

國立交通大學

電機與控制工程學系

碩士論文

全向式移動機器人之
同步定位與環境地圖建立



Simultaneous Localization and Mapping of
an Omni-Directional Robotic Platform

研究生：王兆戊

指導教授：宋開泰 博士

中華民國九十七年七月

全向式移動機器人之
同步定位與環境地圖建立

Simultaneous Localization and Mapping of
an Omni-Directional Robotic Platform

研究生：王兆戊

Student: Chao-Wu Wang

指導教授：宋開泰 博士

Advisor: Dr. Kai-Tai Song

國立交通大學

電機與控制工程學系



A Thesis

Submitted to Department of Electrical and Control Engineering
College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

July 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

全向式移動機器人之 同步定位與環境地圖建立

學生:王兆戊

指導教授:宋開泰 博士

國立交通大學電機與控制工程學系

摘要

本論文提出一使用全向式攝影機(Omni-directional camera)之全向移動式機器人(Omni-directional Mobile Robot)之同步定位與環境地圖建立(Simultaneous Localization and Mapping, SLAM)。藉由全向式攝影機 360°的視野，取得更豐富的影像資訊和更多的環境地標(Landmark)，並結合擴展式卡門濾波器(Extended Kalman Filter, EKF)達成全向式移動平台的 vSLAM。本論文採用參考點轉換與建立的策略，讓機器人能自主建立個別參考點，當走回已到過區域時仍能從資料庫中取回儲存之參考點資訊使用，減少 EKF 的計算負擔。實驗結果顯示特徵點辨識之正確率為 93%，以實驗室之機器人於室內繞行 8 字形軌跡移動行經約 16 公尺後之起點與終點定位誤差均在 0.1 公尺以內。實驗結果證實機器人能藉定位系統達成走廊上的長距離全向式移動及轉動，且建立走廊環境的特徵點地圖，達成全向移動式機器人的室內導航功能。

Simultaneous Localization and Mapping of an Omni-Directional Robotic Platform

Student: Chao-Wu Wang

Advisor: Dr. Kai-Tai Song

Department of Electrical and Control Engineering
National Chiao Tung University

Abstract

This thesis investigates simultaneous localization and mapping (SLAM) of an omni-directional mobile robot. A method is proposed to use an omni-directional camera to realize SLAM algorithm based on extended Kalman filter (EKF). This study focus on the use of 360° of view of the omni-directional camera to reduce the accumulative error from odometer and to achieve simultaneous localization and mapping of the omni-directional platform. The method of visual reference scans is adopted in this design. Features of previously visited places can be used repeatedly to reduce the complexity of extended Kalman filter. Experimental results show that the matching rate of landmark features is 93%. The localization error is less than 0.10m for traveling 16 meters of “8” shaped route. Indoor navigation experiments revealed that the proposed localization system can navigate the omni-directional robot in an indoor environment and build the feature map simultaneously.

誌謝

謹向我的指導教授宋開泰博士致上感謝之意，感謝他兩年來在專業上和生活上待人處事的指導，以他豐富的學識與經驗，配合理論的應用，使得本論文得以順利完成。

也要感謝學長孟儒、嘉豪、格豪、振暘、志昇、裕宏在理論與實驗過程中的指導，以及感謝同學弘哲、仕傑、煥坤在研究過程互相勉勵與打氣，同時要感謝學弟科棟、學妹舒涵、巧敏在生活上與課業中帶來的樂趣。

最後更要感謝我的父母，對我從小的辛苦栽培，他們在研究過程中給了我許多鼓勵與支持，生活上給予溫暖與關愛，是我促使向前的動力，在此我願以此論文獻給我最感激的父母親。



目錄

摘要.....	i
Abstract.....	ii
誌謝.....	iii
目錄.....	iv
圖例.....	vi
表例.....	viii
第一章 緒論.....	1
1.1 研究動機與目的.....	1
1.2 相關研究回顧.....	2
1.3 問題描述.....	4
1.4 系統架構.....	5
1.5 章節說明.....	6
第二章 全向式移動平台之運動控制.....	7
2.1 硬體構造.....	7
2.2 平台運動模型推導.....	10
2.2.1 姿態估測 (Odometry).....	13
2.3 平台之運動控制.....	14
2.4 方形軌跡實驗.....	16
第三章 全向式移動平台之定位設計.....	19
3.1 全向式攝影機.....	19
3.2 vSLAM 系統架構.....	21
3.2.1 基於全向式攝影機之特徵點比對演算法.....	22
3.2.2 水平及垂直位置估測.....	23
3.2.3 基於 Extended Kalman filter 之 SLAM 演算法.....	26

