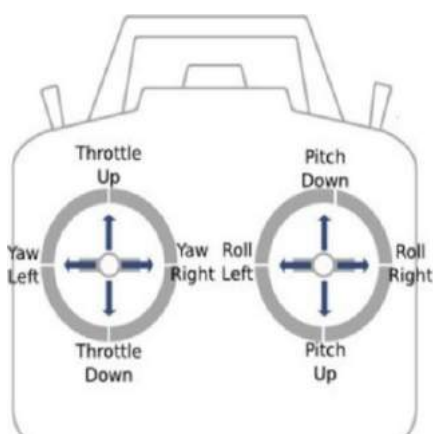


圖 3.2. 遙控器訊號表示。

表 3.2. 歐拉角與遙控器訊號對應表。

左手	上下	Throttle (Throttle)	油門(1 通道)
	左右	Yaw (Rudder)	偏擺(4 通道)
右手	上下	Pitch (Elevator)	俯仰(3 通道)
	左右	Roll (Aileron)	翻滾(2 通道)

3.1.3 角度誤差 *PID* 控制

我們使用 *PID* 控制系統 (proportional-integral-derivative controller) 來對飛行器的飛行姿態做穩定控制，如圖 3.3 所示。飛行器透過九軸姿態模組得到當下的姿態值，並以遙控器輸入數值作為輸入的期望值， $e(t)$ 代表期望值減姿態值，透過比例項、積分項與微分項運算後取總和， $Output(t)$ 代表輸出總和，以減小系統內數值的震盪反應，透過回饋機制將誤差減到最小，以達到穩定地飛行控制。

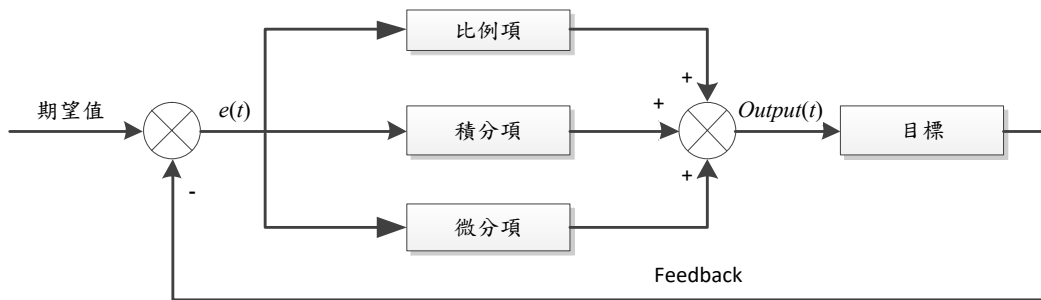


圖 3.3. *PID* 控制系統。

PID 控制器是在工業控制應用中常見的一種反饋迴路機制，控制器將收集到的數據和一個參考值進行比較並計算出誤差，將此誤差用來當作新的輸入值，這個新的輸入值的目的可以讓系統的數據趨向穩定值，方法是利用信號間的誤差做比例 (proportional, *P*) 控制、積分 (integral, *I*) 控制和微分 (derivative, *D*) 控制的運算，產生的結果彼此相加即為控制器的輸出

$$Output(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t), \quad (3.19)$$

其中

K_p ：比例增益調適參數，

K_i ：積分增益調適參數，

K_d ：微分增益調適參數，

$e(t)$ ：誤差 = 設定值 - 回授值，

t ：目前時間，

τ ：積分變數，數值從 0 到目前時間 t 。

- i. 比例控制項：在比例控制中會考慮當前的誤差，比例控制的輸出為誤差值乘以常數 K_p ，但僅有比例控制時，系統輸出容易出現穩態誤差的問題，穩態誤差即是要使受控制系統輸出為非零數值時，存在的非零誤差值或偏移量；可透過加入積分控制消除穩態誤差。

若是比例增益大時，在相同的誤差量下，會有較大的輸出，但若比例增益太大，會使系統不穩定；相反的，若是比例增益小時，在相同誤差量下，其輸出較小，因此控制器會較不敏感的。若是比例增益太小，當有干擾出現時，其控制信號可能不夠大，會有無法修正干擾的情況，比例項輸出公式可以定義為

$$P_{out} = K_p e(t) \quad (3.20)$$

- ii. 積分控制項：在積分控制中會考慮過去誤差，積分控制的輸出為將過去一段時間內產生的誤差和乘以常數 K_i 。 K_i 會從過去的平均誤差值來找到系統的輸出結果和預定值的平均誤差。一個簡單的比例系統會震盪，會在預定值的附近來回變化，因為系統無法消除多餘的糾正。通過加上負的平均誤差值，平均系統誤差值就會漸漸減少。所以，最終這個 PID 迴路系統會在設定值穩定下來，積分控制項輸出可定義為

$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (3.21)$$

- iii. 微分控制項：在微分控制中會考慮將來的誤差，微分控制的輸出為透過計算誤差的一階導數並乘以常數 K_d 。這個導數的控制會對系統的改變作出反應。導數的結果越大，那麼控制系統就對輸出結果作出更快速的反應。這個 K_d 參數也是 PID 被稱為可預測的控制器的原因。 K_d 參數對減少控制器短期的改變很有幫助，微分控制項輸出定義為

$$D_{out} = K_d \frac{d}{dt} e(t) \quad (3.22)$$

綜合上述將控制器離散化以方便運算，輸出可定義為

$$Output(t) = K_p e(t) + K_i \sum_{k=0}^n e(t) \Delta t + K_d \frac{e(t) - e(t-1)}{t} \quad (3.23)$$

Roll 和 pitch 軸按照以上公式計算 PID 輸出，但 yaw 軸比較特殊，因為偏航角法線方向剛好和地球重力平行，這個方向的角度無法由加速度計直接測得，需要增加一個電子羅盤來替代加速度計。如果不使用羅盤的話，我們可以單純的通過角速度積分來測得偏航角，缺點是由於積分環節中存在積分漂移，偏航角隨著時間的推移會偏差越來越大，就會出現航向角漂移的問題。

串級 PID 就是兩個 PID 控制演算法，期望角度就是遙控器控制飛行器的角度值，回饋當前角度就是感測器測得的飛行器角度，這裡的角度指的是 roll、pitch、yaw 三個角度，三個角度在 PID 控制計算時是相互獨立的。串級 PID 可以增強系統的抗干擾性或稱增強穩定性，因為有兩個控制器控制飛行器，它會比單個控制器控制更多的變數，使得飛行器的適應能力更強，串級 PID 的過程如圖 3.4 所示。

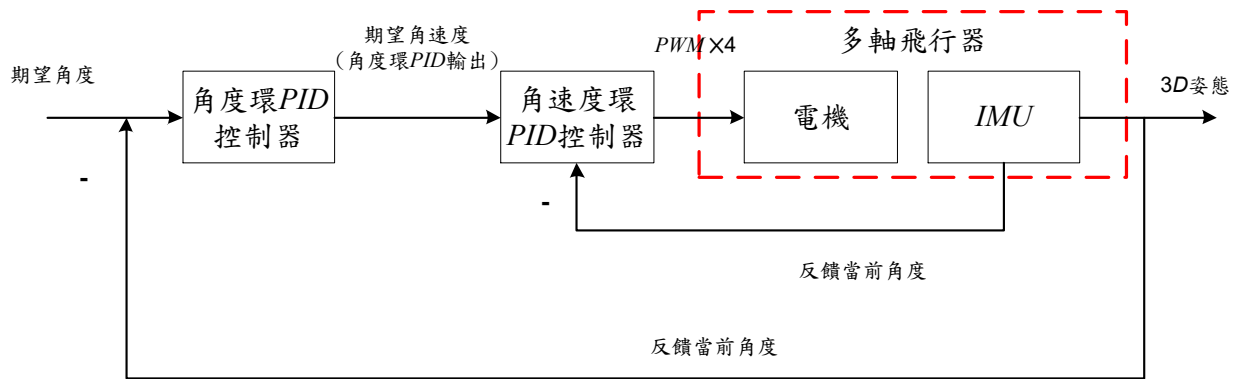


圖 3.4. 串級 PID 控制器。

3.2 障礙物偵測演算法

本論文的障礙物偵測演算法利用超音波感測器資料，利用同步定位與地圖構建方法 (Simultaneous localization and mapping, $SLAM$) 將超音波感測器偵測到的數值來做區域地圖網格構建，並利用航向參考演算法找出飛行器的移動方向與位置，最後利用全域地圖網格構建將飛行器定位。

$SLAM$ 是希望機器人從未知環境的未知地點出發，在運動過程中通過重複觀測到的地圖特徵 (比如，牆角，柱子等) 定位自身位置和姿態，再根據自身位置

增量式的構建地圖，從而達到同時定位和地圖構建的目的。其主要包含四個重點，感測器資料、以 *SLAM* 為基礎發展的障礙物偵測演算法來建模、地圖構建和定位。

3.2.1 區域地圖網格

以飛行器前方範圍 180 度定義為 8×10 的區域地圖網格，每個網格為 80cm×80cm，故區域地圖範圍為長 640cm 與寬 800cm，將配置在飛行器上的 11 個超音波感測器抓取的距離資訊，如表 3.2 所示。

表 3.2 代表超音波 11 個感測器偵測的距離

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
----	----	----	----	----	----	----	----	----	----	-----

