



國立台灣科技大學

機械工程系

實務專題成果報告

---

【多機器人協作收集之應用】

專 題 生：B11103241 陳俊言

B11103244 李家睿

指導教授：莊景歲 博士

中華民國 2025 年 12 月 1 日

# 目錄

摘要 .....	3
壹、 簡介 .....	4
貳、 實驗方法 .....	5
參、 實驗過程與結果 .....	18
肆、 總結檢討 .....	23
伍、 參考文獻 .....	24
陸、 附錄 .....	26

# 摘要

在許多應用情境中，單一移動機器人於收集任務常受限於作業範圍、執行效率與系統容錯能力，當任務數量增加或環境條件不理想時，整體任務完成時間與穩定性易受到影響。為提升系統效率與任務可靠度，多機器人協作系統逐漸成為可行的解決方向。本專題研究延伸自組長陳俊言於大三期間所完成之機器人架構，提出一套低成本之多機器人協作收集系統，旨在解決當初兩台機器人無法協作、執行任務會衝突等問題。系統以 Raspberry Pi 5 相機模組、ESP32-CAM 與中央筆記型電腦為核心，與自行設計之四輪移動機器人，完成多機器人撿拾網球之協作任務。系統採用兩階段式影像伺服控制策略，先由全域視角進行目標與機器人之位置與朝向估測，再透過車載鏡頭進行近距離控制，以確保任務執行之可靠性。在協作策略方面，本研究以預估抵達時間 ETA (Estimated Time of Arrival) 作為任務時間成本，結合匈牙利演算法進行多機器人任務指派。實驗部分透過固定目標配置，分別比較單機器人與多機器人以及 ETA 算法有無考慮機器人轉角的協作情境下之任務完成時間。實驗結果顯示，多機器人協作架構相較於單機器人系統能有效提升整體任務效率，ETA 算法加入轉角同樣能有效提升任務效率，並證實所提出之低成本多機器人協作系統於實際應用上的可行性與效益。

關鍵字：PID 控制、前饋控制、視覺伺服、匈牙利演算法、物件辨識、四輪驅動車、機械手臂、迴歸分析、多機器人

# 壹、簡介

## 1.1 前言

近年來，多機器人系統（Multi-Robot Systems, MRS）已廣泛應用於撿拾、搬運與物料收集等任務中[1, 2]，透過多個移動機器人同時執行作業，以提升整體作業效率與系統彈性。相較於單一機器人系統，多機器人協作能有效分散任務負載，並在大規模或高密度任務情境中展現較佳的時間效率與系統擴展性，因此逐漸成為智慧製造與自動化領域的重要研究方向[3]。

在實際工業應用中，常用自主移動機器人（Autonomous Mobile Robot, AMR）完成撿拾、搬運、組裝等任務。其中四輪移動底盤結合機械手臂之機器人架構，稱為移動式機械手臂(Mobile Manipulator, MM)，如 KUKA 的 KMR iisy (圖 1.1)[4]。此類系統結合移動平台之機動性與機械手臂之操作能力，能在不同工作區域中執行多樣化任務[5]。



圖 1.1 KMR iisy 實體圖

然而，多機器人移動操作系統在實際部署時，仍須同時考量系統成本、運算資源配置與協作效率等因素。特別是在低成本或教學型平台上，若於每一移動操作機器人上配置高效能板載電腦、多種感測器，將顯著提高系統建置成本與複雜度。因此，如何在有限硬體資源條件下，透過適當的感知與定位機制，並結合有效的多機器人協作策略，使系統同時兼顧穩定性與實作可行性，仍為多機器人系統研究中一項重要課題[6, 7]。

## 1.2 研究動機與目的

在成本受限、無高效能板載電腦之移動機器人平台上，如何同時兼顧定位、任務分配與即時控制更顯困難。現有研究中，許多方法仰賴高效能運算平台結合多種感測器，如 Jetson Nano、LiDAR、RGB-D 相機等，進行全域路徑規劃與協作決策[8, 9]，但此類系統通常伴隨較高的硬體成本與系統複雜度。

為降低硬體門檻，室內場域常見的固定式監視器或頂視相機可作為環境感知來源，提供穩定全域視角定位資訊，並將影像處理與決策運算集中於中央運算單元，減少每台機器人對昂貴感測器與高算力板載電腦的依賴。相關研究亦指出，透過室內固定遠端監視相機協助移動機器人定位，可降低機器人端定位設備需求，具備成本導向與可部署性優勢[10]。

另一方面，多機器人系統的效益仰賴適當的任務分配與協調機制；若缺乏分工策略，多機器人反而可能出現重複執行、任務衝突，使整體效率不升反降。基於此，本研究延伸自過去所完成之機器人架構[11]，研究目標聚焦於解決當時兩台機器人同場合作易互相干擾與衝突之問題，使用 Raspberry Pi 5 相機模組，模擬室內監視相機畫面。導入中心式、階層化之協作架構，達成多機器人協作任務。

## 貳、實驗方法

本章節講述為達成多機器人協作收集之研究目的，針對過去已完成的系統架構所進行之改良與變更，並描述本研究之實驗方法與實作流程。主要分為硬體架構與軟體架構兩大部分。在硬體架構中，將說明實驗場地與硬體變更，以及四輪馬達控制器校正與參數調整流程。在軟體架構中，將描述全域定位流程、任務分配與通訊機制，包含兩段式視覺伺服控制架構。

### 2.1 硬體架構

在本專題中主要有兩項硬體變更，分別是在置物箱後增加貼有  $8 \times 8$  AprilTag 的平台，如圖 2.1.1。以及將四輪馬達皆換成帶有編碼器的直流馬達，並四輪都換成麥克納

姆輪，此項變更有效解決過去機器人因負載過重而”外八”的情形，並能進行更順暢的轉向動作。最後在側邊新增一塊 LM2596 降壓模組，穩壓避免馬達運轉不穩定。如圖 2.1.2。

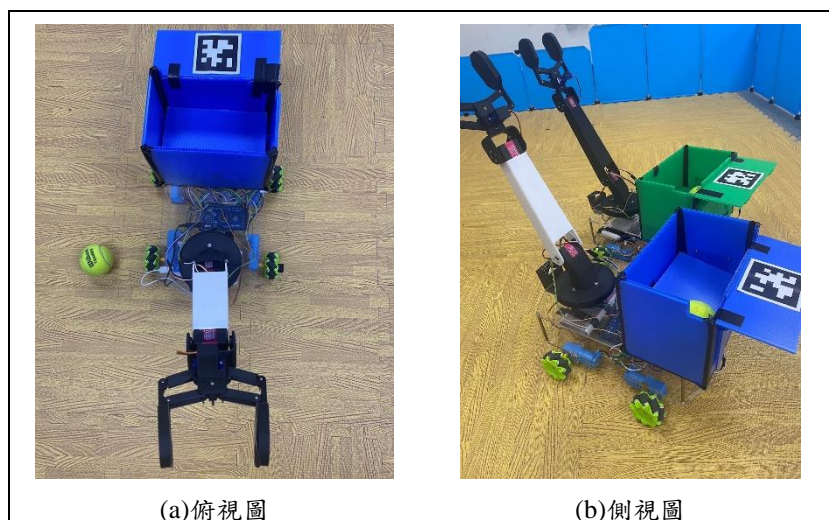


圖 2.1.1 機器人實體圖

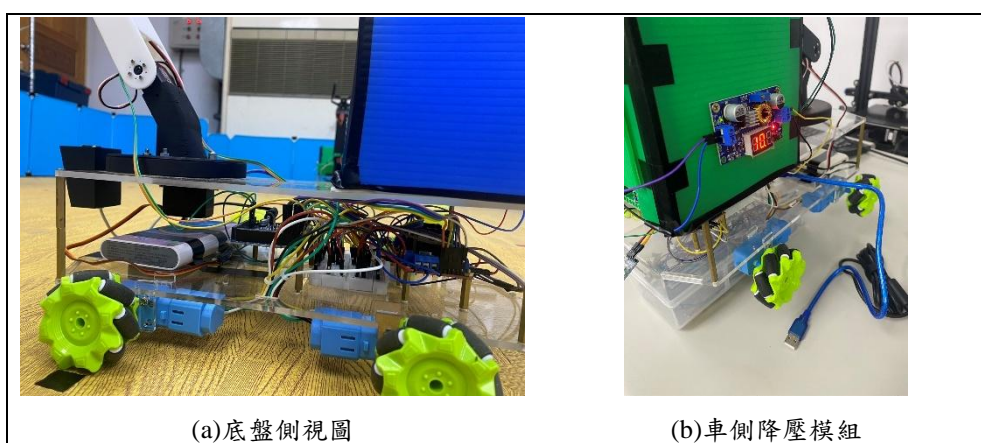


圖 2.1.2 機器人底盤

### 2.1.1 四輪馬達控制器設計

在本專題中機器人以頻繁起停、短距定位為主，因此快速到目標轉速不僅能迅速定位，更能減少行徑角度與距離誤差。過去專題架構直接調整 PWM 佔空比控制機器人移動，此方法無法得到馬達轉速，便無法精確控制機器人轉角與所需速度，會造成直行歪斜，無法控制轉角等問題。因為同一個馬達在相同控制器、參數、PWM 下，會受負載、供電等因素影響，而產生不同的實際轉速，本系統架構僅依靠頂拍相機作為第一階段的機器人定位與任務分配，若機器人無法正確轉彎與直行，會導致整體工作效率低下，因系統需要額外去校正馬達轉速不勻所造成的行走誤差。

