

第一章 緒論

1.1 研究動機

近年來，隨著機器人技術的進步，相較於早期單一機器人系統 (Single-robot System)，有關多機器人系統(Multi-robot System)的研究愈來愈受重視。造成這樣的現象的主要原因為：相較於單一機器人系統，多機器人系統由於藉由團體合作及協調的特性，較單一機器人擁有更多的錯誤容忍率、更多的彈性、更簡單的機器人設計及更好的性能展示等優勢。

由於多機器人系統講求的是團隊所展示的效能，故藉由合作補償的效果可以容許其中部分成員的錯誤判斷或是非預期之行動表現等現象，這是為何多機器人系統能擁有更高的錯誤容忍率的原因。且由於任務的執行有賴於群體的力量，故我們在個別機器人的設計上不須擁有太複雜的功能或者太昂貴的設備即可達成預期的任務。

在多機器人系統的研究中，衍生出了許多的相關主題，如多機器人合作定位[1]、多機器人環境探索[2][3]、多機器人清潔[4]及機器人足球[5]等，其中有關多機器人之協調控制在隊形維持的展示下有著以下的應用：

合作搬運—藉由機器人團隊形成的隊形，可協力搬運指定的物體，使其順利到達目的地。如三個機器人排成前一後二之三角形隊形合力推動夾在前後兩排當中的長方形箱子，使箱子能保持一定的姿態沿著預定的軌跡移動而不至於因為一些轉彎的過程讓機器人隊伍失去對箱子的控制，這點可以應用於家庭、倉庫及其他各種需要搬運功能的工作環境中[6]。

巡邏保全—藉由機器人團隊的自主移動，可以搜尋整個環境區域，當有入侵者進入巡邏區域時，機器人們可以形成如門字形的包圍隊形將其圍繞起來並監視他的行為。當發現入侵者有不法意圖的時候，機器人團隊可以選擇以目前之包圍隊形強制推動驅離入侵者或者是變換成線形隊形阻隔入侵者與重要區域間的通道，使其無法前往從事不法的行為[7][8]。

救災應用—讓機器人團隊深入災區，在人類或大型器械無法進入的斷垣殘壁裡進行搜索待援者的工作。在搜尋任務開始時，各機器人可以分散在各處搜索，以擴大搜索的範圍；當其中有任何一台機器人找到生還者時，可以以無線通訊方式尋求其他機器人協助，協力將受傷者搬動至安全的治療地點[9]。

以上應用皆仰賴於機器人彼此間的協調配合，即每個機器人必須選擇最適當的行為及準確地移動至自己最適當的位置才有辦法達成良好的任務執行效果。因此，只要在行為的切換與隊形的維持上發揮良好的設計的話，不需要價格昂貴及功能複雜的單一機器人，只要搭配一些配備單純且動作靈活的機器人們即可達成上述的實用功能，也因此基於行為模式之多機器人協調控制系統將有相當不錯的功能及經濟價值，這促使了本論文之研究動機。

1.2 相關研究回顧

有關多機器人隊形協調控制之研究早先在 1997 年日本的 Hiroaki Yamaguchi 的研究文獻[8]裡已可發現，他的研究主題是如何讓多個機器人們排成一個弧線的隊形來阻擋入侵者通過監視區域裡的走道(如圖 1-1)。他的做法是在走道的兩側各安置一個地標(landmark)來提供機器人作為定位以及隊形安排的參考點，然後再利用隊形向量(formation vector)來調整每個機器人間的相對位置，藉此形成預期的整體隊形，其中每一個機器人會受到左右互鄰的兩個機器人的位置來修正自己的相對位置及角度，而在隊伍兩端點的兩個機器人則各受到一個相鄰機器人及地標的位置修正之影響。

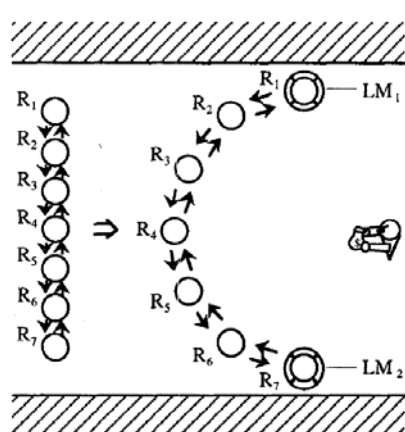


圖 1-1 Obstruction of way by a mobile robot group[8]

Jaydev P. Desai *et al.*[10][11]將研究主題鎖定在如何讓一個具有隊形的多機器人隊伍在碰到障礙物時能夠閃避，且閃避後再度維持一個固定的隊形不被打散，以繼續執行他們的任務。他們設計了 $l-\psi$ 和 $l-l$ 這兩個控制器來實現機器人和機器人間相對距離及角度的控制，然後再利用這兩個控制器間的切換(受障礙物的激發)自動改變隊伍的形狀來達成閃避障礙物的功能，在避障過程完成後，機器人團隊會再度回到原來的隊形以繼續執行之前未完成的任務(如圖 1-2)。

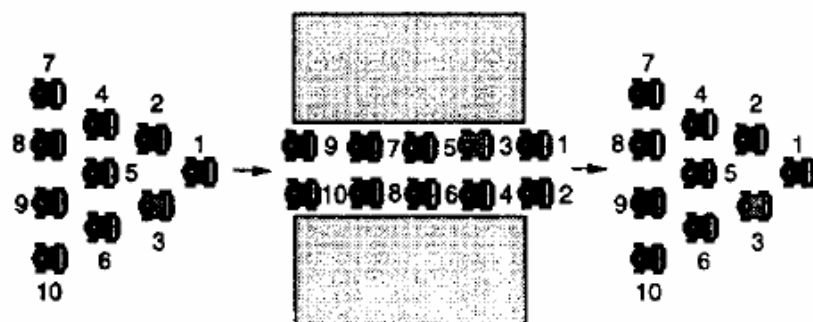
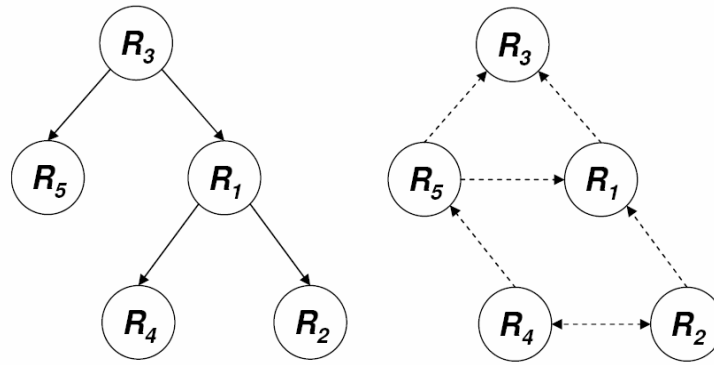


圖 1-2 Formation of robots changing shapes[11]

Guilherme A. S. Pereira *et al.*[12]改善了多機器人隊形若用 Leader-Follower 之方式組成的一項缺點：那就是隊伍領導者(The Leader)的行為不會被跟隨者們(The Followers)所影響。也就是說，當機器人隊伍中若有任何一個跟隨者發生意外或是速度減慢的話，隊伍領導者不會停下來或減緩速度來等它，亦不會做出任何其他反應的機制，這樣會造成隊形的不完整而影響隊伍的運作。因此他們的研究重心放在如何讓領導機器人的行為除了受到環境影響外，同時也會受到其他跟隨機器人們行為的限制。他們解決的方式是針對機器人的隊形結構畫出一張限制圖表(constraint graph，如圖 1-3(b))，而這張圖表規範了機器人們彼此間的限制關係，然後每個機器人再根據它所受到的限制來作隊形控制，這種方式可以保證機器人們間的關聯性，當隊伍中的某一台機器人速度在原本預期之外時，其他的機器人會配合它的速度以求達到隊形的完整。



(a) formation control graph (b) constraint graph

圖 1-3 Graph modeling for a group of 5 robots[12]

Tucker Balch *et al.* 在 1998-1999 年時提出了一套 Behavior-based 的多機器人隊形控制方法[13]，他們首先提出了機器人的四種基本行為設計：avoid-static-obstacle (閃避靜止障礙物)、avoid-robot (閃避機器人)、move-to-goal (移向目的地)以及 maintain-formation (維持隊形)，然後每種行為會有一個權重(weight)來代表每種行為間的相對重要性(此權重數可以根據任務需求的不同由設計者來自由調整)，而每個機器人的最終行為是由這四種基本行為透過各自的權重加權後再總和的高層融合行為(四種基本行為及其權重如表 1-1)。

在 maintain-formation (維持隊形)的行為設計上，一旦機器人得知所指定的隊形位置的話，馬達控制部分會產生一個移動向量(movement vector)指向這個位置。這個移動向量的方向會永遠朝此位置，但大小會根據他距離這個位置的遠近而有所不同，在這裡他們定義了三個區域用來界定現行機器人與指定位置間之距離，好用來做移動向量大小的計算(如圖 1-4)。在圖 1-4 中，Robot3 想要維持在 Robot1 左邊的位置，以這個指定位置為圓心的兩個同心圓區分了三個區域，在最外層的 Ballistic Zone，移動向量的大小被設為最大值，這是由於其距離指定隊形位置最遠的關係；在中央的 Controlled Zone，其移動向量大小由外邊界的最大值線性變化到內邊界的零；值得注意的是，只要機器人一旦處於 Dead Zone 的位置內的話，即認為此機器人已經抵達正確的隊形位置，此時移動向量的大小自動設為零，這個設計是為了容忍位置估測上所產生的誤差。

表 1-1 四種基本行為及其權重大小[13]

Parameter	Value	Units
avoid-static-obstacle		
gain	1.5	
sphere of influence	2.0	meters
minimum range	0.5	meters
avoid-robot		
gain	1.0	
sphere of influence	1.2	meters
minimum range	0.6	meters
move-to-goal		
gain	1.0	
maintain-formation		
gain	2.0	
desired spacing	1.5	meters
controlled zone radius	0.75	meters
dead zone radius	0.1	meters

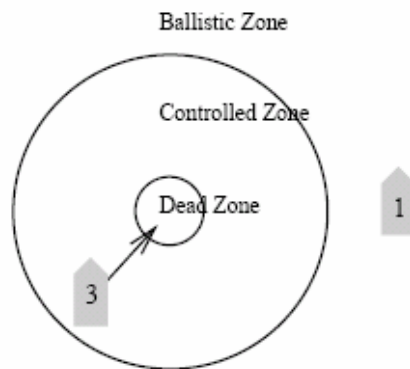


圖 1-4 為了計算移動向量大小所定義的三個區域[13]

在閃避障礙物的設計方面，移動向量的大小會根據距離障礙物距離遠近的不同而有所改變，但方向是恆朝遠離障礙物的方向。閃避機器人的設計方式大致上與閃避障礙物相同(將機器人視為障礙物)，不同的地方是參數的改變。移向目的地的設計方面，移動向量方向恆指向目的地方向，其大小為一個可調整的固定大小。由於以上四種基本行為的融合，使得多個機器人所組成的隊伍能夠在各種不同的指定隊形下順利前往目的地，同時在沿途中能夠處理閃避障礙物及維持隊形的問題(如圖 1-5)。

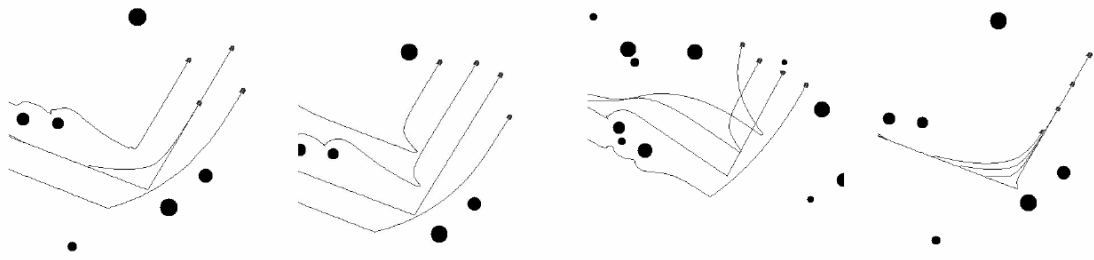


圖 1-5 在各種不同隊形下邊維持隊形邊閃避障礙物的機器人團隊[13]

文獻[14]在 2004 年提出了一套結合 Subsumption 架構(行為之選擇)與 Schema-based 架構(行為之融合)的混合式基於行為模式之控制架構，成功地讓一個機器人團隊能順利完成 Box-Pushing 之任務。但在其行為融合的比例權重上目前是由實驗經驗所得來的固定參數值，未來還仍有待透過學習(learning)能力來自動調整其權重值。

文獻[15]-[18]則是著重在如何讓機器人們用他們身上配備之全向式視覺感測器來達到多機器人隊形協調控制的目的。

1.3 問題描述



在現今之基於行為模式(Behavior-based)的機器人控制架構當中，Subsumption Architecture 由於其階層化的控制架構特色能使系統設計者可隨時增加高階的行為來抑制低層行為的輸出，而不需變動到原先低層行為的設計之優點，因而被廣泛應用在眾多機器人控制領域上。但其各行為間必須有一優先性(Priority)存在，這樣使得系統設計者必須事前考量到所有的環境情況而使得系統設計變得十分複雜；為此，本篇論文希望能提出一基於行為模式的多機器人協調架構，透過機器人間的溝通協調，建立一符合任務需求的機器人行為變換(Behavior Change)機制，使機器人團隊能更加適應各種不同的環境場合及狀態。本篇論文之目的乃提出一個以 Subsumption 架構為基礎之多機器人協調控制架構，將結合以多機器人間無線通訊所實現之 Action Selection 架構特色讓多機器人藉由靈活的行為變換來更容易地完成複雜的團隊任務。

另外在文獻[6]中，其作者欲在機器人缺乏通訊的情況下仍舊完成隊形的控制，其方法在於讓兩個 Follower Robot 分別單獨與 Leader Robot 做 $l - \psi$ 之相對位置的控制，但由

於兩個 Follower Robot 間沒有通訊的緣故，因此在某些情況下這兩個 Follower Robot 有可能會碰撞在一起，若要解決這樣的問題還必須在這兩個 Follower Robot 間增加一個保持距離的控制，但這樣一但當機器人系統數目變大時，其系統複雜度相對地會變得十分龐大。為此本論文希望能設計一個具有一般性(Generality)之基於行為模式的多機器人協調控制架構，並將這樣的架構套用在一個多機器人系統的所有機器人身上，即使當機器人的數目增加時，系統的複雜度也不會受到太大的改變。此外，即使當某一台機器人之行為不適當時，我希望仍然能透過機器人間的無線通訊能力使其回到應當的行為模式上，以增加任務執行(如合作搬運等)之完整度。此外，本文所提出之多機器人協調控制架構必須使機器人具有相當之自主能力(Autonomy)，即使在不同的環境及起始狀態下皆能順利完成任務，使任務的執行更加具有彈性(Flexibility)及效率(Efficiency)。

1.4 章節說明

本論文內容共分為七章，第一章為介紹相關研究背景及研究動機。第二章介紹基於行為模式之多機器人協調控制架構及其個別行為功能之設計。第三章介紹機器人視覺回授與全向式影像系統，包括色彩辨識處理以及相對位置之估測法則。第四章介紹機器人通訊與RF無線傳輸模組，並描述如何將其應用於本實驗室之三台實體機器人來達到溝通協調的目的。第五章為與第六章分別為三台機器人之隊形維持模擬結果與合作搬運實驗成果。第七章為結論與未來展望。

第二章 多機器人協調控制架構與行為設計

2.1 基於行為模式之基本機器人控制理論

有關於基於行為模式之機器人控制架構，以其行為輸出的選擇理論方式大致上可分為以下三類[19]：

(1) Subsumption Architecture — 行為優先權

Subsumption 控制架構(如圖 2-1)首先由 MIT 之 Rodney Brooks 在 1986 年所提出，其為一個階層式的系統架構。對一個以達成任務為目的行為來說，Subsumption 架構可以表示成許多單獨的行為層(Behavior layer)，而每一個行為層皆有其個別的目的(goal)及行為輸出，且這些所有的行為層是同時地在運作的。至於整個系統的最後行為輸出，由於各行為擁有不同的優先權(priority)，因此低階的行為在某些情況條件下會被高層的行為所抑制，造成最終系統輸出只會有其中某一個單一行為，而是哪個行為則要看當時的情況條件而定。總結來說，Subsumption 架構的優點在於增加高層行為(High-level behavior)並不會需要修改已經設計好的低階行為(Low-level behavior)，因為最後仍能夠透過抑制的效果來控制最後的輸出，對於系統未來的擴充性來說有很大的優勢。

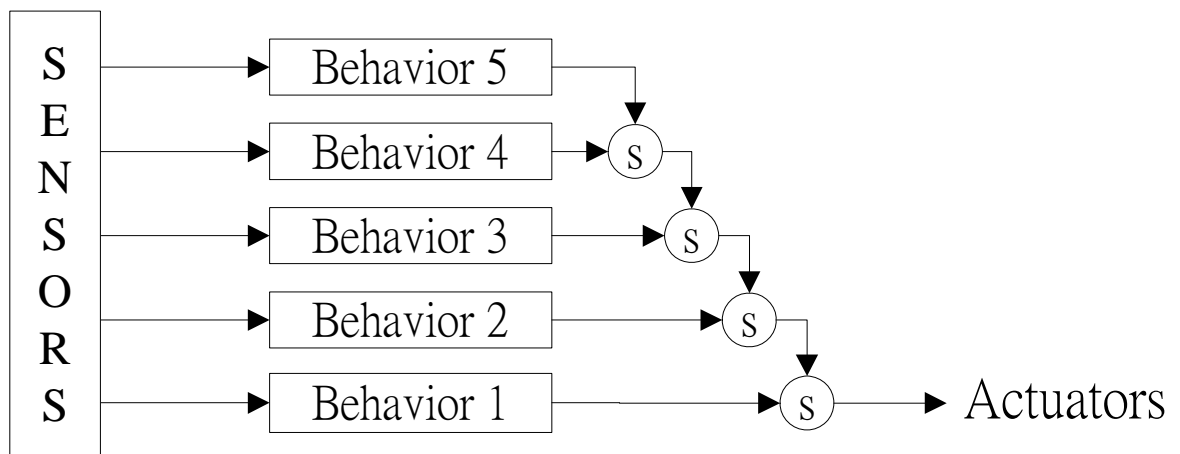


圖 2-1 Subsumption Architecture

(2) Schema-based Architecture — 行為融合

Schema-based 控制架構(如圖 2-2)首先由 Georgia Tech 之 Ronald Arkin 在 1980 年代末期所提出,其為一個行為並行式的系統架構。每一個行為皆可表示成一個單獨的 Motor Schema,擁有自己的目的及輸出(如控制馬達前進之速度大小與方向),其輸出即可表示為一個速度向量。在系統的輸出方面,採用的是向量總和(vector summation)的方式,即每一單獨行為之輸出會乘上一個屬於自己的行為權重(weight)後再加總起來(所有行為之權重數總和為 1),做為最後馬達驅動的輸出。因此在 Schema-based 架構下的整體行為輸出並不是其中某一單獨行為,而是多個行為融合後所得到的結果。

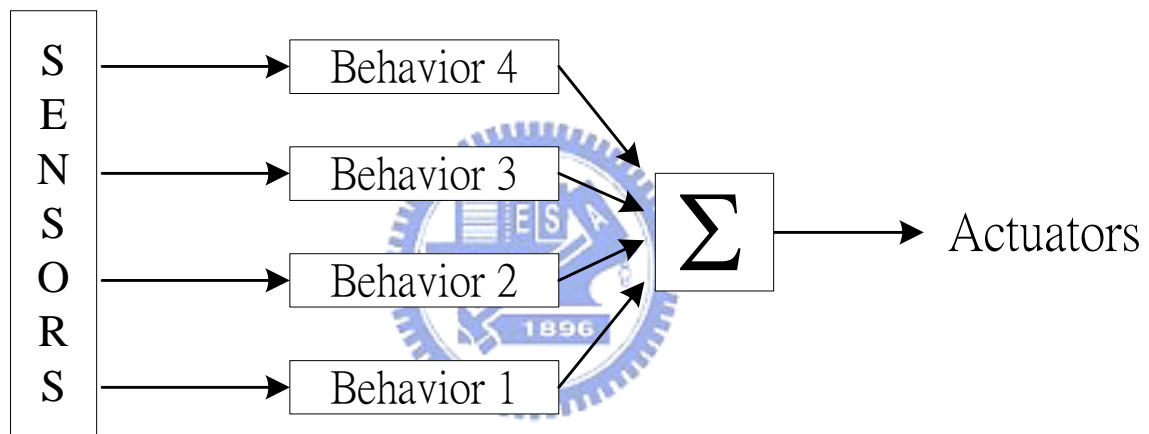


圖 2-2 Schema-based Architecture

(3) Action Selection — 行為競爭

Action Selection 之行為架構最早由 Pattie Maes 在 1980 年代末期所提出,其為一個行為競爭(behavior competition)的系統架構。與 Subsumption 架構類似的是其最後的系統行為輸出為其中的某一個單獨行為;但不同的是,在 Subsumption 架構各行為有不同的優先權(有高低階行為分別),但在 Action Selection 架構裡所有行為是平等競爭的,其每個行為皆有其屬於自己的動機(Motivation),而根據當時的環境情況會影響各行為的動機大小,最後由動機最大的那個行為做為最後的系統行為輸出。

2.2 基於行為模式之多機器人協調控制架構

◆ 設計目的及理念

為提昇機器人團隊在執行需有良好協調之團隊任務(如合作搬運或協力驅逐入侵者)的效率及提高各機器人的自主能力(Autonomy)，本論文因而將提出一個適應性基於行為模式之多機器人協調控制架構 (Adaptive Behavior-based Multi-robot Coordination Control Architecture)，其設計理念如下：

- (1) 為了提昇各機器人間之互動性(Interaction)，因此採用基於行為模式(Behavior-based)為基礎之控制架構，其中機器人之行為(Behavior)設計靈感乃由自然界中動物的群體社交模式啟發得來。
- (2) 為了增加機器人團隊在執行任務時的彈性(Feasibility)，因而衍生出了 Master/Slave 之模式切換設計。即在機器人團隊中，隊伍的領導者(Master)不恆為同一個，會藉由當時之環境狀態再來決定各機器人的角色分配，使其能適應各種不同的任務環境。
- (3) 為結合 Subsumption Architecture(系統擴充性)、Schema-based Architecture(行為模組化設計)及 Action Selection(行為適當性)此三架構之優點，因此本論文欲提出一混合式(Hybrid)之系統架構設計，能同時融入上述三架構的特色。
- (4) 在機器人的行為選擇方面，除了由其自身感測器所讀回來之環境資訊來判斷外；希望透過系統架構的運作，能藉由與其他機器人隊友間之溝通協調來促其選擇適當的任務行為，藉由合作補償來抑制其非預期的行動表現。

本論文之最終目的是希望將由上述四項設計理念所設計出來的多機器人協調控制系統架構確實應用在實際的多機器人系統上，使其展示合作協調的優越性，以突顯此一架構之價值。

◆ 系統架構及運作說明

本文基於上述設計理念所提出之適應性基於行為模式多機器人協調控制架構(如圖

2-3)主要以 Subsumption Architecture 為基礎，先將整個系統中的所有機器人行為依照任務目的之不同而區分為兩個行為層(Behavior Layer)：其底部為**探索任務層(Exploration Task Layer)**，包含了隨機探索(Wandering)、閃避障礙物(Obstacle Avoidance)、閃避機器人隊友(Inter-Robot Collision Avoidance)及趨近目標物(Target Approaching)等四種行為；而高層為**隊形任務層(Formation Task Layer)**，亦包含了前往目的地(Goal Approaching)、原地等待(Waiting)、靠近隊長(Master Approaching)及隊形維持(Formation Maintenance)等四種行為。

I. 探索任務層(Exploration Task Layer)：

探索任務層之任務目的在於讓機器人團隊裡的所有成員在任務環境下皆有自由探索並且尋找目標物(如欲搬運之物體或入侵者)的能力，並且在有碰撞危險時能達成自主的閃避行為，最終希望讓整個機器人團隊能快速地找到目標物的存在進而執行高層之合作協調任務。

在此行為層中，隨機探索(Wandering)之行為會藉由固定週期下之機器人前進角速度的隨機改變而給予其在環境下迴游的能力。在這同時，若其自身判斷出有碰撞危險的可能性發生時，則會受到較高層之物體閃避行為(包含對其它機器人及對障礙物之閃避行為)的抑制，優先執行閃避的動作。當機器人發現目標物在其視覺感測範圍內後，則趨近目標物(Target Approaching)之行為又將抑制其上述之較為底層的行為，在機器人的生存條件下逐漸靠近目標物直到逼近為止(以應用層面來看，相當於機器人發現疑似目標物後，進而靠近確認之)。

II. 隊形任務層(Formation Task Layer)：

隊形任務層之任務目的為當其中有某一台機器人已靠近目標物並成為 Leader Robot 後，開始透過無線通訊功能請其它的 Follower Robot 前來協助，在目標物處共同組成隊形後開始維持隊形並由 Leader Robot 帶隊前往目的地；即使當隊伍在途中遇到障礙物，也會各別執行閃避動作後再回到原先之預定隊形直至目的地為止。

在此行為層中，依據機器人在任務中擔任之角色不同又將其四種行為歸納在兩種模式——隊長模式(Master Mode)和隊員模式(Slave Mode)之下。在系統運作的一開始時，所有機器人的行為模式皆在探索任務層中之一般模式(General Mode)下。當機器人隊伍中有某一台機器人靠近目標物並確認它無法獨力搬運的同時便切換其行為模式至隊形任務層中的隊長模式(Master Mode)下，此時它的任務角色就變成整個團隊裡的 Leader Robot，並同時發送無線訊息請其它的機器人隊友們前來協助。當其它的機器人們接到這樣的求援命令同時，便將自己的行為模式切換至隊形任務層中的隊員模式(Slave Mode)下(任務角色相對變成隊伍裡的 Follower Robot)並執行靠近隊長(Master Approaching)的行為，其中隊長之座標位置同樣已透過無線功能傳送至每台 Follower Robot 中。

在 Follower Robot 在靠近隊長的過程裡，Leader Robot 會執行原地等待(Waiting)的行為，直到所有 Follower Robot 皆進入隊形準備位置為止。當任何一台 Follower Robot 靠近隊長到隊長已在其視覺感測範圍時，則隊形維持(Formation Maintenance)的行為便開始抑制靠近隊長之行為而被此 Follower Robot 所執行，然後此 Follower 便逐漸朝往它自己所屬的隊形位置前進並在隊形位置處原地轉正自己的朝向角與 Leader Robot 一致，在轉正完之後則在傳送無線訊息給 Leader Robot，表示自己已正式進入隊形位置。當 Leader Robot 接收到來自所有 Follower Robot 隊形就位訊息後，其前往目的地(Goal Approaching)的行為將抑制一切行為而被優先執行，此時整個機器人隊伍由 Leader Robot 領隊，其它的 Follower Robot 便以它做為隊形參考點(Formation Reference)，與它維持預定之相對距離及角度以達成隊形維持的目的。

◆ 系統功能描述

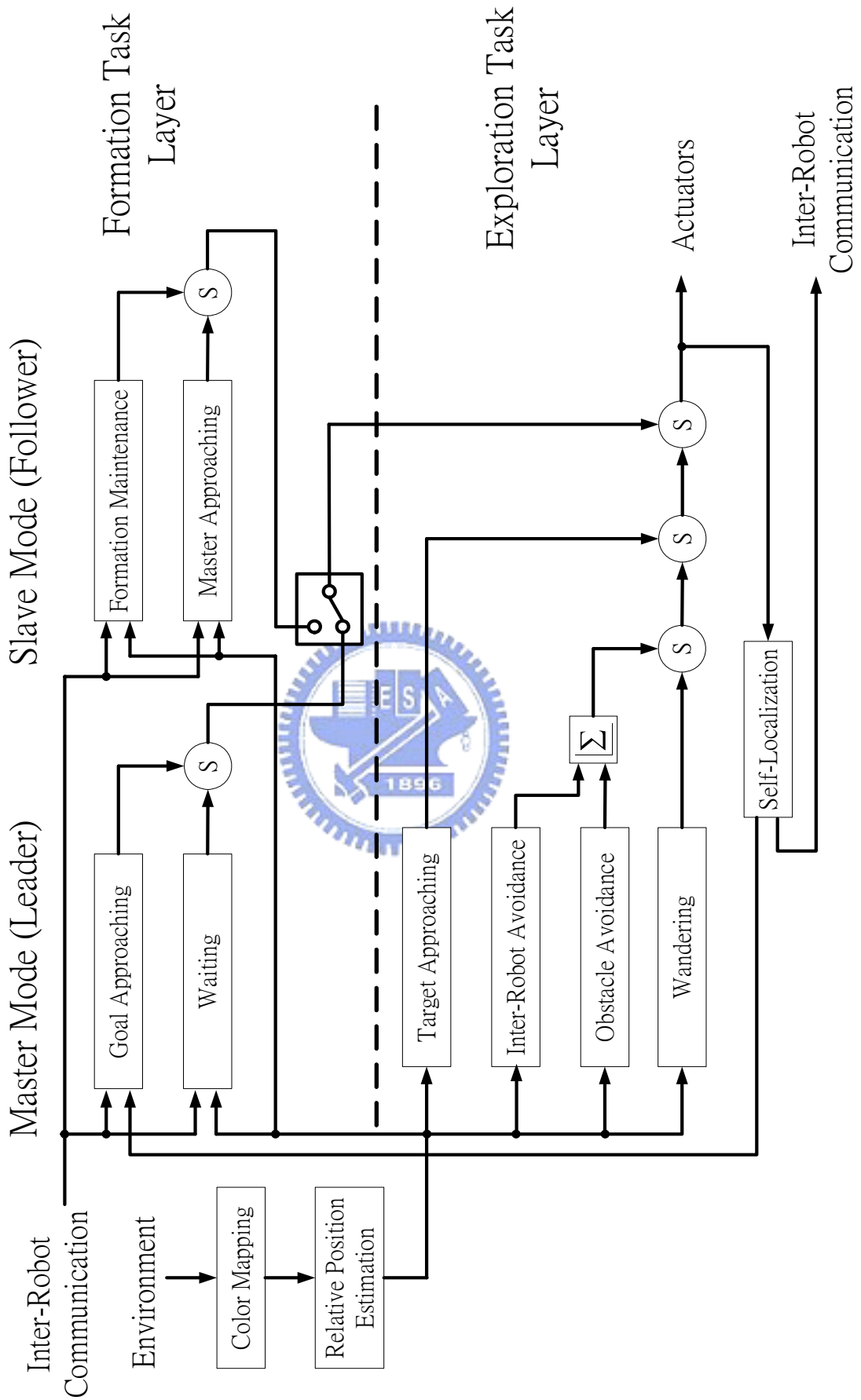
藉由此具有系統設計之一般性(Generality)特色的多機器人協調控制架構，可讓機器人們以更自主且有效率的合作方式來完成任務。在任務開始的同時，所有的機器人成員皆處於探索任務層的行為模式下以便其自由探索是否有疑似目標物的存在；一但當其中有任何一台發現後靠近並確認無法獨力搬運時，則自動切換至 Master 模式，同時以無線

通訊的方式請求其餘機器人隊友的援助。當其餘的機器人們在接收到請求援助命令後則進入 Slave 模式(即在整個環境中除了優先找到目標物的那台機器人角色為 Leader Robot 外，其餘的角色皆為 Follower Robot)，隨後靠近剛才發送求援命令之隊長(Leader Robot)並開始維持隊形直至目標物被合力推送到預定目的地為止。最終，整個任務過程會經過 (A) 尋找目標 (B) 隊形組成 (C) 隊形維持 三個階段而被整個機器人團隊藉由其通訊及協調能力來自主完成，預計此系統之功能將適用在多機器人圍獵(入侵者)或協力搬運(負傷待援者)等應用上。

◆ 系統特色及優點

總結本章所提出之適應性基於行為模式多機器人協調控制架構具有以下之特色：

- (1) 透過無線通訊能力，讓機器人們隨時了解彼此間之狀態及其在團隊任務中的角色定位，藉此來切換其高層之行為模式，藉此我們因而達成了隊形組成之目的。其中 Master 與 Slave 兩種模式之不定性切換更可讓機器人團隊適應各種不同的任務執行環境。
- (2) 此架構可適用在含不同種類機器人之多機器人團隊上，如在閃避障礙物的行為可藉由視覺感測系統或超音波感測器來達成；對於物體之識別可透過影像之顏色識別或 RF ID 辨識系統來實現，對於實際機器人硬體系統有相當大的容忍度(Tolerance)。
- (3) 系統架構設計之一般性(Generality)可套用在數量龐大的多機器人系統上，隨著機器人數量的增加，整體系統之複雜度並不會隨之提升。



2-3 Adaptive Behavior-based Multi-robot Coordination Control Architecture

2.3 個別行為功能設計

此節將描述機器人之個別行為功能的設計，其中包含了 (I) 探索任務層(Exploration Task Layer)中的四種行為 — 隨機探索、閃避障礙物、閃避機器人隊友和趨近目標物 及 (II) 隊形任務層(Formation Task Layer)中的四種行為 — 前往目的地、原地等待、靠近隊長和隊形維持。以下各小節將個別描述其行為功能的設計：

2.3.1 隨機探索 (Wandering)

隨機探索功能讓機器人於環境空間中能四處自由移動並探索是否有目標物(如入侵者等)的存在。其做法在於讓機器人在一段固定的時間週期內隨機選取一預設範圍內之角速度 ω (rad/s)，再輸出到馬達控制器中，如此機器人便會在環境中四處移動，以達到巡邏探索之目的。式(2-1)及式(2-2)為機器人隨機探索行為之運動控制式：

$$v = K_{wand} \quad (2-1)$$

$$\omega = G_{wand} \quad (2-2)$$



where K_{wand} is an adjustable positive value (m/sec).