3.3 平台之運動模型.....	27
3.3.1 Prediction model.....	29
3.3.2 Update model.....	31
3.4 參考點轉換.....	34
第四章 實驗結果.....	37
4.1 環境特徵點辨識實驗.....	37
4.2 EKF-SLAM 定位實驗.....	39
4.2.1 Ground truth 實驗(直行).....	39
4.2.2 Ground truth 實驗(直行加旋轉).....	45
4.2.3 參考點轉換實驗.....	46
4.3 沿方形移動實驗.....	47
4.4 8 字形移動實驗.....	49
4.5 室內導航之情境展示實驗.....	50
4.5.1 室內長距離移動實驗.....	50
4.5.2 室內長距離轉向移動實驗.....	53
4.6 討論.....	54
第五章 結論與未來工作.....	56
5.1 結論.....	56
5.2 未來工作.....	56
參考文獻.....	58

圖例

圖 1-1	實驗室之全向式移動平台結構實體圖	5
圖 2-1	全向輪實體圖	8
圖 2-2	全向式機器人實體圖	9
圖 2-3	機器人硬體架構	9
圖 2-4	平台四輪配置圖	10
圖 2-5	全向輪轉動示意圖	11
圖 2-6	全向輪之幾何關係	12
圖 2-7	平台座標旋轉示意圖	12
圖 2-8	平台運動模型圖	14
圖 2-9	平台控制方塊圖	15
圖 2-10	走廊方形軌跡實驗場景	17
圖 3-1	相鄰影像中之物體大小變化比較	20
圖 3-2	全向式攝影機示意圖	20
圖 3-3	Vstone VS-C450N-TR 全向式攝影機	21
圖 3-4	全向式攝影機所擷取之影像	21
圖 3-5	vSLAM 系統架構圖	22
圖 3-6	基於全向式攝影機之特徵點擷取與比對演算法流程圖	23
圖 3-7	Landmark 水平位置估測示意圖	24
圖 3-8	Landmark 垂直估測示意圖	25
圖 3-9	視角校正及其關係式	26
圖 3-10	EKF 演算法概念圖	27
圖 3-11	參考點轉換流程圖	35
圖 3-12	參考點影像轉換	36
圖 4-1	移動 1m 的特徵點比對結果	38

圖 4-2	旋轉影像 45°後的特徵點比對結果.....	38
圖 4-3	Odometer 與 EKF 紀錄軌跡比較(2m).....	40
圖 4-4	Odometer 與 EKF 紀錄軌跡比較(4m).....	41
圖 4-5	Odometer 與 EKF 紀錄軌跡比較(6m).....	42
圖 4-6	Odometer 與 EKF 紀錄軌跡比較(8m).....	43
圖 4-7	Odometer 與 EKF 紀錄軌跡比較(10m).....	44
圖 4-8	6 公尺來回 EKF 及 Odometer 比較.....	45
圖 4-9	室內沿方形移動實驗場景	47
圖 4-10	室內方形移動實驗結果	48
圖 4-11	8 字形移動軌跡結果	49
圖 4-12	機器人導航實驗結果	51
圖 4-13	機器人導航實驗過程	52
圖 4-14	機器人長距離轉向移動實驗結果	53
圖 4-15	機器人轉向移動實驗過程	54



表例

表 2-1	繞行方形 2 圈實驗結果	17
表 2-2	繞行方形 3 圈實驗結果	18
表 3-1	不同誤差 ΔS 與 dz 及定位之相對關係	25
表 4-1	特徵點比對實驗結果(平移)	39
表 4-2	特徵點比對實驗結果(旋轉)	39
表 4-3	Odometer 與 EKF 量測結果比較(2m)	40
表 4-4	Odometer 與 EKF 量測結果比較(4m)	41
表 4-5	Odometer 與 EKF 量測結果比較(6m)	42
表 4-6	Odometer 與 EKF 量測結果比較(8m)	43
表 4-7	Odometer 與 EKF 量測結果比較(10m)	44
表 4-8	來回行走實驗誤差	46
表 4-9	Reference 影像取 1 公尺實驗結果	46
表 4-10	Reference 影像取 1.5 公尺實驗結果	47
表 4-11	室內方形軌跡實驗結果	48
表 4-12	機器人 8 字形移動誤差	50
表 4-13	機器人導航實驗誤差	52
表 4-14	機器人長距離轉向移動實驗誤差	55