為此，使機器人能被穩定控制行走，為最重要的第一步。本專題採用帶有編碼器的馬達，如圖 2.1.2，其能夠得到當前馬達轉速訊號，再配合比例積分 PI 控制器，回授控制馬達轉速，使其維持在目標轉速，如圖 2.1.3 示意。可看到馬達能在很短的時間內達到目標轉速並維持。

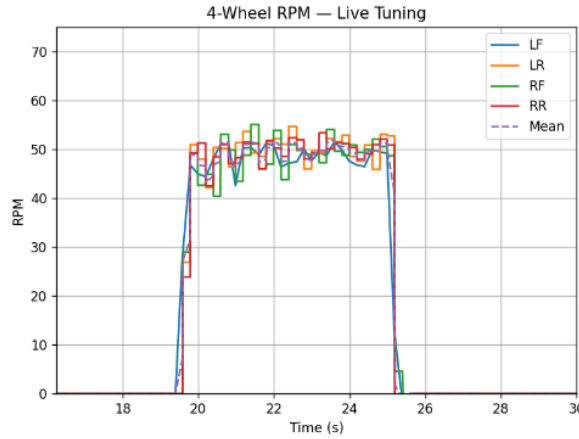


圖 2.1.3 目標 RPM 50 轉速圖

此時系統雖可以穩定維持目標轉速，但仍可看到四個馬達之轉速存在較大的差距，原因除了如本小節前面所述，也因本文所使用編碼器每圈脈衝數僅為 288，在低速情況下量測精度有限。因此，本專題採用具前饋(feedforward)之比例積分(PI)控制器，PI/PID 控制器在工業控制中被廣泛利用，而在應用中，除了調整 PI/PID 增益外，實現高效的目標點追蹤(set-point following task)也至關重要[12, 13]。由於其可主動補償馬達啟動死區(dead zone)，因此能使馬達更快速達到目標轉速，同時避免比例參數增益過大的情形[14]。在穩態轉速下，可將輸出 PWM 與目標 RPM 簡化為一線性關係[15]，如式 1 所示。

$$u(t) = d + \frac{\omega^*(t)}{K_{ff}} + K_p e(t) + K_i \int_0^t e(\tau) d\tau \quad (1)$$

其中， $u(t)$ 為控制輸出(PWM)， $d$ 為馬達啟動死區，主要用作克服馬達啟動靜摩擦， $\omega^*(t)$ 為目標轉速(RPM)， $K_{ff}$ 為前饋增益參數， $e(t)$ 為當前轉速誤差量， $K_p$ 與 $K_i$ 則分別為比例與積分增益。

根據式 1 與參考資料[16]之方法，於馬達穩態運轉下，紀錄當前 PWM-RPM 的對應關係，並透過調整取樣週期、加速間隔時間(確保進入穩態轉速)兩項指標，進行



PWM=0~255 的加速量測實驗，並繪製四個馬達的 PWM-RPM 關係圖，接著以最小平方方法(Least Squares Method, LSM)進行線性迴歸分析，此加速量測進行數十次。以估測四個馬達分別最佳之死區 $d$ 與前饋增益 $K_{ff}$ 。如圖 2.1.4 所示。其中 LF、LR、RF、RR 分別表示左前輪、左後輪、右前輪、右後輪。

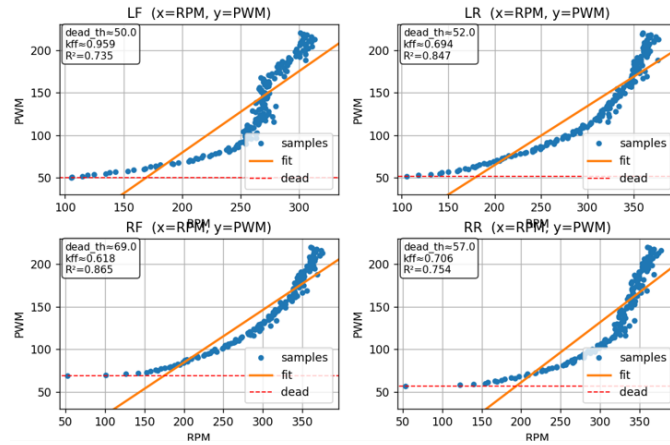


圖 2.1.4 四輪馬達 PWM-RPM 關係曲線與線性迴歸結果 (PWM=0~255)

於圖中可觀察到馬達在 PWM 約 150 以上區間，因受自身最高轉速限制，關係曲線明顯非線性，統計數十次實驗結果，四輪馬達轉速線性模型之決定係數(Coefficient of determination)[17]  $R^2$ 皆未超過0.9，說明單一線性模型並不適合用於 PWM 全區間0~255。此外，本專題中以短距離定位與收集為主，不需要高速運動。其原因於後續章節會詳細闡述。故以 PWM=0~150 區間再次進行加速量測與迴歸分析，如圖 2.1.5 示意。

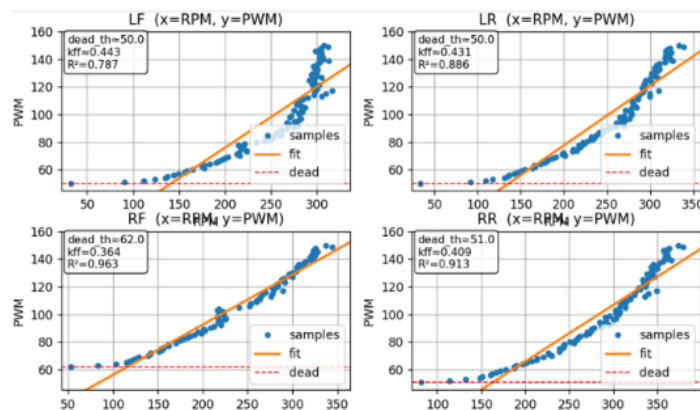


圖 2.1.5 四輪馬達 PWM-RPM 關係曲線與線性迴歸結果 (PWM=0~150)



由圖 2.1.5 可見，PWM=0~150 區間內，PWM-RPM 曲線更符合線性模型假設。其中 RF 馬達之 $R^2 = 0.963$ 。因此本專題設 $R^2 \geq 0.95$  為門檻，同樣進行數十次實驗分析，篩選並得出四輪馬達之最佳化參數，整理如表 1 所示。

表 1 四輪馬達前饋控制參數表

馬達 \ 參數	$d$	$K_{ff}$	$R^2$
LF	50	0.443	0.960
LR	50	0.431	0.981
RF	62	0.364	0.963
RR	51	0.409	0.955

使用此實驗結果參數，以及與圖 2.1.3 測試所用的比例積分增益。再次進行實際轉速測試，目標轉速同樣為 RPM=50，其結果如圖 2.1.6 所示。

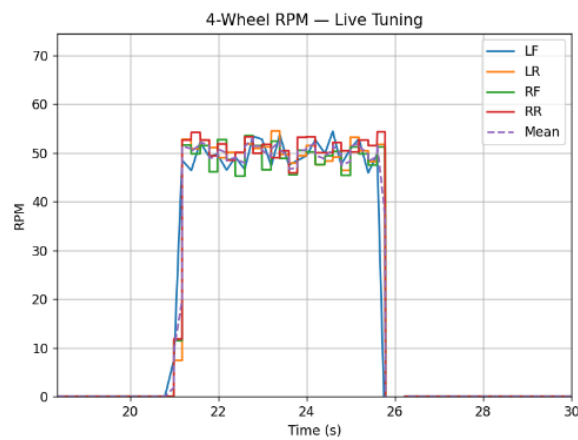


圖 2.1.6 目標 RPM 50 轉速圖(加入前饋)