G_{wand} is a random value of the range of $[- 1.5, 1.5]$ (rad/sec).

由於上述之 ω 每隔一段固定的時間週期就會隨機變動一次，因此每隔一段時間後機器人就會朝不同的方向移動來進行探索之任務。

2.3.2 閃避障礙物 (Obstacle Avoidance)

藉由全向式影像系統的視覺感測，可以讓機器人察覺到它的附近是否有障礙物的存在。以其與障礙物之相對位置來看， θ_{obs} 表示障礙物中心與機器人正前方延長線之夾角(定義逆時針為正，順時針為負)， l_{obs} 表示障礙物與機器人之間的距離(如圖 2-4)。在這裡我們未來可藉由在全向式影像系統平台攫取到之影像平面上製作兩個 90° 扇型的障礙物偵測區域(Obstacle Detection Zone)的方式來判斷與障礙物距離 l_{obs}

是否已小於安全距離 l_{safe} ，且障礙物是在機器人的左前方 ($0^\circ < \theta_{obs} < 90^\circ$) 或者右前方 ($-90^\circ < \theta_{obs} < 0^\circ$) 以做為障礙物閃避行為控制的條件，其運動控制如式(2-3)及式(2-4)：

$$v_{obs} = K_{obs} \quad (2-3)$$

$$\omega_{obs} = \begin{cases} -G_{obs} & \text{if } l_{obs} < l_{safe} \quad \& \quad 0 < \theta_{obs} < \frac{\pi}{2} \\ +G_{obs} & \text{if } l_{obs} < l_{safe} \quad \& \quad -\frac{\pi}{2} < \theta_{obs} < 0 \end{cases} \quad (2-4)$$

where K_{obs} is an adjustable positive value (m/sec).

G_{obs} is an adjustable positive value (rad/sec).

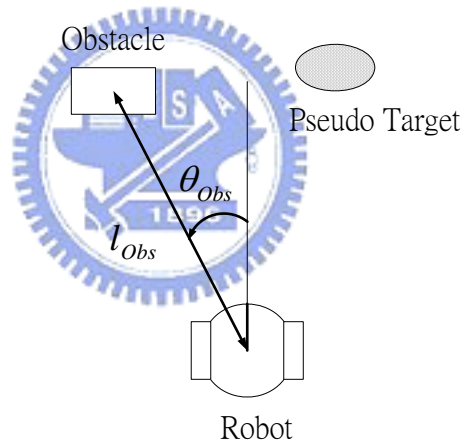


圖 2-4 機器人與障礙物之相對位置示意圖

2.3.3 閃避機器人隊友 (Inter-Robot Collision Avoidance)

機器人間之閃避行為目的為當任意兩台機器人彼此太靠近時，能彼此做閃避對方的動作。以圖 2-5 之 Robot 1 現行機器人為例， θ_{12} 表示它所偵測到 Robot 2 與自己正前方延長線之夾角(逆時針為正，順時針為負)， l_{12} 表示 Robot 2 與自己之間的距離，其 Robot 1 閃避 Robot 2 之運動控制如式(2-5)及式(2-6)：

$$v_{12} = K_{IRC} \quad (2-5)$$

$$w_{12} = \begin{cases} G_{IRC} \cdot (\theta_{12} + \pi) & \text{if } \theta_{12} \leq 0 \\ G_{IRC} \cdot (\theta_{12} - \pi) & \text{if } \theta_{12} > 0 \end{cases} \quad (2-6)$$

where K_{IRC} is an adjustable positive value (m/sec).

G_{IRC} is an adjustable positive gain value (1/sec).

此行為之控制原則為對任意一台機器人，若其偵測到在與自己正前方之相對角度 θ_{12} 處有另外一台機器人已進入到安全距離內時，則開始朝反方向進行閃避的動作。以圖 2-5 之示意為例，當 Robot 1 偵測到 Robot 2 在他右後方且十分靠近時，此時它便會朝向左前方之 Pseudo Target 而去，達成閃避的目的。

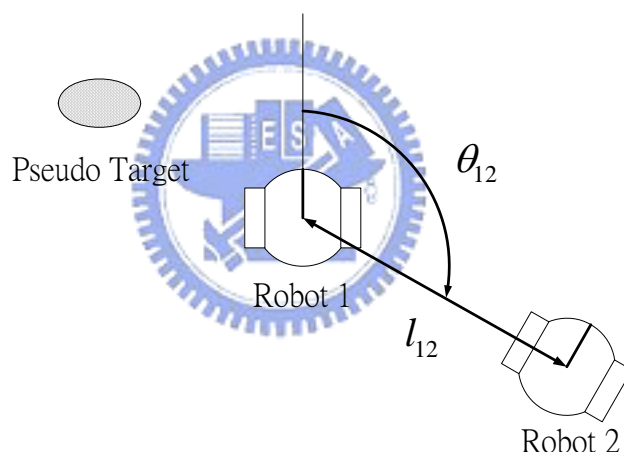


圖 2-5 機器人與機器人之相對位置示意圖

2.3.4 趨近目標物 (Target Approaching)

趨近目標物行為之功能在於當機器人發覺其視覺感測範圍中有疑似目標物存在時，則開始接近它做更進一步的確認(確認是否為入侵者或欲搬動之物體)，隨後透過無線傳輸發送命令讓其餘的機器人們來此處集合，做合力驅逐或搬運之動作。圖 2-6 中 θ_{obj} 表示目標物與現行機器人正前方延長線之夾角(逆時針為正，順時針為負)， l_{obj} 表示目標物與現行機器人之間的距離，其運動控制如式(2-7)及式 (2-8)：

$$v_{Tar} = K_{Tar} \quad (2-7)$$

$$w_{Tar} = G_{Tar} \cdot \theta_{Tar} \quad (2-8)$$

where K_{Tar} is an adjustable positive value (m/sec).

G_{Tar} is an adjustable positive gain value (1/sec).

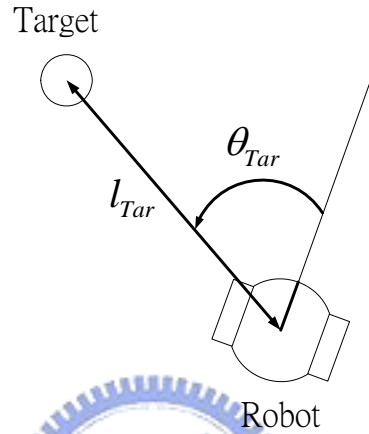


圖 2-6 機器人與目標物之相對位置示意圖

2.3.5 前往目的地 (Goal Approaching)

前往預定目的地之行為功能是當機器人處於隊長模式(Master Mode)下且其餘機器人皆已抵達起始隊形位置後才會啟動，目的在於讓 Leader Robot 帶領機器人團隊將物體協力搬運至預定目的地。在搬運過程當中，Leader Robot 只負責做路徑的規劃，而其餘機器人則負責將自己維持在與隊長之相對位置(即預設之隊形位置)。此預定目的地之座標(X_{Tar}, Y_{Tar})將預先紀錄在每台機器人當中，而機器人再與其自己紀錄(透過馬達軸編碼器)自己所在之位置座標($X_{Robot}, Y_{Robot}, \theta_{Robot}$)做運算，因而得到運動控制器中的兩個輸入項 θ_{Tar} 和 l_{Tar} ，其中 θ_{Tar} 表示預定目的地與 Leader Robot 正前方延長線之夾角(逆時針為正，順時針為負)， l_{Tar} 表示預定目的地與 Leader Robot 之相對距離，此二項之計算如式(2-9)-式(2-14)：

$$\Delta X = X_{Goal} - X_{Robot} \quad (2-9)$$

$$\Delta Y = Y_{Goal} - Y_{Robot} \quad (2-10)$$

$$\Delta X_{Rot} = \cos(\theta_{Robot}) * \Delta X + \sin(\theta_{Robot}) * \Delta Y \quad (2-11)$$

$$\Delta Y_{Rot} = -\sin(\theta_{Robot}) * \Delta X + \cos(\theta_{Robot}) * \Delta Y \quad (2-12)$$

$$\theta_{Goal} = \text{atan} \left(\frac{\Delta Y_{Rot}}{\Delta X_{Rot}} \right) \quad (2-13)$$

where $-\pi < \theta_{Goal} < \pi$

$$l_{Goal} = \sqrt{(X_{Goal} - X_{Robot})^2 + (Y_{Goal} - Y_{Robot})^2} \quad (2-14)$$

接著藉由 θ_{Tar} 即可得到機器人前往預定目的地之運動控制式(2-15)及式(2-16)：

$$v_{Goal} = K_{Goal} \quad (2-15)$$

$$w_{Goal} = G_{Goal} \cdot \theta_{Goal} \quad (2-16)$$

where K_{Goal} is an adjustable positive value (m/sec).

G_{Goal} is an adjustable positive gain value (1/sec).

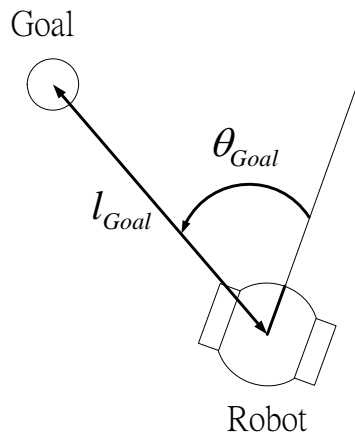


圖 2-7 機器人與預定目的地之相對位置示意圖

2.3.6 原地等待 (Waiting)

原地等待同樣亦為機器人處於隊長模式(Master Mode)下才會發生的行為，發生

條件為當任務執行中其發現目標物並接近後，此時切換至原地等待的行為並透過無線網路通知其餘機器人前來支援(因為自己無法獨力推動目標物)，一直等待到其餘機器人接就定位後才會再切換至別的行為模式，其運動控制如式(2-17)及式(2-18)：

$$v = 0 \quad (2-17)$$

$$w = 0 \quad (2-18)$$

2.3.7 靠近隊長 (Master Approaching)

靠近隊長之行為是當機器人團隊中某一台機器人已靠近到目標物的旁邊(此時那台機器人已切換至 Master 模式且成為 Leader Robot)並發送求援命令後由其它之 Follower Robot 接收到之後，Follower Robot 才會執行此一行為。Follower Robot 其自己與 Leader Robot 間的相對距離 l_{Mas} 和相對角度 θ_{Mas} 可由 Leader Robot 透過無線通訊方式將自己的座標位置 (X_{Master}, Y_{Master}) 傳送給所有之 Follower Robot，接著 Follower Robot 再用自身估測(透過馬達軸編碼器紀錄)其自己的位置座標 $(X_{Slave}, Y_{Slave}, \theta_{Slave})$ 運算來求得，其計算如式(2-19)-式(2-24)：

$$\Delta X = X_{Master} - X_{Slave} \quad (2-19)$$

$$\Delta Y = Y_{Master} - Y_{Slave} \quad (2-20)$$

$$\Delta X_{Rot} = \cos(\theta_{Slave}) * \Delta X + \sin(\theta_{Slave}) * \Delta Y \quad (2-21)$$

$$\Delta Y_{Rot} = -\sin(\theta_{Slave}) * \Delta X + \cos(\theta_{Slave}) * \Delta Y \quad (2-22)$$

$$\theta_{Mas} = \text{atan} \left(\frac{\Delta Y_{Rot}}{\Delta X_{Rot}} \right) \quad (2-23)$$

where $-\pi < \theta_{Tar} < \pi$

$$l_{Mas} = \sqrt{(X_{Master} - X_{Slave})^2 + (Y_{Master} - Y_{Slave})^2} \quad (2-24)$$

接者藉由 θ_{Mas} 即可得到機器人靠近隊長之運動控制式(2-25)及式(2-26)：

$$v_{Mas} = K_{Mas} \quad (2-25)$$

$$w_{Mas} = G_{Mas} \cdot \theta_{Mas} \quad (2-26)$$

where K_{Mas} is an adjustable positive value (m/sec).

G_{Mas} is an adjustable positive gain value (1/sec).

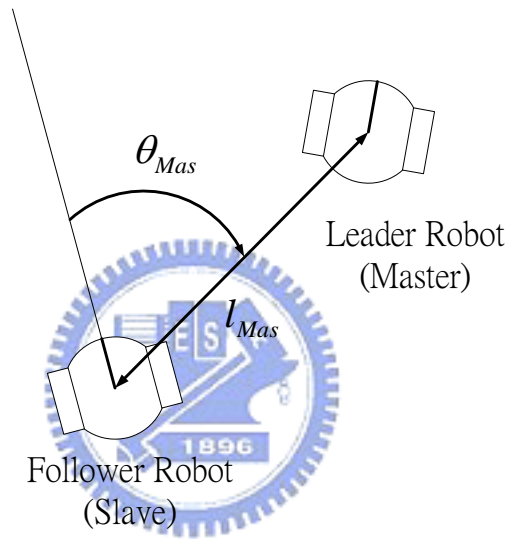


圖 2-8 Follower Robot 與 Leader Robot 之相對位置示意圖

2.3.8 隊形維持 (Formation Maintenance)

在隊形維持之行為下，每個 Follower Robot 會藉由自己偵測其與 Leader Robot 之相對位置及角度(包含朝向角差之計算)來計算出屬於自己的隊形位置。首先，在系統開始執行前預先計算出以 Leader Robot 為基準之座標系上之此 Follower Robot 所對應的隊形位置座標 (X_{D-Mas}, Y_{D-Mas}) ，其中 $\theta_{Desired}$ 和 $l_{Desired}$ 為已知之隊形參數(相對於 Leader Robot 的角度和距離，如圖 2-9)，其計算式如式(2-27)及式(2-28)：

$$X_{D-Mas} = \cos(\theta_{Desired}) * l_{Desired} \quad (2-27)$$

$$Y_{D-Mas} = \sin(\theta_{Desired}) * l_{Desired} \quad (2-28)$$

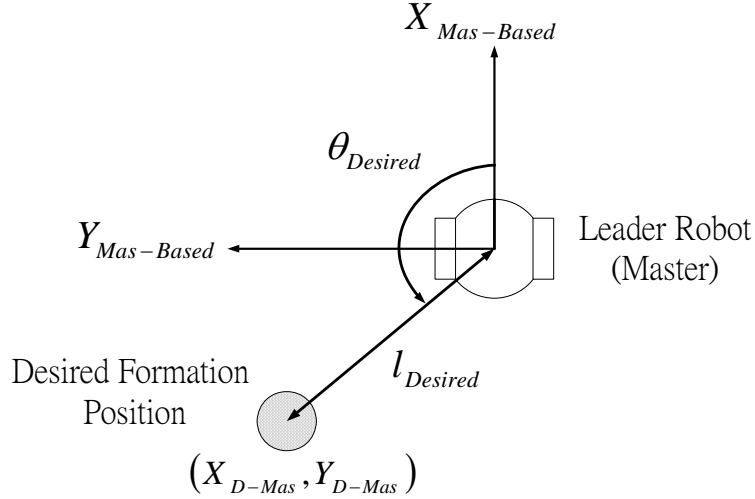


圖 2-9 以 Master 為基準之座標系上各座標位置示意圖

接著 Follower Robot 藉由影像系統可以偵測到自己與 Leader Robot 間之相對距離 l_{LF} 、相對角度 θ_{LF} ，然後再藉由 Leader Robot 透過無線通訊裝置傳送其所量測到自己與 Follower Robot 之相對角度 θ_{FL} (以 Leader Robot 為基準) 即可計算出朝向角差 ψ_{LF} ，其朝向角差的計算公式如式(2-29)。接著將 Leader Robot 及預定隊形位置做座標轉換的動作，將其座標轉換到以自己(Follower Robot)為基準的座標系上，如式(2-30)-式(2-33)，可以得到以 Slave(現行機器人)為基準之座標系上之 Leader Robot 座標 (X_{L-Slv}, Y_{L-Slv}) 及預定隊形位置的座標 (X_{D-Slv}, Y_{D-Slv}) ：

$$\psi_{LF} = \theta_{LF} - \theta_{FL} - \pi \quad (2-29)$$

$$X_{L-Slv} = \cos(\theta_{LF}) * l_{LF} \quad (2-30)$$

$$Y_{L-Slv} = \sin(\theta_{LF}) * l_{LF} \quad (2-31)$$

$$X_{D-Slv} = \cos(\psi_{LF}) * X_{D-Mas} - \sin(\psi_{LF}) * Y_{D-Mas} + X_{L-Slv} \quad (2-32)$$

$$Y_{D-Slv} = \sin(\psi_{LF}) * X_{D-Mas} + \cos(\psi_{LF}) * Y_{D-Mas} + Y_{L-Slv} \quad (2-33)$$

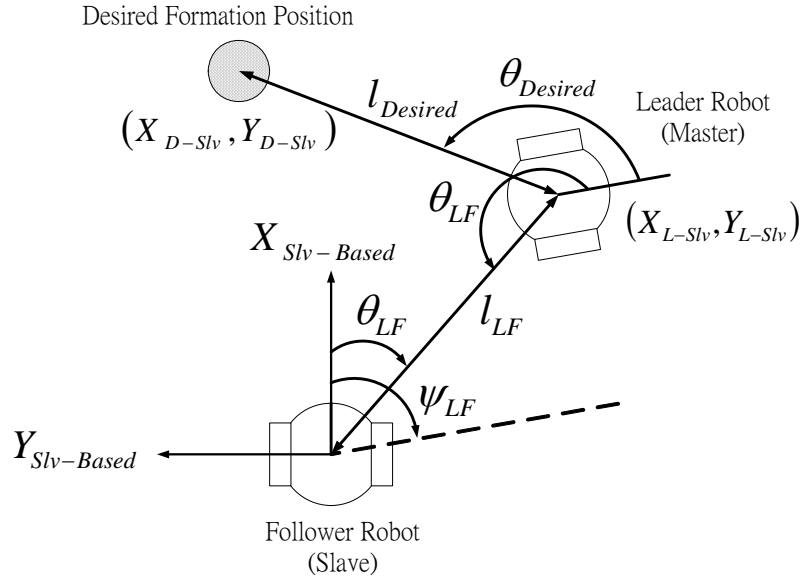


圖 2-10 以 Slave 為基準之座標系上各座標位置示意圖

以下為式(2-29)的推導過程，考慮一般性的 Leader Robot (Master)及 Follower Robot (Slave) 之相對關係，並將 Leader Robot 之原點平移與 Follower Robot 重疊如圖 2-11 所示。由幾何關係可發現 θ_{LF} 、 θ_{FL} 與 ψ_{LF} 此三角度會合成一水平線(角度大小為 π)，且我們定義之角度順時針為正，逆時針為負，因此得到下式(2-34)：

$$\theta_{LF} + (-\psi_{LF}) + (-\theta_{FL}) = \pi \quad (2-34)$$

將式(2-34)整理過後即推得式(2-29)： $\psi_{LF} = \theta_{LF} - \theta_{FL} - \pi$

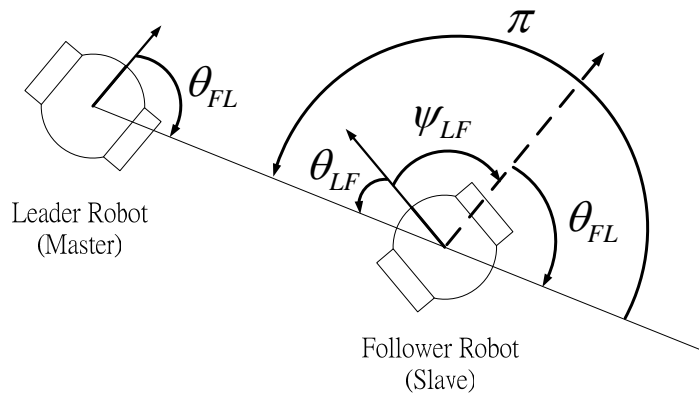


圖 2-11 ψ_{LF} 之推導關係圖

然後藉由預定隊形位置座標 (X_{D-Slv}, Y_{D-Slv}) 來計算出其與現行機器人(Follower Robot)間之相對距離 l_{Fmt} 及相對角度 θ_{Fmt} ，如式(2-35)及式(2-36)：

$$l_{Fmt} = \sqrt{(X_{D-Slv})^2 + (Y_{D-Slv})^2} \quad (2-35)$$

$$\theta_{Fmt} = \text{atan}\left(\frac{Y_{D-Slv}}{X_{D-Slv}}\right) \quad (2-36)$$

where $-\pi < \theta_{Fmt} < \pi$

最後再由 l_{Fmt} 及 θ_{Fmt} 可得到此 Follower Robot 抵達預定隊形位置之運動控制式(2-37)

及式(2-38)：

$$v_{Fmt} = \begin{cases} K_{Fmt} * l_{Fmt} & \text{if } l_{Fmt} \geq l_{Dead} \\ 0 & \text{if } l_{Fmt} < l_{Dead} \end{cases} \quad (2-37)$$

$$w_{Fmt} = G_{Fmt} \cdot \theta_{Fmt} \quad (2-38)$$

where K_{Fmt} is an adjustable positive gain value (1/sec).

G_{Fmt} is an adjustable positive gain value (1/sec).

l_{Dead} is the dead zone distance for formation maintenance (m).

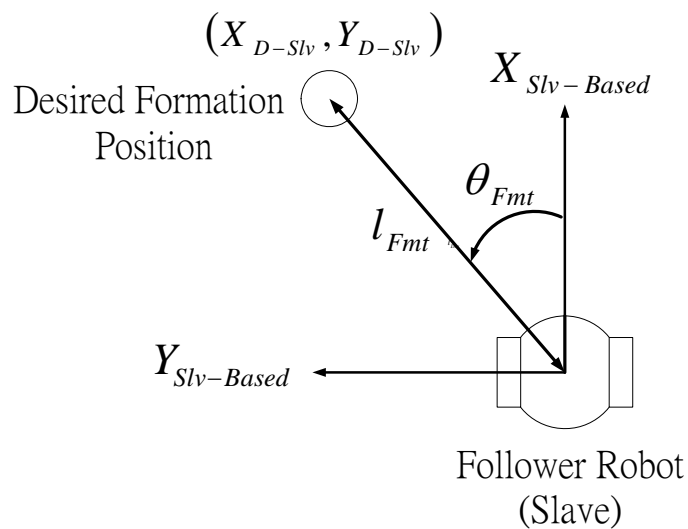


圖 2-12 Follower Robot 與預定隊形位置之相對位置示意圖

第三章 視覺回授與全向式影像系統

本章將介紹應用於實體機器人上之全向式視覺感測系統，其目的在於讓各機器人能取得自己與目標物、其他機器人亦或者是障礙物間的相對距離及角度，以便做隊形維持的控制。下圖 3-1 為全向式影像系統的實體圖，主要由 (1) DSP 嵌入式影像平台 (2) 全向式反射鏡面 此兩部分所組成，反射鏡架設在機器人的最上方，而影像平台中的 CMOS Camera 的鏡頭朝上面對此反射鏡，周圍 360 度的環境影像會透過光線的反射而被此 CCD Camera 所擷取下來(示意圖如圖 3-2)，故此全向式影像系統可以輕易地幫助 Robot 擷取到周圍 360° 的環境影像並且即時做處理，以分擔 PC 端的計算處理負擔。在本論文的實驗裡，此全向式攝影系統個別架設在三台機器人上，視野範圍是以機器人中心為圓心之半徑 1.2 公尺內的圓形區域(視野範圍可依據使用者的需求由反射鏡與 Camera 間的相對距離及反射鏡離地高度來做調整)，經由 DSP 影像平台即時之影像處理流程後，PC 端可隨時得到與周圍物體(包含辨識目標物、機器人隊友亦或者障礙物)之相對距離及角度資訊，以作為運動控制器的輸入。目前，類似這樣全向式影像系統(大部分仍用 Webcam 取代影像擷取平台)之理論研究與應用控制也同樣地被應用在機器人的快速追蹤上[20][21]。



圖 3-1 全向式影像系統實體圖

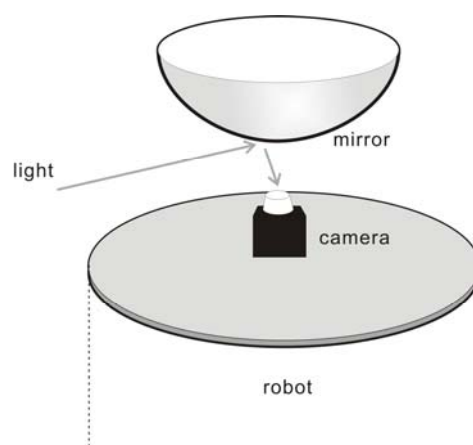



圖 3-2 全向式影像系統原理示意圖

3.1 硬體描述

本論文中之全向式影像系統所採用的影像擷取系統乃為本實驗室所開發之 DSP 嵌入式影像平台，圖 3-3 為其整體系統架構圖，主要包含了一個 CMOS 影像感測板、FPGA Board、DSK6416T 影像發展板和一顆緩衝記憶體(Frame Buffer)。CMOS sensor 部分為銳相(IC-media)公司所生產之型號 EVT 202 CMOS 感測器，其最大有效像素為 640 x 480，最大更新率為 30 frames/sec，影像輸出格式包含了 8-bit raw data、8/16-bit YCrCb(本系統所選用的即為此)、16-bit RGB 和 24-bit RGB。此外，此 CMOS 感測板尚包含一些影像調整功能，包括了自動白平衡(Auto White Balancing)及 Gamma 參數調整可對整張影像的亮度及色彩飽和度進行調整等功能。在 FPGA 板方面所使用的為 ALTERA 公司生產，型號為 FLEX10K50E240-3，主要功能為處理緩衝記憶體之讀出與寫入的控制訊號、資料排序及產生讓 DSK6416T 讀取影像資料之觸發訊號，另外要透過 I2C 來做 CMOS 感測器的初始化過程，其初始化內容如下：

- 
- 影像更新速率：每秒 30 張 Frames
 - 影像輸出格式：8-bit 4:2:2 YCrCb QVGA (320 x 240 像素大小)
 - 輻射校正：Gamma = 2.2
 - 色彩飽和度：Saturation=1.5

在影像處理板方面選用的為德州儀器(TI)公司所出產之 TMS320C6416T DSK 影像發展板，主要做為影像資料搬運及即時運算的功能，其工作頻率為 1GHz。另外在緩衝記憶體(AL422B)的部分主要做為影像資料儲存之用，其記憶體長度為 384Kbytes 可以完整儲存兩張 QVGA 大小之影像，且資料存取動作為 FIFO(First In First Out)，可以以較低的速度接收資料，而以非常快的速度放出(此緩衝記憶體之最快工作時脈為 50MHz)。

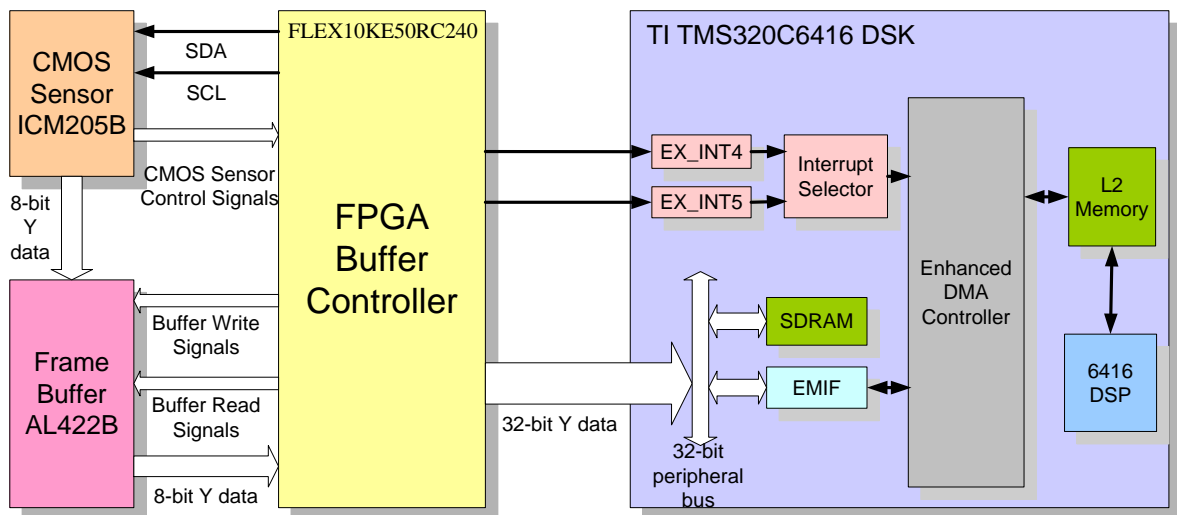


圖 3-3 DSP 影像平台系統架構圖[22]