第一章 緒論

1.1 研究動機與目的

輪式機器人移動平台由於在平面移動速度快且平穩，因此已成為各種服務型機器人廣泛採用：如雙獨立驅動輪機構，於平台前方或後方置一被動自由輪的三輪式之設計。然而此種結構為 Non-Holonomic，限制了機器人全向性的移動性，其轉向利用兩驅動輪的速度差來達成，在靈活度上便受到限制。一般家用或服務型機器人往往需要較高的機動性以完成一些複雜的路徑；能夠 S 形曲走或是突然的改變行進方向且正確的到達工作站的位置。為因應各種應用的需求，近年來有許多 Holonomic 或具有全向式移動(Omni-directional)之機器人被發展出來，如雙足機械人(Biped Robots)、球型輪移動平台(Ball Wheel Robots)[1]、全向輪式移動平台.....等類型。其中雙足機械人其機械結構及控制方法較為複雜，且移動速度低於輪型移動平台；球型輪移動平台受限於輪子與地板摩擦力過小，使得平台的定位困難，進一步影響控制結果；而以全向輪構成之全向式移動平台是較佳的選擇[2]-[8]。全向式移動平台之輪子經過特殊設計，除了能夠在行進中同時旋轉機器人朝向角，也能夠即時的朝任一方向前進而不需轉向，使平台達成全方位移動。為了使全向性移動的結構正確且穩定的達到目標點，必須有高性能之運動控制系統。

全向式移動平台，其移動具有全向性的優勢，擁有 Holonomic 的特性，雖然機動性更高，但相較於傳統二輪式平台它的輪子與地面的打滑問題和累積誤差也更嚴重，因此關於平台的定位控制也是很受重視的議題。本論文的研究目的即是希望藉由全向式攝影機 360°的影像資訊搭配全向式移動平台的高機動性，並修正平台的累積誤差，達成機器人能夠在室內完成自主導航的功能，且對於周遭的環境能探索了解並能快速移動至設定的地點，完成全向式移動機器人在室內環境下的巡航及建立環境地圖。

1.2 相關研究回顧

近年來對於全向式移動平台的研究朝向以影像定位[2]-[5]及平台的控制[6]-[9]為多。其中[2][3]是以全向式攝影機為主要感測器搭配里程計(Odometer)為三輪全向式平台的定位。他們主要的用途為足球機器人，活動範圍為固定場地。利用場地中的球門、邊緣以及角落旗幟的顏色作為影像定位的依據。其三輪主軸外另安插了三個小輪，增加與地面的接觸性，減少打滑造成的誤差。但此種設計當機器人到了新地點，影像便無法定位。[4]是以基於軌跡線性化(Trajectory Linearization)的理論實現全向平台的導航功能，其控制器也是以此理論為基礎設計；使用車上感測器(Onboard sensor)和視覺系統作量測，以擴展式卡門濾波器(Extended Kalman filter)融合，提供可靠且正確的機器人位置資訊。[8]則是探討使用支撐滑輪(Caster Wheels)的全向式移動平台之導航，透過里程計回傳的資訊得到機器人本身位置資訊後，再由雷射掃描儀作路徑導航規劃。[5]主要是探討三輪全向式移動平台的控制方法，如 Resolved acceleration control、PID control、Stochastic Fuzzy Servo control[6]等，[7]也是以三輪全向式平台作控制，利用點對點與軌跡追蹤，使用 PI Kinematics control，以機器人目前的位置和目標位置差作速度控制命令，機器人再將目前位置以里程計回授。模擬的結果雖然可行，但並無考慮實際全向輪打滑造成的誤差問題。[9]是使用光流感測器對全向平台作控制，由於打滑問題，使得平台的動態模型失去準確性；考慮到輪子與地面的摩擦力，藉由一 Spatial tracking 演算法以修改機器人的速度命令，以期能平台對軌跡追蹤達成較好的效能。最後並以基於光流感測器達成閃避障礙物之演算法。

當前對於使用影像作全向式移動平台的同時定位與地圖建立(Simultaneous Localization and Mapping, SLAM)的研究越來越受重視。傳統作法是先讓機器人自我定位，再藉由機器人在自我定位時獲得的資訊來建立環境地圖。SLAM 作法與其他定位方法不同的地方，在於當機器人依所觀測的地圖環境資訊而定位出本身相對位置的同時，也能夠逐漸建立起環境的週遭地圖。解決了 SLAM 的問題