將超音波 11 個感測器偵測的距離分別根據飛行器配置感測器的方向填入數值如圖 3.5 所示，依照距離配置位置方法定義為

$$\begin{aligned} \text{飛行器位置 } X \text{ 座標} &= \frac{\text{超音波測距偵測距離} \times \cos\theta}{\text{網格大小}} \\ \text{飛行器位置 } Y \text{ 座標} &= \frac{\text{超音波測距偵測距離} \times \sin\theta}{\text{網格大小}}。 \end{aligned} \quad (3.24)$$

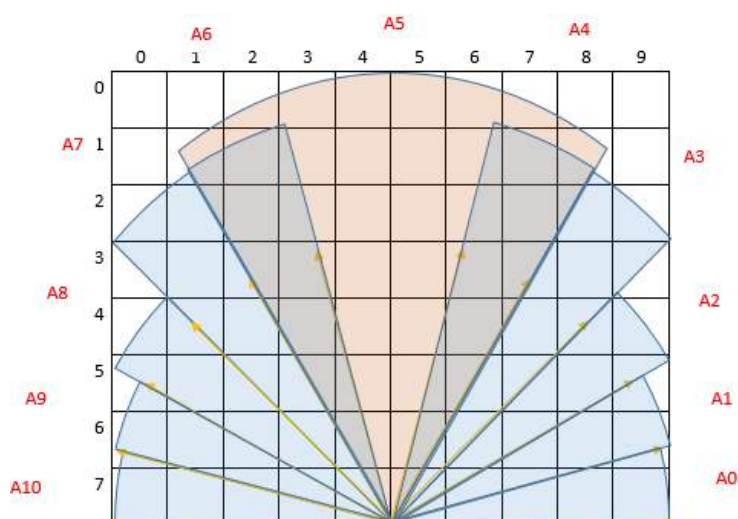


圖 3.5. 超音波感測器配置圖。

根據超音波感測器的特性，讀取一次距離資訊會代表在偵測角度範圍 15 度內都代表有物體，將其依照感測器覆蓋區域地圖網格裡的面積來代表障礙物可能存在的機率，如圖 3.6 所示，將資訊填入後的超音波感測器配置圖，如圖 3.7 所示。

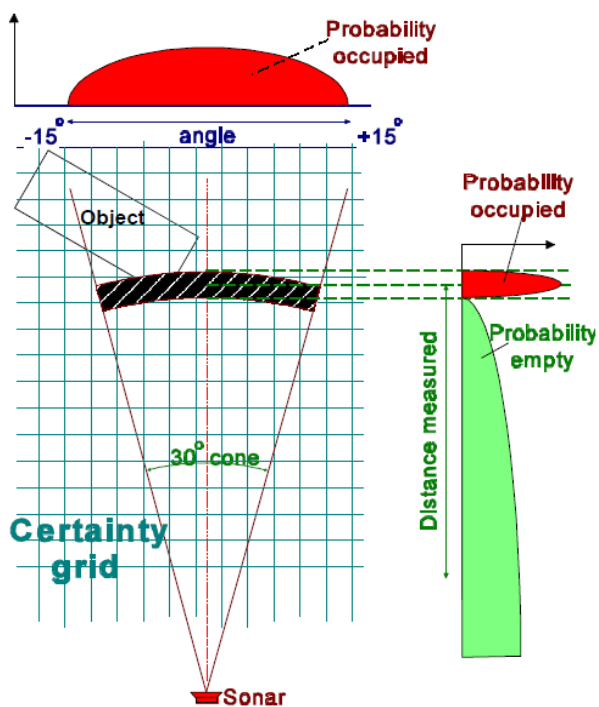


圖 3.6. 超音波測距決定障礙機率數值圖。

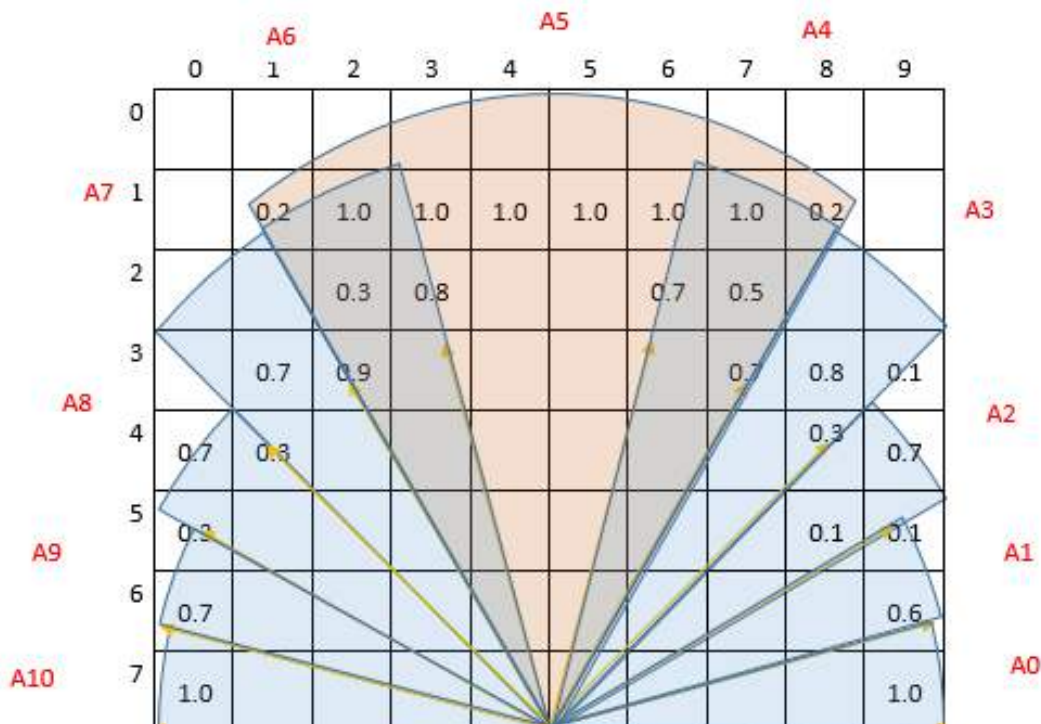


圖 3.7. 區域地圖配置機率圖。

3.2.2 航向參考演算法

Goodwin [17] 提出半正矢公式 (Haversine formula)，是從球面三角學中計算球面三角形側面和角度的半徑定律，也是球面上任意兩點之間的距離計算公式，

$$S = 2 \arcsin \sqrt{\sin^2 \frac{dLat}{2} + \cos(Lat1) \times \cos(Lat2) \times \sin^2 \frac{dLon}{2}} \times 6378.137, \quad (3.25)$$

其中

$dLat$: 為兩次時刻的緯度相差值，

$Lat1$: 為上一時刻的緯度值，

$Lat2$: 為現在時刻的緯度值，

$dLon$: 為兩次時刻的緯度相差值。

本論文以區域地圖為相對座標，全域地圖為絕對座標，全域地圖每個網格對應現實距離為 80cm×80cm，根據航向參考演算法找出在 24×24 的全域地圖網格裡的飛行器位置，以飛行器起飛位置的 GPS 座標當作起始點，並開始定位在全域地圖中；在每個時刻更新後，透過 GPS 座標經度與緯度數值的變化，就能計算出位置更新後的距離變化，並在全域地圖網格中更新其定位資料。

3.2.3 全域地圖網格

利用九軸姿態模組量測出的偏航角找出當下時刻的飛行器朝向，將移動的方向與距離對應網格的大小，找出飛行器在全域地圖裡的位置，如圖 3.8 所示，其中 Ws 代表全域地圖的長與寬格數皆為 24 格，每個格子對應真實環境為 80cm×80cm 大小，Certainty value 為可能障礙物的數值，Active Window 代表飛行器移動的活動窗口，active cells 代表其中的格子。

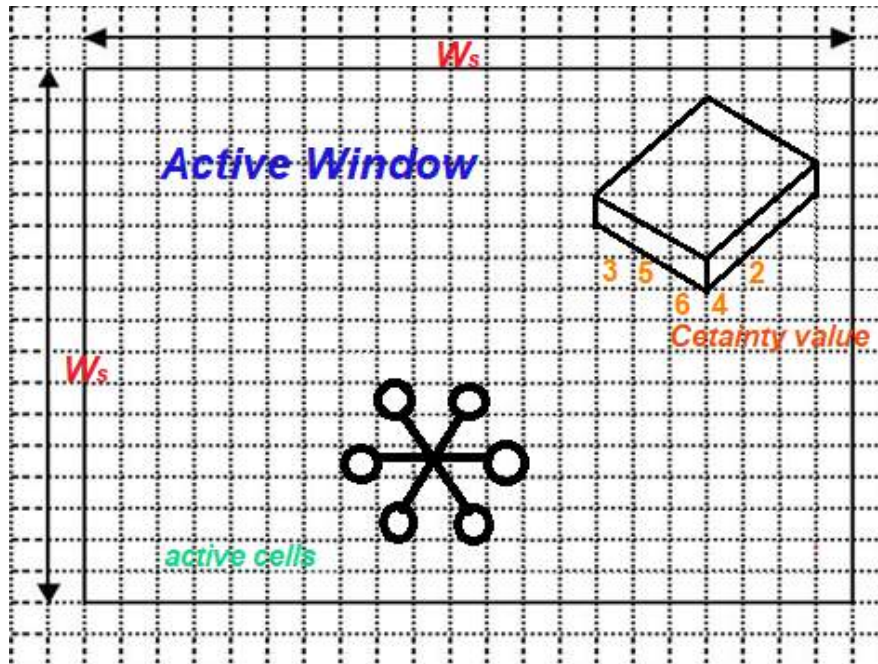


圖 3.8. 飛行器對應於全域地圖位置。

每時刻依照飛行器旋轉的偏航角，與 *GPS* 偵測的距離，更新飛行器移動到的新位置，將飛行器移動時偵測的區域地圖更新到全域地圖裡，將區域地圖裡的四個邊點影像旋轉到全域地圖裡，將其覆蓋的每個網格座標依照影像旋轉的逆運算找出原始區域地圖裡的網格座標，並將其網格裡的障礙物機率值更新到全域地圖裡的網格。