3.2 環境影像擷取

由上述之影像平台再搭配微星(MSI)公司生產之全向式鏡面模組可藉由光線的折射原理擷取機器人周圍環境 360 度角之環境影像，其原始影像如圖 3-4，大小為 320 x 240 之 QVAG 彩色影像(8bit YCrCb 格式)，在未經影像處理前每秒可擷取 30 張影像。

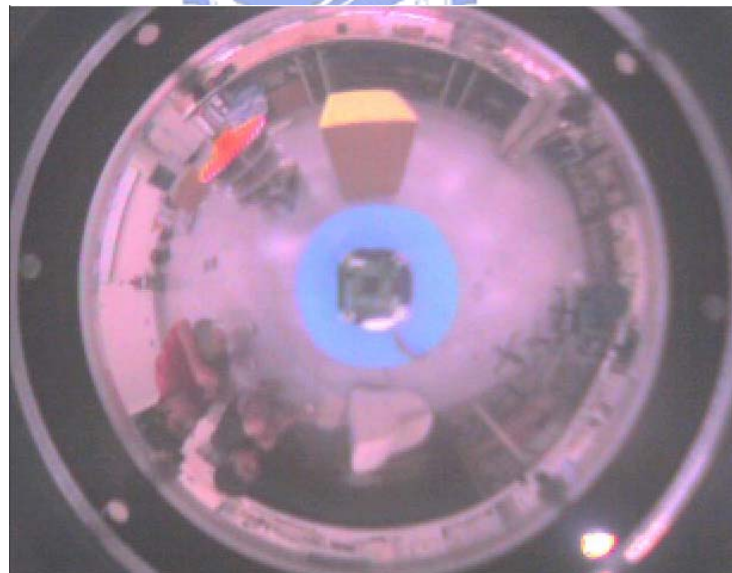


圖 3-4 原始環境影像

上述擷取到之環境影像為 YCrCb(即 YUV)格式，每一單一像素分別由 2Bytes 的 Y 及各 1Byte 的 U 和 V 所構成，其 YUV 分量最後分別儲存在以 qvga_y、qvga_u 及 qvga_v 為名之起始位置的記憶體陣列中，未來便是以上述三記憶體陣列來做後續的影像處理。

3.3 色彩辨識處理

為了進行機器人之間及機器人對於目標物的辨識，我們將實驗中的三台機器人及目標物(紙箱)分別標示了顏色，分別為：Robot 1(橘色)、Robot2(紅色)、Robot 3 (藍色)及目標物(黃色)。針對 DSP 影像處理板的程式撰寫部分，我撰寫了兩種功能模式— Color Threshold 模式以及 Color Detection 模式。在 Color Threshold 模式下，使用者可自行輸入所欲做顏色統計的辨識區塊(Detection Window)之起始位置(X_n, Y_n)和其長寬大小(Width, Height)，如圖 3-5。在輸入完 Detection Window 的起始資訊後啟動 DSP 處理板，接著統計此 Window 內之所有像素的 YUV 值範圍，以找出各種顏色的 YUV 臨界值(Threshold Value)。表 3-1 為各個機器人上之影像系統在 Color Threshold 模式下所找出之各顏色之 YUV 範圍，圖 3-6 為整個色彩範圍尋找之工作流程圖：

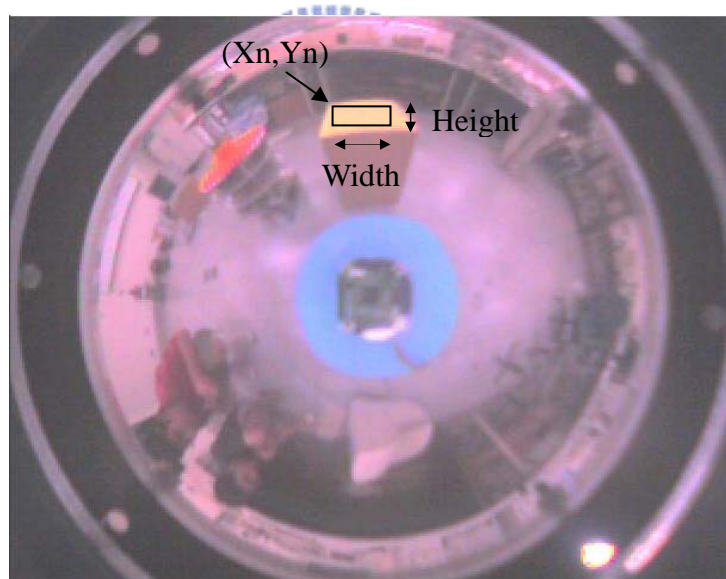


圖 3-5 Detection Window 之各項參數示意圖

表 3-1 在 Color Threshold 模式下所找出的各顏色 YUV 範圍

Robot 1 (Orange)						
	Y_min	Y_max	U_min	U_max	V_min	V_max
Red (Robot 2)	149	163	101	122	176	203
Blue (Robot 3)	200	213	176	203	68	89
Yellow (Target)	230	255	68	98	110	128

Robot 2 (Red)						
	Y_min	Y_max	U_min	U_max	V_min	V_max
Orange (R1)	148	187	74	107	206	255
Blue (R3)	163	196	167	209	77	125
Yellow (Target)	240	255	56	89	173	215
Robot 3 (Blue)						
	Y_min	Y_max	U_min	U_max	V_min	V_max
Orange (R1)	162	207	71	119	218	255
Red (R2)	131	149	107	140	164	206
Yellow (Target)	220	246	71	95	176	203

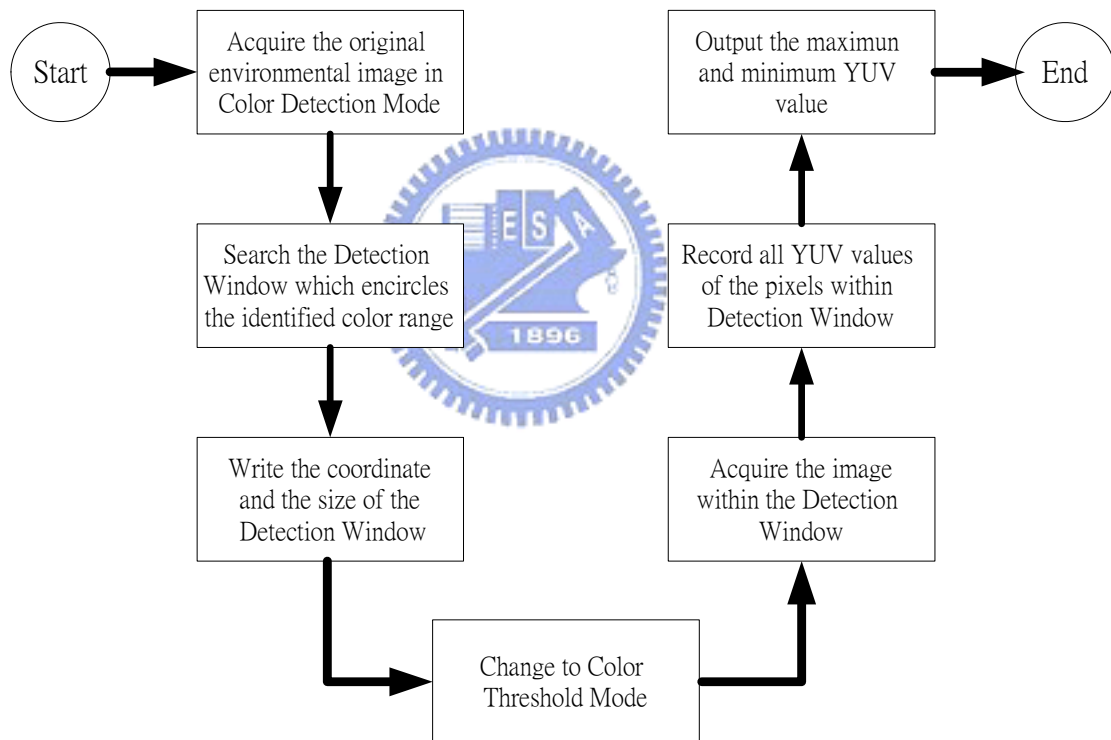


圖 3-6 色彩範圍尋找工作流程圖

藉由已經找出來的色彩臨界值(參照表 3-1),接著我們將要在原始環境影像中框選出各顏色識別物(含目標物及三台機器人)的範圍,並找出其中心位置,以做為下一節所要描述之相對位置及角度估測的原始數據。以 Robot 3 找尋目標物(黃色紙箱)為例,首先將 Robot 3 擷取到的整張原始影像(如下圖 3-7)做逐一像素的掃描,若其 YUV 值皆符合

在 Robot 3 上校正之 Yellow (Target)範圍，則定此像素值為 1；反之，若有其中任何一項不符合則定像素值為 0。經由上述之二值化過程後，只有在原始影像中為黃色的範圍會以白色像素顯示，其餘部分皆為黑色(如下圖 3-8)，接著再以一矩形框來框選住圖中白色的部分，並回傳其中心位置座標，此座標對應到原始影像上即為黃色目標物之中心位置，圖 3-9 為此矩形框與實際黃色目標物之位置關係，證實了此矩形框確實框選到了欲辨識的部分。



圖 3-7 Robot 3 擷取之原始環境影像

圖 3-8 經由二值化過程後之環境影像

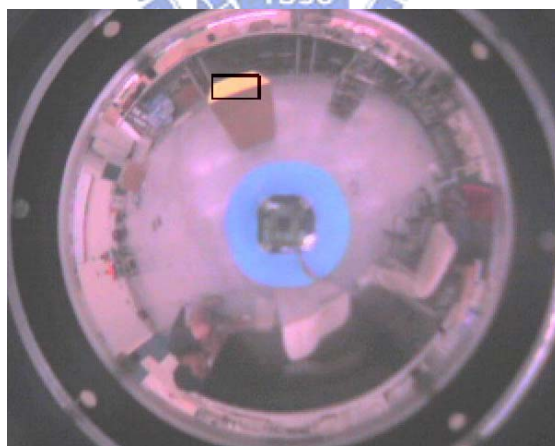


圖 3-9 目標物(黃色部分)在經過二值化流程後已被正確地框選出來

3.4 相對距離及角度估測

全向式影像系統其影像平面上像素位置與反射鏡曲面間會有個函數關係存在[16]：

$$z = f_{\xi}(x, y) = \frac{-1 + \xi^2(x^2 + y^2)}{1 + \xi\sqrt{1 + (1 - \xi^2)(x^2 + y^2)}} \quad (3-1)$$

其中 $\xi \in [0, 1]$ 為曲面參數，會因反射鏡曲面的不同而有不同的代表值。此時一個在實際空間座標 (X, Y, Z) 與經反射鏡反射而投影在影像平面上的 (x, y) 座標，兩者之間會有以下的關係：

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{-Z + \xi\sqrt{X^2 + Y^2 + Z^2}} \begin{bmatrix} X \\ Y \end{bmatrix} \stackrel{\Delta}{=} \frac{1}{\lambda} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (3-2)$$

由於我們已知所要追蹤之機器人目標物之高度 Z (這點已由架設已知高度的顏色識別物來達成)，即可由影像平面的座標來推算回其目標物在實際空間中的位置，這樣可以讓每一個機器人估算其周圍機器人之相對位置(圖 3-10 為實際空間座標與影像平面之投影關係示意圖，由實際空間座標 $(X, Y, Z)^T$ 可對應到影像平面之座標 $(x, y)^T$)。

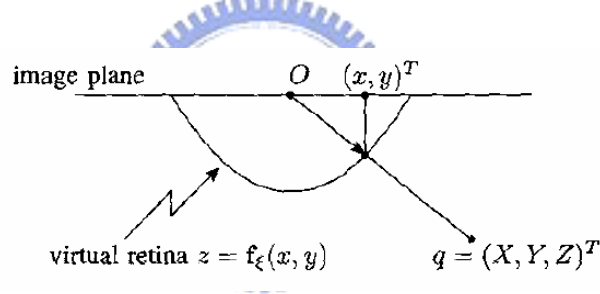


圖 3-10 實際空間座標點在全向式影像系統之影像平面上的投影關係[16]

在圖 3-8 裡，我們首先來計算目標物(黃色部分)在影像平面上與影像正中央(即 CMOS sensor 鏡頭所在位置)之距離(單位為像素)，若目標物之中心在影像平面上之中心座標為 Target (x_{center}, y_{center}) ，我們定義此 320x240 QVGA 影像之正中央座標為 Mid (x_{mid}, y_{mid}) ，其中 $x_{mid} = 160, y_{mid} = 120$ 。因此從 Target 到 Mid 在影像平面上之距離計算如下式(3-3)：

$$D_{Image} = \sqrt{(x_{center} - x_{mid})^2 + (y_{center} - y_{mid})^2} \quad (3-3)$$

下一步驟是將影像平面之距離 D_{Image} (單位為像素)還原成實際空間座標系之距離 D_{Real} (單位為公尺)，這部分可以由實際校正步驟來求得全向是影像平台架設後之實際轉

換公式。由於全向式反射鏡面是 360 度全對稱的，因此我們只要沿著其中一條線來做校正即可求得準確的轉換函數，下圖 3-11 為 Robot 1 對於黃色目標物經由實際量測之影像平面距離與實際量測距離的關係圖(共 16 個校正點)：

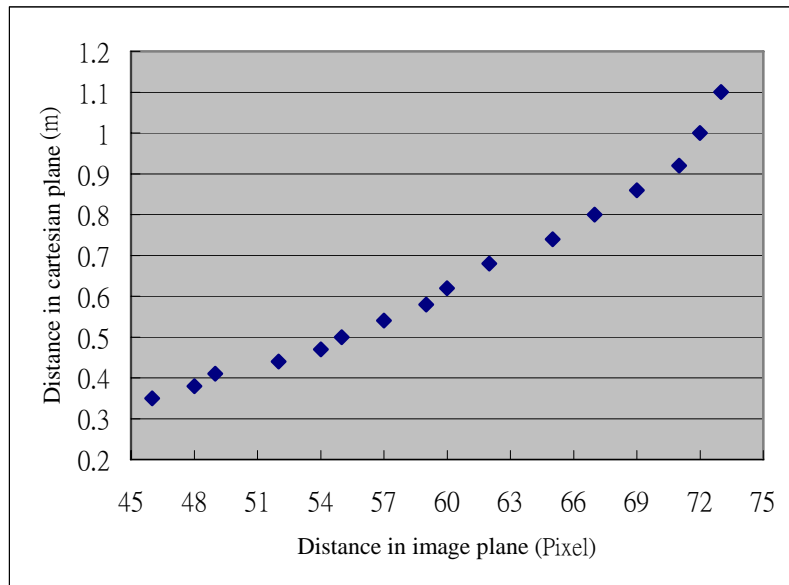


圖 3-11 Robot 1 之影像平面距離(Pixel)與實際空間距離(m)之關係圖

將圖 3-11 中之所有校正點以一個二次函數曲線來近似可得圖 3-11 與式(3-4)之轉換公式：

$$y = 0.0005987x^2 - 0.046155x + 1.2203 \quad (3-4)$$

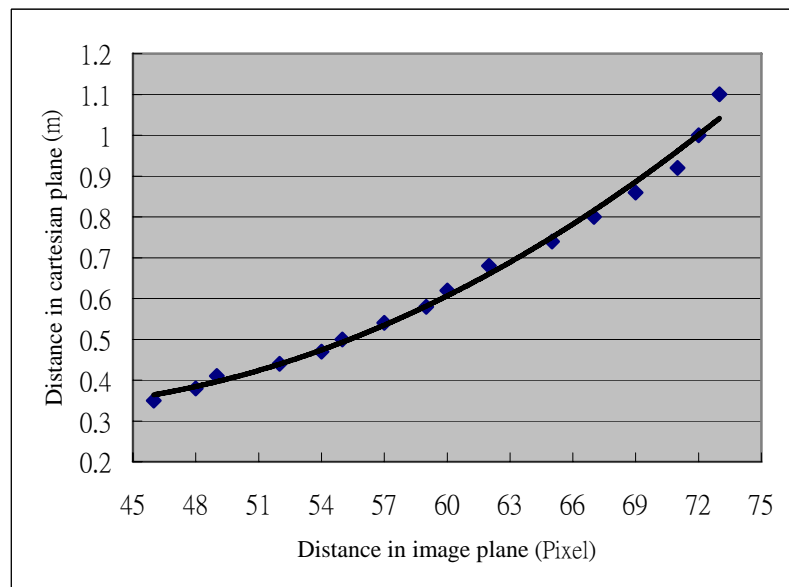


圖 3-12 Robot 1 之影像平面距離與實際空間距離之二次近似曲線

同理，依照同樣的校正步驟，我們依序求得 Robot 2 及 Robot 3 的距離轉換公式 (3-5)-(3-6) (式中之 x 項為影像距離， y 項為估測所得之實際空間距離)：

$$\text{Robot 2: } y = 0.0003765x^2 - 0.041034x + 1.5281 \quad (3-5)$$

$$\text{Robot 3: } y = 0.0005458x^2 - 0.05175x + 1.6172 \quad (3-6)$$

在角度估測方面，我們欲求得以 Robot 正前方(即影像正上方)為基準，辨識物中心與影像正中央延長線與此基準線之夾角 θ (逆時針為正，順時針為負，範圍為 $\pm 180^\circ$ 之間)，如下圖 3-13：

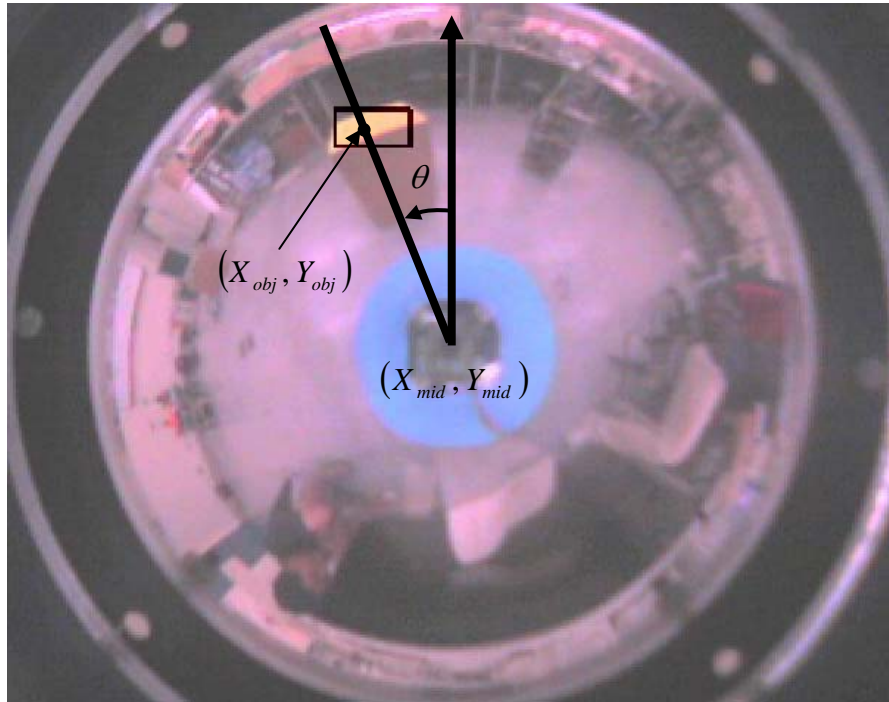


圖 3-13 黃色目標物與 Robot 間之相對角度 θ 定義圖

此處我們將藉由辨識物中心座標 (X_{obj}, Y_{obj}) 及影像正中央座標 (X_{mid}, Y_{mid}) 來計算出我們所定義之 θ 夾角，如式(3-7)：

$$\theta = \text{atan}\left(\frac{Y_{obj} - Y_{mid}}{X_{obj} - X_{mid}}\right) - \frac{\pi}{2} \quad (\text{rad}) \quad (3-7)$$

$$\theta \in (-\pi, \pi]$$

3.5 與 PC 端之資料溝通

藉由 DSK6416T 之通用 I/O 埠(GPIO)，我們實現了一組 RS232 串列埠介面，使其能將影像處理完後所得到的相對距離及角度資訊傳送至 PC 端來做機器人的行為運動控制，其資料輸出格式如下圖 3-14。對每個機器人上之全向式影像系統而言，每次會送出 6 Bytes 的資料至 PC 端，其中包含了其他兩個機器人隊友及目標物的相對距離及角度資訊(每筆資料皆為 1 Bytes)。

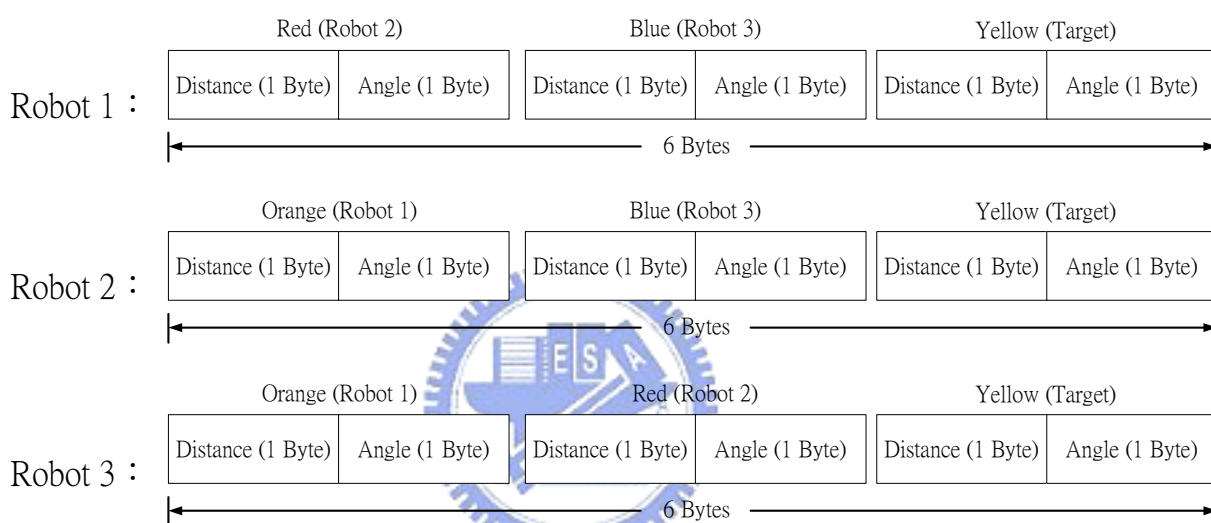


圖 3-14 全向式影像系統輸出至 PC 端之資料格式

在相對距離(Distance)資料方面，影像系統所輸出的為其在影像平面上之像素距離，如距離為 75 個像素，則輸出 0x4B；在相對角度(Angle)資料方面，由於 1 Byte 所能表示之十進位數值介於-128 ~ 127 之間，因此我們把實際影像平台所得到之-180° ~ 180°的角度範圍做一個比例上的轉換，使其能夠落於-127~127 區間內，而以 1 Byte 表示之。其轉換公式如式(3-8)，若原始得到角度 θ (單位 Degree) 範圍在 ± 180 間，轉換後之輸出資料數值大小為 θ' ，則：

$$\theta' = \theta / 180 * 127 \quad (3-8)$$

之後再四捨五入取其整數值傳送出去，即若相對角度為 60 度，則輸出 $60/180*127 = 42.3$ 之四捨五入值 42 (即 0x2A) 至 PC 端，其後再由 PC 端還原為實際之角度大小。

3.6 結果與測試

在 PC 端透過 RS232 埠接受到由影像平台所傳送過來之相對位置資料後(資料型態在上一節已描述)，在相對距離方面透過公式(3-4)-(3-6)還原成 Cartesian 平面上之距離(單位 m)；接著再將接收得到之相對角度數值 θ' 透過下式(3-9)還原成單位為 Degree，數值範圍在 ± 180 間之角度 θ ：

$$\theta = \theta' / 127 * 180 \quad (\text{Degree}) \quad (3-9)$$

最後我們在 PC 端觀察經過估測及還原兩步驟所得到之相對距離(m)及角度(degree)，透過 Robot 3 對於目標物之實際量測，可得到這兩項資訊之平均誤差如下：

- 距離誤差： ± 0.06 (m)
- 角度誤差： ± 3 (Degree)

上述之相對距離及角度資訊在啟動影像平台功能的同時即不斷透過 RS232 埠傳送至 PC 端，每秒至少可傳送 15 張影像處理後所得到的資訊(每張影像各含 6 Bytes 之資訊)，有效探測範圍為以鏡頭中心為圓心，半徑 30cm ~ 1.2m 內之區域(30cm 內為機器人本體所造成之死角範圍)，藉此我們便能以全向式之視覺回授功能來達成隊形控制及合作搬運之目的。



第四章 機器人通訊與 RF 無線傳輸模組

本章將介紹應用於三台實體機器人上之 RF 無線傳輸模組電路及其應用設計，其目的在於讓各個機器人能與其他機器人隊友間做無線方式之訊息溝通，以做為其高層行為選擇之輸入。在下面 4.1 節我將介紹此模組的元件特性與硬體規格，4.2 節描述我們所設計之其與 PC 端通訊的介面電路設計，4.3 節將詳細描述其應用在實體機器人上之使用方式及機器人間之無線通訊協定。

4.1 元件特性

在本論文實驗中，所有機器人皆裝備 RF 無線傳輸模組做為機器人彼此間的訊息傳遞功能。其硬體電路部分採用艾威 (Awin) 公司編號 AM2400BS-RA 的 RF 模組 (IC 部分為 Nordic 公司發展之 nRF2401)。其實體圖如圖 4-1，且元件特性如以下所列：

1. 工作電壓介於 1.9V~3.6V 之間，若元件設定在傳輸模式其最大電流為 10.5mA；若設定為接收模式其最大電流則為 18mA。整體體積為 30mm × 15mm，其 IC 部分為體積 5mm × 5mm 的 24pin 腳位 IC。
2. 此模組之傳送頻率範圍為 2400~2524 MHz 之 ISM 頻帶，最大支援 125 個通道，每個通道間隔約為 1MHz。但若欲同時使用超過 83 個通道時只在某些特定區域內有效，如日本等。
3. 提供 ShockBurst 模式，資料傳輸速率(Data Rate)可達 1Mbps，另外只要經由 PWR_UP、CS 和 CE 三根腳位，再藉由串列介面即可完成此模組之初始組態設定。另外支援雙通道模式，在接收模式下，可同時接收兩個獨立通道之傳送模組所傳送之資料 (示意圖如圖 4-2)；若欲使用通道模式接收資料時，第二個通道的頻率必須比第一個通道的頻率高出 8MHz (如圖 4-3)。最後此模組並提供 8-bit 和 16-bit 的 CRC 資料檢查功能，

以確認資料傳輸的正確性。

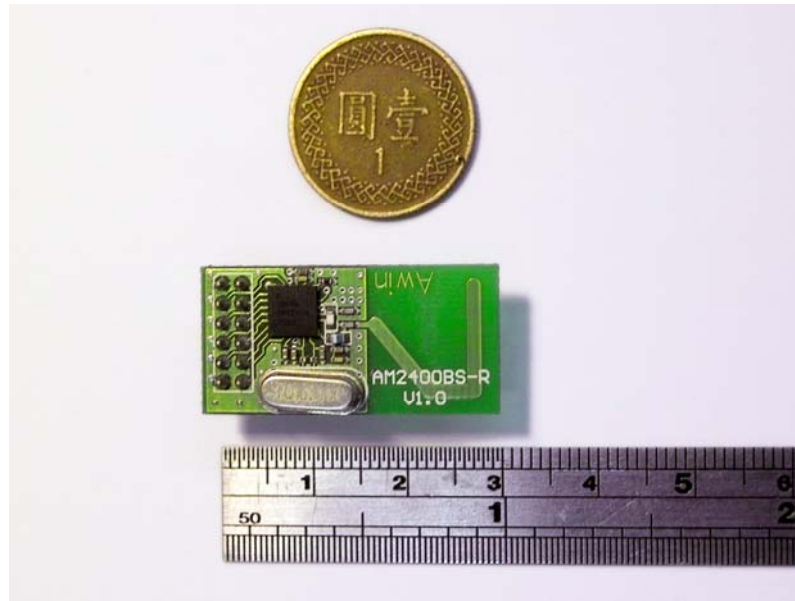


圖 4-1 RF 無線傳輸模組實體圖

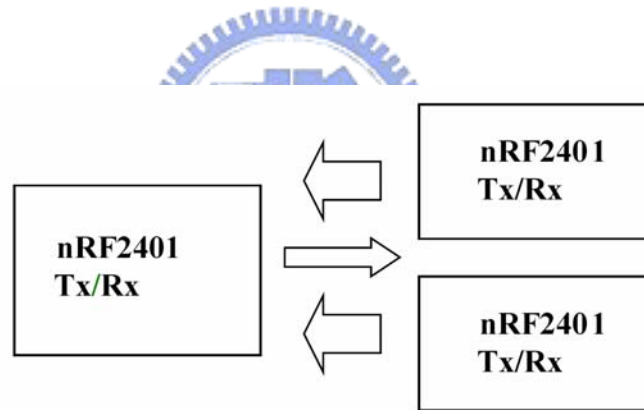


圖 4-2 雙通道模式示意圖

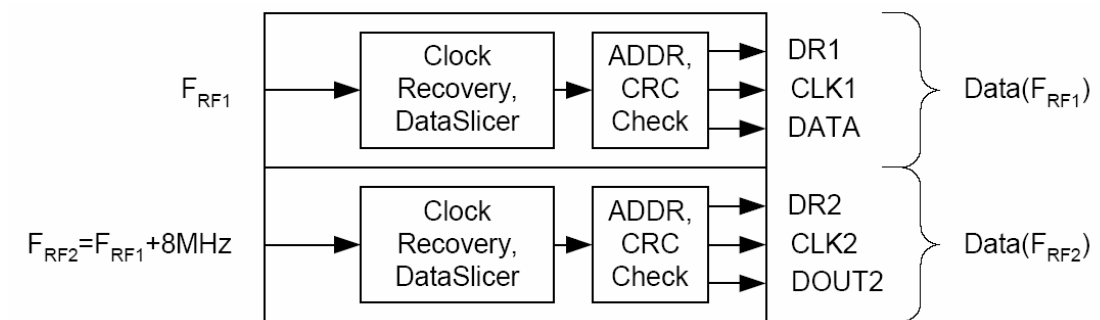


圖 4-3 雙通道模式下同時接受兩個獨立通道所傳送之資料示意圖

在使用 RF 無線傳輸模組時，必須先由微控器 (如 8051)將初始組態設定資料，如傳送或接收模式、頻帶設定、CRC 模式 (8-bit 或 16-bit)、資料 Address 及資料長度 (bit 數)等 12 Bytes 的資料透過 PWR_UP、CS 和 CE 之控制訊號腳位傳送到 nRF2401 晶片上。其初始設定之控制訊號動作如圖 R-4 所示，首先將 PWR_UP (Power Up)拉為高態，接著將 CS (Chip Select)也拉為高態，並同時將 CE (Chip Enable)拉為低態，這樣就表示此時要做初始設定組態的資料傳送，然後 12 Bytes 的初始設定資料就會透過 Clock 時脈依序寫入到 RF 模組的晶片上。