[10][11]，實際上也解決了機器人導航的問題，因此受到了許多學者的重視。本論文的重點便著重在使用全向鏡影像作全向式移動平台的同步定位與環境地圖建立，因藉由影像可得到豐富的环境資訊，擷取到更多環境特徵點(Feature)，應用更為廣泛。也可稱為基於全向式移動平台的 vSLAM。

vSLAM 的主要感測器為攝影機，在攝影機的選取上，考慮到全向式移動平台的全向性，適當的視覺感測器便相當重要。[12]使用單眼攝影機，導入了 Inverse depth 的概念。能在一個影像 Frame 中定義出特徵點的相對距離和角度，並對環境特徵作 State 以及 Covariance 的初始化。即使相對距離的不確定性高，相對角度卻有一定可靠度，能對 SLAM 中的機器人朝向角修正有幫助。[13]是兩台攝影機的立體視覺系統，利用了立體視覺能估測畫面景深的特性，對影像中的特徵距離估測。然而在攝影機視角有限的情形下，使用視覺系統來建立環境資訊時，如何整合並比對各畫面中的特徵點是個重要的問題。無論單眼或雙眼攝影機，同一場景在旋轉超過一定角度後，便要重新建立視覺參考點，使得全向式移動平台由於影像的侷限性導致移動性受限。而全向式攝影機由於具有 360°視角，能同時取得更多環境特徵，且特徵點的描述向量有抗旋轉的功能，並增加追蹤的時間，使得視覺參考點即使在旋轉的情形下也不會改變，用於全向式移動機器人的定位是個很好的感測器。

使用電腦視覺最大的問題，在於無法即時的偵測環境距離。物體的距離決定其位於影像上的位置，但若是物體的高度未知，要精準的根據影像量測其相對位置是很困難的。[14]是將三個全向式攝影機架設在等邊三角形的頂點，作為足球機器人的定位系統，以顏色為標籤，利用三個攝影機觀測到物體的角度差，統計並整合機器人與物體的距離，與單一或兩個全向式攝影機的距離估測相較，不但穩定且誤差也減少許多。[15]為利用兩個全向式攝影機，使用 Structure from motion (SFM) algorithm 來對環境作距離估測。其主要是結合相鄰的數張影像，重建出 3D 景物。[16]同樣是使用全向式攝影機，以小波轉換的方法擷取出特徵點後，建立數個 Reference image 供機器人比對；並利用 Particle filter 作機器人定

位，接著再延伸至建立環境地圖，加上 Loop closing 修正多次行走後的累積誤差，達成機器人之 Appearance-based 的 vSLAM。

以上數篇文獻可知，以基於全向式攝影機對全向式移動平台作定位控制是近年來受許多學者重視的議題。在影像平面中，如何對特徵物體進行距離的估測是首要解決的問題，並利用影像資訊搭配合適的濾波器減少里程計的累積誤差。本論文使用基於 EKF-SLAM 全向式移動平台搭配全向式攝影機，EKF 具有收斂速度快、運算精度佳、穩定性高等優點，其演算法適用於非線性系統中[17]，且較為直接，容易實現，並有效克服數據的不確定性，得到系統的最佳估測值，且利用全向式攝影機全方位視角的特性，觀測周遭環境，使全向式移動平台能展現其優勢，發揮全向移動的特性，達成自我定位同時建立環境特徵點地圖。

1.3 問題描述

由文獻中可知，相對於二差動輪式機器人，全向式移動平台要作精準的控制相對挑戰更大；雖然具高機動性，但它在移動時與地面打滑所造成的累積誤差 (Accumulated Error) 也比傳統的差動式輪子更為顯著。全向平台藉由四軸相互作用產生的合力分力來形成運動路徑，但各軸為獨立驅動，特性上有所差異，容易造成運動時的誤差，進而影響位置準確性，因此無法長時間使用 Odometer 做定位，勢必要藉由其他 Sensor 改善。以全向式攝影機搭配全向式移動平台是近年來相當受重視的研究議題。但使用單一全向式攝影機，其最重要的問題就是如何在影像平面中偵測出環境特徵點並對其作位置的估測，以減少全向移動式機器人的累積誤差。再者，如何修改固有的演算法以展示出全向式攝影機的優勢所在，對其作有效的利用，辨識出環境特徵點及估測出其空間位置，使其能有效幫助全向式機器人定位，即是在基於全向式影像的機器人定位議題中必須解決的問題。

在許多關於 vSLAM 的研究中，如[12]使用 EKF 演算法，會因所觀測到的特徵點持續增多，造成 Covariance matrix 持續增大，不確定性增加，提昇運算的複雜度。[13]中因為攝影機的視角有限，故旋轉超過一限定值便需要再建立一新的