比較圖 2.1.3 與圖 2.1.6 可觀察到，加入前饋控制後，馬達到達目標轉速之暫態時間縮短，且穩態誤差也有所降低。雖然須耗費時間根據馬達特性估測前饋參數，但可在不更換馬達的前提下提升轉速的控制精度。自此底盤馬達控制器設計完畢。

## 2.1.2 實驗場域與 Raspberry Pi 5 相機模組

本專題由安裝於天花板上的 Raspberry Pi 5 相機模組提供全域視角，其固定於實驗場地正上方，採用 IMX219 相機模組，相關鏡頭規格如[18]所述。實際視角如圖 2.1.7。

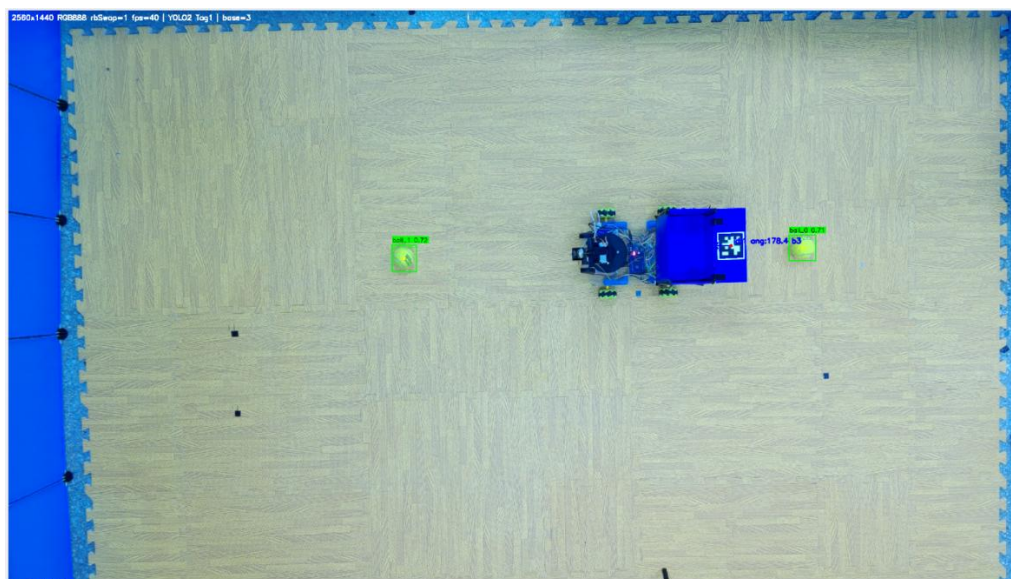


圖 2.1.7 真實實驗場域

實驗場域面積大小為  $287 \times 173$  (cm<sup>2</sup>)，畫面解析度為  $2560 \times 1440$  (pixels)，Raspberry Pi 5 會進行 YOLO 辨識網球與 AprilTag 辨識，再整理資訊後送至中央電腦，詳細過程將於下一章節敘述。

## 2.2 軟體架構

本章節為全文核心，闡述本專題基於過去專題成果所完成之系統架構，針對多機器人協作時會產生衝突等問題所進行的軟體架構改良。整體架構同樣以中央電腦(筆電)為主要決策與運算核心，整合頂拍相機所提供之全域資訊，建立機器人與目標物之全域座標與朝向，並使用匈牙利演算法進行任務分配邏輯，實現兩階段式定位。達成中心式、階層化的控制，整體系統流程與通訊如圖 2.2.1 所示。

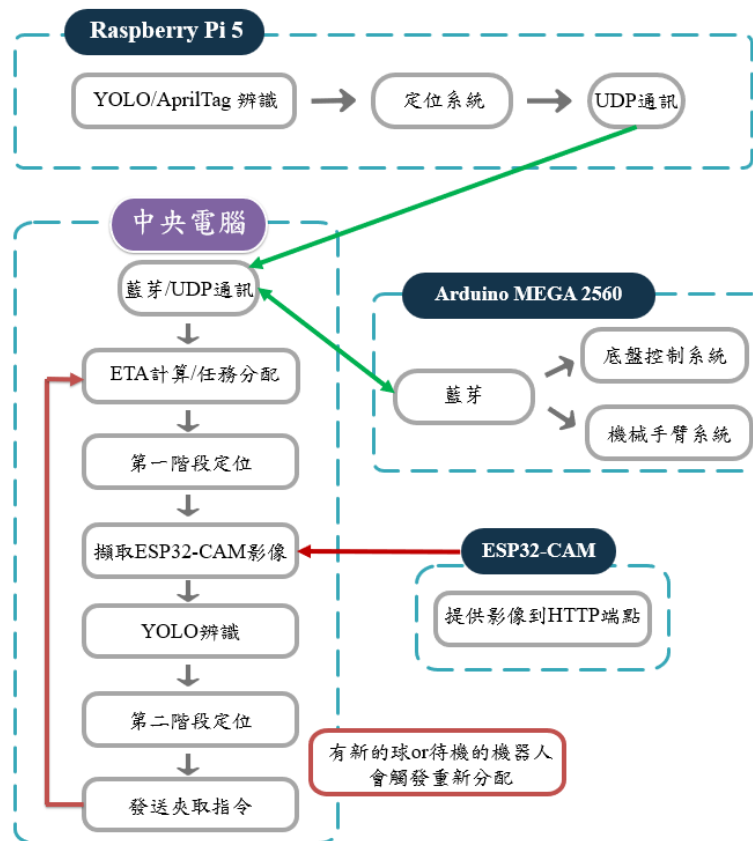


圖 2.2.1 系統流程與通訊圖

### 2.2.1 AprilTag 與全域座標定位

由圖 2.2.1 可知，系統運作時首先由 Raspberry Pi 5 相機擷取全域影像，並進行 YOLO 與 AprilTag 辨識，YOLO 所用模型為[11]之改良，將資料集改為 9000 張網球圖片，並加入約 1000 張純背景圖片，同樣進行訓練曲線分析。車載 ESP32-CAM 鏡頭畫面同樣以此模型進行辨識。最終辨識結果佳，分析過程與過去專題相似，因此將於本文略過。

除了目標物網球外，系統也須得知機器人的位置與姿態，最初是以機械手臂的外觀輪廓進行辨識，此方法雖可得知機器人位置，但姿態卻無法掌握，當機械手臂直行夾取時，更會使機器人變得不可控。因此本專題導入 AprilTag 作為視覺化標記(fiducial marker)，將其安裝於機器人後方的水平平台上，其可提供機器人位置與姿態，作為穩定的全域定位依據[19, 20]。AprilTag 實體圖如圖 2.2.2 所示，來源自[21]。

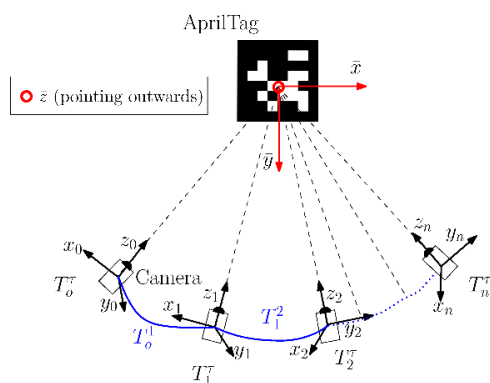


圖 2.2.2 AprilTag 的姿態與空間座標關係圖

由於 OpenCV 影像座標系為原點位於左上角，x 軸向右、y 軸向下，與常用的笛卡兒坐標系不同[21, 22]，若直接以影像座標推導朝向角度，將產生 y 軸符號反轉的問題。因此在機器人朝向與轉角問題時需謹慎處理此座標差異。並且 OpenCV 的色彩排序與常見 RGB 不同，其採用 BGR 排序，因此進行影像預處理，包括色彩空間轉換和通道重新排序(RGB/BGR swap)。再透過 AprilTag 中心與四個邊去渲染出機器人朝向箭頭，提供轉角、系統可視化等等用途，如圖 2.2.3 所示。