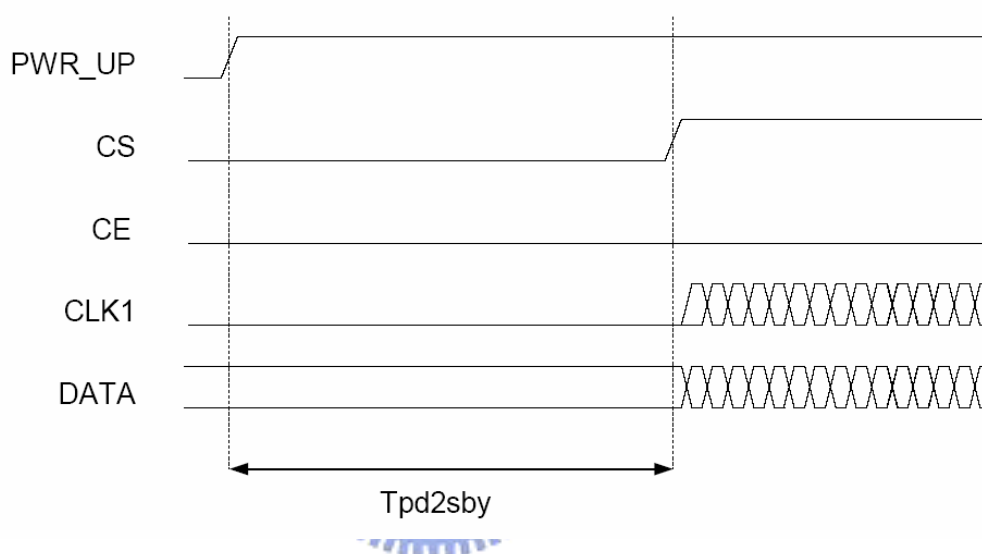


圖 4-4 RF 無線傳送、接收模組之初始組態設定時序圖

另外，此 RF 無線傳輸模組在傳輸模式下，可選擇設定為 ShockBurst 模式或 Direct 模式。ShockBurst 模式提供了 on-chip 的 FIFO，可以以較低的資料速率接收微控器端傳來的資料，並再將這些資料以非常高的速率 (最高可達 1Mbps)透過無線方式傳送給其它的 RF 無線接收模組，因而大幅降低了功率的損耗。圖 4-5 為 ShockBurst 模式下之資料速率示意圖，圖 4-6 為在 ShockBurst 模式下之 RF 傳送模組的資料傳送時序圖 (同樣透過 PWR_UP、CS 和 CE 三根腳位來做資料傳送的控制)。

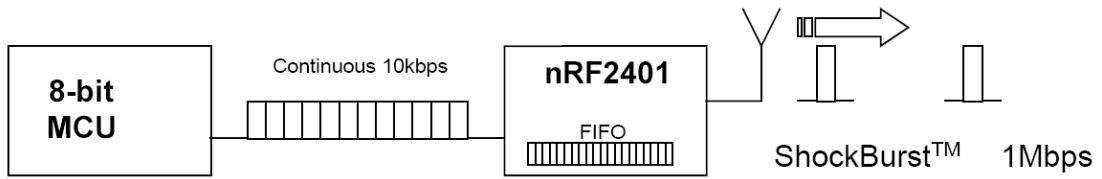


圖 4-5 ShockBurst 模式下的資料傳送速率示意圖

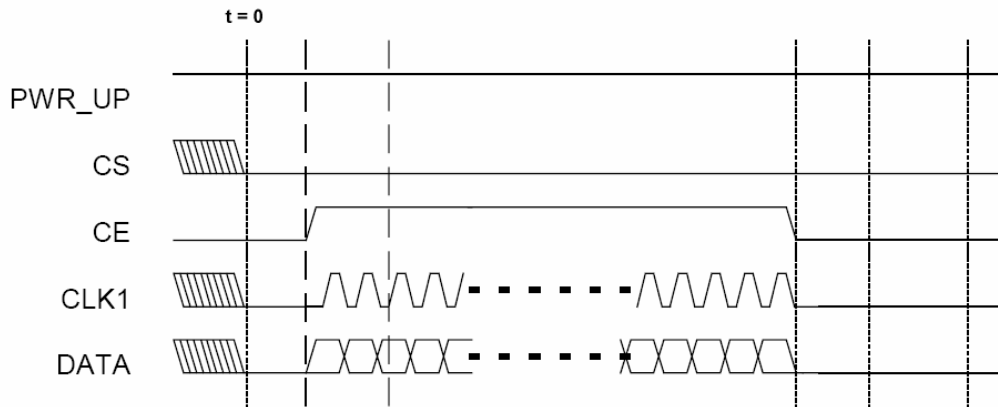


圖 4-6 ShockBurst 模式下之 RF 傳送模組資料傳送時序圖

RF 無線傳輸模組之傳送端在 ShockBurst 模式下資料處理流程如下圖 4-7：

- 微控器首先透過 Configure 過程將 RF 模組設定為傳送端，並將 CE 腳位先拉為高態，這樣便開始啟動 nRF2401 晶片的資料處理功能。
- PC 端將欲傳送的資料 (Payload Data) 透過串列埠介面送到微控器後，再由微控器將 Data Address 和 Payload Data 透過 clock 時脈依序寫入到 nRF2401 晶片中。
- nRF2401 晶片開始計算 CRC (Cyclical Redundancy Check)，並將 CRC 加在欲傳送之資料的最後面，此過程完成再將 CE 設為低態。
- 在欲傳送之資料的最前面加上一 Preamble，做為接收端判斷此筆資料是否為應當接收之資料的依據。
- RF 傳送端模組開始以 1Mbps 之速度將資料以無線傳送方式送到接收端模組。

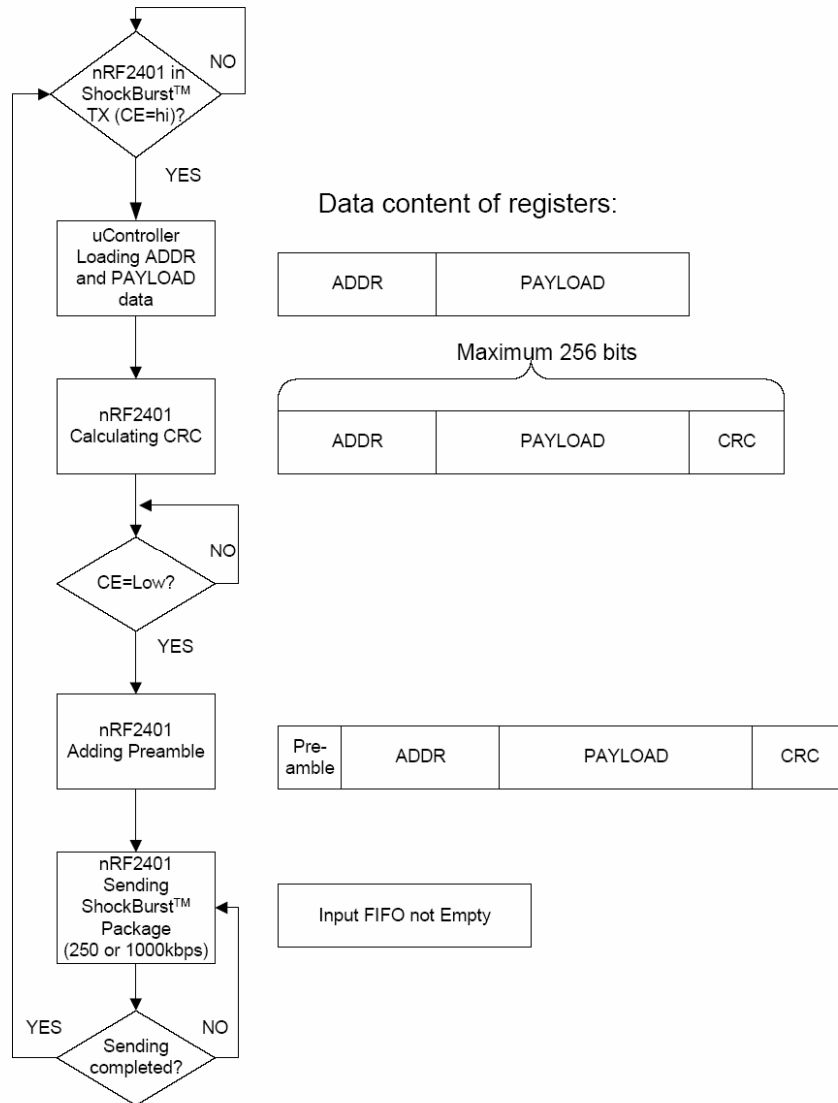


圖 4-7 RF 無線傳輸模組之傳送端資料處理程序流程圖

RF 無線傳輸模組之接收端在 ShockBurst 模式下資料處理流程如下圖 4-8：

- a. 微控器首先透過 Configure 過程將 RF 模組設定為接收端，這樣便開始啟動 nRF2401 晶片的資料接收功能。
- b. nRF2401 晶片偵測是否有資料需要被接收；若在資料串列的最前端偵測到 Preamble 則開始接收後續的資料。
- c. nRF2401 晶片確認 Data Address 無誤後，接收後續的資料並開始檢查 CRC。當 CRC 也確認無誤後再將 Data Address 和 CRC 分別從資料串列的前後移除，只留下 Payload

Data，再將 DR1 (Data Ready at data channel 1)腳位設為高態，通知微控器來將資料取走。

- d. 當微控器偵測到 nRF2401 晶片上之 DR1 腳位被拉為高態後，開始透過 clock 時脈依序將儲存在 nRF2401 晶片之資料取出，並存在自己的暫存器裡，直到 DR1 腳位再度變為低態為止(表示 nRF2401 晶片裡的資料已全部被取完)。
- e. 最後微控器再將接收得來的資料透過串列埠介面傳送到 PC 端。

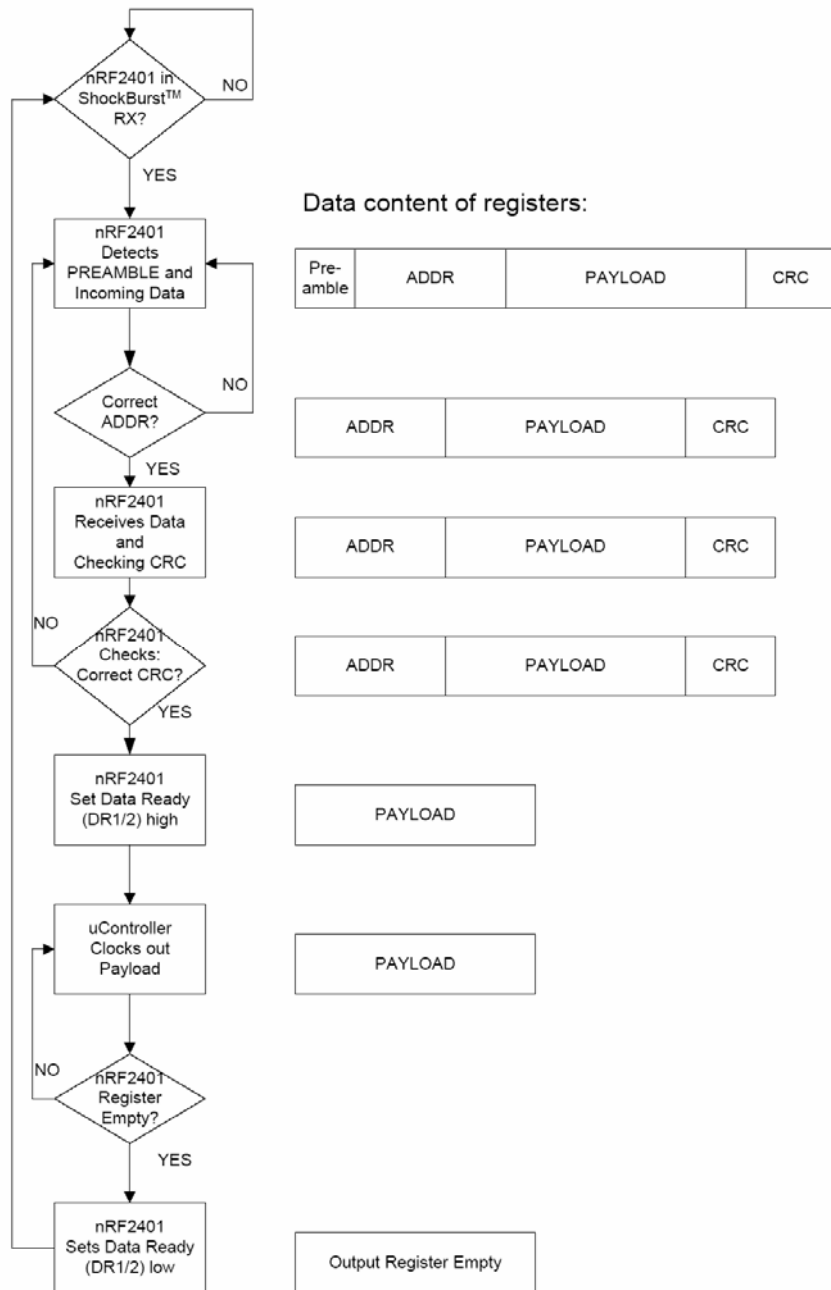


圖 4-8 RF 無線傳輸模組之接收端資料處理程序流程圖

4.2 微控器介面電路設計

本論文搭配 RF 無線傳輸模組的微控器所採用為 89C51，其包含專為控制應用所設計的 8 位元中央處理單元，4 個 Byte 雙向且每條都可以被單獨定址的 I/O 埠，內部 128 Byte 可供讀/寫的 RAM，2 組 16 位元的計時器/計數器以及一個通信用的全雙工 UART (串列 I/O)。圖 4-9 為此微控器之介面電路圖，其中由於 RF 模組的工作電壓介於 1.9V~3.6V 之間，因此必須先將 5V 之外接電源經過 3 顆二極體降壓後再供給之。在控制訊號部分，我們應用 89C51 的 I/O 埠 2.0~2.5 分別連接到 RF 模組的各根腳位，再透過 8051 的組語撰寫產生出傳送資料所需要的 Clock 訊號以及 PWR_UP、CS 和 CE 三根控制訊號並傳送 12 Bytes 的資料來設定 RF 模組的組態。最後在 89C51 的週邊加上一個 MAX232 IC，功能為微控器與 PC 端之間串列埠通信的控制，以做為 RF 模組與 PC 間資料傳輸的管道，而串列傳輸介面的鮑率 (Baud Rate) 設定為 2400bps，其整個介面電路的實體圖如下圖 4-10。



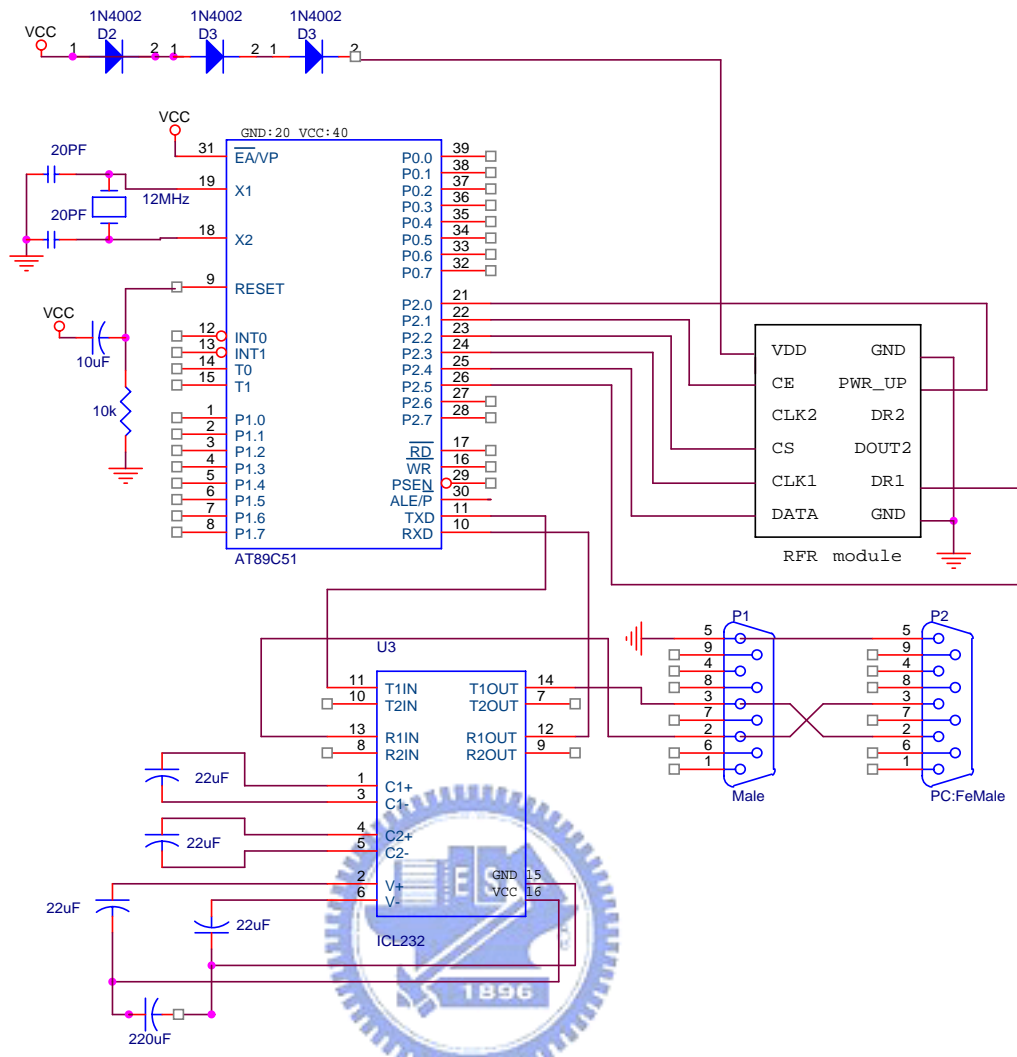


圖 4-9 RF 模組之微控器介面電路圖

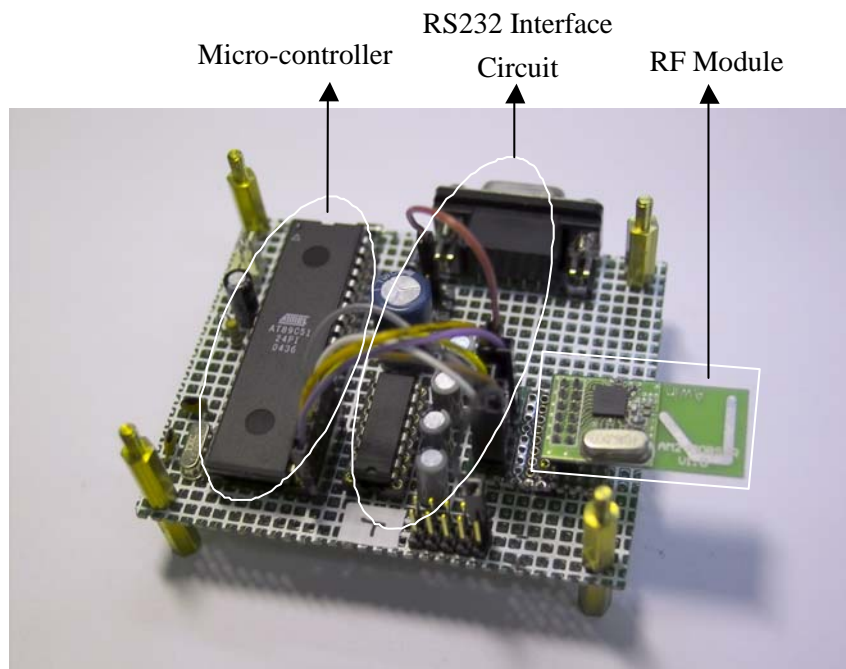


圖 4-10 RF 模組與微控器介面電路實體圖

4.3 多機器人無線通訊協定

在多機器人之無線通訊網路的實現上，每台實體機器人接裝備兩個如 4.2 節所述之 RF 無線傳輸模組電路，其中一個做為接受器，另一個則做為傳送器，已達成全雙工之通訊功能。在通訊頻道上，所以之 RF 模組接設定在頻道 0 (channel #0) 的模式下，因此任何一台機器人皆可透過廣播(broadcast)的方式將訊息傳送給其他所有的機器人隊友，其最大可靠傳輸距離為 5m。

在通訊協定上，無線傳輸訊息的命令格式為 1 Byte 大小之代碼，其種類包含了：(1) 請求協助命令 (2) 隊形就位訊息 (3) 隊形起步命令。當任何一台機器人首先找到目標物後，此時便進入 Master 模式並傳送求援命令，由於要讓其他隊員了解是誰成為了 Leader，因此對於每一台機器人而言各有一個屬於自己的請求協助命令代碼。當其他隊員(Follower Robot)靠近隊長且已就隊形準備位置後再各自傳送隊形就位訊息；最後當 Leader Robot 收到來自所有 Follower Robot 之隊形就位訊息時則再傳送隊形起步命令，通知所有 Follower Robot 要進入隊形維持的行為中並起步帶領全隊前往目的地，其詳細之無線通訊代碼如表 4-1 所示(所有情況之 Master-Slave 關係如表 4-2 所列)。

表 4-1 機器人無線通訊協定之訊息代碼

Code	Type	Sender	Receiver	Information Meaning
0xAA	Request for help	Master (Robot 1)	All Slaves	Robot 1 becomes the leader robot and needs assistance.
0xAB	Request for help	Master (Robot 2)	All Slaves	Robot 2 becomes the leader robot and needs assistance.
0xAC	Request for help	Master (Robot 3)	All Slaves	Robot 3 becomes the leader robot and needs assistance.
0xBB	Formation Ready	Slave 1	Current Master	Slave 1 is already on the formation position.
0xCC	Formation Ready	Slave 2	Current Master	Slave 2 is already on the formation position.
0xDD	Formation Go	Current Master	All Slaves	Start to go to the desired location with formation.

為了避免產生通訊雜訊所產生之誤動作，因此我們在實際設計上當訊息傳送者要傳送訊息代碼時會連續送出五次，當訊息接收者接受到三次以上同樣之訊息時則判斷此為正確之機器人無線通訊。經由這樣的偵錯判斷機制可大幅降低通訊雜訊所帶來的負面影響，經由實際測試結果證明其訊息可靠程度在 5m 的範圍內將近 100%。

表 4-2 不同 Master 情況下之 Slave 相對關係

	Master (Leader)	Slave 1 (Follower 1)	Slave 2 (Follower 2)
Case 1	Robot 1	Robot 2	Robot 3
Case 2	Robot 2	Robot 3	Robot 1
Case 3	Robot 3	Robot 1	Robot 2



第五章 模擬結果

本章將以軟體模擬來驗證本文在 2.2 節所提出之適應性基於行為模式之多機器人協同控制架構的可行性。在下面的 5.1 節至 5.5 節裡將分別以不同模擬環境場景來實現三台機器人(Robot 1 – Robot 3)之協調隊形組成與維持控制，在不同的場景環境裡各自有不同的機器人起始位置及目標物位置和障礙物組成，在最後 5.6 節將針對此四個不同的模擬做結果分析與討論。所使用之模擬環境為利用 Borland C++ Builder 軟體所撰寫出來之多機器人協同控制模擬程式，由即時的圖形介面輸出可以觀察到各個機器人的動向及其狀態，對於使用者在調整參數及偵錯方面有相當大的幫助性。

下面之模擬環境場景為 10m x 10m 之封閉環境，四面之邊界可視為障礙物，地圖中每格長為 50 cm。由模擬環境之場景狀態圖之左方狀態列可看出各機器人現在所處之狀態模式(Mode)、線速度(m/sec)、角速度(rad/sec)以及目前所執行之行為(Behavior)。

在模擬中，三台機器人會隨機在不同的起始位置及朝向角，啟動後即各自自由探索目標物的所在，當其中有任何一台發現目標物時會先接近後再發送求援命令給另外兩台機器人隊友，在目標物之地點共同組成預定隊形後隊長開始等領團隊前往預定目的地，沿途中另外兩個機器人隊員會盡量與隊長保持預定的隊形，即使在有碰撞物的情況下也會自主式地先執行閃避再企圖回到隊形位置。

5.1 模擬環境一

■ 任務說明

一開始各機器人之起始位置分別如下：Robot 1 (0, 1, -90°)、Robot 2 (-1, -2, 90°)及 Robot 3 (1.5, -2, 90°)；目標物擺放在座標(3, 3)的位置，在場景圖 5-1 中以右上角之紅色實心圓表示；預定目的地之座標為(-3, -3)，即地圖中之左下角位置；預定之機器人隊形為三角形(Triangle Formation，機器人間隔為 1.0m)，而六個障礙物之座標位置分別在座標(0, 0)、(-2, -1)、(1, -3)、(3, -2)、(-3, 2)、(-1, 3)的位置上。此模擬將觀察在本文 2.2 節所提出的多機器人協同控制架構運作下之各機器人的行為反應及隊形任務執行的效果。

■ 模擬結果及討論

下圖 5-2 與 5-3 為模擬一裡三台機器人的軌跡紀錄(實心圓點表軌跡起點，空心圓點表軌跡終點)：在時間 $t=0$ 時，系統開始運作，此時三台機器人分別由不同之起始位置出發，由於 Robot 1 十分靠近障礙物的緣故，此時它優先執行閃避障礙物的行為，同時間的另外兩台機器人則在進行自由探索的動作，由圖 5-5 即可看出各機器人在時間下的行為變化，其符號表示如下：

Goal : Goal Approaching

Waiting : Waiting


Format : Formation Maintenance

Master : Master Approaching

Target : Target Approaching

Avoid : Object Avoidance (Obstacle Avoidance & Inter-Robot Collision Avoidance)

Wander : Wandering



時間 $t=10$ ，Robot 1 和 Robot 3 由於彼此太接近的緣故，同時執行閃避機器人之行為。時間 $t=33$ ，Robot 3 在右上角發現目標物後靠近它，此時其狀態由一般模式變成 Master 模式，而 Robot 1 及 Robot 2 則由一般模式變成 Slave 模式(各機器人之模式變化如圖 5-4 所示)。時間 $t=37$ ，Robot 1 (Follower 1)閃避完障礙物後開始執行靠近隊長的行為。時間 $t=43$ ，Robot 2 (Follower 2)在閃避完障礙物後也同樣開始執行靠近隊長的行為。時間 $t=66$ ，Robot 1 及 Robot 2 皆進入其隊形位置(已轉正其朝向角與 Robot 3 一致)，此時 Robot 3 (Leader Robot)開始起動，朝左下角之預定目的地移動。時間 $t=82$ ，Robot 3 發現前方有障礙物，開始執行閃避之功能。時間 $t=85$ ，Robot 1 及 Robot 2 分別朝障礙物的兩側執行閃避動作。時間 $t=91$ ，Robot 2 在閃避完障礙物後試圖回到原先之隊形位置，此時 Robot 1 則正在保持隊形中。時間 $t=98$ ，Robot 2 回到隊形位置，輪到 Robot 1 閃避最下方之障礙物。時間 $t=108$ ，Robot 1 亦回到其隊形位置，此時三台機器人在沒有障礙物的情況下持續維持住三角形的隊形。時間 $t=111$ ，Robot 3 帶隊抵達預定目的地，最終三台機器人維持住預定之三角形隊形，順利地結束了任務。