以實驗結果說明，每時刻程式計算出的偏航角 (yaw) 將相對應的區域地圖作影像旋轉更新到全域地圖裡，如圖 3.9 所示，程式計算出的偏航角為 -40° ，且因 *GPS* 座標的更新，計算出經緯度差的實際距離，將數值乘以 $\cos \theta$ 得到 X 軸偏移位置，將數值乘以 $\sin \theta$ 得到 Y 軸偏移位置， θ 為更新 *GPS* 位置的移動方向，定位出飛行器的位置後，將偏航角對應的區域地圖更新到全域地圖裡，如圖 3.10 所示，以此類推每一時刻感測器偵測到的距離數值都會更新至全域地圖裡。


```

*****Rotate2*****
yawAngle =30.000000
6.449100
*****print Global map result*****
0| 1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12| 13| 14| 15| 16| 17| 18| 19| 20| 21| 22| 23|
1|
2|
3|
4|
5|
6|
7|
8|
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|

```

圖 3.9. yaw angle = 30 度的全域地圖環境。

```

*****Rotate2*****
yawAngle =-40.000000
3.873000
*****print Global map result*****
0| 1| 2| 3| 4| 5| 6| 7| 8| 9| 10| 11| 12| 13| 14| 15| 16| 17| 18| 19| 20| 21| 22| 23|
1|
2|
3|
4|
5|
6|
7|
8|
9|
10|
11|
12|
13|
14|
15|
16|
17|
18|
19|
20|
21|
22|
23|

```

圖 3.10. yaw angle = - 40 度的全域地圖環境。

第四章 避障分析與姿態控制

在本章中，我們將介紹如何進行避障分析與姿態控制，首先建立出 *SLAM* 模型的架構，Khatib [18] 提出一種機器人路徑規劃法稱人工勢場法 (Artificial potential field, *APF*)，該算法將目標和障礙物分別看做對機器人有引力和斥力的物體，機器人沿引力與斥力的合力來進行運動。該方法結構簡單，便於簡單環境的即時控制，在即時避障和平滑的軌跡控制方面，得到了廣泛應用，其不足在於存在局部最佳解時，容易產生死結現象，因而可能使移動機器人在到達目標點之前就停留在局部最佳點。由於人工勢場法無法很好地處理在目標點附近的陷阱干擾問題，因此 Borenstein and Koren [19] 在 1991 年提出了 *VFH* 算法。

VFH 算法核心思想，算法會計算各個方向的行進代價，該方向的障礙越多，代價越高，並且會累加該方向不同距離的障礙物 (根據距離，權重不同)。根據不同方向的行進代價，可以直觀的用一個柱狀圖表示。橫坐標為 0-360 度的方向，縱坐標是該角度下的行進代價。柱狀圖越高，表示向該方向行進的代價越高，也表明越不可能通過。理論上，這個柱狀圖低的區域是便於行進的，但可能會偏離目標方向，因此，需要一個平衡函數來平衡行進代價和目標方向。最終，會選擇一個相對最適宜的方向行進。

本章的第一節將介紹避障演算法，其中第一節的第一項介紹與距離相關的障礙物可能值，第二項介紹轉換成平滑一維極性機率密度圖，第三項介紹選擇候選區塊，第四項介紹選擇候選區塊內飛行方向，第二節將介紹飛行器姿態控制，其中第二節的第一項將介紹飛行策略，第二項介紹將角度轉換成 *PWM* 訊號。

4.1 避障演算法

在本章中，我們將介紹如何進行避障分析，Borenstein and Koren [19] 提出的虛擬力場法 (Virtual force field; *VFF*)。為了解決虛擬力場的死結局限性問題，於是 Borenstein and Koren [20, 21] 提出方向區間柱圖法 (Vector Field Histogram, *VFH*)，此方法不需使用複雜的計算，只需要超音波測距感測器 (ultrasonic range finder, *URF*)，所獲得的資訊計算出障礙物可能存在之方向，對於運用於避障功能上來說是相當適合的，本研究將方向區間柱圖法分成四個流程，如圖 4.1 所示。

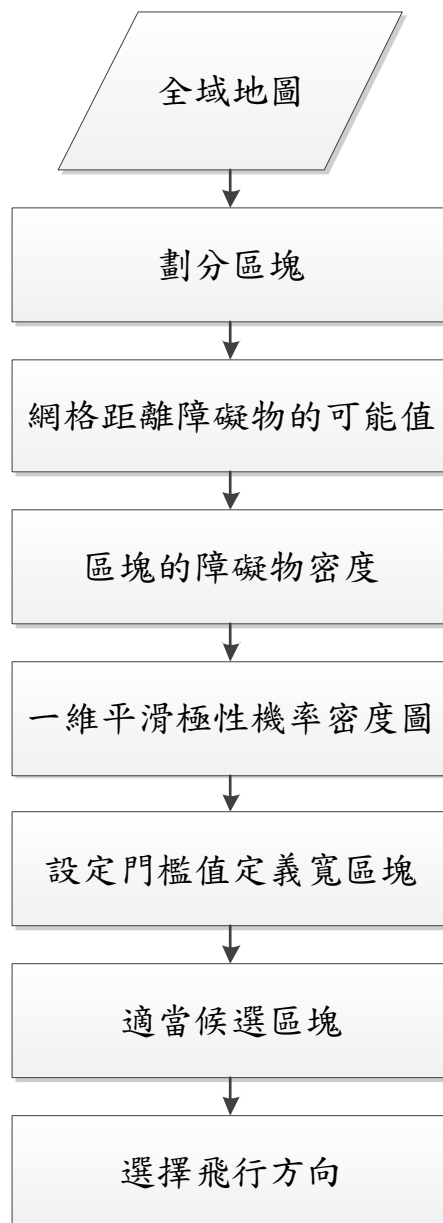


圖 4.1. 方向區間柱圖法流程。

4.1.1 與距離相關的障礙物可能值

將每個扇形區塊中，每個格子 (grid cells) 內的可能障礙物值 (certainty value, C^*) 透過公式計算出 $m_{i,j}$ ， i, j 為機率密度網格 (histogram grid) 中的 X 座標及 Y 座標， $a-b \times d_{max}=0$ ，其中 d_{max} 為飛行器與活性網格之最遠距離， d_{max} 為最遠的活性網格到飛行器中心點的距離，最遠時 $m_{i,j}=0$ ，若靠近時會增加其數值，距離障礙物可能值定義為

$$m_{i,j} = (C^*_{i,j})^2 \times (a - b \times d_{i,j})。 \quad (4.1)$$

$m_{i,j}$ 與 $-d$ 成正比，當活動網格產生大的障礙物向量，代表他們在飛行器附近，較小的障礙物向量代表離飛行器較遠，其中 d_{max} 是代表最遠的活動網格到飛行器的距離，這樣 $m_{i,j}$ 會等於 0，最遠距離定義為

$$d_{max} = ws\sqrt{2}。 \quad (4.2)$$

4.1.2 平滑一維極性機率密度圖

為了將全域地圖轉換成圍繞飛行器瞬間位置的一維極性直方圖 (polar histogram)，需先將全域地圖內出現經常性的讀值代表為障礙物，而單一出現的讀值可能為雜訊所引起，以飛行器所在位置將全域地圖以每隔 15 度為一區塊劃分，劃分成 24 個區塊，將各個扇型區塊內的距離障礙物可能值加總在一起，得到這區塊的障礙物密度，將所有區塊的障礙物密度做平滑化後，得到每區塊的平滑化障礙物密度，即可找到整個地圖障礙物較少的地方，進一步找到避障方向。

i. 區塊劃分

二維座標活動窗口 (Active Window) 轉換成圍繞飛行器瞬間位置的一維極性機率密度圖，首先活動窗口範圍內出現經常性的讀值代表為障礙物，而單一出現的讀值可能為雜訊所引起，全域地圖中切割扇型區塊內的障礙物可能性數值圖如圖 4.2 所示。