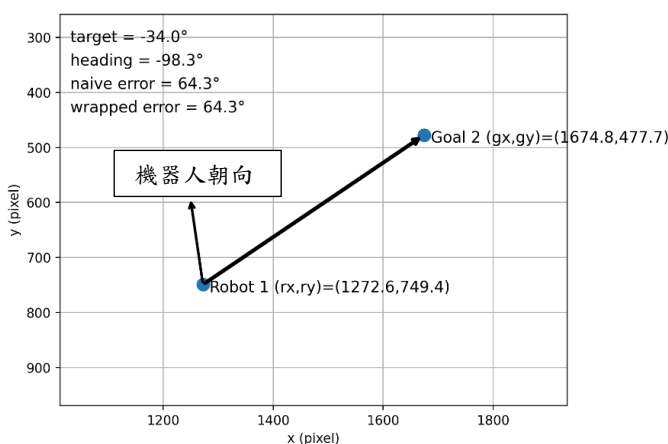


圖 2.2.3 基於像素點全域座標系的機器人與目標點朝向示意圖

於圖 2.2.3 可觀察到機器人朝向為 $-98.3^\circ$ ，對應到 $[0, 2\pi]$ 也就是 $261.7^\circ$ ，視覺上非笛卡爾坐標系下的 $98.3^\circ$ 。在處理轉角問題上會遇到超過 $180^\circ$ 的情形，因此在運算上導入了 atan2 與 wrap 的函數，將角度規一化在一定區間內 $(-\pi, \pi)$ ，使機器人能以最小的轉角運行[23]，提升工作效率。

## 2.2.2 UDP 通訊協定

由上一章節辨識與運算後，可得到場域中所有 AprilTag(機器人)與網球的中心點座標、角度，如圖 2.2.4 所示。其中包含時間戳記、機器人數量、目標物數量，本文所用的 AprilTag ID 為 1 與 168，以下內文稱為機器人 1 (R1)、機器人 2 (R2)。此外網球角度未在本文中被使用，因系統僅需透過座標與反正切函數計算轉角即可，但此角度資訊可快速判斷網球相對於影像畫面中心之方位，故保留。

```

=====
Timestamp: 1763133520.52 | Robots: 2 | Balls: 6
Robots Detected:
- ID: 1 | cx: 2054.1 | cy: 1197.1 | angle: 166.3
- ID: 168 | cx: 2112.6 | cy: 302.1 | angle: 152.3
Balls Detected:
- ID: ball_0 | cx: 1304.7 | cy: 708.2 | angle: 0.6
- ID: ball_1 | cx: 1056.3 | cy: 648.8 | angle: -5.4
- ID: ball_2 | cx: 1488.8 | cy: 1319.4 | angle: 5.1
- ID: ball_3 | cx: 762.3 | cy: 483.4 | angle: -12.6
- ID: ball_4 | cx: 866.8 | cy: 965.6 | angle: -10.0
- ID: ball_5 | cx: 1475.7 | cy: 419.4 | angle: 4.8
=====

```

圖 2.2.4 全域座標與角度

系統利用第一幀的中心點座標對網球進行排序，如圖 2.2.4 中的 ball\_0 ~ ball\_5，此排序設計為全程固定，避免每幀跳動誤差而造成重新排序的問題，本文稱其為”凍結機制”。再透過 UDP 通訊協定，以 JSON 格式播送於自訂 Port。使中央電腦能夠無線處理座標資訊。

### 2.2.3 ETA 計算與匈牙利演算任務分配

中央電腦以 UDP 接收訊息後，會以座標資訊利用畢氏定理與反正切函數，求出每台機器人與每顆球之間所需距離  $\Delta d$  與轉角  $\Delta \theta$ 。再用預估抵達時間(Estimated Time of Arrival, ETA)定義為任務所需時間成本  $C_{ETA}$ ，見式 2，其中  $V_{avg}$  為平均速度； $\omega_{avg}$  為平均角速度。

$$C_{ETA} = \frac{|\Delta d|}{V_{avg}} + \frac{|\Delta \theta|}{\omega_{avg}} \quad (2)$$

為使 ETA 具真實時間成本之量綱，其運算中單位一致性尤為重要。在影像視覺領域，光流法(Optical Flow)是一項核心技術，其可測出影像前後兩幀之間像素移動的瞬時速度與方向[24, 25]，在光流法運算上，需滿足假設條件:連續幀之間物體的像素強度(pixel

intensities)是不變的。並且光流以影像距離(pixels)除以光流大小(pixels/frame)，得到光流以(pixels per frame)表示，若再乘上相機的幀率(Frames Per Second, FPS)，即為 $\frac{\text{pixels}}{\text{frame}} \times \frac{\text{frame}}{\text{second}}$ 可得到像素每秒(pixels/s)。受光流法[26, 27]等重要文獻啟發，在本專題中不額外進行像素(pixels)與公尺(m)的關係轉換，其理由如下：

1. 本專題頂拍相機固定於室內天花板上，且固定幀率 FPS 為 30，解析度為  $2560 \times 1440$  (pixels)，像素強度不變假設條件成立。且 FPS 固定，容易計算像素每秒(pixels/s)。
2. 本專題不需要得到機器人絕對物理速度，僅需得到機器人的像素每秒，即可透過機器人與目標物的距離(pixels)計算出移動所需時間(s)，並建立成本矩陣。此方法比起由公尺和車體運動學換算得出之移動時間更為直覺且能避免換算中的誤差影響。

基於以上兩點理由，本文將移動所需距離 $\Delta d$ ，定義為影像平面上的位移(pixel，以下簡稱 px)，並令 $V_{avg}$ 為影像平面平均位移率，單位定義為(px/s)。

完成單位定義後，使機器人於固定轉速 RPM 進行直線行走、原地旋轉實驗並記錄行走時間內的座標、角度變化數據，最後去掉啟動前與停止後的量測延遲，以 LSM 迴歸分析。並考慮了移動中物體所產生之影像模糊的情形與參考 ShadowCam 等研究中對於相機運動與標記偵測率之討論[28]，將整體速度設在不會造成模糊的區間(RPM = 0~60)，此設計也可降低因辨識不佳而無法監管機器人的情形發生，實驗結果如圖 2.2.5。

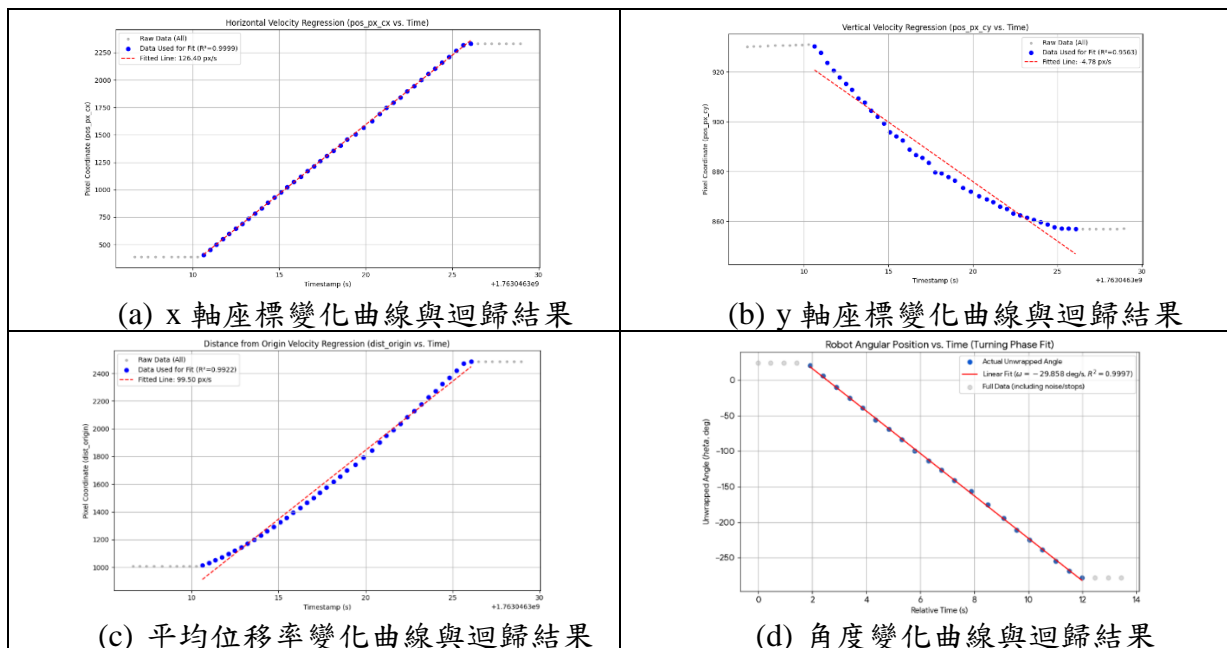


圖 2.2.5 平均位移率與平均角速度實驗(RPM = 50)