參考點，當機器人走回之前走過的地點時，仍需再次建立參考點，隨著參考點的增加，系統的運算複雜度也跟著提升。故本論文希望藉由全向式攝影機的 360° 影像資訊，減少參考點建立的數量，降低位置估測的不確定性；使用 EKF-SLAM 的機器人定位方法，減少全向式平台因打滑偏移帶來的誤差問題影響。

1.4 系統架構

本論文使用實驗室研製的全向式移動平台如圖 1-1 所示，此平台採用四個獨立驅動輪式之運動機構，利用四個馬達控制卡實現機器人在平面上的移動。所使用主要的感測器為全向式攝影機，利用影像辨識演算法擷取影像中的特徵點作為機器人位置估測用的 Landmark 以及建立環境地圖，並以 EKF 融合機器人 Odometer 的資訊，同時更新機器人本體座標及環境特徵點的位置；第一次見到的 Landmark 加入地圖中，若是已見過的 Landmark，則代表此區域來過，從資料庫中取出先前的資訊使用。同時為了延長機器人定位的距離範圍，本論文採用一參考點建立及轉換的流程，當機器人移動到新地點時則建立此處參考點，回程時則取之前建立的參考點資訊使用，讓機器人能在大範圍環境中持續重覆行走，達成長距離導航及建立環境地圖的功能。

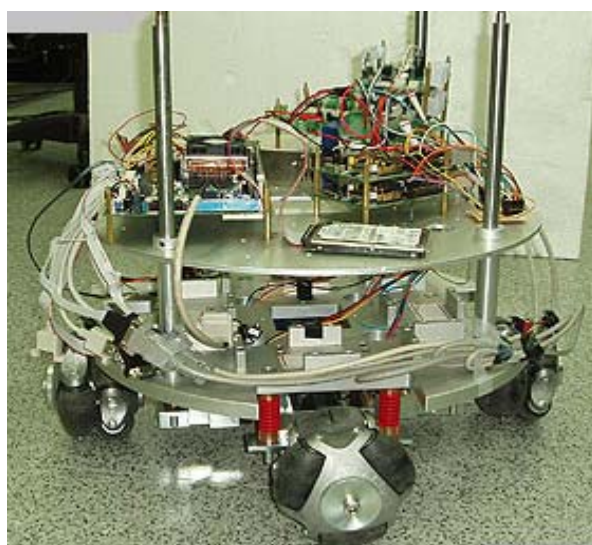


圖 1-1 全向式移動平台結構實體圖

1.5 章節說明

本論文一共分成五章，第一章介紹研究動機及其相關回顧並說明機器人之系統架構。第二章介紹全向式移動平台的架構，以及其定位和控制相關實驗。第三章則介紹所使用之基於 Extended Kalman filter 的 vSLAM 演算法。第四章為實驗結果，用以驗證各演算法的可行性與正確性。第五章為結論與未來工作。



第二章 全向式移動平台之運動控制

全向式移動平台為一種在平面上同時具有 x 、 y 及 θ 三自由度移動的平台，它是以此結構為基礎，提供更好的移動性，使得機器人更具機動性。全向輪有多種形式：[8]是使用支撐滑輪作為其平台，其輪子主要特色，在於除了有水平方向的軸承外，多加了垂直方向的軸承，使得平台除了可於平面穩定行走，同時也藉由垂直方向的軸承轉動達成平台的全方位移動。由卡內基美隆大學的 Ralph Hollis[18]所設計之球形移動平台其特色為在類似倒單擺的二輪式移動平台(沒有輔助輪)一般，可使機器人的重心落在高點而不受傳統其他機器人的限制需把重心壓低避免翻覆情況產生，同時又具有全向式平台優點，可直接朝目標方向移動而不需經過自行旋轉定向的暫態，使此項研究具有一定的優勢。但也由於要求能達到高處重心的條件，使得重心與輪子間的力臂增長，這在要確保平台能平衡的目的下勢必要求馬達需較一般機器人使用之馬達要有較大之扭力輸出，這意味著其電力的損耗將遠大於同級機器人所需，尤其當在有傾斜地面維持靜止時其可能尚須有相當能量支持，方可維持不倒，此點將是在面臨實用時的一項挑戰。除此之外，相較於全向輪式移動平台，其移動速度可能要受到許多限制。

2.1 硬體構造

本論文之實驗用全向式機器人採用全方向性的輪子，如圖 2-1 所示，於輪子的周圍排列了六個滾軸，同時擁有橫向及縱向相互垂直的輪子，可以透過輪子間合力分力的關係，決定任何移動方向；並於輪軸附近，安插了彈簧式的機構，增加輪子與地面的接觸性。比起差動輪平台，全向式移動平台有更好的移動性，其優點為：(1)改變行進方向時，輪軸不需移動。(2)可在原地旋轉平台，不需要旋轉半徑。(3)可同時自旋與曲線運動。此類的移動平台，已被廣泛的討論。