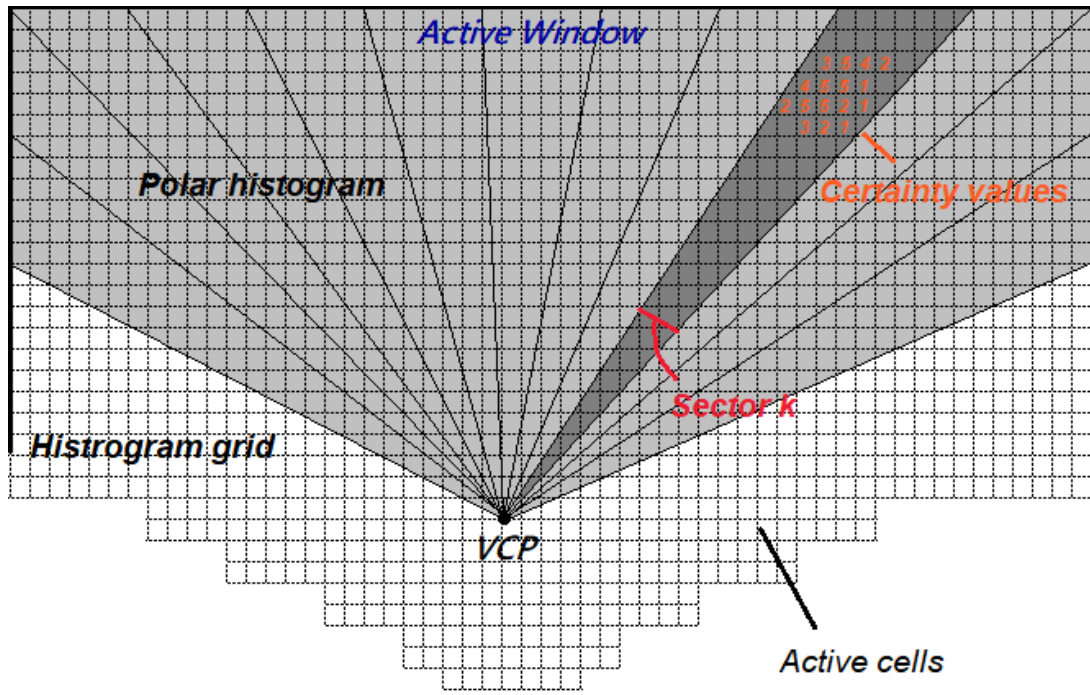


圖 4.2. 全域地圖中切割扇型區塊內的障礙物可能性數值圖。

並根據飛行器所在位置 (Vehicle Center point, VCP)，將活動窗口 (Active Window) 以每隔 α 度為一區塊劃分，劃分成若干個區塊，區塊定義為

$$K = INT(\beta_{ij}/\alpha), \quad (4.3)$$

其中

K ：扇形區塊編號，

β ：飛行器幾何中心對活性網格的角度，

α ：區塊角度，

INT：數值取整數。

ii. 障礙物密度圖

計算全域地圖裡每個網格的座標裡的障礙物密度圖， $i, j \in$ 第 k 區塊的 X 座標及 Y 座標，障礙物密度定義為

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

iii. 平滑障碍物密度圖

a 、 b 在獲得活性網格的 m_{ij} 大小後，將各個區塊內的網格 m_{ij} 全部加總在一起，即可獲得這區塊的障碍物密度 (Polar Obstacle Density, POD) h_k 值， k 值為扇區塊編號，之後再將所有區塊 h_k 值作平滑化得到每區塊的平滑化障碍物密度 (Smoothed Polar Obstacle Density, $SPOD$) h'_k 值，其中 l 為平滑化參數，飛行器對照平滑障碍物密度，如圖 4.3 所示，將平滑化障碍物密度對應到一維極性機率密度圖，如圖 4.4 所示。

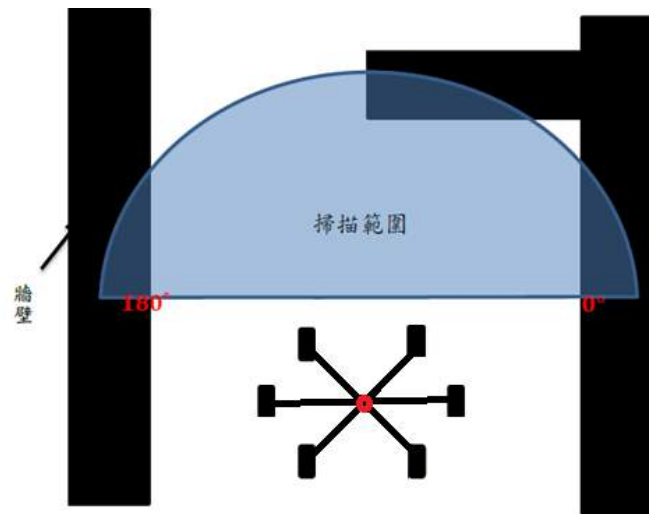


圖 4.3. 每區塊的平滑障碍物密度之示意圖。

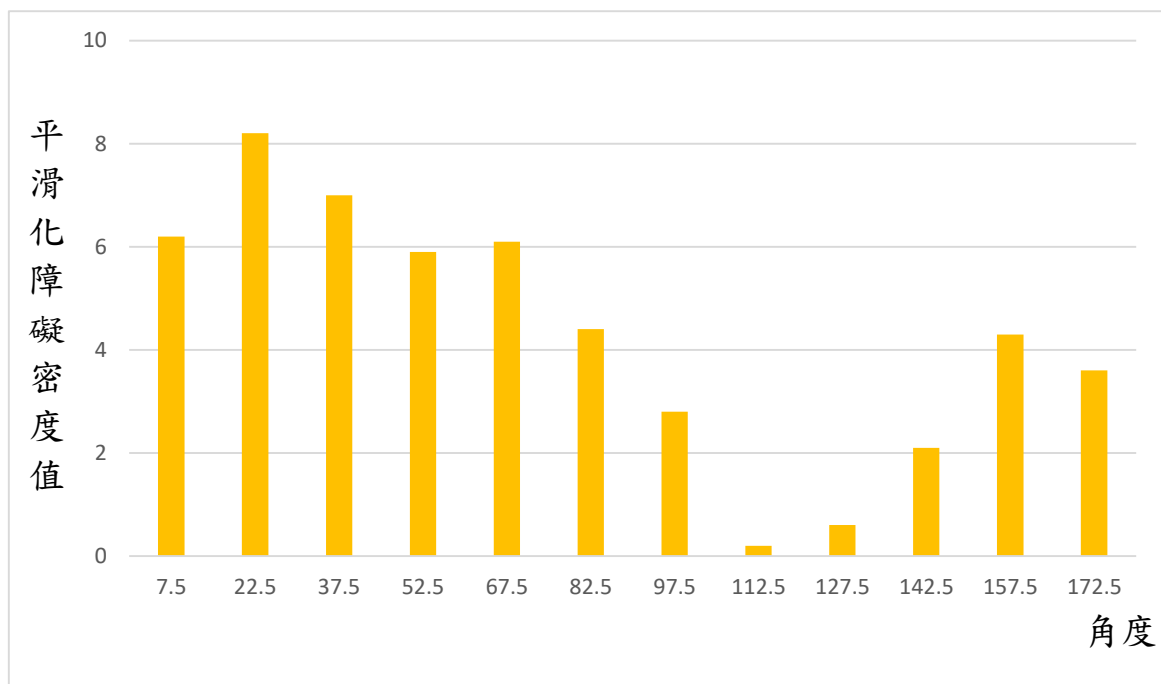


圖 4.4. 一維極性機率密度圖。

iv. 設定門檻值

閾值是採自適應閾值法，適當的門檻值可以使飛行器在外知環境中避開障礙物，若是將門檻值設定過大時，則會使飛行器誤判障礙物導致相撞；而是門檻值設定過小時，則導致飛行器因此停滯不前，門檻值設定如圖 4.5 所示。

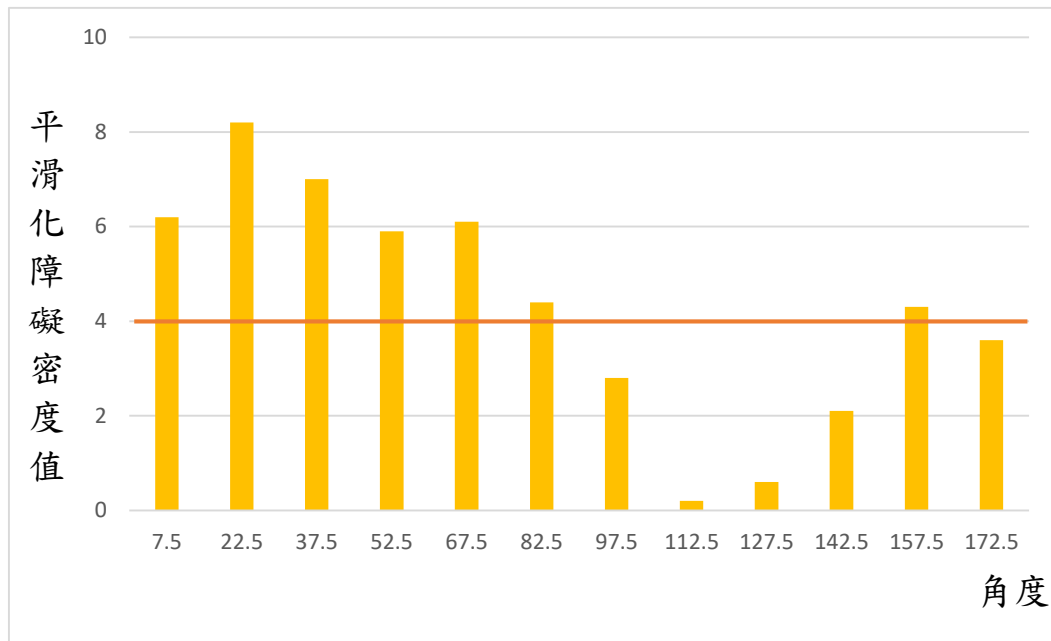


圖 4.5. 一維極性機率密度閾值圖。

4.1.3 選擇適當候選區塊

在向量場直方圖算法會計算候選區塊大小，候選區塊分成寬區塊及窄區塊，而在寬窄區塊的評斷，若是低於閾值的連續扇區數目大於 5 個，則會將此區塊判定為寬區塊，反之為窄區塊；後會選擇適合的寬區塊，其區塊為靠近飛行方向之區塊。