圖 2.2.6 直行量測實驗

實際實驗場景如圖 2.2.6，畫面左方黑色膠帶處為起點，右方黑色膠帶處為終點。對照圖 2.2.5(a)(b)與前面章節之坐標系定義，可驗證機器人穩定向右直行，但發生些微的角度偏差，以圖中起終點的像素變化數據計算，可得此偏差約為  $2.05^\circ$ ，但此直線移動距離幾乎涵蓋整個實驗場域，因此在控制計算上不考慮此移動誤差。

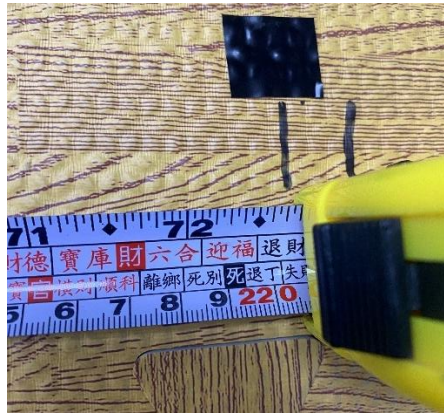


圖 2.2.7 直行量測實驗-真實距離(cm)

在迴歸分析出  $V_{avg}$  後，如圖 2.2.5(c)。可得出在  $RPM = 50$  下，機器人的  $V_{avg}$  為  $99.50(px/s)$ 。直線位移距離約為  $1500(px)$ ，算出行走時間約為  $15.07$  秒，對應時間戳記紀錄約為  $15.16$  秒。再如圖 2.2.7 所示，量測得真實距離為  $220.5$  (cm)，又因起始時是前輪對準膠帶，停止時是後輪對準膠帶(量測點)，因此加上兩輪間距  $19$ (cm)，共移動  $s = 239.5$ (cm)，再以式 3 計算移動所需時間，代入  $RPM=50$ 、車輪半徑  $r = 2.9$  (cm)

$$t = \frac{s}{v} = \frac{s}{r\omega} = \frac{s}{r \left( 2\pi \frac{RPM}{60} \right)} \quad (3)$$



計算後可得到理論行走時間約為 15.77 秒，與 15.07 秒之差異源自影像採樣誤差、地面摩擦力、行走偏差角度等等，但兩方法數值相近，自此證實本文所提出之量測方法有效。再進行 RPM=10、30、60 後，同樣進行 LSM 分析並平均，得出機器人的 $V_{avg}$ 與 $\omega_{avg}$ 如表 2，兩常數值與任務過程中的 $\Delta d$ 與 $\Delta \theta$ 用於全域的 ETA 計算。

表 2 ETA  $V_{avg}$ 與 $\omega_{avg}$ 參數表

類型	平均值	R <sup>2</sup>
$V_{avg}$	97.025 (px/s)	0.9879
$\omega_{avg}$	20.785 (deg/s)	0.9998

接著以實際測試 ETA 算法有無參考角度的影響，如圖 2.2.8 所示，在式 2 中有考慮轉角下，機器人會前進並夾取前方的球，再轉 180 度夾取後方的球。然而在沒有考慮轉角下，機器人會先轉 180 度夾取後方的球，再轉 180 度夾取前方的球。



圖 2.2.8 轉角、距離與 ETA 關係實驗

在這簡易兩球任務下，兩者差距 4 秒，且機器人需要多轉 180 度完成任務，因此考慮轉角可提升效率與降低機械損耗，在文末實驗結果將會比較 6 球任務下所需時間差異。最後中央電腦計算 ETA 後，會產生如表 3 之時間矩陣，並以匈牙利演算法分配這輪的最佳任務給機器人。

表 3 時間成本矩陣

標號	ball_0	ball_1	ball_2	ball_3	ball_4
R1	10.52	16.45	5.38	1.70	9.50
R2	6.24	13.99	6.42	13.71	1.20

從表 3 可知第一輪機器人 R1 將收集 ball\_3，；R2 則收集 ball\_4，同時本系統設計之「凍結機制」會自動排序以及取出網球座標作為本輪 ETA 計算所需，因此在任務分配

中，網球座標是固定的；但機器人位置則透過 AprilTag 動態追蹤，因此能夠依據即時的全域座標重新計算成本矩陣，並再次透過匈牙利演算法求得當前局部最佳任務分配[29, 30]，其數學式如式 4

$$C_{\text{best}} = \min_X \left\{ \sum_{(r_i, b_j) \in X} C_{\text{ETA}}(r_i, b_j) \right\} \quad (4)$$

其中配對的集合  $X \subset R \times B$ ， $(r_i, b_j)$  為機器人與網球的一組配對。

## 2.2.4 第一階段定位

第一階段定位係透過上一小節所敘，完成任務分配後，利用凍結機制之座標參數，配合一簡易 PI 控制器，使機器人能夠由任意位置、姿態，粗略定位至任務目標，再進行第二階段定位，如圖 2.2.9 所示。



圖 2.2.9 第一階段過程示意圖

第一階段定位其容許距離與角度，能夠使 ESP32-CAM 視角辨識到網球，順利進行第二階段定位。並且在圖 2.2.1 有提到觸發重新分配任務的情況，本系統設有保護機制，只有在第一階段完成定位前才會觸發重新分配，此設計旨在避免機器人臨時改變路徑而帶來的安全疑慮與錯誤發生。

## 2.2.5 第二階段定位

在第一階段定位會發送”APPROACH”表示完成，之後系統會發送 AT\_BALL，並自動開啟 EPS32-CAM 畫面進行交接，狀態切換如圖 2.2.10。

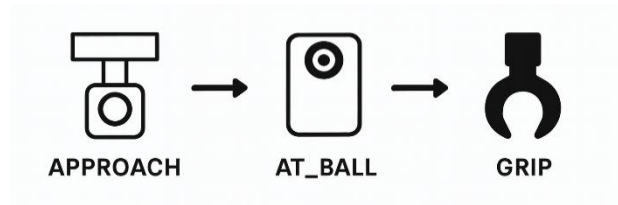


圖 2.2.10 定位流程示意圖

第二階段定位與[11]文中 4.3.2 節內的 PID 控制運算章節所述相同，透過相同的 PID 控制器，藉由 ESP32-CAM 畫面中心與目標物中心之像素點誤差，進行離散的 PID 控制。惟將系統輸出由直接輸出目標 PWM 更改成目標 RPM 值，並將 RPM 值設定在 0~60 區間，因此未來能夠配合麥克納姆輪的車體運動學模型[31, 32]，進行更進一步的分析，第二階段定位過程如圖，詳細原理將於本文略過。

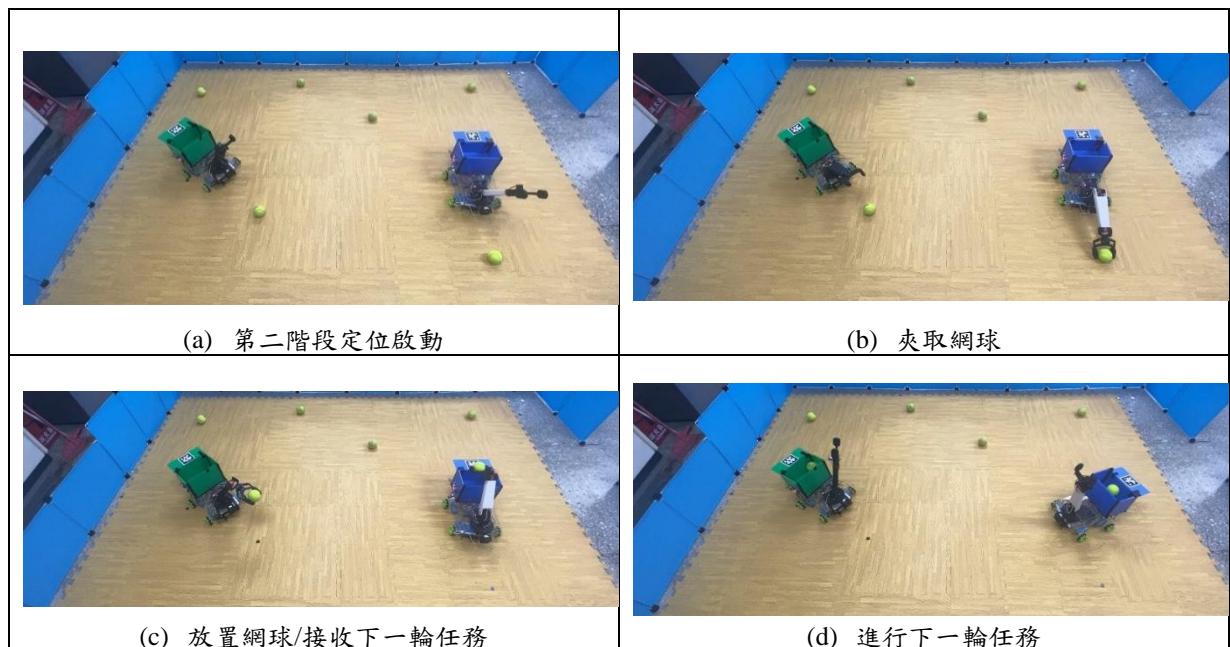


圖 2.2.11 第二階段定位實際流程圖

## 參、實驗過程與結果

最終實驗如圖 3.1 所示，任意擺放兩台機器人與六顆網球，並且將機器人起始位置、姿態固定後重複數次實驗，主要有 ETA 算法(有無考慮角度)、單與多機器人進行收集網球任務時，對總任務時間的影響比較。最後會展示本文提出之系統架構能夠處理動態環境(任意添加網球)。