目前常見的全向式移動平台，大多以三軸到五軸的平台為主，在不同的考量下，這三類平台各有本身的優缺點，三軸在以 60 度為倍數的移動角度上，可



圖 2-1 全向輪實體圖

以達到較好的移動效率，四軸則是在 90 度為倍數的角度上，能獲得遠大於其他兩者的效果，至於五軸的效能，則是在各種角度效能都相當平均。此外，在負重能力上，軸數越多負重能力越好。

本論文之實驗用全向式機器人外觀如圖 2-2 所示，其主要規格如下：

- 外型(長×寬×高)：450×450×1050(mm) (加裝全向鏡後)
- 線速度：最高約 1m/sec
- 旋轉速度：最高約 4.44rad/sec
- 輪徑：約 60mm
- 電池規格：12V 2.2Ah / 7.2Ah 鉛酸電池

此全向式移動平台系統是以四個全方位輪所組成，以四個馬達驅動，採用 Faulhaber 公司所提供之馬達，其額定轉速為 6500 rpm，搭配減速齒輪組，其減速比為 14：1。此全向式平台除了能提供與其主軸垂直方向上轉動的方向外，還能提供沿著其主軸平行方向上移動的方向，因此我們可以藉由適當的設計移動機器人的底座，讓其具有任意方向平移或在移動時可以進行旋轉等動作。機器人運算核心為威強公司出產之工業電腦主機板(NOVA-6612)，搭載運算速度為 2GHz 之 CPU，主要用於影像處理及機器人之運動控制。