4.1.4 選擇候選區塊內飛行方向

最後步驟為決定飛行器旋轉前進方向的判斷，在上一步驟選訂好適合之區塊後，會在此區塊內更進一步選擇適合的扇區方向 θ ，而飛行器將會旋轉至此扇區前進。

在障礙物環境中產生並選擇合適的寬闊區塊，在這此區塊中上一時刻飛行方向的扇區本文會定義為 K_{FD} ，而離 K_{FD} 最近且低於門檻值的扇形稱為 K_n ，最遠的扇區可定義為

$$K_f = K_n + S_{max}, \quad (4.5)$$

最後飛行器決定旋轉的方向 θ ，

$$\theta = (K_f + K_n) / 2, \quad (4.6)$$

飛行器移動方向為 K_f 與 K_n 的合力方向，如圖 4.6 所示。

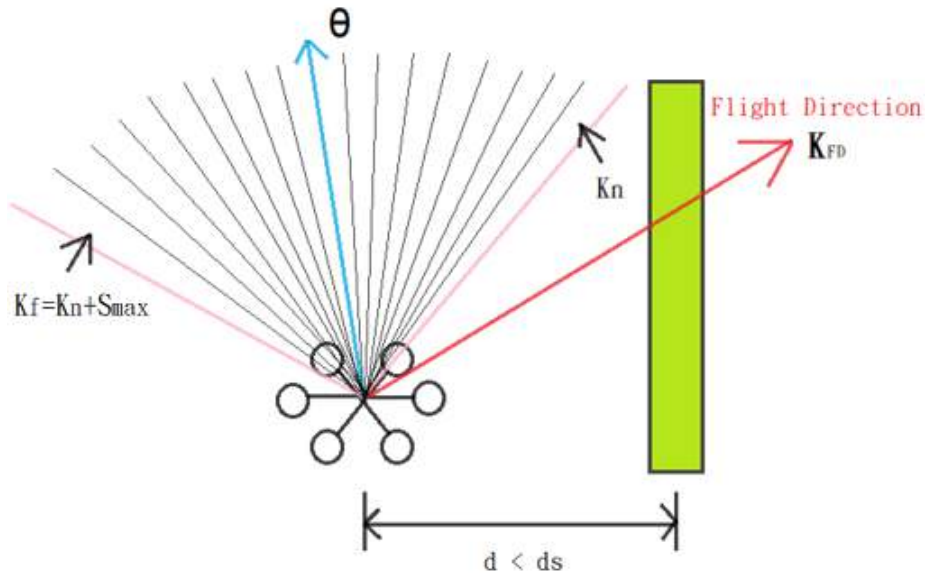


圖 4.6. 飛行器移動方向圖，綠色代表障礙物環境。

4.2 飛行器姿態控制

利用飛行器姿態解算後得到的俯仰角、滾動角和偏航角和避障分析後，根據飛行策略與將其飛行移動方向角度轉換成相對應的 *PWM* 訊號已達到飛行器姿態控制。

4.2.1 飛行策略與控制

依照連續低於門檻值的區塊，定義為寬區塊後，再找出目前靠近飛行器的飛行方向且低於門檻值的區塊。

控制飛行器 *PWM* 訊號，主要以飛行方向的 roll 與 yaw 角度去做控制，依照飛行器前方定義障礙物移動訊號，飛行器控制方式可如圖 4.7 所示。

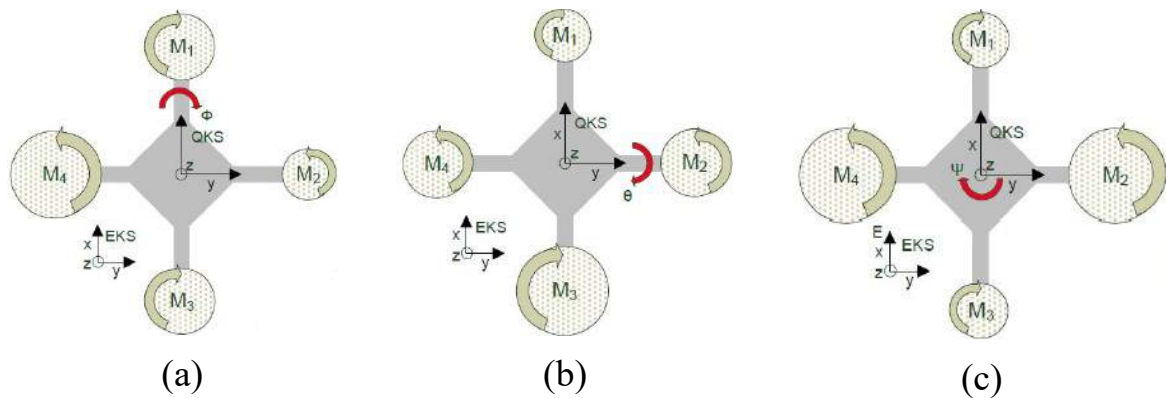


圖 4.7. 飛行器控制方式。(a) 翻滾 (roll) (b) 俯仰 (pitch) (c) 偏航 (yaw)。

4.2.2 將角度轉換成 *PWM* 訊號

將飛行策略決定後的角度訊號，轉化為 yaw, pitch, roll 信號的相對應角度，再將所有的角度換算為飛行器遙控器最小值與最大值的相對應轉換，如圖 4.8 所示；角度換算可定義為

$$\frac{Angle}{90} = \frac{PWM - PWM_{Average}}{PWM_{Max} - PWM_{Average}} \quad (4.7)$$

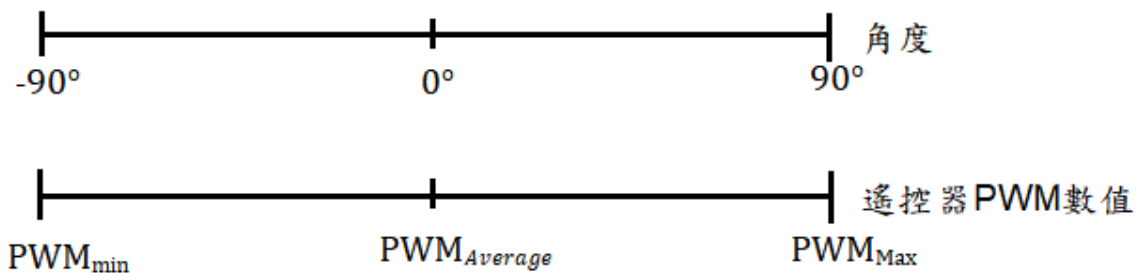


圖 4.8. 飛行器角度換算對照圖。

若 yaw 角度算出為 40 度，則將 yaw *PWM* 的最小值和最大值取平均，以角度 0 度為基準 ± 90 度，即可找出對應的 *PWM* 數值大小，再將數值從 Arduino 傳送至飛控板，以達成飛行器姿態控制。


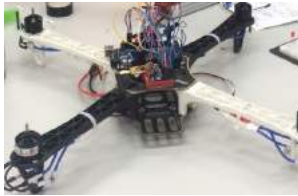


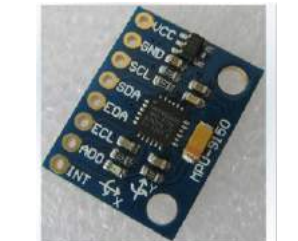






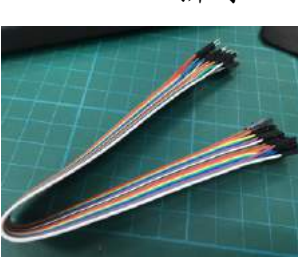
第五章 實驗

本章將介紹我們使用的實驗設備與姿態解算和障礙物偵測實驗，以及避障分析與姿態控制實驗，最後將結果展示並與分析。

5.1 實驗設備介紹

實驗設備簡表，如表 5.1 所示，主要實驗在六軸飛行器上，如圖 5.1 所示，四軸飛行器、飛行器測試用重物是作姿態控制實驗時測試的設備，而超音波測距的平台是由 3D 印表機列印出來的 PLA 素材，所有模組與 Arduino 的排線與洞洞板皆是用焊接工具與焊錫完成接線，電池以及遙控器是供飛行器飛行的設備。

表 5.1. 實驗設備簡表

六軸飛行器 	四軸飛行器 	ArduinoMega2560 	GPS NEO 6M V2 
MPU9250 IMU 	飛行測試用重物 	超音波測距模組 	銲接相關工具 
螺絲起子組 	3D 印表機 	鋰聚合物電池 	Arduino 排線 