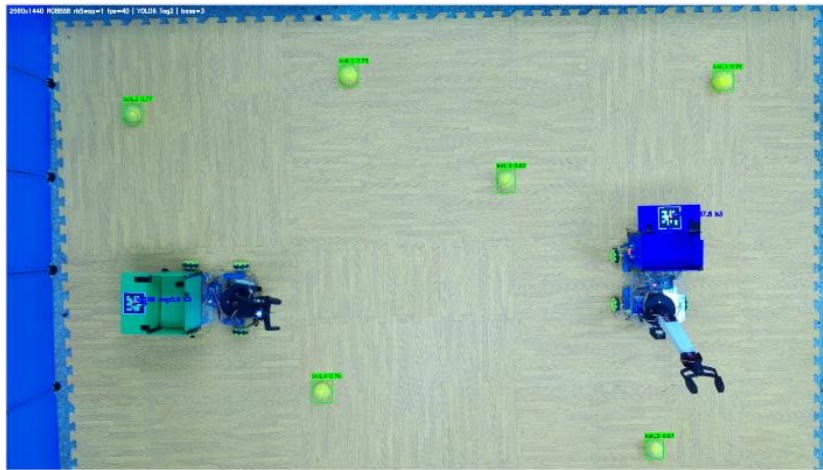


圖 3.1 實驗場域

### 3.1 ETA 算法實驗

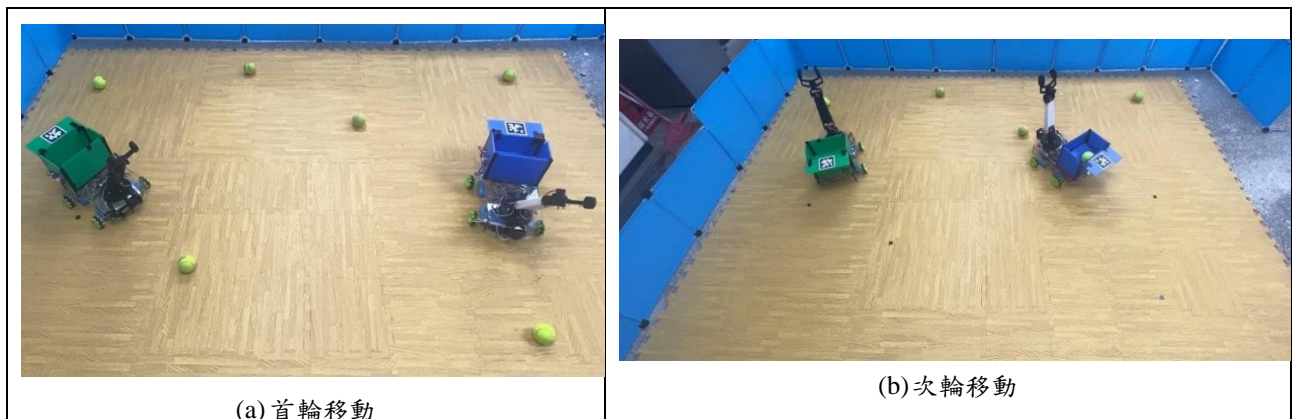
在 2.2.3 節已說明 ETA 有無考慮角度，對任務效率與機械損耗之影響，因此在本節將比較 6 球複雜任務下 ETA 算法對任務時間的影響。圖 3.1.1 為有考慮角度任務分配，包含預估時間計算。圖 3.1.2 則為實際移動情形，並可對照圖 3.1 之網球編號。

```
正在監聽 Pi 5 的 UDP 廣播，埠: 5566...
使用速度參數: v_linear=97.025 px/s, v_angular=20.785 deg/s
=====
Time: 1763555016.8 | 偵測到 2 台機器人, 6 顆球

--- ETA 成本矩陣 (秒) ---
      ball_0  ball_1  ball_2  ball_3  ball_4  ball_5
R1      10.87   8.55   16.38   5.89   9.41   1.60
R168    6.22   15.01   4.18   7.28   2.20   12.41

--- 最佳分配 (匈牙利演算法) ---
[分配] 機器人 ID 1 => ball_5
[總ETA]: 1.60 秒
[細節]: 旋轉 0.33 秒 (轉 6.8 度) + 移動 1.27 秒
[分配] 機器人 ID 168 => ball_4
[總ETA]: 2.20 秒
[細節]: 旋轉 1.37 秒 (轉 28.4 度) + 移動 0.83 秒
[總成本] 最小總時間: 3.80 秒
```

圖 3.1.1 有考慮角度-首輪任務分配







(c) 末輪移動

圖 3.1.2 ETA 有考慮角度-實際移動過程

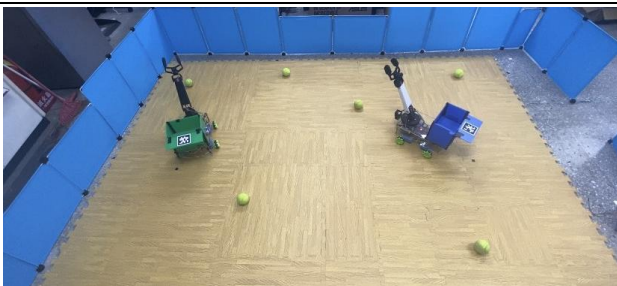
可看出首輪任務分配與實際移動符合，次輪與末輪移動則會根據機器人位置動態調整，驗證前一章節的系統設計可行性。圖 3.1.3 則為無考慮角度下的任務分配，圖為實際移動情形。

```
=====
Time: 1763554993.0 | 偵測到 2 台機器人, 6 顆球

--- ETA 成本矩陣 (秒) ---
      ball_0  ball_1  ball_2  ball_3  ball_4  ball_5
R1      5.12   1.17  11.28   0.66   6.15   1.27
R168    4.25  14.27   0.20   6.64   0.82  11.43

--- 最佳分配 (匈牙利演算法) ---
[分配] 機器人 ID 1 => ball_3
[總ETA]: 0.66 秒
[細節]: 旋轉 5.23 秒 (轉 108.8 度) + 移動 0.66 秒
[分配] 機器人 ID 168 => ball_2
[總ETA]: 0.20 秒
[細節]: 旋轉 3.98 秒 (轉 -82.8 度) + 移動 0.20 秒
[總成本] 最小總時間: 0.86 秒
```

圖 3.1.3 無考慮角度-首輪任務分配



(a) 首輪移動



(b) 次輪移動



(c) 末輪移動

圖 3.1.2 ETA 有考慮角度-實際移動過程

由此可明顯看出有無考慮角度下，每一輪的分配都不同，也能推論出無考慮角度下有多餘的轉彎，造成機械損耗。本輪實驗任務時間如表 4，可看出同樣任務下，單次實驗總任務時間差異為 12 秒。說明 ETA 算法考慮角度是能夠增加任務效率的。

表 4 ETA 算法對任務時間差異

	總任務時間 (秒)
有考慮角度	98
無考慮角度	110
<u>差異</u>	<u>12</u>

為避免單次第一階段、第二階段定位的耗時影響結果，本專題共進行三次相同實驗。表 5、表 6 為 ETA 算法有無考慮角度的三次實驗任務時間與平均結果，可看出兩者平均差異為 11.3 秒，由此可得 ETA 算法考慮角度能夠增加任務效率。