圖 2-3 為機器人之硬體架構，攝影機是主要感測器，以 Onboard IPC 為運算

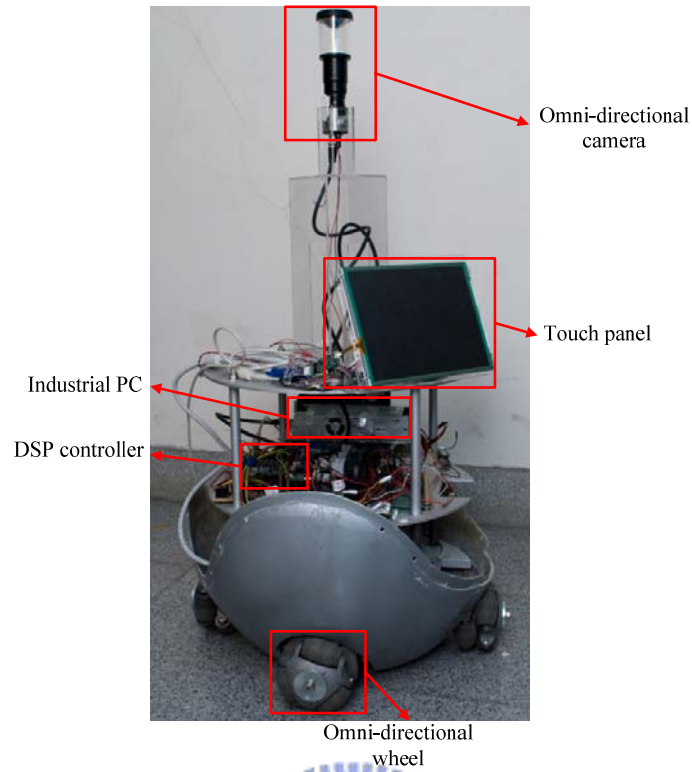


圖 2-2 全向式機器人實體圖

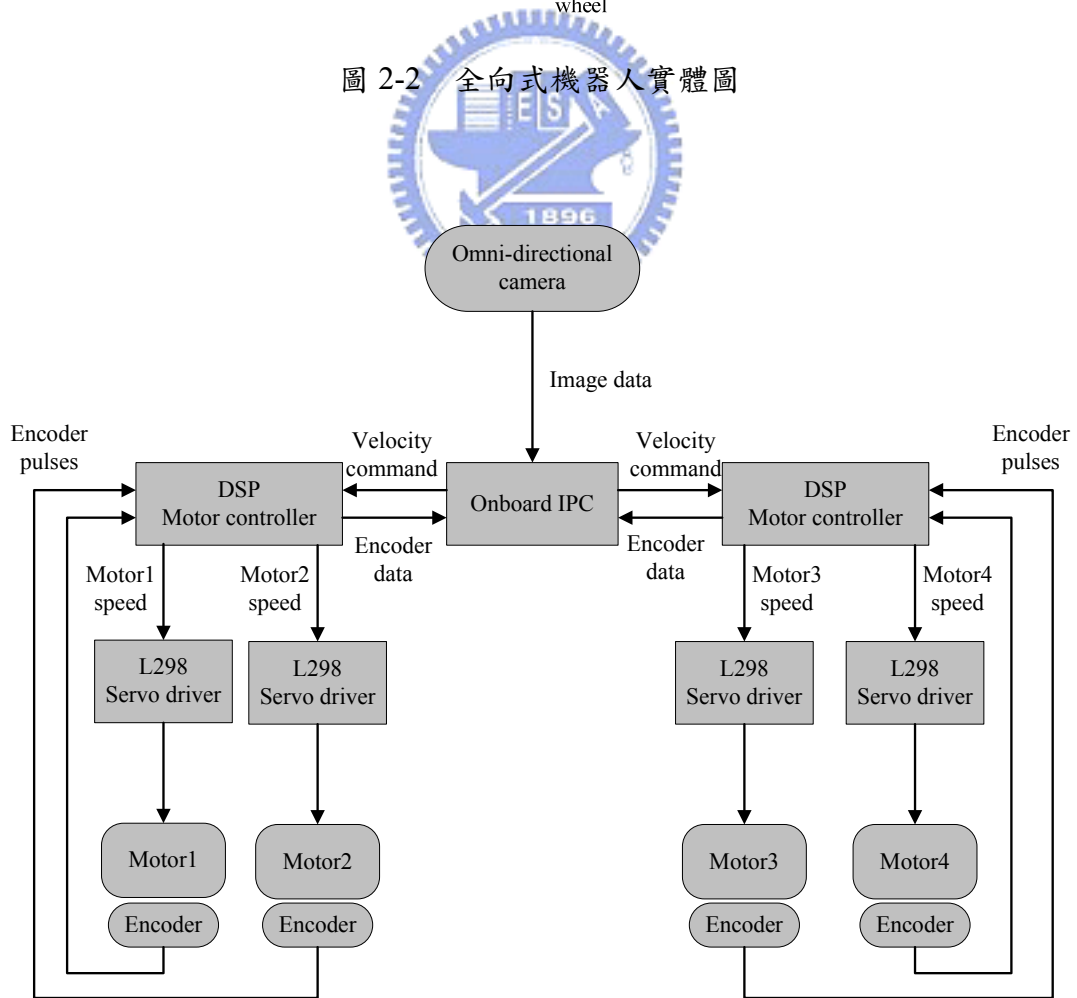


圖 2-3 機器人硬體架構

核心，馬達之伺服控制是由實驗室研製之雙軸運動控制卡完成，此運動控制卡採用 TI 之 DSP 運動控制晶片，DSP 晶片接收到速度命令後傳送給四輪馬達控制卡，經由伺服驅動器驅動馬達的運轉，馬達上之軸編碼器(Shaft encoder)可回傳機器人之移動資訊，經由里程計計算機器人之姿態，其取樣時間為 51ms。

2.2 平台運動模型推導

圖 2-4 為全向式移動平台四輪配置圖，其中標號 1~4 的方框表示四個相隔 90 度輪子的排列位置，L 為從全方位輪至全方位底盤中心點的距離，而 v_i 表示第 i 個輪子方向的速度。這樣的設計安排將使得四個全向輪具有對稱性，使得無論是在機構重心上的考量或是馬達運轉控制上都較為穩定與容易。