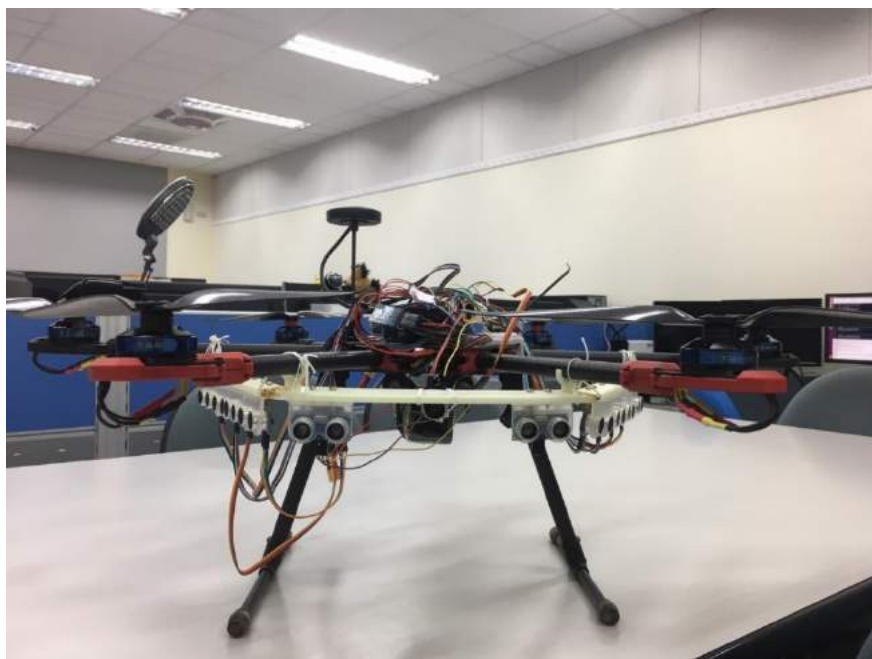


圖 5.1. 實驗障礙物偵測與避障的六軸飛行器。

九軸姿態模組：本論文使用 MPU-9250 模組，其包含三軸陀螺儀,三軸加速度計和三軸磁場感測器。

GPS 模組：本論文使用 GPS NEO-6M V2 是一款高性能 GPS 定位模組。該模組採用 U-BLOX NEO-6M 模組，模組自帶高性能無源陶瓷天線，並自帶可充電後備電池（支援溫啟動或暖開機，後備電池在主電源斷電後，可以維持半小時左右的 GPS 接收資料保存）。

超音波測距模組：本論文利用超音波模組作為多軸飛行器物件偵測與避障時的感測器，利用超音波感測器求得飛行器與周圍環境之間的距離，再將這些資訊作為飛行器做姿態調整，並依據感測器數據來建構出環境障礙物地圖，再將地圖資訊做進一步剖析以擬定飛行策略，使用的超音波感測器為 HC-SR04，探測的距離為 2cm-400cm，感應角度為 15 度，另外前方使用的超音波感測器為 MaxBotix 1240，感測的距離為 0 cm-765 cm，感應角度為 60 度。

遙控器控制系統：本論文使用 Futaba T10J 10-channel 遙控器，遙控器為美國手，左手上下為油門 (throttle)，左右為尾舵 (rudder)，右手上下為升降 (elevator)，左右為副翼 (aileron)，遙控器操作時發射器送出信號，當飛行器接收

到裝置於飛控板上的接收器訊號，訊號再傳遞給飛控板，操控遙控器的組合可改變飛行器的姿態變化，如圖 5.2 所示。



圖 5.2. 遙控器 Futaba T10。

飛控系統：本論文使用 Pixhawk 飛控板，飛控板有著 *ARM cortex M4* 處理器；立體空間的三軸 (axis) 是大家熟悉的 *XYZ* 軸，而飛機有專用的三軸名稱：*X* 軸稱為 *longitudinal* 軸 (縱軸)；*Y* 軸稱為 *lateral* 軸 (橫軸)；*Z* 軸稱為 *vertical* 軸 (垂直軸)，如果飛機相對 *longitudinal* 軸 (縱軸)，作機翼左右旋轉動作，稱為翻滾 (roll)。如果飛機相對 *lateral* 軸 (橫軸)，作機頭上下抬頭低頭動作，稱為俯仰 (pitch)。如果飛機相對 *vertical* 軸 (垂直軸)，作機頭左右偏擺動作，稱為偏擺 (yaw)，飛控系統根據資料處理後，產生這些動作已完成飛行動作，如圖 5.3 所示。



圖 5.3. Pixhawk 飛行控制板。

本研究使用 Arduino Mega 2560 基於 ATmega2560 處理器的微控制器板。它有 54 個數位輸入/輸出引腳 (其中 15 個可用作 *PWM* 輸出)、16 個模擬輸入、4 個 UART (硬體序列埠)，將感測器數值輸入 Arduino Mega 2560 微控制板，並將遙控接收器的信號攔截輸入微控制板，運行姿態解算與障礙物偵測和避障分析與姿態控制的演算法，再將輸出的 *PWM* 訊號送回遙控接收器使其控制飛行器。

5.2 實驗結果與展示

本論文目的是能夠讓飛行器能夠即時地達成障礙物偵測與避障，為了使飛行器飛行時偵測到障礙物時立即避障，穩定地流暢飛行避障姿態是很重要的，故我們做了一些實驗來驗證，例如飛行器在不同角度時，可即時使飛行器回歸平穩狀態的測試，與在實際飛行環境偵測到障礙物時避障的結果。

5.2.1 飛行器姿態解算實驗

利用飛行器姿態控制方式，測量姿態解算成果是否符合實際的角度反映量，如圖 5.4 所示，將 MPU9250 裝設在飛行器上，向右 roll 旋轉 θ 角，觀看 Arduino Serial port 的輸出是否與姿態解算出來的結果角度相同。

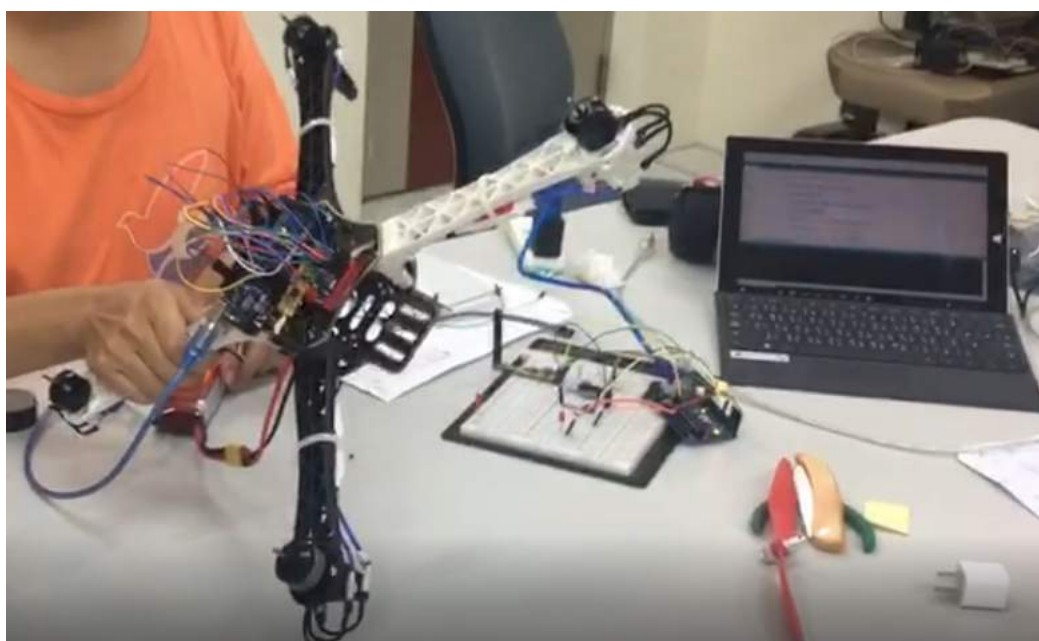


圖 5.4. 姿態解算實驗。

5.2.2 障礙物偵測與避障實驗

在飛行器的飛行過程，以飛行器前方偵測的環境，如圖 5.5 所示，飛行器往障礙物的方向移動，以障礙物與飛行器的距離遠近，樹為最近，電線桿為中距，房子為遠距。



圖 5.5. 障礙物環境與飛行器起飛前狀態。

5.2.3 飛行器避障分析實驗

飛行器會逐一將偵測環境轉換為飛行避障控制，如圖 5.6 所示，飛行器偵測到前方樹障礙物時，會比較飛行器方向寬區域，找到左方區域的障礙物密度比右方的障礙物密度還小，故飛行器方向判斷為往左側避障，如圖 5.7 所示，飛行器經過了樹之後判斷前方的電線桿與右側的房子障礙物，找到左方區域的障礙物密度比右方的障礙物密度還小，故飛行器方向判斷為往左側避障，飛行過程若無偵測到障礙物就仍往飛行器前方移動，如圖 5.8 所示，最後飛行器安全降落，完成飛行實時避障實驗，如圖 5.9 所示。