表 5 有考慮角度-三次任務時間

有考慮角度	總任務時間 (秒)
第一次	98
第二次	100
第三次	96
平均	98

表 6 無考慮角度-三次任務時間

無考慮角度	總任務時間 (秒)
第一次	110
第二次	106
第三次	112
平均	109.3

## 3.2 單與多機器人實驗

如摘要所敘，多機器人協作能提高任務效率，因此在同樣任務條件下進行單與多機器人執行任務實測，並採用考慮角度之 ETA。如圖 3.2.1 示意；表 7 比較。

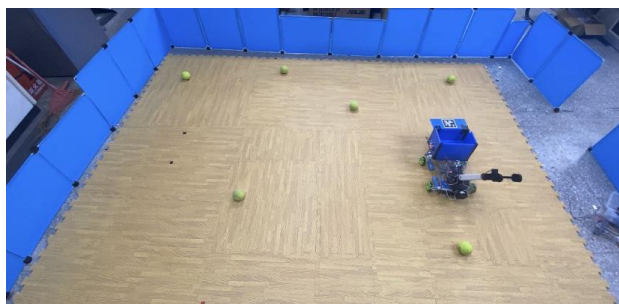


圖 3.2.1 僅 R1 執行任務

表 7 單與多機器人總任務時間比較

機器人數量	總任務時間 (秒)	時間相較倍率
R1 + R2	98	1.0
R1	226	2.31
R2	214	2.18

表 7 可看出單台機器人相較於兩台，所花費時間不只兩倍，而是 2.18~2.31 倍，因為多機器人能減少彼此的行走距離、轉角成本。同樣為避免單次第一階段、第二階段定位的耗時影響結果，接著進行三次相同實驗。由表 8、表 9 可得在相同任務條件下，兩台機器人比起單台機器人，任務時間能夠縮短 115.6~130.3 秒。

表 8 單機器人(R1)-三次任務時間

僅 R1	總任務時間 (秒)
第一次	226
第二次	231
第三次	228
平均	228.3

表 9 單機器人(R2)-三次任務時間

僅 R2	總任務時間 (秒)
第一次	214
第二次	217
第三次	210
平均	213.6



### 3.3 動態環境任務

最後實測系統在動態環境中之可行性，在本實驗過程中，任意添加網球至實驗場域，系統仍可順利完成應對所有臨時任務，此功能為整合”凍結機制”與重新分配任務之條件而成。如圖 3.3.1 所示，可看出有隨機任務出現時，機器人也能夠被分配任務並執行。

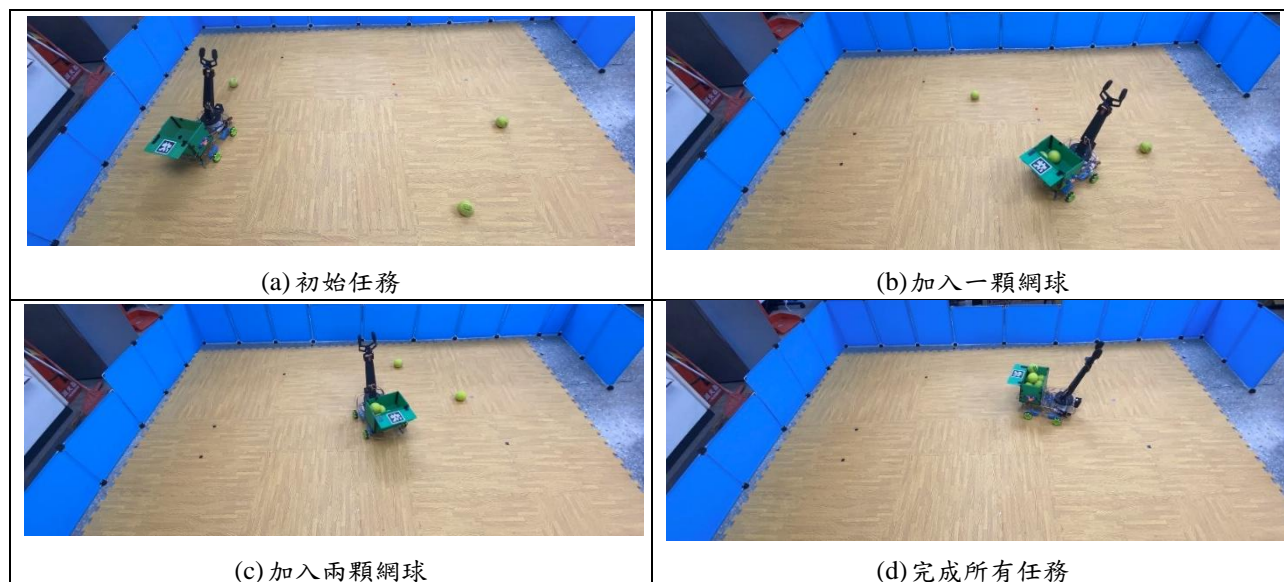


圖 3.3.1 實際動態任務執行過程

自此透過實驗證明本專題所提出架構，能使用低成本的多機器人基於兩階段式影像伺服控制達成定位、任務分配、協作等工作，完成網球收集任務，同時能夠兼顧效率與實用性。

## 肆、總結檢討

本專題延伸自既有之單機器人收集系統架構，進一步完成一套可實際運作之低成本多機器人協作收集系統。硬體面以自製四輪移動平台搭配機械手臂，並整合 Raspberry Pi 5 頂拍相機與車載 ESP32-CAM，建立「全域座標定位-任務分配-粗略定位-近距離定位-夾取」的兩段式視覺伺服流程。控制端則由馬達轉速回授出發，加入具前饋之 PI 控制器以提升起停與低速條件下的系統響應。降低因供電狀態與負載差異造成的行走誤差對系統效率的影響。

軟體端本研究採用中心式且階層化之架構，以中央電腦整合全域定位結果並執行任務指派。任務分配以 ETA 作為時間成本，結合匈牙利演算法完成多機器人之目標分派，避免多台機器人同時趨近相同目標而產生的衝突與重複行為。最後實驗結果顯示，相較單機器人作業，多機器人並行可有效提升整體任務效率；同時，將轉角因素納入 ETA 成本能使指派更貼近實際任務需求，進一步改善任務完成時間與系統運作流暢度，並驗證本研究架構於室內收集任務之可行性。

目前系統仍以 AprilTag 中心點為機器人之移動依據，並未考慮機器人的體積，此會導致兩台機器人若任務路徑上有所重疊，會造成碰撞，抑或機器人移動過程中會碰到其他網球，而導致系統需應對網球移動而重新計算一次，加重系統負擔。因此未來可利用中心點映射成一個面，進行路徑規劃，以及能在頂拍相機上加入對於障礙物等等的辨識，使系統能夠更加實用，應對更多實際場景。

## 伍、參考文獻

- [1] B. P. Gerkey and M. J. Mataric, "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [2] L. E. Parker, "Multiple Mobile Robot Systems," in *Springer Handbook of Robotics*, Springer, 2016.
- [3] S. Liu et al.,\*\*  
"A Survey on Multi-Robot Task Allocation,"  
*Robotics and Autonomous Systems*, vol. 152, 2022.
- [4] KUKA AG, "KMR iiwa" [官方網站](#), 2021.
- [5] A. Heins, M. Jakob, and A. P. Schoellig, "Mobile Manipulation in Unknown Environments with Differential Inverse Kinematics Control," in *2021 18th Conference on Robots and Vision (CRV)*, 2021, pp. 64–71, doi: 10.1109/CRV52889.2021.00017.
- [6] J. McLurkin, A. Lynch, S. Rixner, T. Barr, A. Chou, K. Foster, and S. Bilstein, "A low-cost multi-robot system for research, teaching, and outreach" *Proc. of the Tenth Int. Symp. on Distributed Autonomous Robotic Systems DARS-10*, November, pp. 597–609, 2010.
- [7] J. McLurkin, A. McMullen, N. Robbins, G. Habibi, A. Becker, and A. Chou, "A robot system design for low-cost multi-robot manipulation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, doi: 10.1109/ICRA.2014.6942668.
- [8] Y. Liu, S. Wang, Y. Xie, T. Xiong, and M. Wu, "A Review of Sensing Technologies for