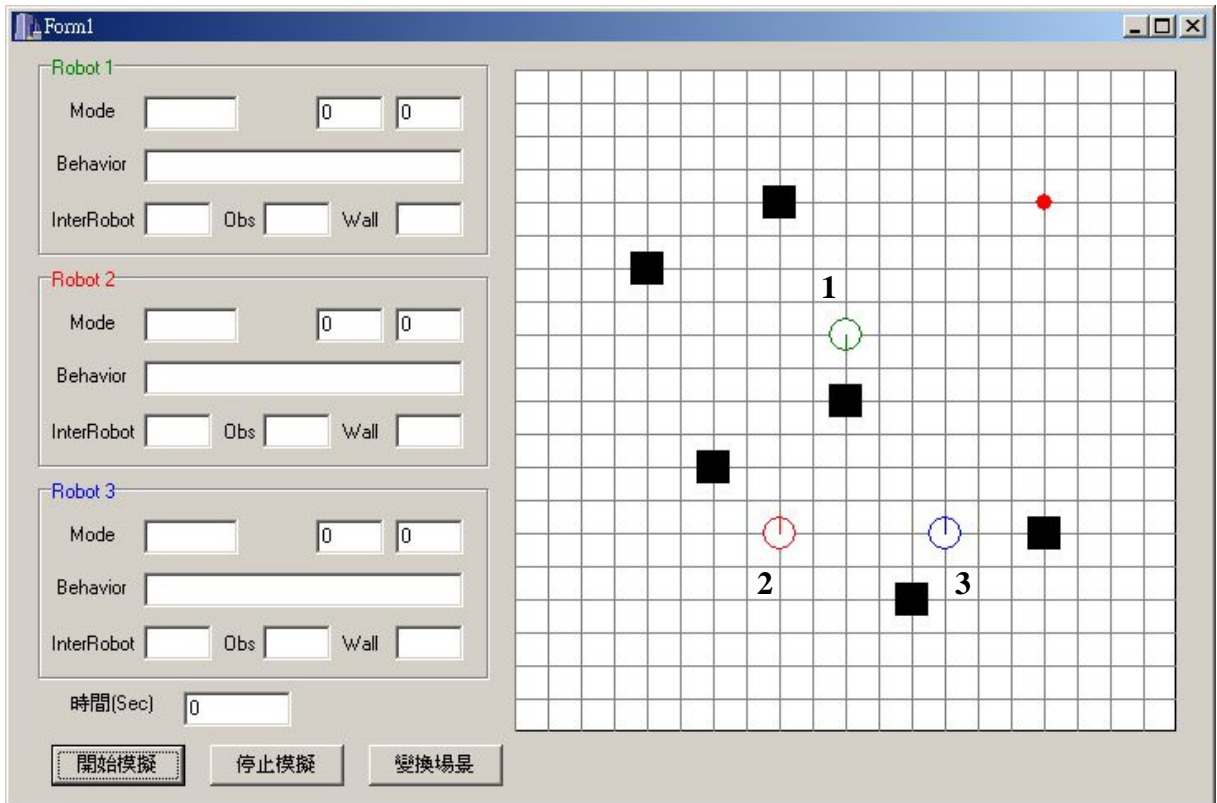


圖 5-1 模擬環境一之場景圖

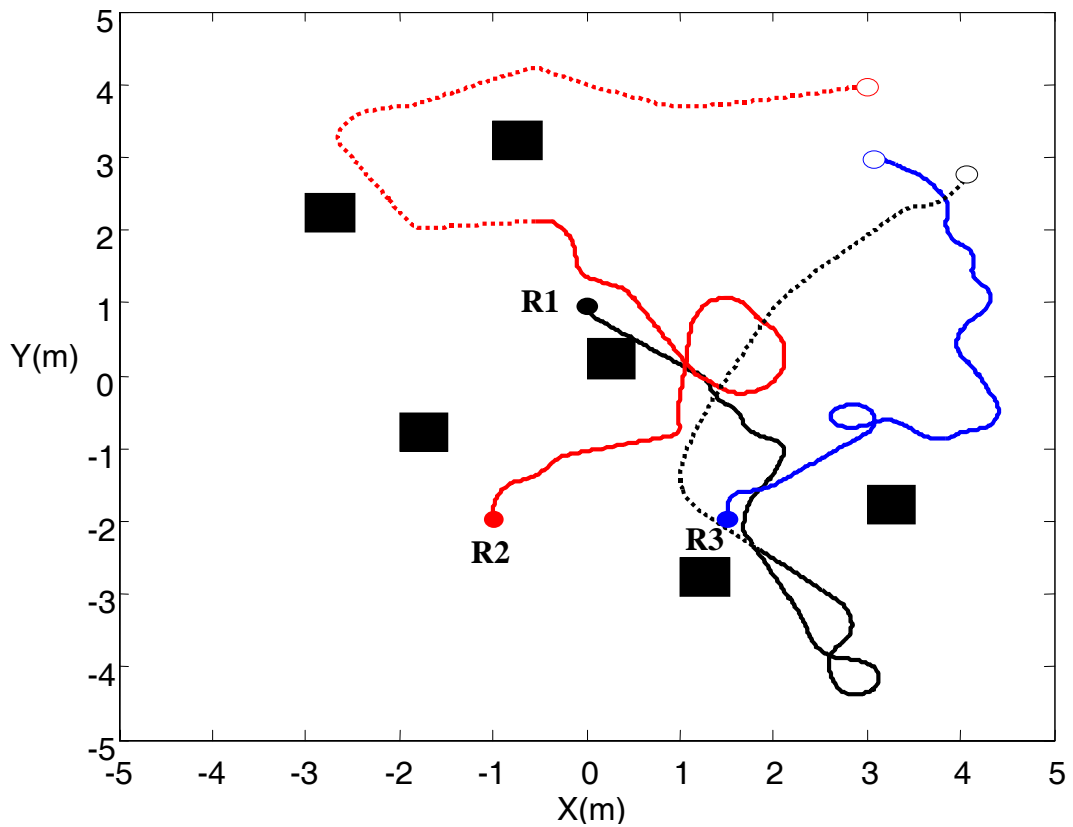


圖 5-2 模擬環境一之各機器人軌跡圖(由開始至隊形組成，虛線部分為靠近隊長之軌跡)

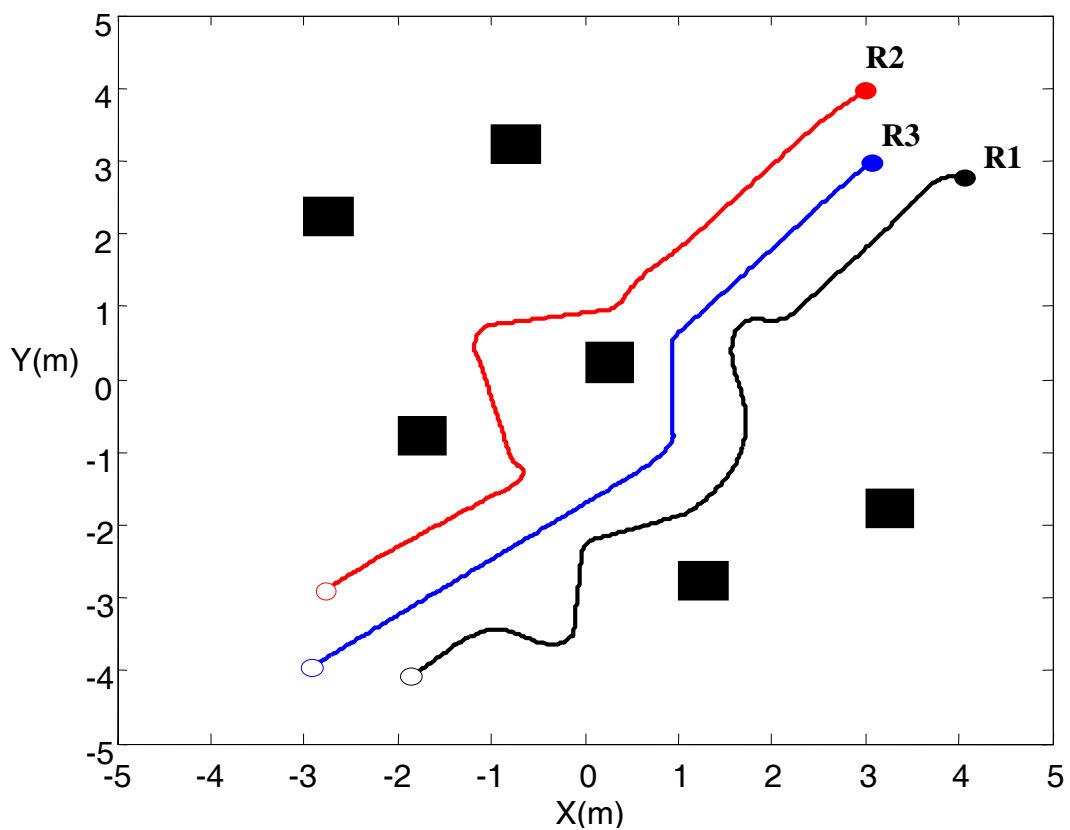


圖 5-3 模擬環境一之各機器人軌跡圖(隊形維持過程)

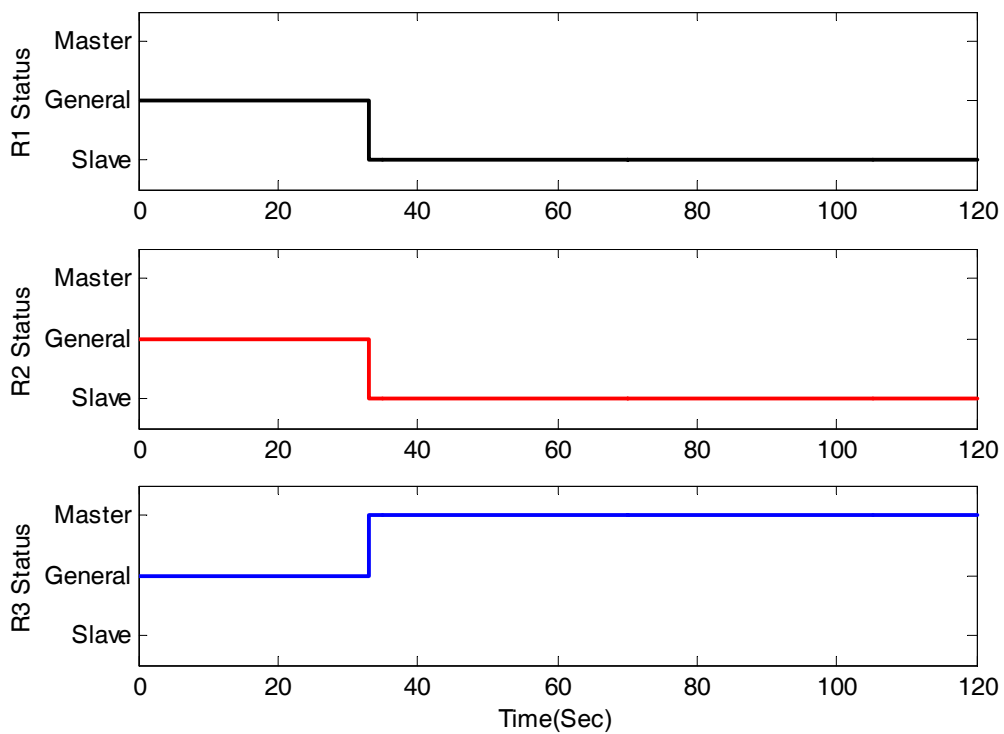


圖 5-4 模擬環境一之各機器人模式狀態變化

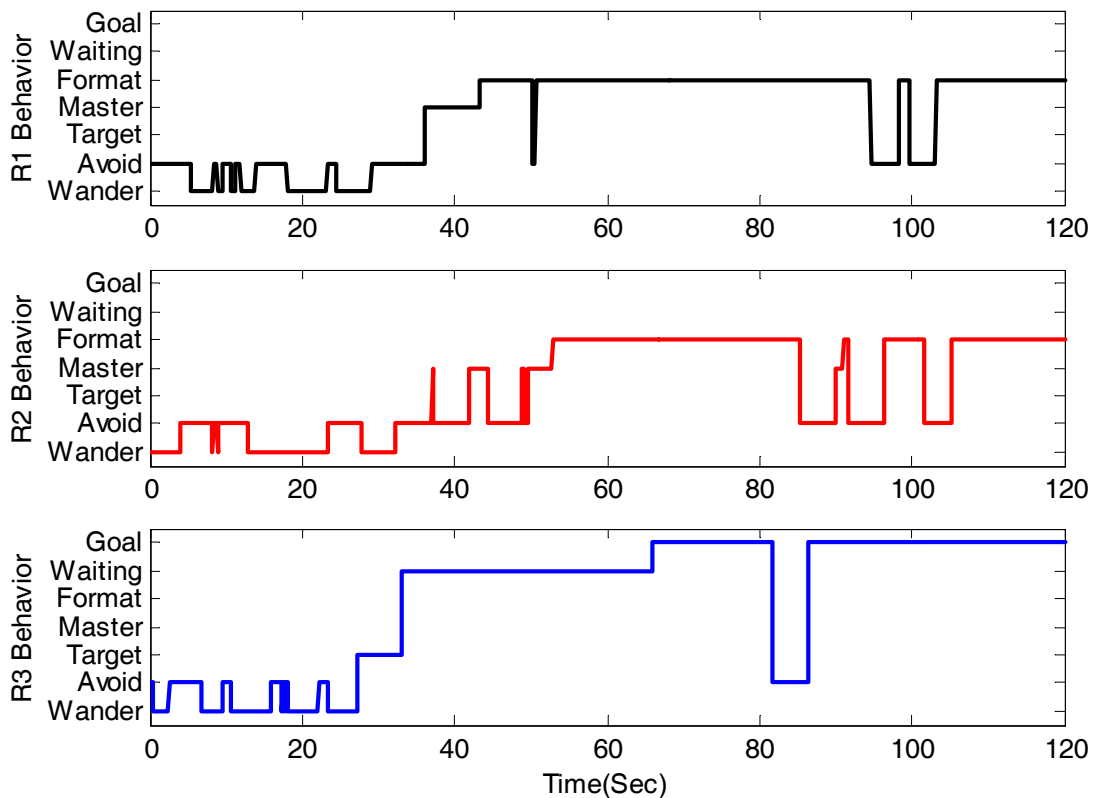


圖 5-5 模擬環境一之各機器人行為狀態變化



5.2 模擬環境二

■ 任務說明

一開始各機器人之起始位置分別如下：Robot 1 $(-4, 1, 90^\circ)$ 、Robot 2 $(2, -4, 45^\circ)$ 及 Robot 3 $(2, 2, 0^\circ)$ ；目標物擺放在座標 $(-3, -3)$ 的位置，即場景圖 5-6 中左下角之紅色實心圓；預定目的地之座標為 $(4, 3)$ ，即地圖中之右上角位置；預定之機器人隊形為三角形 (Triangle Formation，機器人間隔為 1.0m)，而七個障礙物之座標位置分別在座標 $(0, 0)$ 、 $(-3, 2)$ 、 $(3, -2)$ 、 $(-1, 3)$ 、 $(0, -3)$ 、 $(-3, -0.5)$ 、 $(1.5, 3.5)$ 的位置上。本模擬同樣將觀察本文 2.2 節所提出的多機器人協調控制架構運作下之任務執行效果。

■ 模擬結果及討論

下圖 5-7 與 5-8 為模擬二裡三台機器人的軌跡紀錄(實心圓點表軌跡起點，空心圓點

表軌跡終點)：在時間 $t = 0$ 時，系統開始運作，此時三台機器人分別由不同之起始位置出發，由圖 5-10 可看出各機器人在時間下的行為變化。在時間 $t = 46$ 之前，三台機器人皆在執行隨機探索與閃避障礙物兩種行為之切換。時間 $t = 46$ ，Robot 1 在左下角發現目標物後靠近它，此時開始轉正自己之朝向角對向預定目的地，同時其狀態由一般模式變成 Master 模式，而 Robot 2 及 Robot 3 則由一般模式變成 Slave 模式(各機器人之模式變化如圖 5-9)。時間 $t = 55$ ，Robot 2 (Follower 1)閃避完障礙物後開始執行靠近隊長的行為。時間 $t = 59$ ，Robot 3 (Follower 2)在閃避完障礙物後也同樣開始執行靠近隊長的行為。時間 $t = 85$ ，Robot 2 及 Robot 3 皆進入其隊形位置(已轉正其朝向角與 Robot 1 一致)，此時 Robot 1 (Leader Robot)開始起動，朝右上角之預定目的地移動。時間 $t = 100$ ，Robot 1 發現前方有障礙物，開始朝右執行閃避的行為。時間 $t = 105$ ，Robot 2 朝左方執行對障礙物的閃避動作。時間 $t = 114$ ，Robot 2 在閃避完障礙物後試圖回到原先之隊形位置，此時 Robot 1 正在保持隊形中。時間 $t = 118$ ，Robot 2 回到隊形位置，此時三台機器人開始維持住三角形之隊形。時間 $t = 132$ ，Robot 1 帶隊抵達預定目的地，最終此三台機器人以三角形之隊形順利結束任務。

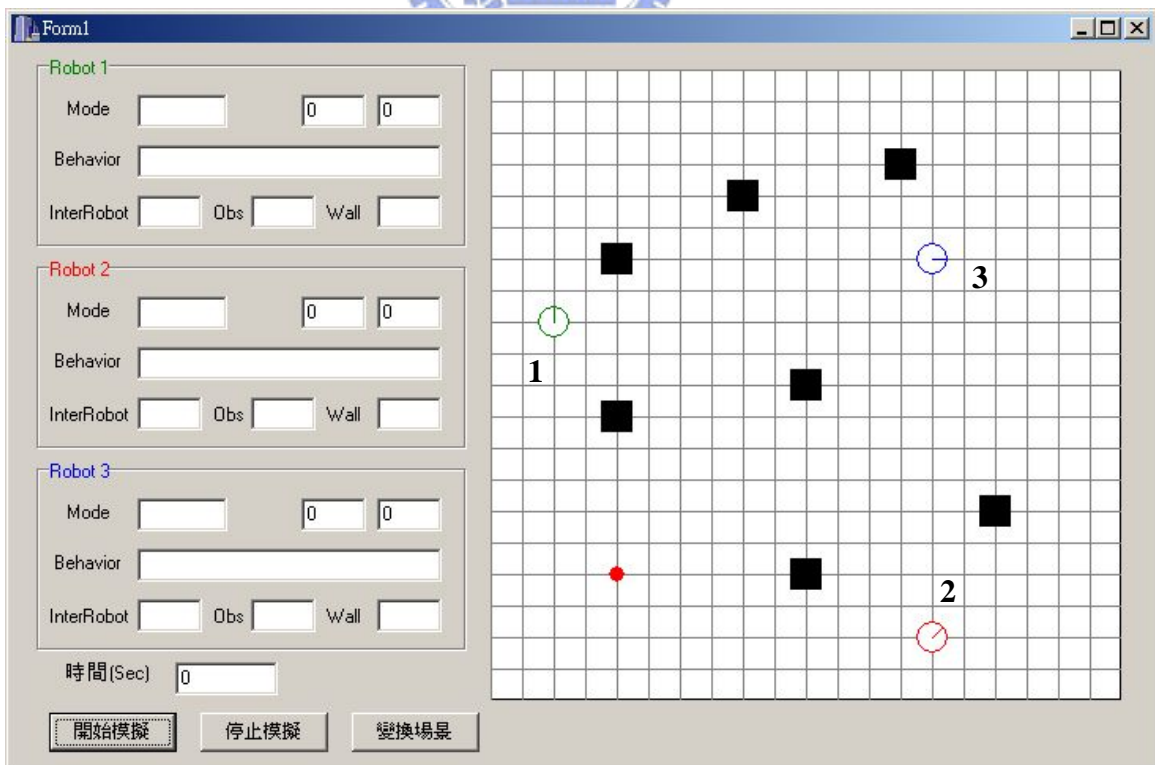


圖 5-6 模擬環境二之場景圖

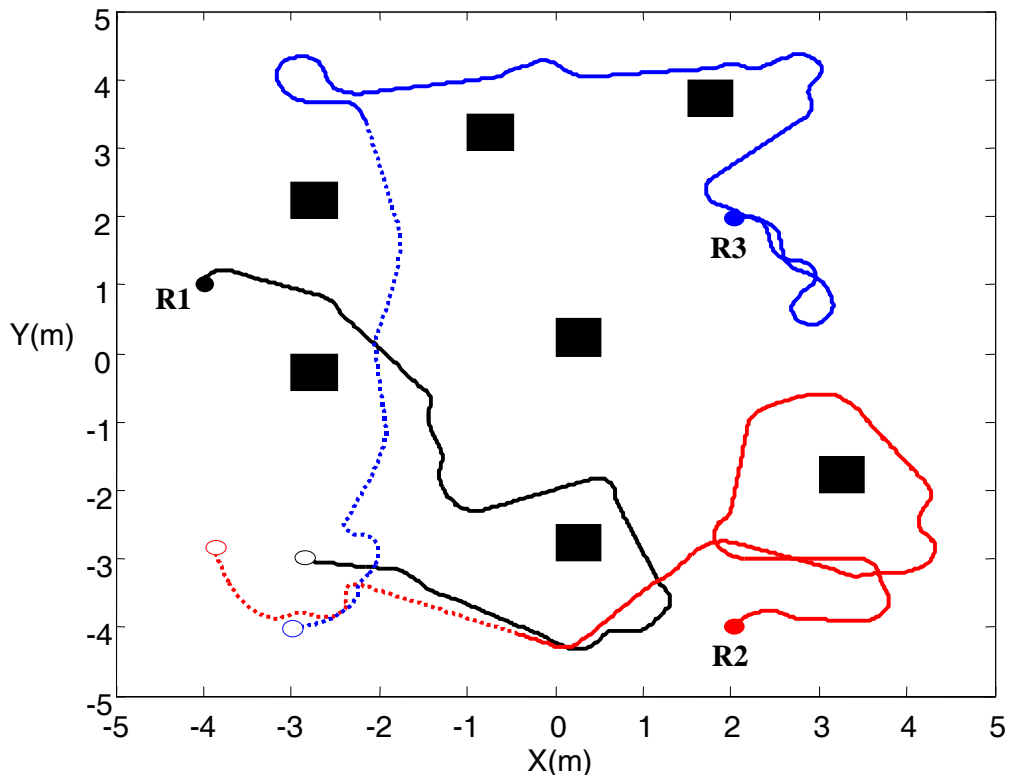


圖 5-7 模擬環境二之各機器人軌跡圖(由開始至隊形組成，虛線部分為靠近隊長之軌跡)

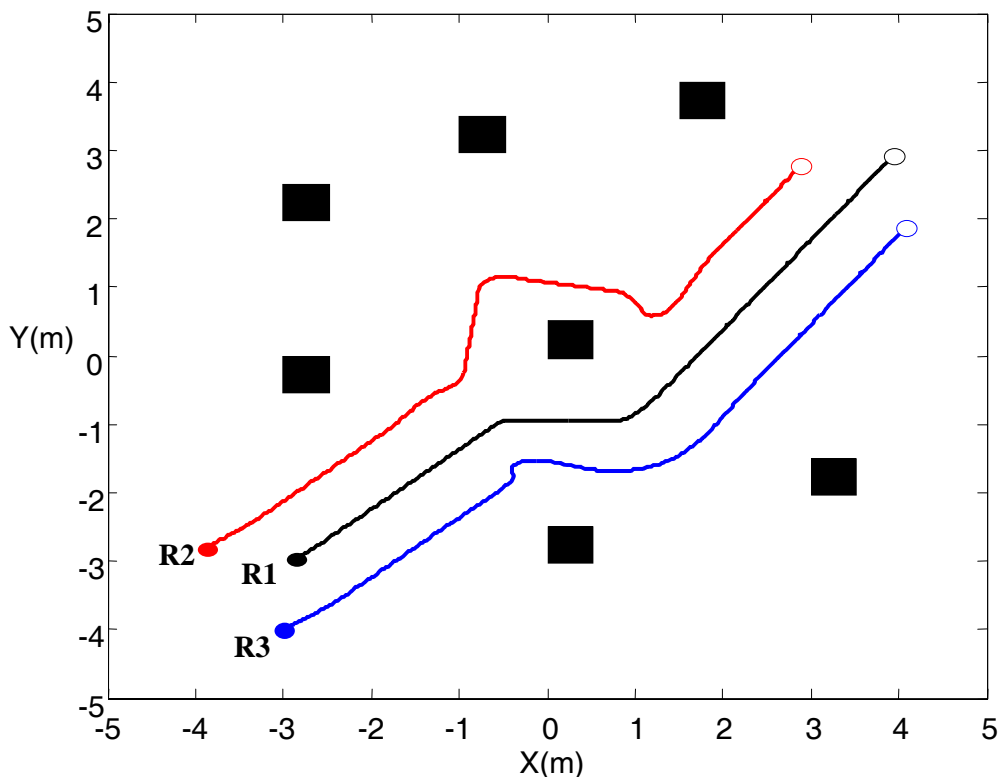


圖 5-8 模擬環境二之各機器人軌跡圖(隊形維持過程)

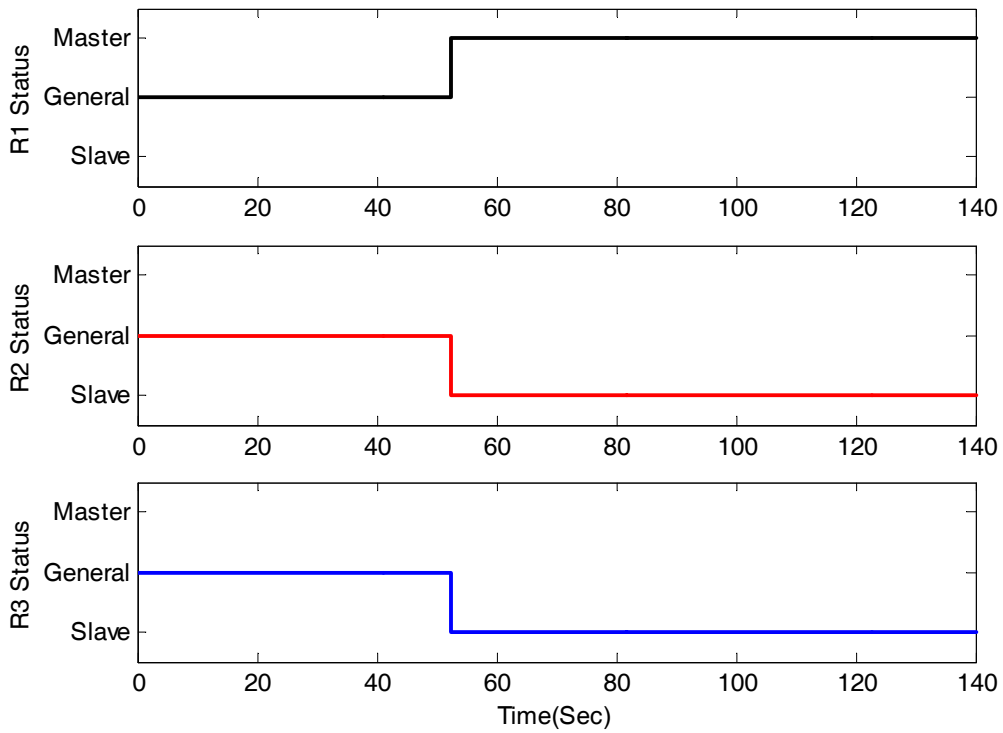


圖 5-9 模擬環境二之各機器人模式狀態變化

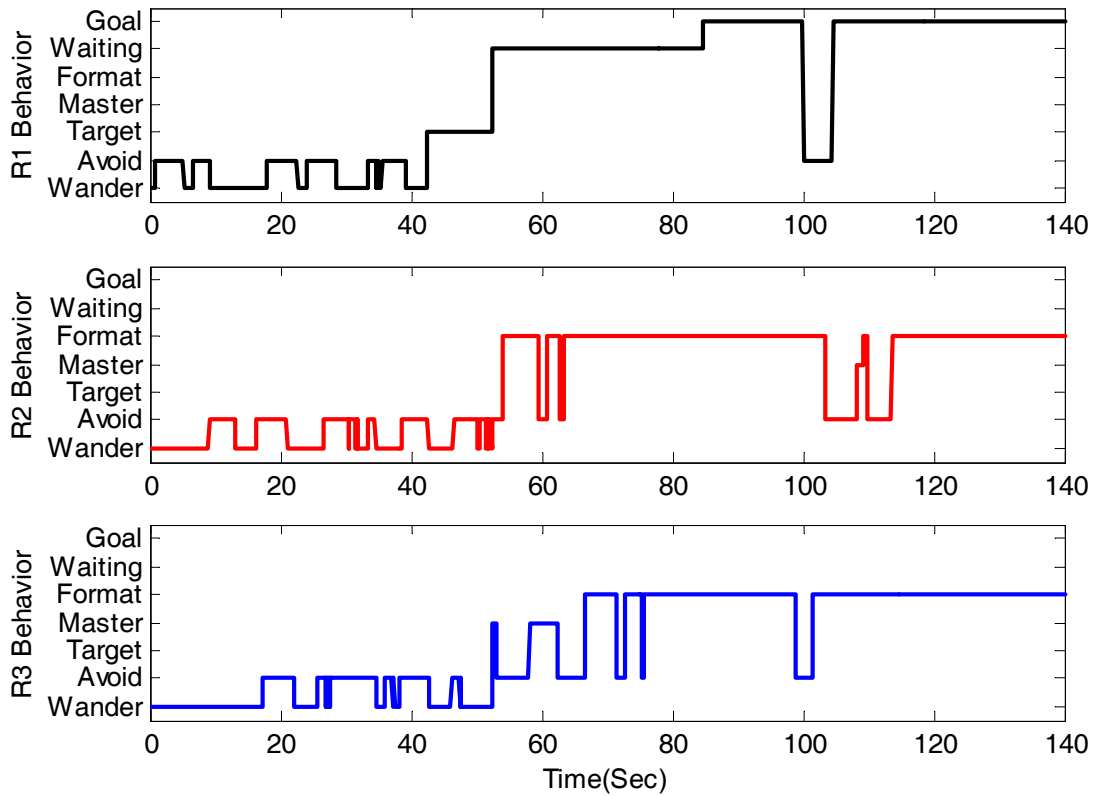


圖 5-10 模擬環境二之各機器人行為狀態變化

5.3 模擬環境三

■ 任務說明

一開始各機器人之起始位置分別如下：Robot 1 (0, 3, 45°)、Robot 2 (2, -2, 45°) 及 Robot 3 (1, -1, -90°)；目標物擺放在座標(0, -4)的位置，即場景圖 5-11 中正下角之紅色實心圓；預定目的地之座標為(0, 4)，即地圖中之正上角位置；預定之機器人隊形為線形隊形(Line Formation，Follower Robot 與 Leader Robot 之間隔為 1.2m)，而六個障礙物之座標位置分別在座標(2, 1)、(-3, 2)、(3, 3)、(4, -1)、(-1, -1)、(-3, -3)的位置上。

■ 模擬結果及討論

下圖 5-12 與 5-13 為模擬三裡三台機器人的軌跡紀錄(實心圓點表軌跡起點，空心圓點表軌跡終點)：在時間 $t = 0$ 時，系統開始運作，此時三台機器人分別由不同之起始位置出發，由圖 5-15 可看出各機器人在時間下的行為變化。在時間 $t = 20$ ，由於 Robot 1 和 Robot 3 由於彼此太接近的關係，此時同時執行閃避機器人之行為。時間 $t = 44$ ，Robot 3 在地圖下方發現目標物的存在並靠近它，此時開始轉正自己之朝向角對向預定目的地(理應對準正上方，不過由於在行為中加了臨界角之誤差模擬的緣故，實際大約有 8° 之角度偏差)，同時其狀態由一般模式變成 Master 模式，而 Robot 1 及 Robot 2 則由一般模式變成 Slave 模式(各機器人之模式變化如圖 5-14)，此刻同時由於 Robot 2 (Follower 2) 其視覺範圍內已探測到 Leader Robot (Robot 3)，因此直接朝向其所對應的隊形位置而去。時間 $t = 55$ ，Robot 1 (Follower 1) 閃避完障礙物後開始執行靠近隊長的行為。時間 $t = 66$ ，Robot 1 及 Robot 2 皆進入其隊形位置(已轉正其朝向角與 Robot 3 一致，但仍有 8° 左右之角度偏差)，此時 Robot 3 (Leader Robot) 開始起動，朝地圖上方之預定目的地移動。時間 $t = 77$ ，Robot 1 發現前方有障礙物，開始朝左執行閃避的行為。時間 $t = 78$ ，Robot 3 亦朝右執行閃避障礙物之行為。時間 $t = 87$ ，Robot 1 在閃避完障礙物後試圖回到其原先之隊形位置，此時 Robot 2 正遇到右方之障礙物並開始閃避之。時間 $t = 99$ ，三台機器人開始維持住線形隊形。時間 $t = 106$ ，Robot 3 帶隊抵達預定目的地，最終此三台機