圖 5.6. 飛行器偵測到障礙物環境時避障反應前。



圖 5.7. 飛行器避開樹障礙物後往左側避障。



圖 5.8. 飛行器避開電線桿後往左側避障。



圖 5.9. 飛行器執行完避障後安全降落。

5.2.4 姿態控制實驗

在 Arduino 環境裡，Serial port 顯示輸入的遙控器 *PWM* 訊號數值，以及距離數值，和當下時間，如圖 5.10 所示，in_aileron 是遙控器 roll 方向的訊號數值，in_elevator 是遙控器 pitch 方向的訊號數值，out_aileron 為 Arduino 偵測完障礙物環境產生的 roll 方向 *PWM* 訊號給飛行控制板上的 *RC* 接收器，out_elevator 為 Arduino 偵測完障礙物環境產生的 pitch 方向 *PWM* 訊號給飛行控制板上的 *RC* 接收器，distance 為飛行器上超音波感測器偵測不同方向的距離，time 為當下時間。

```

INAIL=1033 INELE=1911 OUTAIL=127 OUTELE=93 distance1=300 distance2=295 distance3=300 distance4=300Time=12031
INAIL=1493 INELE=1462 OUTAIL=93 OUTELE=90 distance1=300 distance2=249 distance3=300 distance4=300Time=12631
INAIL=1453 INELE=1669 OUTAIL=90 OUTELE=111 distance1=300 distance2=249 distance3=300 distance4=300Time=12631
INAIL=1487 INELE=1708 OUTAIL=93 OUTELE=114 distance1=300 distance2=247 distance3=300 distance4=300Time=12632
INAIL=1448 INELE=2007 OUTAIL=89 OUTELE=144 distance1=300 distance2=250 distance3=300 distance4=300Time=12632
INAIL=1481 INELE=1991 OUTAIL=92 OUTELE=142 distance1=300 distance2=248 distance3=300 distance4=300Time=12632
INAIL=1010 INELE=2007 OUTAIL=47 OUTELE=144 distance1=300 distance2=247 distance3=300 distance4=300Time=12632
INAIL=1481 INELE=1981 OUTAIL=92 OUTELE=141 distance1=300 distance2=117 distance3=300 distance4=300Time=12632
INAIL=1951 INELE=1948 OUTAIL=138 OUTELE=138 distance1=300 distance2=139 distance3=300 distance4=300Time=12633
INAIL=2007 INELE=1636 OUTAIL=144 OUTELE=107 distance1=300 distance2=300 distance3=300 distance4=300Time=12633
INAIL=1616 INELE=1010 OUTAIL=105 OUTELE=47 distance1=300 distance2=300 distance3=300 distance4=300Time=12633
INAIL=1489 INELE=1010 OUTAIL=93 OUTELE=47 distance1=300 distance2=300 distance3=300 distance4=300Time=12633
INAIL=1010 INELE=1010 OUTAIL=47 OUTELE=47 distance1=300 distance2=300 distance3=300 distance4=300Time=12633
INAIL=1480 INELE=1010 OUTAIL=92 OUTELE=47 distance1=300 distance2=300 distance3=300 distance4=300Time=12634
INAIL=1010 INELE=1010 OUTAIL=47 OUTELE=47 distance1=300 distance2=248 distance3=300 distance4=300Time=12634
INAIL=1031 INELE=1472 OUTAIL=49 OUTELE=91 distance1=300 distance2=142 distance3=300 distance4=300Time=12634
INAIL=1913 INELE=1475 OUTAIL=134 OUTELE=92 distance1=300 distance2=149 distance3=300 distance4=300Time=12634
INAIL=1010 INELE=1288 OUTAIL=47 OUTELE=74 distance1=300 distance2=248 distance3=300 distance4=300Time=12635
INAIL=1010 INELE=1010 OUTAIL=47 OUTELE=47 distance1=300 distance2=248 distance3=300 distance4=300Time=12635
INAIL=1505 INELE=1630 OUTAIL=95 OUTELE=107 distance1=300 distance2=247 distance3=300 distance4=300Time=12635
INAIL=1487 INELE=1010 OUTAIL=93 OUTELE=47 distance1=300 distance2=247 distance3=300 distance4=251Time=12635
INAIL=1487 INELE=1010 OUTAIL=93 OUTELE=47 distance1=300 distance2=137 distance3=300 distance4=300Time=12635
INAIL=1010 INELE=1010 OUTAIL=47 OUTELE=47 distance1=300 distance2=151 distance3=300 distance4=300Time=12636
INAIL=1480 INELE=1010 OUTAIL=92 OUTELE=47 distance1=300 distance2=123 distance3=300 distance4=300Time=12636
INAIL=2007 INELE=1508 OUTAIL=144 OUTELE=95 distance1=300 distance2=154 distance3=282 distance4=300Time=12636
INAIL=1496 INELE=1010 OUTAIL=94 OUTELE=47 distance1=300 distance2=247 distance3=279 distance4=300Time=12637
INAIL=1010 INELE=1010 OUTAIL=47 OUTELE=47 distance1=300 distance2=247 distance3=300 distance4=249Time=12637
INAIL=1348 INELE=1490 OUTAIL=79 OUTELE=92 distance1=300 distance2=247 distance3=300 distance4=300Time=12637
INAIL=1480 INELE=1010 OUTAIL=92 OUTELE=47 distance1=300 distance2=247 distance3=300 distance4=300Time=12637

```

圖 5.10. Serial port 顯示姿態控制完輸出的 *PWM* 訊號。

測試姿態控制的設備為自組的四軸飛行器與飛行測試用重物，信號控制為利用 APC220 無線傳輸模組，將自己設定的 *PWM* 訊號模擬為遙控器訊號作飛行控制，並將飛行器綁上重物作起飛測試，是否能夠平穩起飛與達到飛行後穩定控制，如圖 5.11 所示，飛行器起飛後的穩定飛行控制，如圖 5.12 所示。飛行器起飛後的自調整平穩狀態，如圖 5.13 所示。



圖 5.11. 飛行器起飛前綁上重物的實驗環境。



圖 5.12. 飛行器起飛後的穩定飛行控制。



圖 5.13. 飛行器起飛後的自調整平穩狀態。

5.3 實驗結果分析

為了達到障礙物偵測與避障的穩定飛行控制，做了以下的實驗分析，飛行器的九軸姿態模組在沒經過姿態解算方法，純粹從感測器提取的 pitch 資料為藍色線條數值，因沒有將雜訊去除，數值在 47 到 55 度間晃動，經過姿態解算方法的穩定控制後的資料為橘色線條數值，數值皆落在 51-52 度間，如圖 5.14 所示。

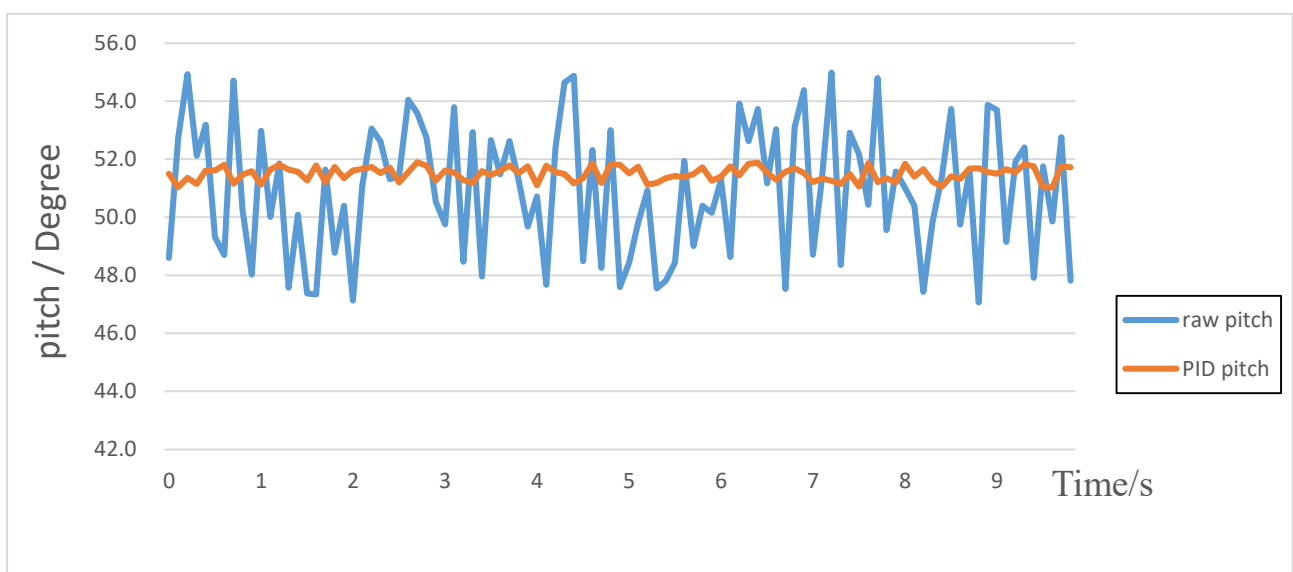


圖 5.14. 在震動下濾波效果顯示圖。

飛行器的九軸姿態模組在沒經過姿態解算方法，純粹從感測器提取的 yaw 資料為藍色線條數值，因沒有將磁場感測器的誤差和陀螺儀偏移問題去除，數值在 160 到 175 度間晃動，經過姿態解算方法的穩定控制後的資料為橘色線條數值，數值皆落在 168 度的小數點間，如圖 5.15 所示。