- Indoor Autonomous Mobile Robots,” *Sensors*, vol. 24, no. 4, Art. no. 1222, 2024, doi: 10.3390/s24041222.
- [9] A. A. Adil, S. Sakhrieh, J. Mounsef, and N. Maalouf, “A multi-robot collaborative manipulation framework for dynamic and obstacle-dense environments: integration of deep learning for real-time task execution,” *Frontiers in Robotics and AI*, vol. 12, Art. no. 1585544, Jul. 2025, doi: 10.3389/frobt.2025.1585544.
- [10] J. H. Shim and Y. I. Cho, “A Mobile Robot Localization via Indoor Fixed Remote Surveillance Cameras,” *Sensors*, vol. 16, no. 2, Art. no. 195, Feb. 2016, doi: 10.3390/s16020195.
- [11] 陳俊言、鍾宏葦、劉育豪、謝秉緯，「多機器人協作收集之應用：專題報告」，專題報告，GitHub 專案文件，2025。[Online]. Available: [GitHub repository](#). Accessed: Dec. 24, 2025.
- [12] A. Visioli, “A new design for a PID plus feedforward controller,” *Journal of Process Control*, vol. 14, no. 4, pp. 457–463, Jun. 2004, doi: 10.1016/j.jprocont.2003.09.003.
- [13] J. Sánchez, A. Visioli, and S. Dormido, “A two-degree-of-freedom PI controller based on events,” *Journal of Process Control*, vol. 21, no. 4, pp. 639–651, Apr. 2011, doi: 10.1016/j.jprocont.2010.12.001.
- [14] S. J. Elliot and T. J. Sutton, “Performance of feedforward and feedback systems for active control” *IEEE Transactions on Speech and Audio Processing*, 1996.
- [15] 施予中。“直流馬達 PWM 驅動與摩擦效應之建模與辨識”。國立臺灣科技大學機械工程所碩士論文。2013 年。
- [16] WPILib Contributors, “Combining Feedforward and PID control”, 2025.
- [17] M. K. Chung, “R-squared (R<sup>2</sup>),” (sometimes called the *coefficient of determination*), in *Encyclopedia/Reference entry (PDF)*, Univ. of Wisconsin–Madison, [Online]. Available: <https://pages.stat.wisc.edu/~mchung/papers/R2.pdf>. Accessed: Dec. 25, 2025.
- [18] Waveshare, “Pi5-IMX219,” *Waveshare Wiki*, accessed: Dec. 25, 2025. [Online]. Available: <https://www.waveshare.com/wiki/Pi5-IMX219>.
- [19] J. Wang and E. Olson, “AprilTag 2: Efficient and robust fiducial detection,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2016, pp. 4193–4201.
- [20] OpenCV, “Detection of ArUco Markers,” 2023.
- [21] S. M. Abbas, S. Aslam, K. Berns, and A. Muhammad, “Analysis and improvements in AprilTag based state estimation,” *Sensors*, vol. 19, no. 24, Art. no. 5480, Dec. 2019.
- [22] NVIDIA, “Isaac ROS AprilTag,” 2023.
- [23] C. Samson, “Control of chained systems: Application to path following and time-varying point stabilization of mobile robots,” *IEEE Transactions on Automatic Control*, vol. 40,

no. 1, pp. 64–77, Jan. 1995.

- [24] Z. Teed and J. Deng, “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow,” in *Computer Vision – ECCV 2020*, LNCS vol. 12347, pp. 402–419, 2020, doi: 10.1007/978-3-030-58536-5\_24.
- [25] B. D. Lucas and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision,” in *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 1981, pp. 674–679.
- [26] Lee, D. N. (1976). A theory of visual control of braking based on information about time-to-collision.
- [27] F. Chaumette and S. Hutchinson, “Visual servo control. I. Basic approaches,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec. 2006, doi: 10.1109/MRA.2006.250573.
- [28] F. Naser, I. Gilitschenski, G. Rosman, A. Amini, F. Durand, A. Torralba, G. W. Wornell, W. T. Freeman, S. Karaman, and D. Rus, “ShadowCam: Real-time detection of moving obstacles behind a corner for autonomous vehicles” in *Proc. IEEE Int. Conf. Intelligent Transportation Systems (ITSC)*, 2018.
- [29] Z. Wang, Y. Wang, M. Chen, T. Yuan, H. Chen, and Y. Yang, “A task allocation strategy based on Hungarian algorithm in RoboCup rescue simulation” *Journal of Physics: Conference Series*, 2003.
- [30] H. W. Kuhn, “The Hungarian method for the assignment problem”, *Naval Research Logistics Quarterly*, 1955.
- [31] B. E. Ilon, “Wheels for a course stable selfpropelling vehicle movable in any desired direction on the ground or some other base,” U.S. Patent 3,876,255, Apr. 8, 1975.
- [32] A. Gferrer, “Geometry and kinematics of the Mecanum wheel,” *Computer Aided Geometric Design*, vol. 25, no. 9, pp. 784–791, 2008, doi: 10.1016/j.cagd.2008.07.008.

## 陸、附錄

### [任務分工]

隊員	分工項目
陳俊言 (組長)	(50%) 機構設計改良 (80%) Arduino 程式設計 (100%) RaspberryPi5 程式設計 (50%) YOLO 資料集收集/模型改良 (100%) Feed-Forward +PI 控制器設計實驗

	(100%) 影像伺服 PID 控制器設計實驗 (100%) 匈牙利演算法設計實驗 (100%) ETA(Estimated Time of Arrival)參數實驗 (90%) 專題報告
李家睿	(20%) Arduino 程式設計 (30%) Raspberry Pi 5 位置校正 (60%) AprilTag 校正 (50%) 機構設計改良 (20%)匈牙利演算法程式優化 (50%) YOLO 資料集收集/模型改良 (10%) 專題報告(協助製作系統流程圖) (50%) 成果錄影、材料採購

## [組員心得]

### 陳俊言：

在這次的專題實作中，我負責整套系統的發想、設計、實驗，除了機構設計之外我把所有的工作都包辦了，我也因此意識到機器人領域中機構設計與控制架構之間的合作重要性。

從編碼器馬達的轉速校正、雙鏡頭校正、UDP 通訊協定、透過訓練曲線的YOLOv8n 模型改良、匈牙利演算法、視覺伺服控制器設計等等，我花了非常大量的時間在閱讀文獻與做實驗驗證，這學期我每周大約耗費 70 小時在專題上，但我絲毫不覺得疲憊，因為我對這個領域越做越有興趣，雖然偶爾會有一整天都實驗失敗的情況，但我又會不停思考哪個環節有問題，並在隔天修改他，一旦我的想法被我驗證了，我就會非常非常開心，我想這就是做專題的核心所在，讓我們能從簡單的專案中培養思考問題、設計實驗解決的能力。

另一方面，這項專題橫跨了控制器設計、演算法應用、視覺伺服等等領域，因此經過這次專題的實作後我對機器人領域有更深刻的理解，我已更加知道自己有興趣的方向，日後我打算進行視覺伺服相關的研究。是這次的專提經驗讓我培養基礎的能力，

同時也讓我找到自己的短板，還有找到自己有興趣的領域。甚至同時培養團隊合作以及獨立研究的實力，這對我日後的發展都非常有幫助。

### 李家睿:

在這次專題中，因為我加入時隊友已經設計好穩定的第二階段定位跟夾取流程，因此一開始我對整體架構還不熟悉，需要花一些時間去理解前期所留下的程式碼與硬體設計。

但由於先前的架構沒有提供全域資訊，兩台機器人也不知道彼此的存在，因此會有搶球的情形，這樣雖然能夠完成任務，但整體效率不高且危險，更無法展現多機器人協作的真正價值。因此，我們著手建立一套新的集中式階層化的控制架構，讓整個系統能從單獨控制兩台器人，提升到多機器人共享資訊、分配任務的層級。

我主要負責 Raspberry Pi 5 的位置校正、AprilTag 的標與姿態修正，以及任務分配中成本矩陣的調整。AprilTag 的標定看似枯燥，但卻是整個全域定位能否穩定運作的關鍵。我透過調整相機內參、濾除角度跳動、校正偏移量等方法，使每台機器人的位置與角度都能更準確地被中央電腦掌握。同時我也針對匈牙利演算法重新整理資料格式，並加入凍結機制避免球的位置在影像中跳動造成錯誤分配，使整體任務分配更貼近真實場域需求。

總結來說，這次專題不僅讓我接觸到真正的系統整合，也讓我理解多機器人架構的核心精神:資訊共享、角色分工與有效協作。這樣的經驗對我而言非常重要，也讓我更確立自己在未來想繼續深耕定位、協作與演算法設計的方向。