器人以線形隊形結束此次的任務。

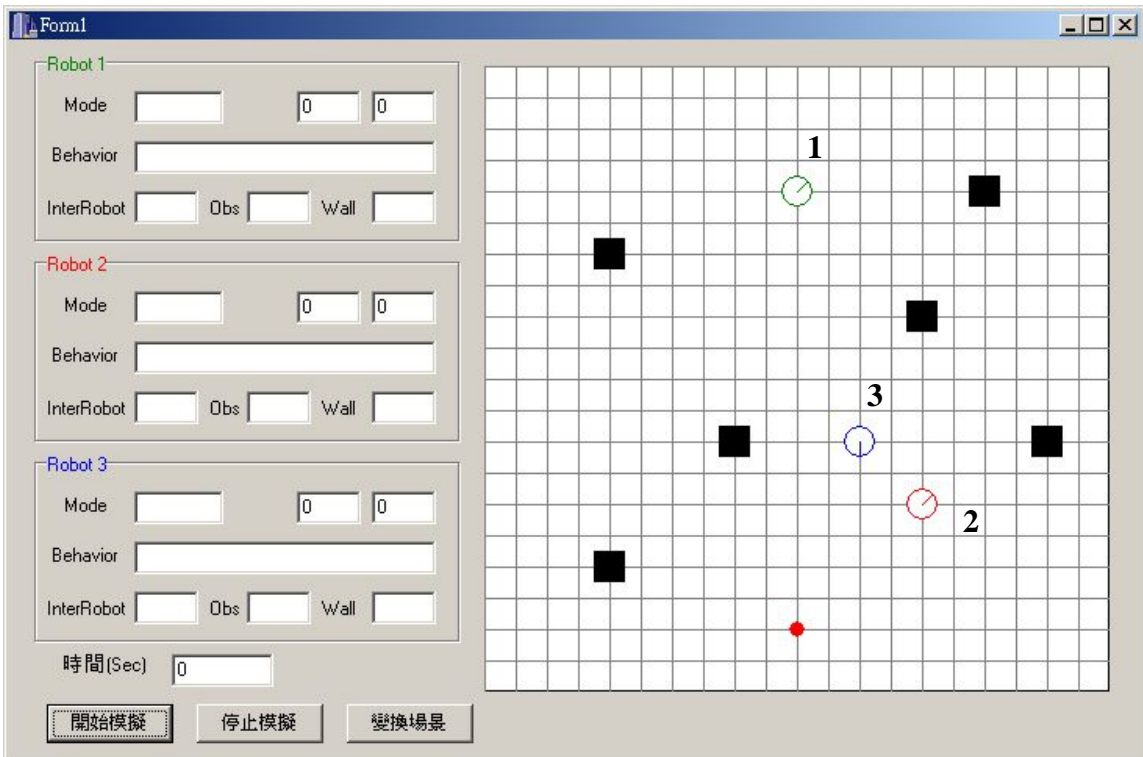


圖 5-11 模擬環境三之場景圖

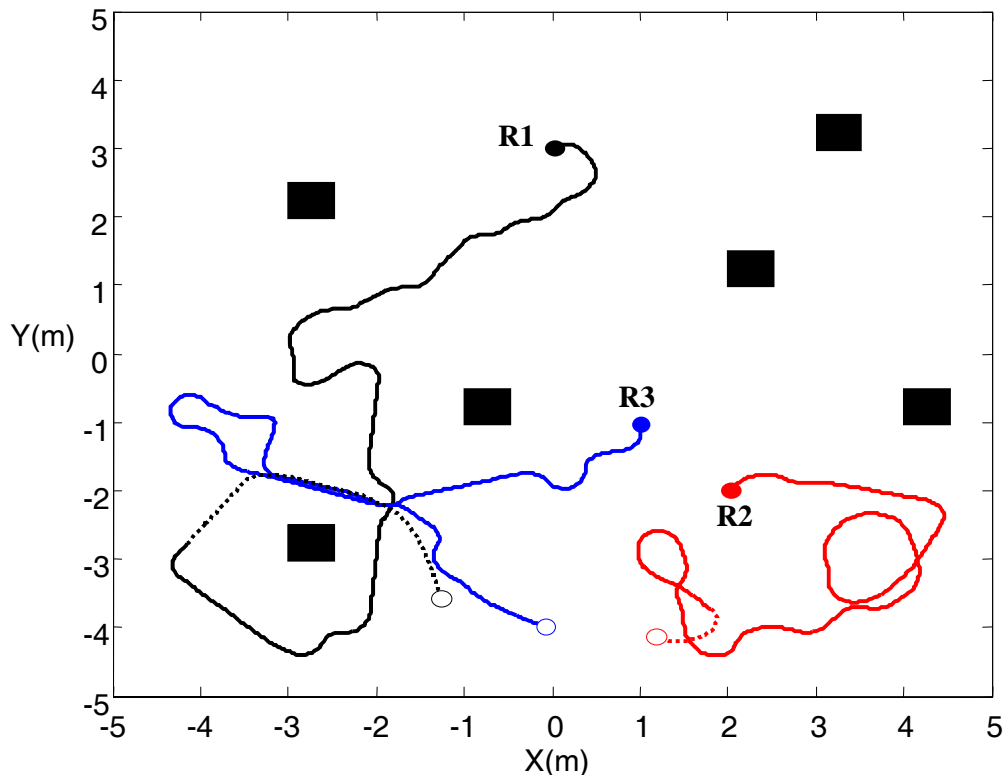


圖 5-12 模擬環境三之各機器人軌跡圖(由開始至隊形組成, 虛線部分為靠近隊長之軌跡)

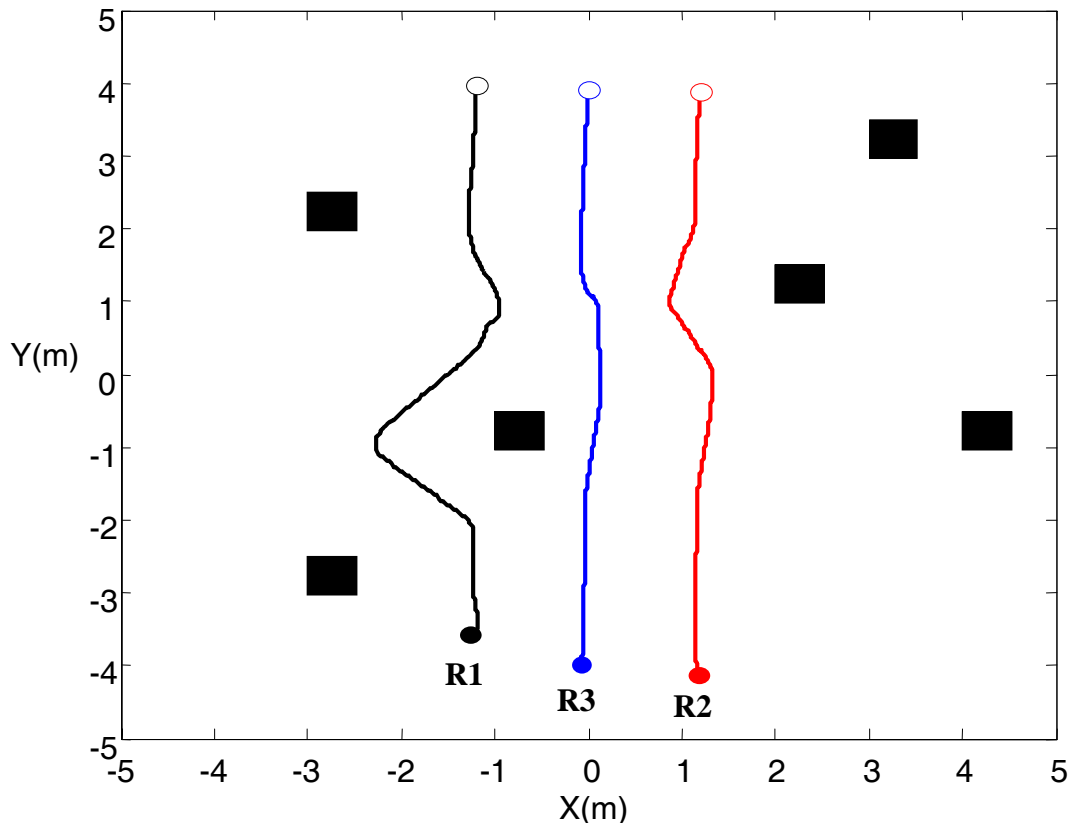


圖 5-13 模擬環境三之各機器人軌跡圖(隊形維持過程)

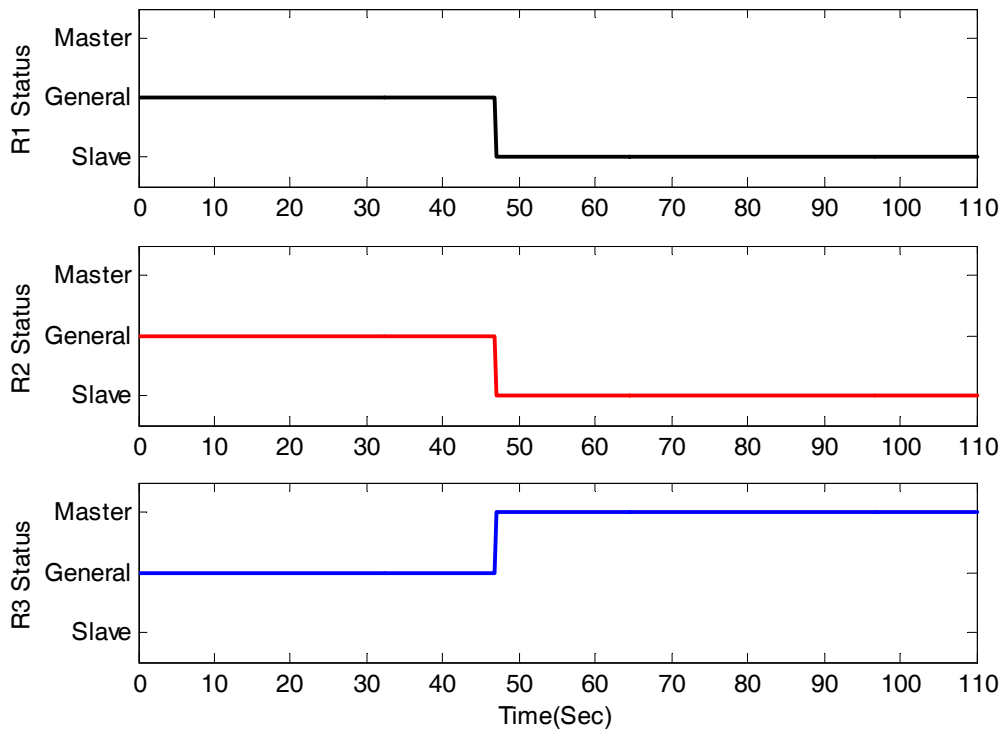


圖 5-14 模擬環境三之各機器人模式狀態變化

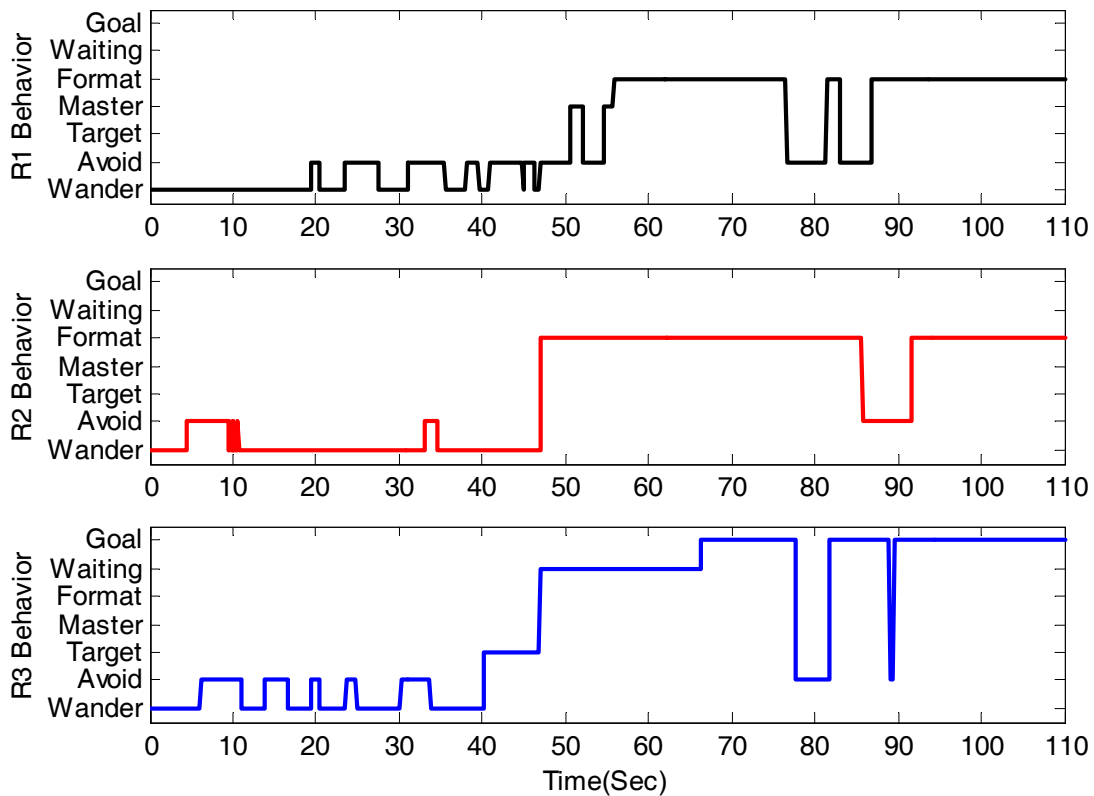


圖 5-15 模擬環境三之各機器人行為狀態變化



5.4 模擬環境四

■ 任務說明

一開始各機器人之起始位置分別如下：Robot 1 (2, -4, 45°)、Robot 2 (2, 2, 0°)及 Robot 3 (-4, 1, 90°)；目標物一開始擺放在座標(-3, -3)的位置，即場景圖 5-16 中左下角之紅色實心圓；預定目的地之座標為(4, 3)，即地圖中之右上角位置；預定之機器人隊形為三角形(Triangle Formation，機器人間隔為 1.0m)。本模擬環境將模擬當目標物突然消失時，機器人團隊之狀態是否會重置(Reset)並回到一般模式下再次進行分散自主式探索目標物的功能，此模擬將驗證在本文 2.2 節所提出的多機器人協調控制架構之任務重複再生性(Task Recycling)。

■ 模擬結果及討論

下圖 5-17 ~ 圖 5-20 為模擬四裡三台機器人的軌跡紀錄(實心圓點表軌跡起點, 空心圓點表軌跡終點): 在時間 $t=0$ 時, 系統開始運作, 此時三台機器人分別由不同之起始位置出發, 由圖 5-22 可看出各機器人在時間下的行為變化。在時間 $t=41$ 之前, 三台機器人皆在執行隨機探索與閃避障礙物兩種行為之切換。時間 $t=41$, Robot 3 在左下角發現目標物後靠近它, 此時開始轉正自己之朝向角對向預定目的地, 同時其狀態由一般模式變成 Master 模式, 而 Robot 1 及 Robot 2 則由一般模式變成 Slave 模式(各機器人之模式變化如圖 5-21)。時間 $t=43$, Robot 1 (Follower 1) 閃避完障礙物後開始執行靠近隊長的行為。時間 $t=46$, Robot 2 (Follower 2) 在閃避完障礙物後也同樣開始執行靠近隊長的行為。時間 $t=80$, Robot 1 及 Robot 2 皆進入其隊形位置(已轉正其朝向角與 Robot 3 一致), 此時 Robot 1 (Leader Robot) 開始起動, 朝右上角之預定目的地移動。時間 $t=94$, Robot 1 發現前方有障礙物, 開始朝左執行閃避的行為。時間 $t=105$, Robot 2 朝左方執行對障礙物的閃避動作。時間 $t=111$, Robot 2 在閃避完障礙物後試圖回到原先之隊形位置, 此時 Robot 1 正在保持隊形中。時間 $t=122$, Robot 2 回到隊形位置; 時間 $t=126$, 三台機器人開始維持住三角形之隊形。時間 $t=129$, Robot 3 帶隊抵達預定目的地, 最終此三台機器人以三角形之隊形結束第一次任務。時間 $t=136$, 目標物消失, 此時三台機器人再次進入一般模式各自進行探索目標物的功能。時間 $t=181$, 目標物再次出現在座標(3, 3)的位置上, 即地圖中之右上角位置。時間 $t=270$, Robot 3 在探索的過程中再次發現左上角之目標物並靠近它。時間 $t=277$, Robot 3 接近目標物後再次發送請求援助命令給 Robot 1 及 Robot 2 (此兩台機器人再度由一般模式變成 Slave 模式, 如圖 5-21)。時間 $t=315$, 此三台機器人再次在目標物之新位置處組成三角形之隊形並開始朝下一次預定目的地座標(-4, -4)處進行移動。時間 $t=364$, Robot 3 帶隊抵達預定目的地, 三台機器人結束第二次的任務時間(第二次隊形維持過程如軌跡圖 5-20), 最終驗證了此多機器人協調控制架構之任務重複再生性(Task Recycling)。

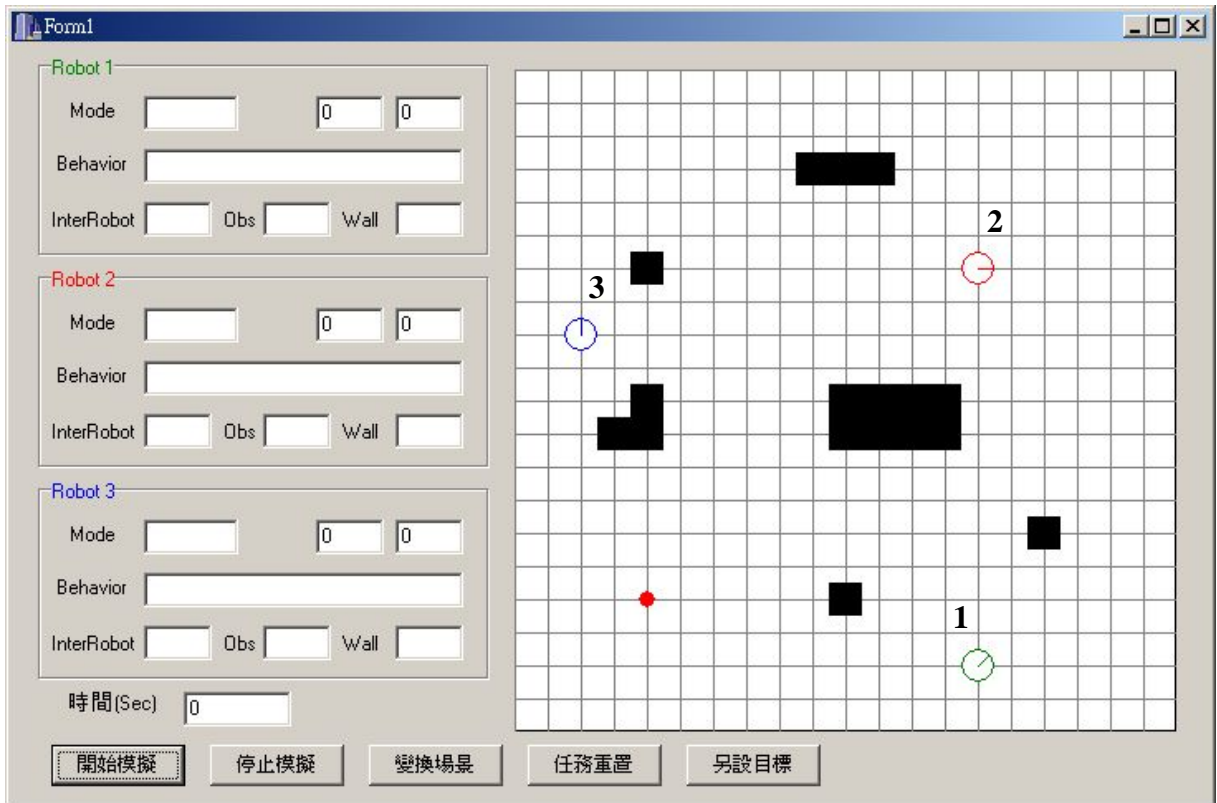


圖 5-16 模擬環境四之場景圖

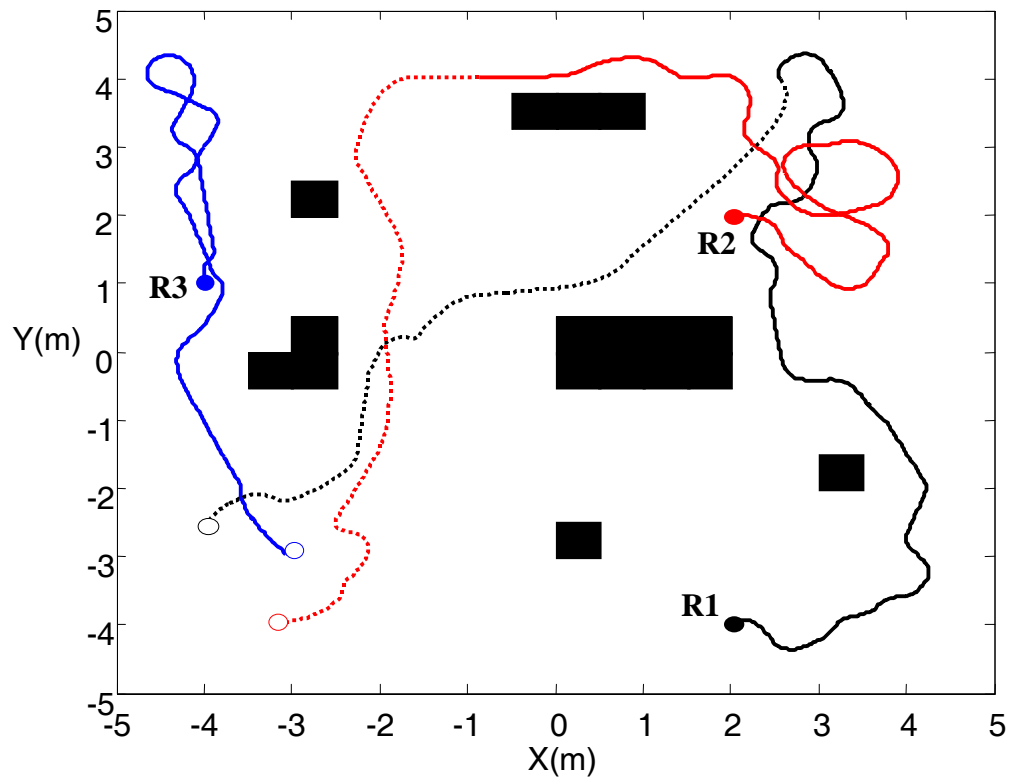


圖 5-17 模擬環境四之各機器人軌跡圖(第一次開始至隊形組成，時間 $t = 0 \sim 80$ 秒)

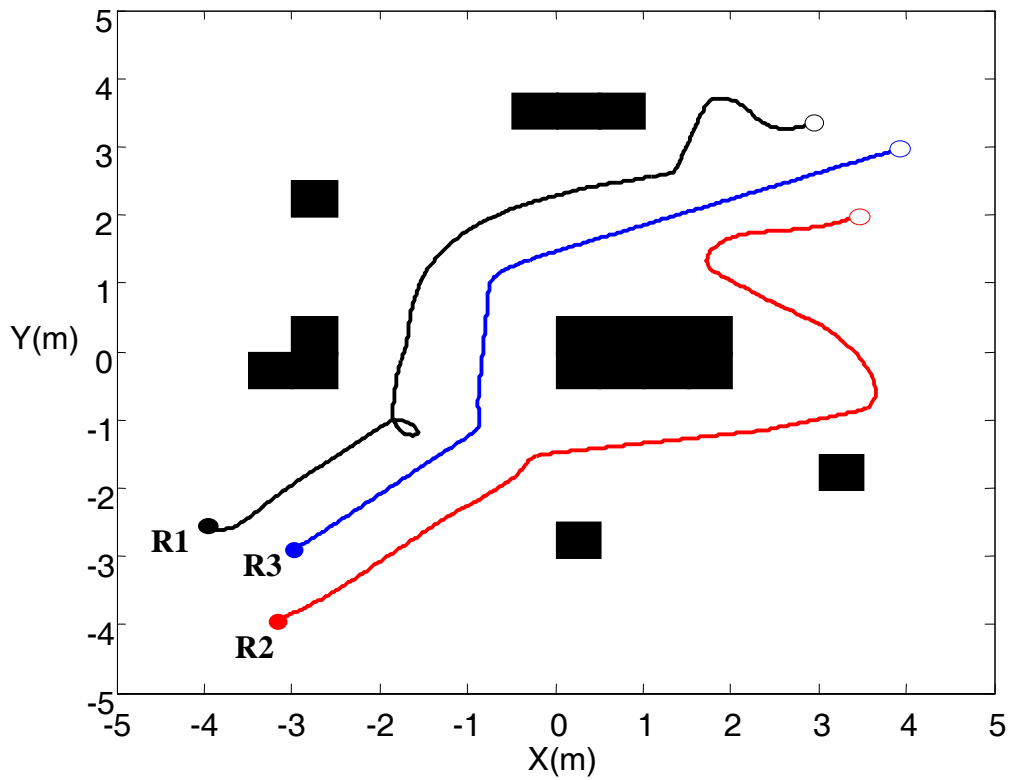


圖 5-18 模擬環境四之各機器人軌跡圖(第一次隊形維持過程，時間 $t = 80 \sim 128$ 秒)

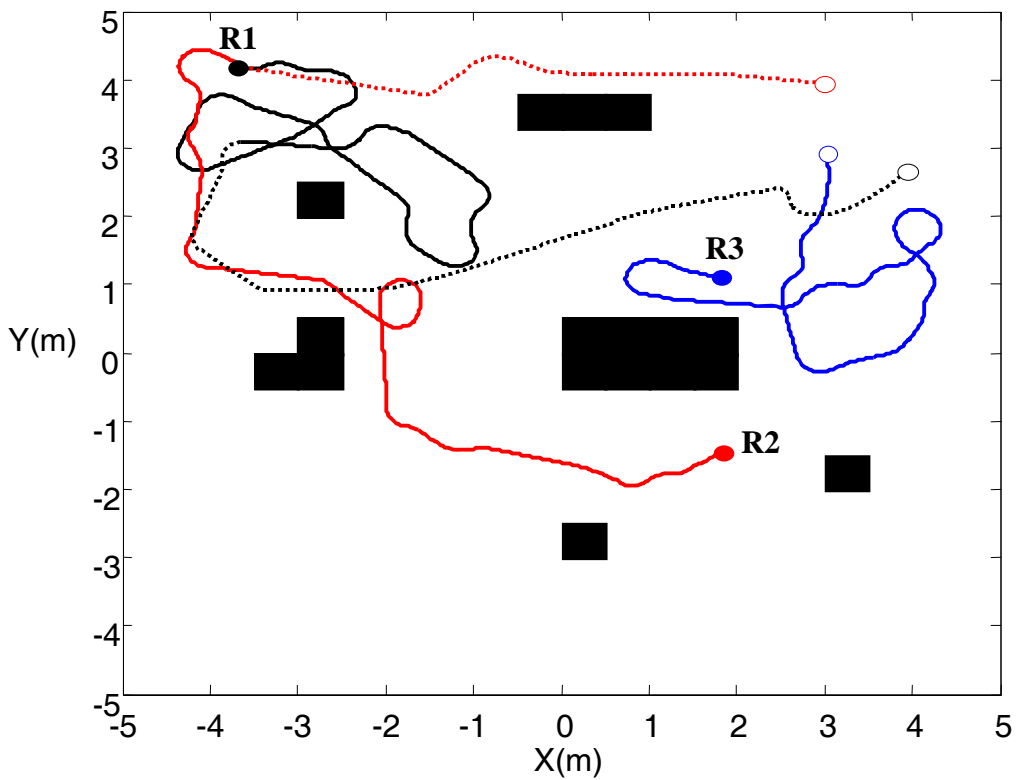


圖 5-19 模擬環境四之各機器人軌跡圖(第二次隊形組成，時間 $t = 230 \sim 312$ 秒)

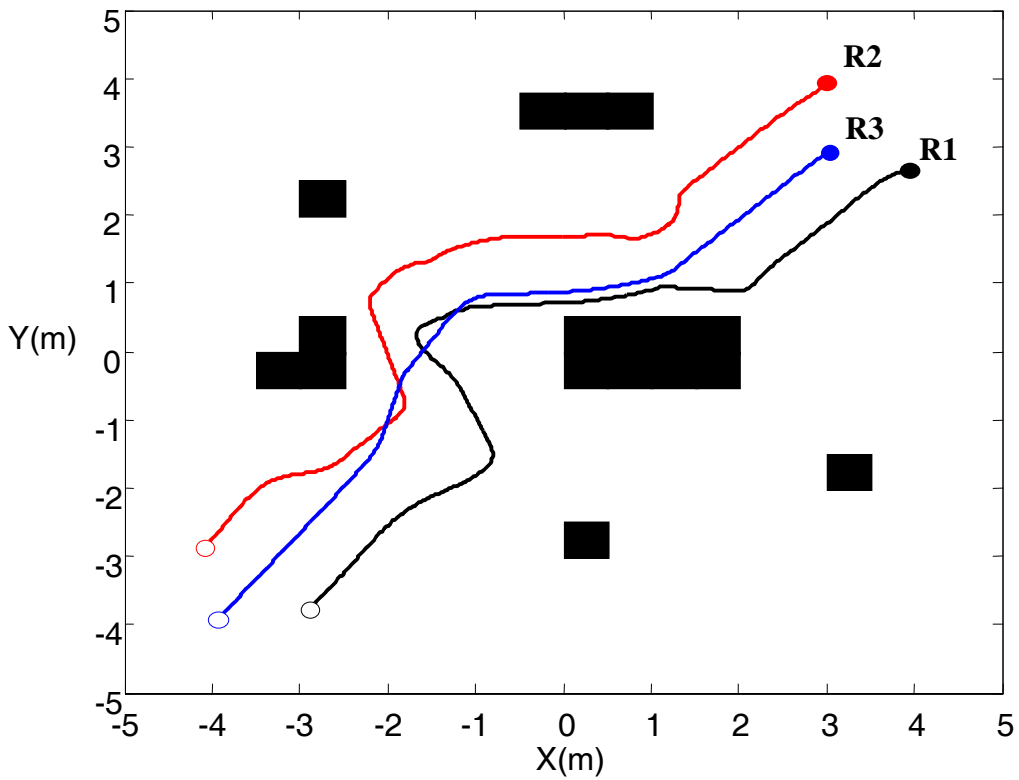


圖 5-20 模擬環境四之各機器人軌跡圖(第二次隊形維持過程，時間 $t = 312 \sim 364$ 秒)