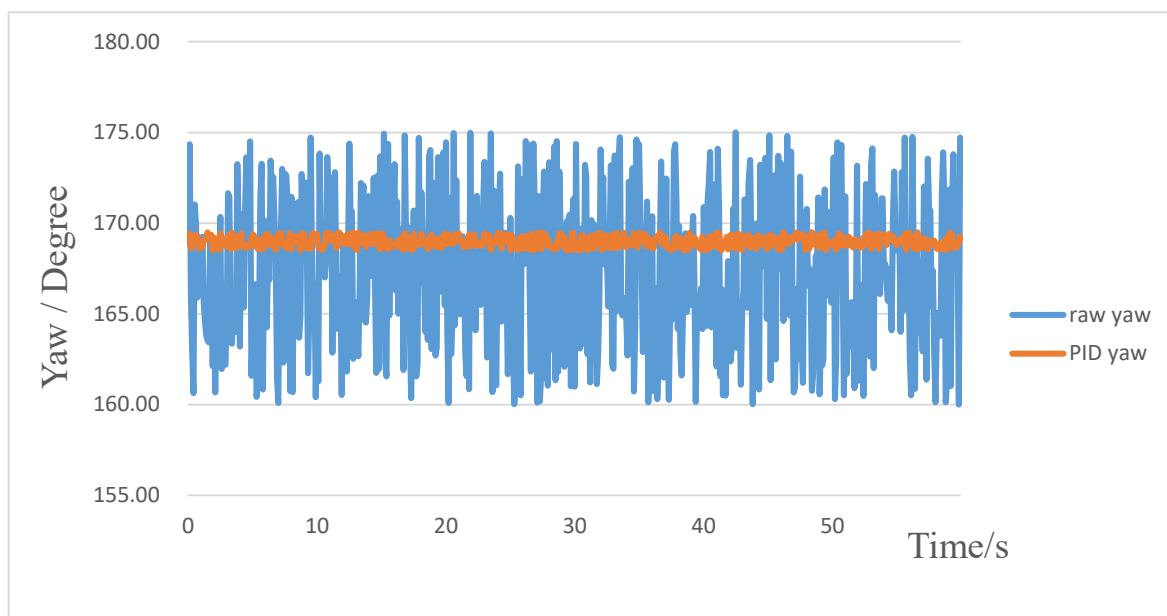


圖 5.15. 在姿態解算整合角度前磁場感測器朝向數值，與姿態解算後橘色的穩定航向數值。

飛行器經過姿態解算後，以 *SLAM* 方法做障礙物偵測，以 *VFH* 方法做避障分析，並利用穩定地飛行控制達成姿態的控制，表 5.2 為演算法的執行時間。

表 5.2. 障礙物偵測與避障演算法的執行時間。

<i>SLAM</i> 障礙物偵測	<i>VFH</i> 避障演算法
0.093 秒	0.223 秒

透過實驗可發現雖飛行器在電腦端執行演算法效率佳，但因 Arduino 微控制板處理器硬體的限制，使得障礙物偵測與避障處理效果並沒有達到完全即時性。

第六章 結論及未來展望

在本章中，我們將描述本研究的結論及未來展望。

6.1 結論

本論文研究利用九軸姿態模組得到三軸加速度、三軸角速度與三軸磁場值，計算出目前機體坐標系的俯仰角、滾動角、偏航角，再將 *RC* 接收器接收到的使用者操作遙控器的信號做數值穩定控制，得到的數值是誤差量，根據角度誤差量大小決定要多少修正量以供姿態控制時使用。

在障礙物偵測上使用同步定位與地圖構建 (*SLAM*) 的方法並結合超音波感測器，首先利用超音波感測器偵測到的數值來做區域地圖網格構建，並利用航向參考演算法找出飛行器的移動方向與距離，最後利用全域地圖網格構建將飛行器定位，得到飛行器位置與障礙物的環境地圖後做實時地 *VFH* 避障分析，在將飛行器的偵測環境先做區塊劃分，算出地圖裡每個網格距離障礙物可能值，找出每個區塊的障礙物密度，分析成平滑一為極性機率密度圖，將此設定門檻值若低於門檻值五個區塊即表示為寬區塊，若找不著寬區塊即執行飛行器懸停，若找著寬區塊後再選擇適當地候選區塊，並與飛行器前次的飛行方向找出合力方向，將合力方向角度轉換成 *PWM* 訊號，將 *Arduino* 微控制板偵測的環境產生的避障 *PWM* 訊號輸出至 *Pixhawk* 飛控板完成姿態避障。

最後透過姿態解算實驗，飛行器在其他角度時皆可依據 *IMU* 姿態的變化，利用角度誤差 *PID* 控制使飛行器得到合適地飛行角度，*Arduino* 回傳 *Pixhawk* 相對應地 *PWM* 訊號，使飛行器回歸平穩狀態。透過障礙物偵測與避障實驗，雖飛行器在電腦端執行演算法效率佳，但因 *Arduino* 微控制板處理器硬體的限制，使得障礙物偵測與避障處理效果並沒有達到完全即時性。

6.2 未來展望

我們的飛行器結合 Arduio 微控制板與超音波 *SLAM* 方法，將運算完的避障結果傳送回飛控板，因 Arduino 硬體上的限制，使得飛行器運算效能並不如預期，因此在未來希望可以運用於更先進的系統，針對運算效能做提升以加以改進，並結合視覺 *SLAM* 或雷達 *SLAM* 等其他偵測方法，藉由搭配相機的電腦視覺技術，達到更精準的偵測效果，和不同的路徑規劃演算法使避障策略更完善，並結合 Kim and Ostrowski [22] 的飛行器動力學模型使飛行器飛行與路徑規劃時更符合運動姿態。

參考文獻

- [1] F. Hoffmann, N. Goddemeier, and T. Bertram, “Attitude estimation and control of a quadcopter,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, Oct.18-22, 2010, pp.1072-1077.
- [2] A. Chan, S. Tan, and C. Kwek, “Sensor data fusion for attitude stabilization in a low cost quadrotor system,” in *Proc. IEEE 15th Int. Symp. on Consumer Electronics (ISCE)*, Singapore, Jun.14-17, 2011, pp.34-39.
- [3] D. Luo, F. Wang, B. Wang, and B. M. Chen, “Implementation of obstacle avoidance technique for in indoor coaxial rotorcraft with scanning laser range finder,” in *Proc. of the 31st Chinese Control Conf.*, Hefei, China, July 25-27, 2012, pp.5135-5140.
- [4] J. Wagster, M. Rose, H. Yaralian, and S. Bhandari, “Obstacle avoidance system for a quadrotor UAV,” in *Proc. of Conf. Infotech@Aerospace (American Institute of Aeronautics and Astronautics)*, Garden Grove, CA, Jun.19-21, 2012, pp.1-8.
- [5] N. Gageik, T. Müller, and S. Montenegro, “Obstacle detection and collision avoidance using ultrasonic distance sensors for an autonomous quadcopter,” in *Proc. Conf. on UAVweek 2012*, Siegen, Germany, Nov.20-21, 2012, pp.1-6.
- [6] N. Gageik, P. Benz, and S. Montenegro, “Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors,” *IEEE Journal of Access*, vol.3, 2015, pp.599-609.
- [7] A. Ferrick, J. Fish, E. Venator, and G. S. Lee, “UAV obstacle avoidance using image processing techniques,” in *Proc. IEEE Int. Conf. on Technologies for Practical Robot Applications (TePRA)*, Woburn, MA, Apr.23-24, 2012, pp.73-78.
- [8] M. Nieuwenhuisen, M.Schadler, and S. Behnke, ” Predictive potential field -based collision avoidance for multicopters,” in *Proc. Int. Archived of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Rostock, Germany, Sep.4-6, 2013, vol.XL-1/ W2, pp.293-298.

- [9] M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke, "Autonomous navigation for micro aerial vehicles in complex GNSS-denied environments," *Journal of Intelligent and Robotic Systems*, vol.84, no.1-4, Dec. 2016, pp 199-216.
- [10] T. Mori and S. Scherer, "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 6-10, 2013, pp.1750-1757.
- [11] S. Saha, A. Natraj, and S. Waharte, "A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment," in *Proc. IEEE Int. Conf. on Aerospace Electronics and Remote Sensing Technology (ICARES)*, Yogyakarta, Indonesia, Nov.13-14, 2014, pp.189-195.
- [12] C. Berger, P. Rudol, and M. Wzorek, "Evaluation of reactive obstacle avoidance algorithms for a quadcopter," in *Proc. IEEE Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, Phuket, Thailand, Nov.13-15, 2016, pp.1-6.
- [13] R. Kuc and B. Barshan, "Navigating vehicles through an unstructured environment with sonar," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, vol.3, May 14-19, 1989, pp.1422-1426.
- [14] M. M. Saleem, "An economic simultaneous localization and mapping system for remote mobile robot using sonar and an innovative AI algorithm," *Int. Journal of Future Computer and Communication (IJFCC)*, vol.2, no.2, Apr. 2013, pp.147-150.
- [15] V. Varveropoulos, *Robot Localization and Map Construction Using Sonar Data*, The Rossum Project: 2000. Available online: <http://www.rossum.sourceforge.net/papers/Localization>
- [16] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: modeling, estimation, and control of quadrotor," *IEEE Robot and Automation Magazine*, vol.19, is.3, pp.20-32, Sep. 2012.
- [17] H. B. Goodwin, "The haversine in nautical astronomy," *United States Naval*

Institute Proceedings, vol.36, no.3, 1910, pp.735-746.

- [18] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, Mar.25-28, 1985, pp.500-505.
- [19] J. Borenstein and Y. Koren, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, Apr.7-12, 1991, pp.1398-1404.
- [20] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots in cluttered environments¹," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, May 13-18, 1990, pp.572-577.
- [21] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Journal of Robotics and Automation*, vol.7, no.3, Jun.1991, pp.278-288.
- [22] J. Kim and J. P. Ostrowski, "Motion planning of an aerial robot using rapidly-exploring random trees with dynamic constraints," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA '03)*, Taipei, Taiwan, Sep.14-19, 2003, pp.2200-2205.