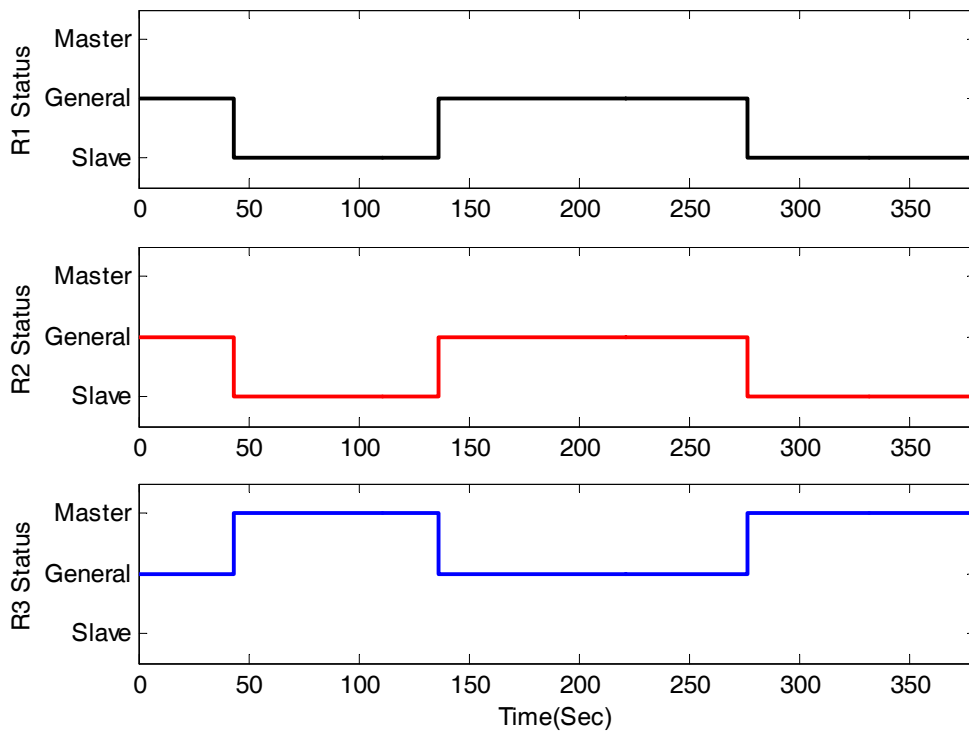


圖 5-21 模擬環境四之各機器人模式狀態變化

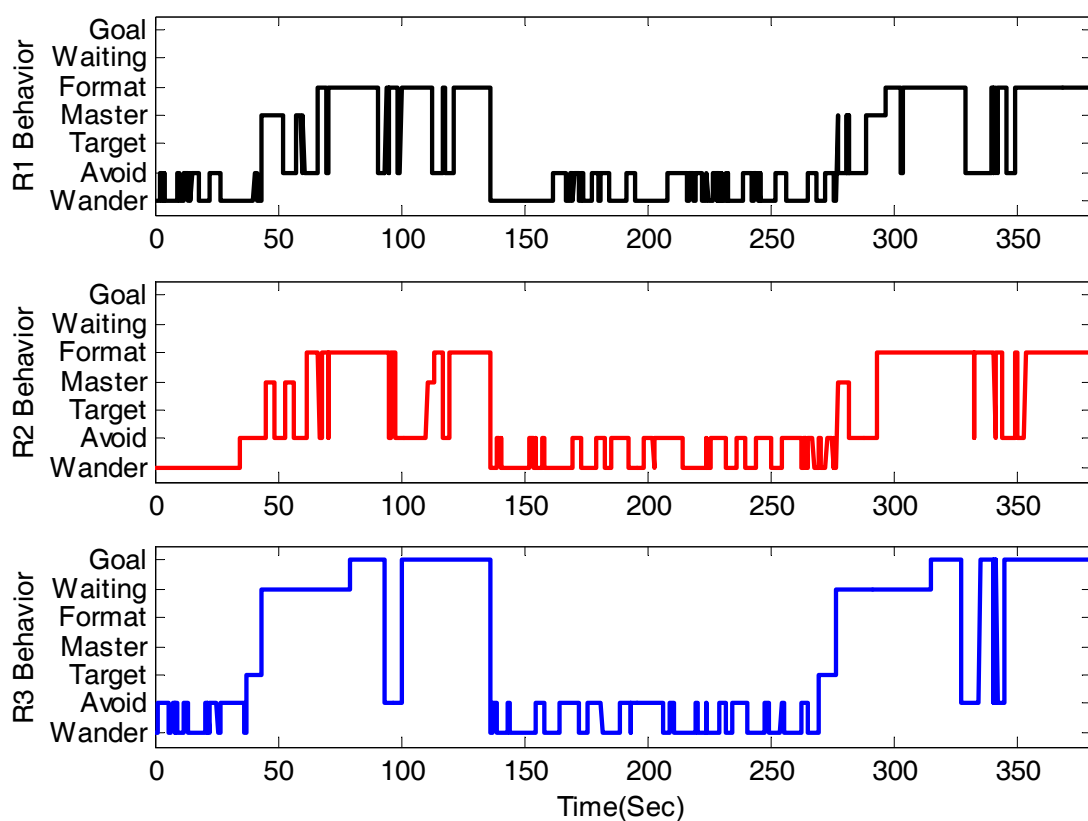


圖 5-22 模擬環境四之各機器人行為狀態變化



5.5 模擬環境五

■ 任務說明

一開始各機器人之起始位置分別如下：Robot 1 ($2, 2, 0^\circ$)、Robot 2 ($-4, 1, 90^\circ$)、Robot 3 ($3, 1, -90^\circ$)及 Robot 4 ($2, -4, -45^\circ$)；目標物擺放在座標 $(-2, -2.5)$ 的位置，即場景圖 5-23 中左下角之紅色實心圓；預定目的地之座標為 $(3, 3)$ ，即地圖中之右上角位置；預定之機器人隊形為三角形(Triangle Formation，機器人間隔為 1.0m)。本模擬將驗證在本文 2.2 節所提出之多機器人協調控制架構能適用在為數眾多的機器人團隊上(前四個模擬為三台機器人，本模擬將機器人數字提昇為四台)。

■ 模擬結果及討論

下圖 5-24 與 5-25 為模擬五裡四台機器人的軌跡紀錄(實心圓點表軌跡起點，空心圓

點表軌跡終點)：在時間 $t = 0$ 時，系統開始運作，此時四台機器人分別由不同之起始位置出發，由圖 5-27 可看出各機器人在時間下的行為變化。在時間 $t = 13$ ，由於 Robot 1 和 Robot 4 由於彼此太接近的關係，此時同時執行閃避機器人之行為。時間 $t = 32$ ，同樣 Robot 2 和 Robot 3 因為彼此太接近而同時採取閃避機器人的行為。時間 $t = 34$ ，Robot 3 閃避過 Robot 2 後在地圖左下方發現目標物的存在並靠近它。時間 $t = 43$ ，Robot 3 靠近目標物後開始轉正自己之朝向角對向預定目的地，同時其狀態由一般模式變成 Master 模式，而 Robot 1、Robot 2 及 Robot 4 則由一般模式變成 Slave 模式(各機器人之模式變化如圖 5-26)。同一時間，由於 Robot 2 (Follower 3) 已在 Robot 3 (Leader Robot) 的附近，因此直接執行就隊形準備位置的行為，而 Robot 1 (Follower 2) 則切換至靠近隊長的行為。時間 $t = 45$ ，Robot 4 (Follower 1) 閃避完障礙物後開始執行靠近隊長的行為。時間 $t = 80$ ，三台 Follower Robot 皆進入其隊形位置(已轉正其朝向角與 Robot 3 一致)，此時 Robot 3 (Leader Robot) 開始起動，朝右上角之預定目的地移動。時間 $t = 88$ ，Robot 3 發現前方有障礙物，開始朝左執行閃避的行為，此時 Robot 2 和 Robot 4 隨著 Robot 3 朝障礙物左側進行閃避；而 Robot 1 則朝障礙物右側閃避。時間 $t = 105$ ，Robot 1 在閃避完障礙物後試圖回到原先之隊形位置，此時 Robot 2 及 Robot 4 正尾隨在 Robot 3 之後。時間 $t = 118$ ，Robot 1 回到隊形位置，此時四台機器人開始維持住三角形之隊形。時間 $t = 122$ ，Robot 3 帶隊抵達預定目的地，最終此四台機器人以三角形之隊形順利結束任務，而本模擬驗證了此一般性之多機器人協調控制架構可直接套用在數量龐大的多機器人團隊上。

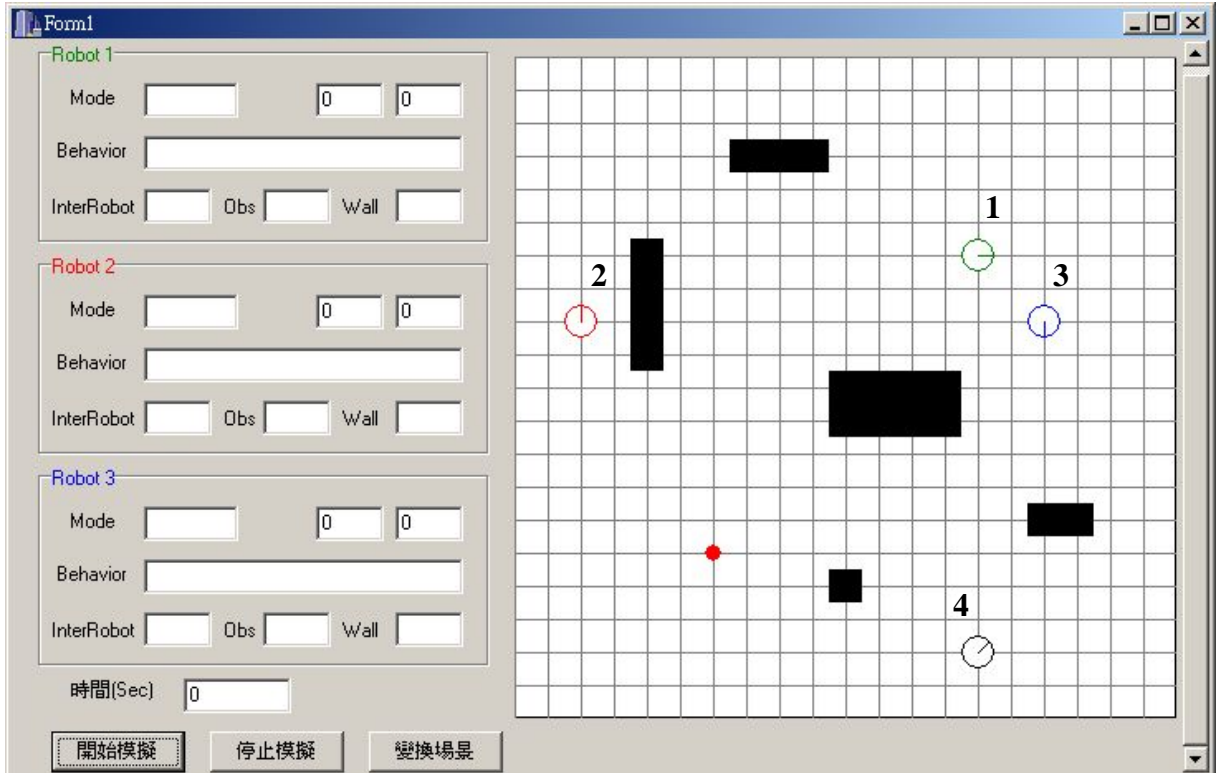


圖 5-23 模擬環境五之場景圖

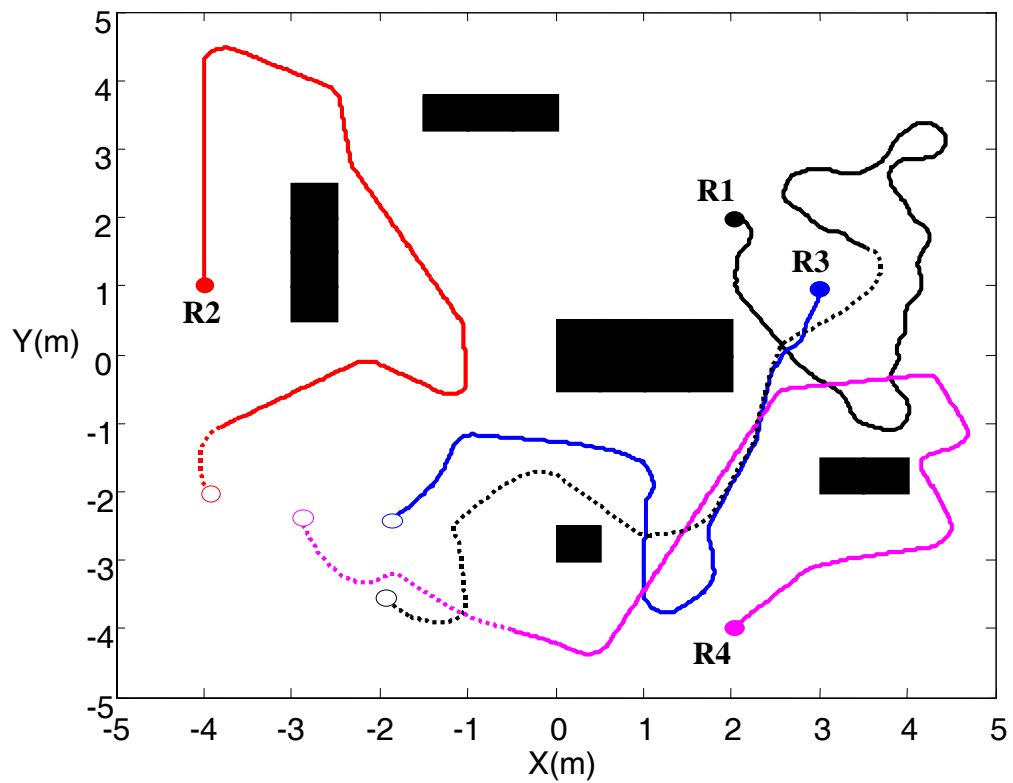


圖 5-24 模擬環境五之各機器人軌跡圖(由開始至隊形組成, 虛線部分為靠近隊長之軌跡)

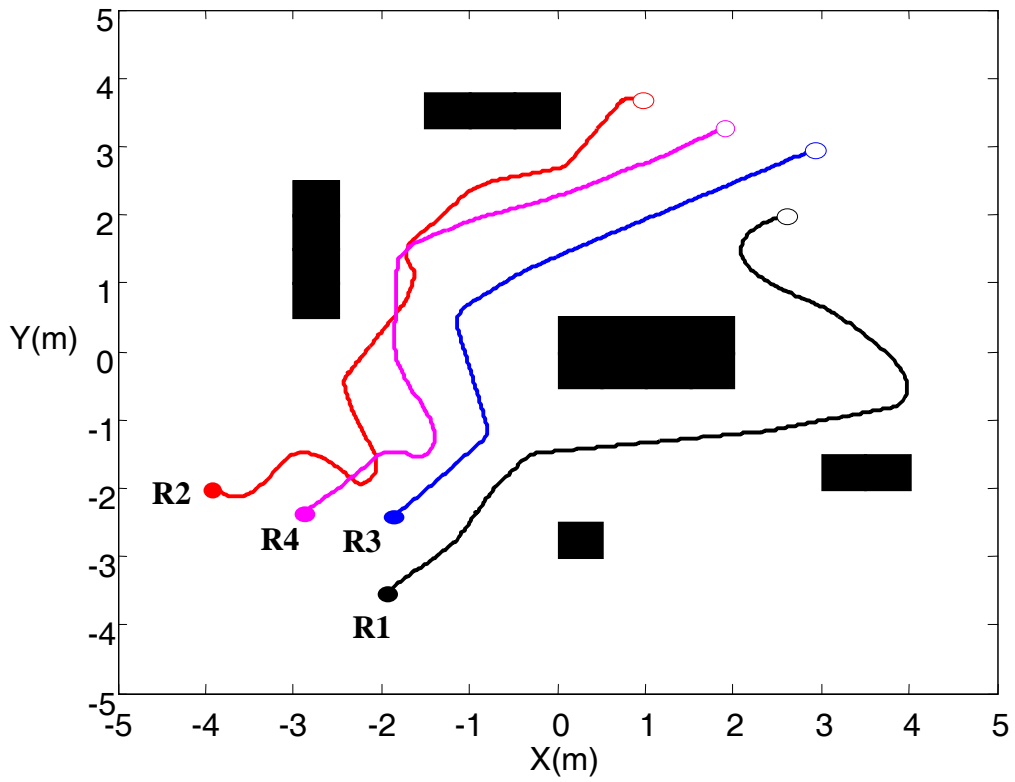


圖 5-25 模擬環境五之各機器人軌跡圖(隊形維持過程)

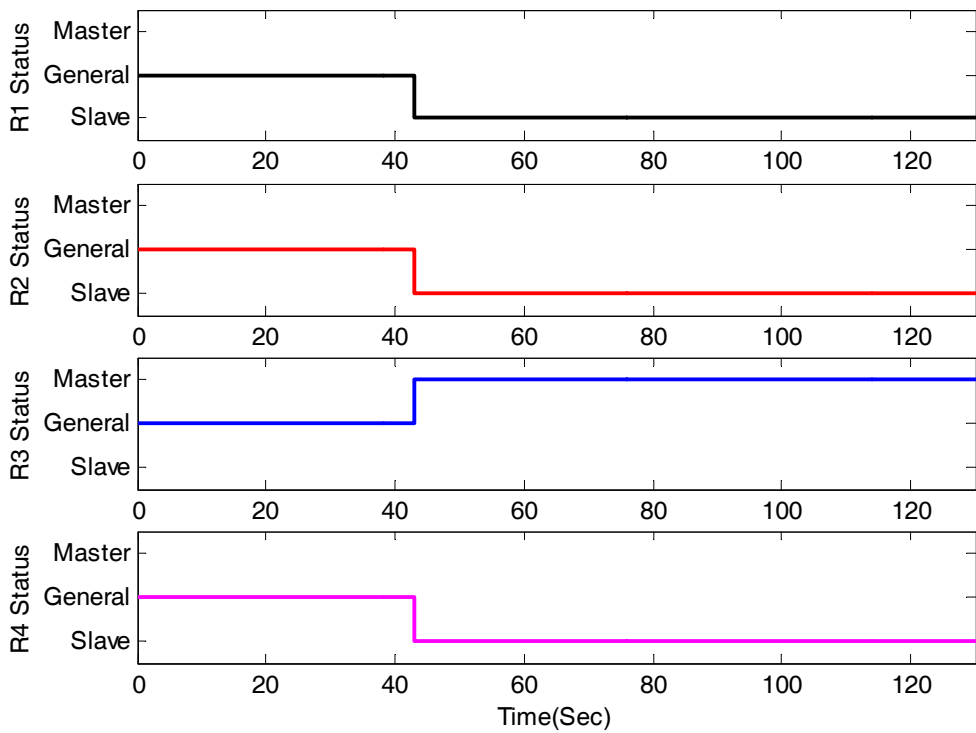


圖 5-26 模擬環境五之各機器人模式狀態變化

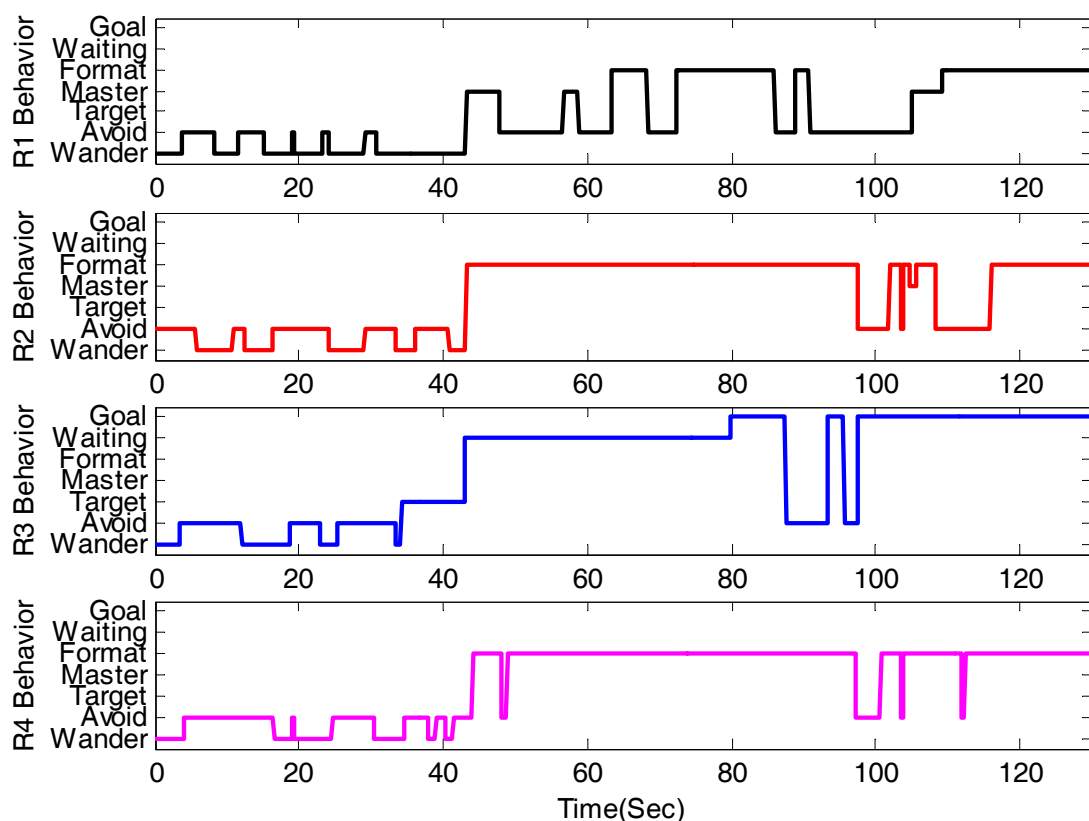


圖 5-27 模擬環境五之各機器人行為狀態變化



5.6 模擬結果與討論

在上述五個環境場景模擬中，最後機器人團隊皆完成其預定之任務內容(探索目標物→組成預定隊形→維持隊形至目的地)，即使因為各機器人之初始位置不同及環境地形(如障礙物之位置)的改變，三台(或四台)機器人仍然能夠藉由其自主式的行為選擇能力來稱職地扮演好自己在任務中的角色。針對本文提出之架構中的 Master/Slave 模式切換機制，在上述四個模擬結果中，我們由各機器人之模式狀態變化與行為狀態變化的時間對應關係中發現當機器人在執行靠近目標物的行為後確實產生一般模式→Master 模式的狀態變化，且其同時另外兩台機器人也產生了一般模式→Slave 模式的變化，且在模式改變後皆選擇了與其任務角色所相對應的行為輸出(如 Leader Robot 的前往目的地和原地等待、Follower Robot 的靠近隊長和隊形維持)，由這部分的結果可顯示出透過機器人間之無線溝通能力，可促使機器人個體執行其最適當之行為，以增加團體任務完成

的可能性。

透過對於全向式視覺回授的模擬，證實其功能的確能讓一個多機器人團隊具有隊形組成與維持的能力。此外在本章的五個模擬當中，我們一共設定了兩種不同的機器人隊形(含三角形隊形及線形隊形)，且在每種隊形任務執行當中，目標物及預定目的地的位置設定皆不相同，但最終機器人團隊仍藉由本論文所提出之多機器人協調控制架構，經由分散式之自主探索及合作協調之隊形組成與維持兩大過程，順利地執行完任務。上述的部分則驗證了本論文所提出的適應性基於行為模式之多機器人協調控制架構確實在任務執行上擁有相當大的彈性(Flexibility)，即便因為環境及任務內容(隊形種類)的不同也不會影響任務執行的困難。

表 5-1 為上述五個模擬之不同的任務內容及機器人團隊其相對反應出的情況比較：在五個模擬結果中，其中 Robot 1 及 Robot 3 各分別擔任過一次以上的 Leader Robot，表示在不同的環境下藉由此一架構之一般性(Generality)的設計優點能讓機器人團隊靈活地變換其適當的任務角色，以增加任務執行的效率。最後，在模擬中之機器人視覺回授及無線網路通訊的部分我們未來將以本文第三章與第四章中所描述之全向式影像系統及 RF 無線傳輸模組來實現之。

表 5-1 五個模擬之任務內容及結果比較

Simulation No.	Formation	Target	Destination	Leader Robot	Task Time
Simulation 1	Triangle	(3, 3)	(-3,-4)	Robot 3	111 Sec
Simulation 2	Triangle	(-3,-3)	(4, 3)	Robot 1	132 Sec
Simulation 3	Line	(0, -4)	(0, 4)	Robot 3	106 Sec
Simulation 4	Triangle	(-3,-3)	(4, 3)	Robot 3	129 Sec
	Triangle	(3,3)	(-4,-4)	Robot 3	228 Sec
Simulation 5	Triangle	(-2, -2.5)	(3, 3)	Robot 3	122 Sec

第六章 實驗結果

在本章裡我將以實驗室之三台實體機器人來進行協調隊形維持及合作搬運之實驗，來驗證全向式影像系統之視覺回授機制和 RF 無線傳輸模組實現之多機器人無線通訊功能是否發揮其預定之功效，同時也將驗證 2.2 節所介紹之多機器人協調控制架構在實際任務執行之可行性。6.1 節為此三台實體機器人之硬體描述，6.2 節為多機器人協調隊形維持實驗之結果與討論，6.3 節為多機器人合作搬運實驗之結果與討論。

6.1 機器人系統描述

在 6.2 節及 6.3 節的兩個實驗裡，我們一共用到了實驗室裡三台實體機器人平台 (Robot 1- Robot 3，實體如圖 6-1)，其系統核心為工業電腦主機板，負責接收各個感測器所傳來的資訊並將計算後之運動控制命令透過序列埠介面傳送至 DSP 馬達控制系統中，以完成機器人的行為控制。三台機器人的詳細規格如表 6-1，個別皆架設了上面章節所介紹的全向式影像系統及 RF 無線傳輸模組(全雙工功能)，兩者之資料皆透過序列埠介面傳送至 PC 端(硬體系統架構如圖 6-2)，圖 6-3 為機器人之硬體裝置架構圖。

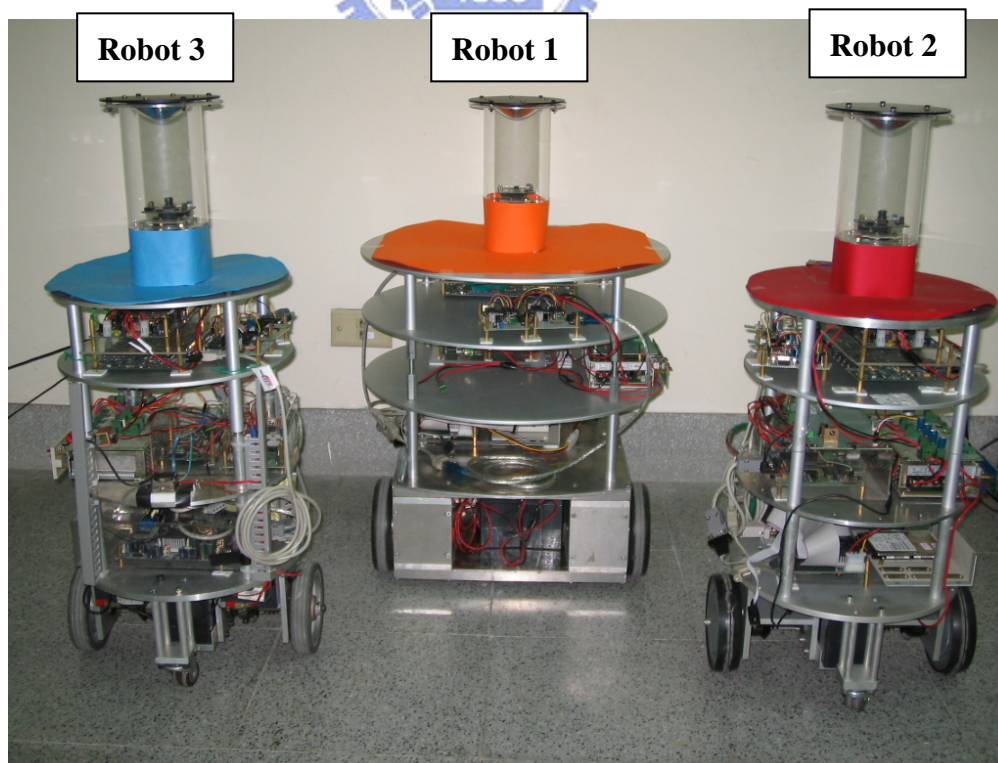


圖 6-1 三台機器人實體

表 6-1 三台機器人之硬體規格

	Robot 1	Robot 2	Robot 3
IPC Mainboard	Advantech PCM-9373	Advantech PCM-9373	VIA EPIA-TC10000
CPU	C3/Eden 1GHz	C3/Eden 1GHz	1GMhz
Wheel Diameter	15 cm	12 cm	12 cm
Gear Ratio	27.2 : 1	19.7 : 1	19.7 : 1
CMOS Sensor	IC-media EVT202		
Omnidirectional Mirror	MSI		
Wireless RF Module	Awin AM2400BS-RA		

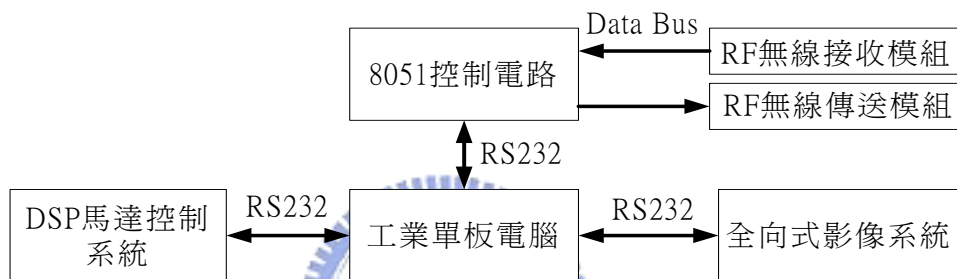


圖 6-2 機器人之硬體系統架構

方塊說明：

1. 全向式反射鏡
2. CMOS Sensor
3. DSP 影像平台
4. RF 無線傳輸模組
5. 機器人電力系統
6. DSP 馬達控制卡
7. 工業電腦主機板

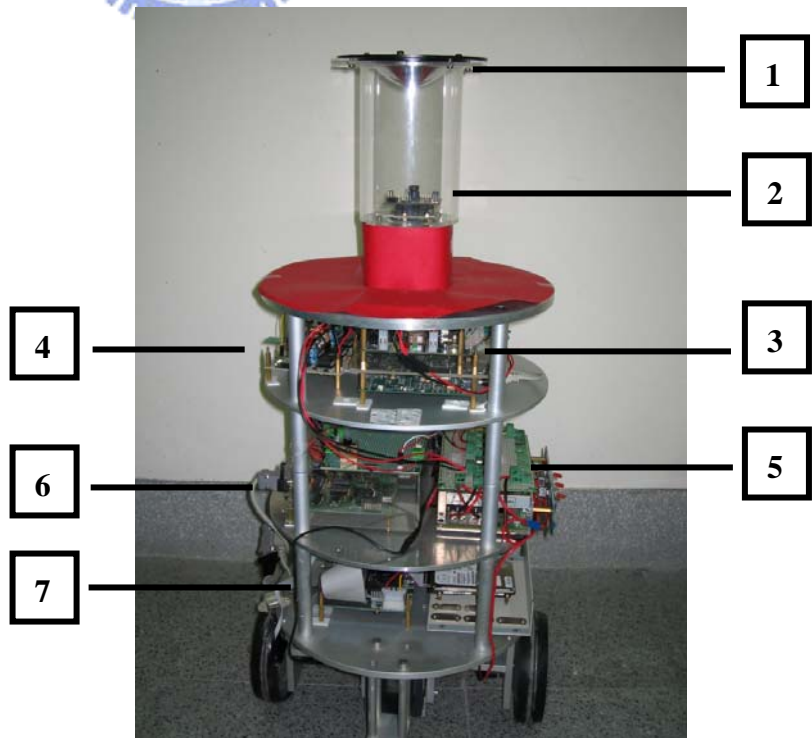


圖 6-3 機器人之硬體裝置架構圖

在機器人之馬達控制系統部分採用實驗室之簡辰竹學長所開發完成之 DSP 馬達控制卡[23]，再加上兩塊 L298 驅動電路，來做為機器人的兩輪移動控制(速度回授控制)，其與 PC 端之訊息傳遞透過 RS232 串列埠介面來實現(鮑率為 19200bps)，而整體控制方塊流程圖如下圖 6-4：

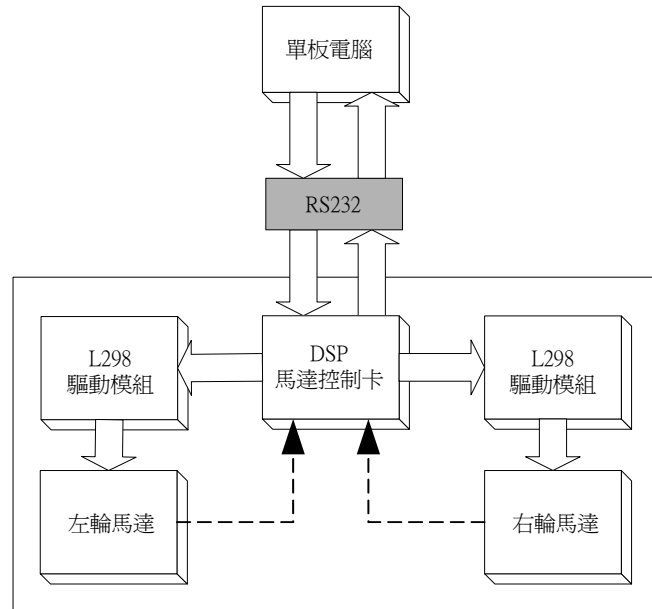


圖 6-4 DSP 馬達控制系統架構圖

6.2 三台機器人之協調隊形維持實驗

6.2.1 實驗說明與描述

此實驗之目的在於透過機器人之視覺回授功能及無線通訊機制來達成多機器人協調隊形維持的行為效能。實驗場景如圖 6-5，畫面中央之機器人為 Robot 1(橘色)，右方之機器人為 Robot 2(紅色)，左方之機器人為 Robot 3(藍色)；設定上 Robot 1 為機器人團隊之 Leader Robot (Master)，Robot 2 為 Follower Robot 1 (Slave 1)，而 Robot 3 為 Follower Robot 2 (Slave 2)，最終藉由兩個 Follower Robot 分別與 Leader Robot 維持一相對位置(距離與角度)來達成三台機器人隊形維持的目的。其方法為 Leader Robot 透過 RF 無線傳輸模組將自己所量測到其與 Follower Robot 間之相對角度 θ_{FL} (Leader-based，如圖 6-6)傳送到 Follower Robot 上，再讓 Follower Robot 再藉由自己所量測出之其與 Leader Robot 的



圖 6-5 三台機器人之協調隊形維持實驗場景圖

相對角度 θ_{LF} (Follower-based) 來計算出其與 Leader Robot 間的朝向角差 ψ_{LF} (兩台 Follower Robot 皆用相同的方式計算)，如 2-29 式所示：

$$\psi_{LF} = \theta_{LF} - \theta_{FL} - \pi \quad (\text{rad})$$

$$\text{where } \psi_{LF} \in (-\pi, \pi)$$

最後再由朝向角差 ψ_{LF} 來即時計算目前更新後之隊形位置 (控制公式在 2.3.8 節已有描述)，經由即時計算所得之 ψ_{LF} ，可由式(2-32)、(2-33)分別計算出在當時 Follower Local 座標系之預期 X 及 Y 座標 (X_{D-Slv} , Y_{D-Slv})，然後依據式(2-35)及(2-36)計算出 Follower 當時之運動距離與方向。本實驗實現三台機器人之協調隊形維持行為功能，而預定之機器人隊形為三角形 ($l_{Desired} = 0.6$, $\theta_{Desired} = \pm \frac{2\pi}{3}$)。在相對角度的資料傳輸部分，Robot 1 (Master) 偵測到其個別與 Robot 2 (Slave 1) 和 Robot 3 (Slave 2) 之相對角度 θ_{FL_Slave1} 與 θ_{FL_Slave2} 後，再透過 RF 無線傳輸模組將角度資訊(以 1 Byte 表示，其資料壓縮與還原方式在 3.5 節已有所描述)傳送給另外兩台機器人。由於 Robot 1 是一次同時將 θ_{FL_Slave1} 與

θ_{FL_Slave2} 送出，且三台機器人之無線通訊方式為廣播(Broadcast)，因此我們在每筆相對角度資訊的前面各加上了 1 Byte 之辨識位元(傳輸協定如圖 6-7)，使 Robot 2 及 Robot 3 能正確拿到自己所需要的角度資訊，而不至於錯拿到對方的。

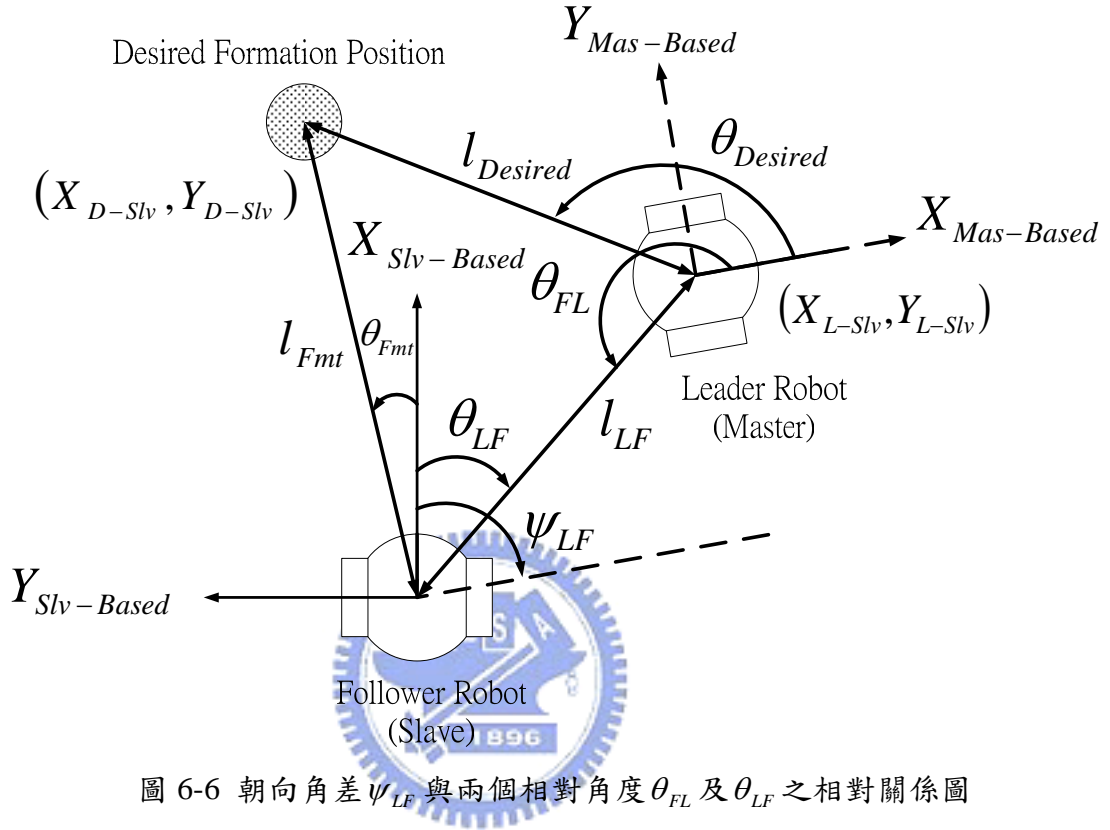


圖 6-6 朝向角差 ψ_{LF} 與兩個相對角度 θ_{FL} 及 θ_{LF} 之相對關係圖

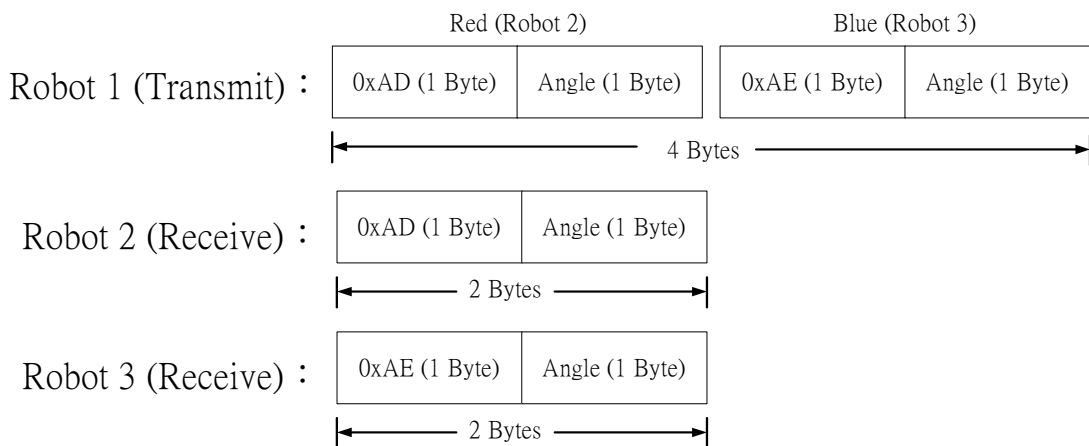


圖 6-7 相對角度之無線傳輸協定

6.2.2 實驗結果與討論

實驗過程照片如圖 6-8，三台機器人由起始隊形開始，Robot 1 (Master)朝預定目的地(黃色紙箱位置)移動，過程中 Robot 2 (Slave 1)及 Robot 3 (Slave 2) 不斷藉由由 Robot 1 透過無線通訊所傳來的相對角度資訊 (Master 為基準之座標系上的 Slave 相對角度) 與自身全向式感測器所估測到之 Robot 1 的相對角度 (Slave 為基準之座標系上的 Master 相對角度)來計算出自己與 Robot 1 的朝向角差，以便即時更新自己所應處於的隊形位置並趨近之，來達成三個機器人隊形維持的目的。圖 6-9 為三台機器人由其自身兩輪馬達軸編碼器所計算出來自己之行徑軌跡，Robot 1 (Master)之起始位置為(0, 0, 90°)，Robot 2 (Slave 1)之起始位置為(0.3, -0.6, 90°)，而 Robot 3 (Slave 2)之起始位置為(-0.3, -0.6, 90°)，預定隊形目的地為座標(0, 3)的位置(即黃色紙箱的所在處前方)。由軌跡結果可觀察出三台機器人由起始隊形位置開始，途中不斷透過視覺回授機制及 RF 無線傳輸協調機制來更新自己的隊形位置並趨進之；雖然由於全向式視覺感測系統在人為校正上之誤差(包含相對角度及相對距離誤差)及 RF 無線傳輸模組接收端所收到的少許雜訊的影響會造成 Slave Robot 在某些時候稍微偏離其隊形軌道，但由於我們在計算 Master Robot 及 Slave Robot 之朝向角差 ψ_{LF} 時，所用到的資訊(θ_{FL} 、 θ_{LF} 及 l_{LF})並不會有累積誤差的產生，因此透過不斷迴圈式的回授控制，等到正確的資料進來時能使 Slave Robot 修正回其所屬的隊形位置上，最終讓這三台機器人維持前一後二的三角形隊形，直到抵達目的地。由這個實驗結果驗證了透過我們在第三章所描述之全向式影像系統所實現的機器人視覺回授能力及第四章所描述之 RF 無線傳輸模組所實現的多機器人無線通訊機制能夠讓一個多機器人團隊達成協調式的隊形維持行為功能。未來若再搭配 6.3 節多機器人合作搬運實驗中之各機器人的行為模式切換機制即可實現在 2.2 節所提出之多機器人協調控制架構的完整功能。



(a) 三台機器人之起始隊形為前一後二之三角形，中央的為 Leader Robot (Robot 1)。



(b) Robot 1 開始朝預定目的地(黃色紙箱位置)前進，Robot 2 及 Robot 3 負責維持隊形。



(c) 透過視覺回授及 RF 無線通訊協調功能，兩個 Slave 分別保持在 Master 的左後及右後方。



(d) 由於感測器的自然誤差，Robot 3 稍微偏離其隊形軌道。



(e) 透過視覺回授功能，Robot 3 逐漸修正回其隊形位置。



(f) 透過視覺回授功能，兩台隨時修正回其所屬之隊形位置。



(g) 由於感測誤差, Robot 2 稍偏離其隊形軌道。(h) 透過視覺回授, Robot 2 修正回其隊形位置。

圖 6-8 三台機器人之協調隊形維持實驗

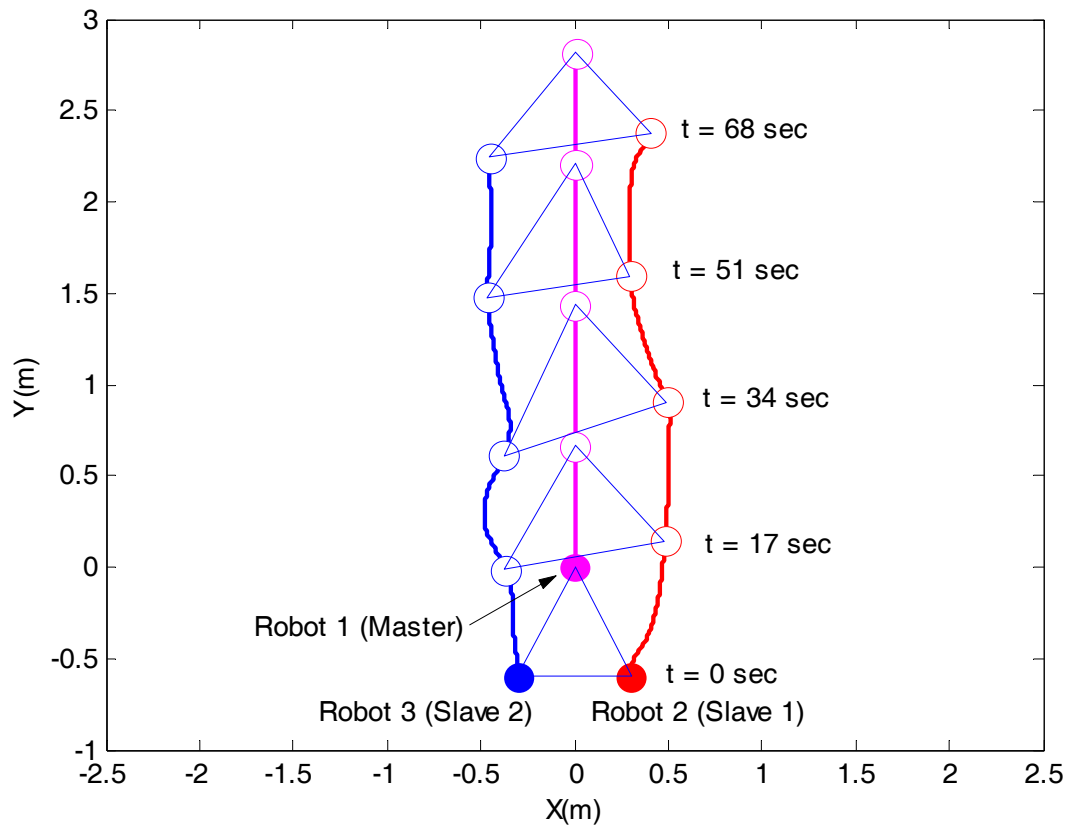


圖 6-9 協調隊形維持之三台機器人軌跡

6.3 三台機器人之合作搬運實驗

6.3.1 實驗說明與描述

此實驗之目的在於透過機器人之視覺回授機制及無線通訊功能來達到多機器人合作搬運的效果。實驗照片如圖 6-10，畫面一開始下方之機器人為 Robot 1(橘色)，左方之機器人為 Robot 2(紅色)，右方之機器人為 Robot 3(藍色)，而正中央的紙箱(黃色)即為此次欲由三台機器人合力搬運推動的目標物。在這次實驗中，我們將觀察當 Robot 1 發現目標物——黃色紙箱時，是否會正確地藉由全向式影像系統所得到之紙箱相對位置而靠近它，在順利靠近後是否又會透過 RF 無線傳輸模組發送正確的請求協助命令來請兩旁之 Robot 2 及 Robot 3 前來支援，最終合力推動紙箱。





(a) 三台機器人及目標物分別在其起始位置，
下方之機器人為 Robot 1。



(b) Robot 1 發現黃色紙箱，並開始靠近它



(c) Robot 1 持續靠近黃色紙箱。



(d) 當 Robot 1 到達剛好接觸到的距離後停止
前進，並開始發送請求協助命令。



(e) Robot 2 及 Robot 3 收到 Robot 1 發送之請求
協助命令後開始趨近黃色紙箱。



(f) 右方之 Robot 3 已就定位，左方的 Robot 2
正在調整自己的朝向角。



(g) 三台機器人皆已就定位，此時 Robot 1 已收到分別來自 Robot 2 及 Robot 3 的就位訊息。

(h) Robot 1 開始朝目的地前進，此時另外兩台機器人跟 Robot 1 保持預定之相對距離。



(i) 三台機器人合力搬運紙箱中。

(j) 三台機器人合力搬運紙箱中。



(k) 三台機器人合力搬運紙箱中。

(l) 抵達目的地後三台機器人停止動作。

圖 6-10 三台機器人之合作搬運實驗

6.3.2 實驗結果與討論

圖 6-11 為三台機器人由其自身兩輪馬達軸編碼器所計算出來自己之行徑軌跡，由軌跡可看出三台機器人由不同之起始座標開始，在紙箱之中心位置座標(0.2, 1.5)處集合後開始呈 V 字型隊形合力搬運之。最終紙箱被推動到座標系 $Y = 2.7$ 之所在，即成功被搬運了 1.2m 之距離，最終三台機器人仍維持在紙箱其左、右、後三方呈 V 字型之隊形。

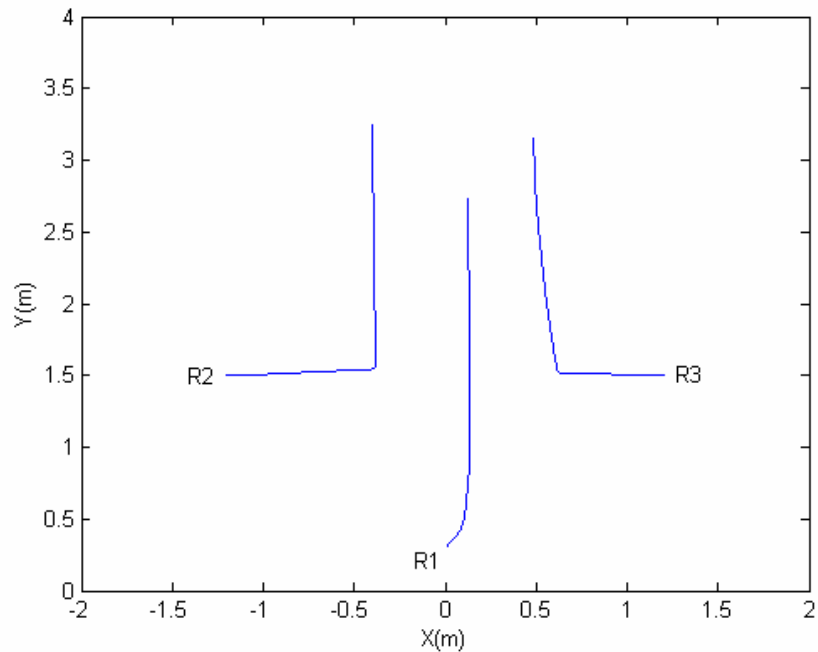


圖 6-11 合作搬運之三台機器人軌跡

下圖 6-12 為三台機器人在實驗過程中紀錄自己之狀態模式的變化，以數字代碼表示：

- 0：未看到目標物（一般模式）。
- 1：已看到目標物，正在接近中（一般模式）。
- 2：已接近目標物，成為隊長（Master 模式）。
- 1：收到請求協助命令，成為隊員（Slave 模式）。
- 2：已就準備位置，且收到來自隊長之隊形起步命令（Slave 模式）。

由圖中我們可以明顯觀察到這三台機器人有兩個明顯的狀態變化：

- (1) 在時間 13 秒附近，Robot 1 由於已經靠近目標物因此其狀態由一般模式提升為 Master 模式(表示它的角色成為 Leader Robot)。此時 Robot 2 與 Robot 3 由於收到它所傳送出來的請求協助命令而切換為 Slave 模式(表示它們的角色成為 Follower Robot)。
- (2) 在時間 28 秒附近，Robot 1 已經分別收到來自 Robot 2 及 Robot 3 的已就定位訊息，因此再次發送隊形起步命令給 Robot 2 和 Robot 3，因此這兩台機器人在收到命令後再次產生狀態變化(表示此時 Leader Robot 開始動作後，這兩個 Follower Robot 會開始與它維持隊形)。

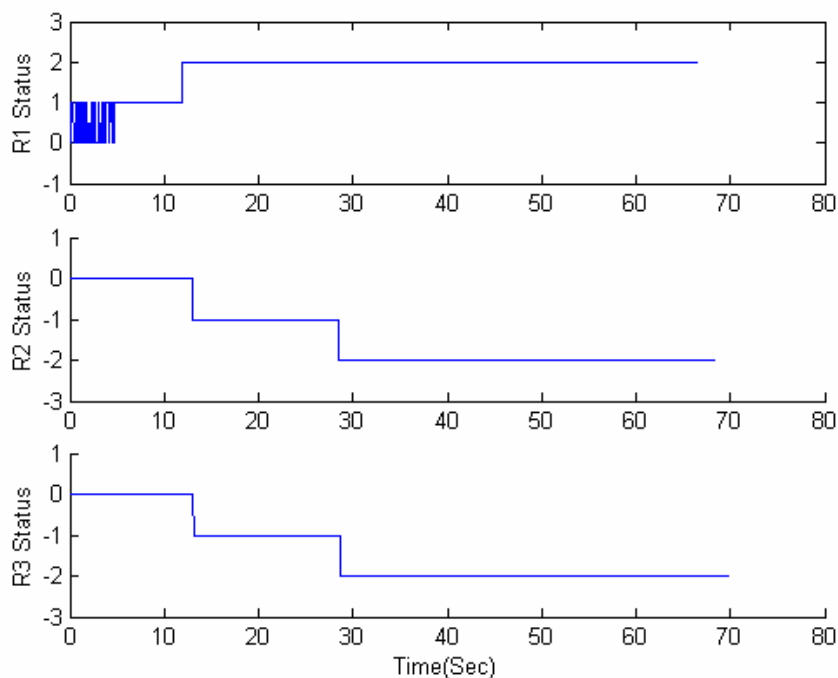


圖 6-12 合作搬運之三台機器人狀態模式

藉由上述之實驗結果與紀錄數據我們可以證實此三台機器人之視覺回授與無線通訊功能的確發揮了功效，讓這三台機器人間在協調的過程藉由行為模式的適當切換順利地完成了合作搬運的任務。此實驗裡之合力搬運的行為設計上，是採用讓兩台 Follower Robot 各自在紙箱的左右兩側原地轉正與紙箱呈 90 度之方位角(靠視覺回授來修正，轉正後三台機器人皆朝向同一方向)後再與 Leader Robot 保持一預設之距離的方式來維持彼此間的距離關係而達成合作搬運之目的。

第七章 結論與未來展望

7.1 結論

本論文一共完成了：

- (1) 以全向式影像系統平台實現了機器人之視覺回授功能，其中包含了色彩辨識之影像處理動作，提供機器人與機器人或與目標物之間的辨識能力；此外還有相對距離及角度的估測及校正，以做為多機器人隊形之組成與維持的依據。另外由於環境影像是先經由影像平台之處理動作後才將必要資訊傳送至 PC 端，因此大幅降低了 PC 端的計算負擔(computing load)，所以更加達成了機器人視覺回授運動控制之即時性。
- (2) 以 RF 無線傳輸模組實現了多機器人間的無線通訊能力，再透過預先制定的無線通訊協定，各機器人間能在適當的時刻了解彼此之狀態並適當地改變自己的行為模式以達成最佳之團隊任務執行效果。
- (3) 以三台機器人之隊形維持模擬結果與協調隊形維持及合作搬運之實驗成果來驗證我在本篇論文所提出的基於行為模式之多機器人協調控制架構其可行性與整體系統在執行任務時的彈性(Flexibility)與機器人之自主能力(Autonomy)。

相信透過我所提出多機器人合作式協調控制架構，能讓一個多機器人系統藉由其團體合作及協調的特性，在任務的執行上擁有更多的彈性、更好的效率及更簡化之機器人設計，以上對於巡邏保全及合作搬運等應用將具有相當大的經濟及實用價值。

7.2 未來展望

雖然透過軟體的模擬結果可以充分展現我所提出之多機器人協調控制架構的功能及其特色，但整體架構中之某些部分在合作搬運的實驗中我尚未加以實現，其中包含機器人對於障礙物的偵測部分。關於此點我們未來可以考慮同樣透過全向式影像平台再加

上邊緣偵測(Edge Detection)的法則來偵測機器人之附近是否有障礙物的存在，亦或者可透過其它種類之感測器，如超音波等感測器等來取得其各方位之與障礙物的距離資訊。另外在個別行為的設計方面我認為仍有可以加強的空間，我相信利用這樣的合作式基於行為模式之協調架構再搭配考慮更周詳的行為設計將大大有利於提升機器人團隊在執行任務時的穩健性(Robustness)。



參考文獻

- [1] Cheng-Hsien Chiu Huang, "Localization and Map-Building Using Multi-Robot Cooperative Sensing," M.S. thesis, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 2003.
- [2] Chomchana Trevai, Tusuke Fukazawa, Jun Ota, Hideo Yuasa, Tamio Arai, and Hajime Asama, "Cooperative Exploration of Mobile Robots Using Reaction-Diffusion Equation on a Graph," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2269-2274, September 2003.
- [3] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E. Schneider, "Coordinated Multi-Robot Exploration," *IEEE Transactions on Robotics and Automation*, vol. 21, no. 3, pp. 376-386, June 2005.
- [4] H. Subramanian, and C. H. Dagli, "Cooperative Cleaning for Distributed Autonomous Robot systems using Fuzzy Cognitive Maps," *Proceedings of the 22nd International Conference on North American Fuzzy Information Processing Society*, pp. 287-290, July 2003.
- [5] Chih-Ching Tang, "Design and Implementation of a Multirobot System for Team Competition," M.S. thesis, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 1999.
- [6] J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar, and J. P. Ostrowski, "Cooperative Localization and Control for Multi-Robot Manipulation," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 631-636, November 2001.
- [7] Hiroaki Yamaguchi, "A Distributed Motion Coordination Strategy for Multiple Nonholonomic Mobile Robots in Cooperative Hunting Operations," *Proceedings of IEEE International Conference on Decision and Control*, vol. 3, pp. 2984-2991, December 2002.
- [8] Hiroaki Yamaguchi, "Adaptive Formation Control for Distributed Autonomous Mobile Robot Groups," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2300-2305, April 1997.
- [9] "Rescue Robotics in Japan," Asian Technology Information Program (ATIP), January 2003.
- [10] Jaydev P. Desai, Jim Ostrowski, and Vijay Kumar, "Controlling formations of multiple mobile robots," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 2864-2869, May 1998.

- [11] Jaydev P. Desai, Vijay Kumar, and James P. Ostrowski, "Control of changes in formation for a team of mobile robots," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1556-1561, May 1999.
- [12] Guilherme A. S. Pereira, Aveek K. Das, Vijay Kumar, and Mario F. M. Campos, "Formation Control with Configuration Space Constraints," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2755-2760, October 2003.
- [13] Tucker Balch, and Ronald C. Arkin, "Behavior-Based Formation Control for Multirobot Teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926-939, December 1998.
- [14] Lounis ADOUANE, and Nadine LE FORT-PIAT, "Hybrid Behavioral Control Architecture for the Cooperation of Minimalist Mobile Robots," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3735-3740, May 2004.
- [15] Aveek K. Das, Rafael Fierro, Vijay Kumar, James P. Ostrowski, John Spletzer, and Camillo J. Taylor, "A Vision-Based Formation Control Framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, October 2002.
- [16] Rene Vidal, Omid Shakernia, and Shankar Sastry, "Formation Control of Nonholonomic Mobile Robots with Omnidirectional Visual Servoing and Motion Segmentation," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 584-589, September 2003.
- [17] Noah Cowan, Omid Shakernia, Rene Vidal, and Shankar Sastry, "Vision-based Follow-the-Leader," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1796-1801, October 2003.
- [18] John R. Spletzer, and Camillo J. Taylor, "A Framework for Sensor Planning and Control with Applications to Vision Guided Multi-robot Systems," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I-378 - I-383, 2001.
- [19] Ronald C. Arkin, *Behavior-Based Robotics*, The MIT Press, 1998.
- [20] Kostas Daniilidis, and Christopher Geyer, "Omnidirectional Vision: Theory and Algorithms," *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 1, pp. 89-96, September 2000.
- [21] Andrea Bonarini, Paolo Aliverti, and Michele Lucioni, "An Omnidirectional Vision Sensor for Fast Tracking for Mobile Robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 3, June 2000.
- [22] Andrian Henry, "An Embedded Image Processing Platform for a Mobile Manipulator," M.S. thesis, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, July 2004.

- [23] Chen-Chu Chien, “Design and Implementation of a Visual Tracking System for a Home Robot,” M.S. thesis, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, July 2002.