在輪子轉動時，我們可得知每一個輪子方向轉動速度與其半徑、輪子自身角速度的關係如下；

$$v_i = r\dot{\phi}_i, i=1 \sim 4 \quad (2-1)$$

其中 r 表示輪子半徑， $\dot{\phi}_i$ 表示輪子自身角速度，如圖 2-5 所示。

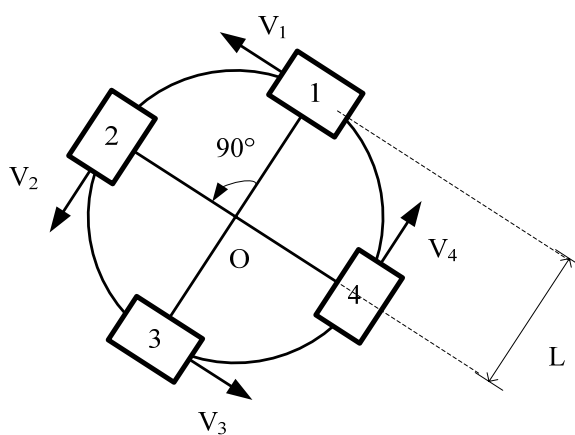


圖 2-4 平台四輪配置圖

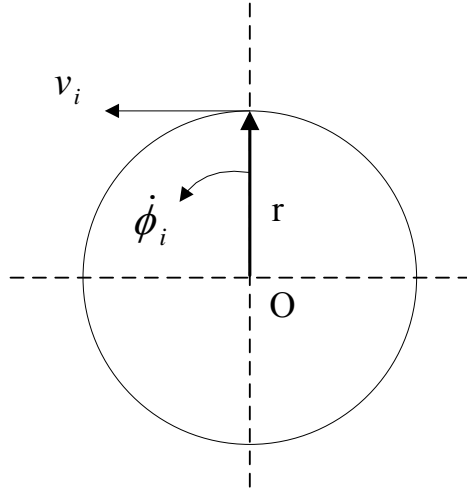


圖 2-5 全向輪轉動示意圖

現假設機器人前進的方向速度為 V ，同時要機器人之角速度為 $\dot{\theta}$ ，此時首先把向量 V 拆解成所定義之座標平面上的二向量合 \dot{x} 和 \dot{y} ：

$$V = \dot{x} + \dot{y} \quad (2-2)$$

然後藉由如圖 2-6 的幾何關係將可分別計算出 k 時刻 \dot{x} 與 \dot{y} 二向量在 v_{1k} 方向的速度而得到如下式：

$$v_{1k} = -\sin\left(\frac{\pi}{4}\right)\dot{x}_k + \cos\left(\frac{\pi}{4}\right)\dot{y}_k + L\dot{\theta}_k \quad (2-3)$$

同理由以上方式，藉由四個輪子在空間中排列的對稱性，推得如下於 k 時刻之平台速度 $\begin{bmatrix} \dot{x}_k & \dot{y}_k & \dot{\theta}_k \end{bmatrix}^T$ 轉換成四輪速度 $v_{1k} \sim v_{4k}$ 的轉換方程式：

$$\begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} = \begin{bmatrix} -\sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & L \\ -\sin\left(\frac{\pi}{4}\right) & -\cos\left(\frac{\pi}{4}\right) & L \\ \sin\left(\frac{\pi}{4}\right) & -\cos\left(\frac{\pi}{4}\right) & L \\ \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) & L \end{bmatrix} \begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} \quad (2-4)$$

在這裡等號右邊的第一個矩陣為常數，因此由(2-4)式我們將可用來對此全向式平台做任意方向速度移動的控制。

為了之後平台的定位控制，要對平台本身座標和世界座標作清楚的定義，

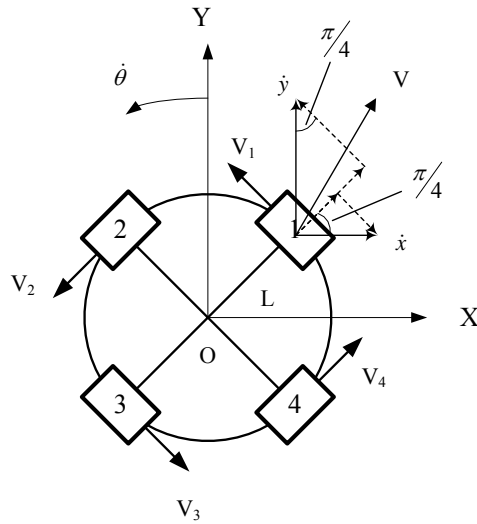


圖 2-6 全向輪之幾何關係

以便估測平台於世界座標的位置。本論文將世界座標(Global)軸定義為 X 和 Y ，平台局部座標(Local)軸為 X_R 和 Y_R 。當平台以旋轉角速度 $\dot{\theta}$ 旋轉之後，平台本身的座標軸 X_R 和 Y_R 相對於世界座標軸 X 和 Y 會產生如圖 2-7 所示一朝向角 θ ，若機器人朝著 θ 方向以同一速度 $[\dot{x}_k \quad \dot{y}_k \quad \dot{\theta}_k]^T$ 繼續移動，根據先前運動方程式的推導方法，同樣可以推導出此時全方位移動機器人的運動方程式如下，藉由此式可由平台速度 $[\dot{x}_k \quad \dot{y}_k \quad \dot{\theta}_k]^T$ 推得四輪速度 $v_{1k} \sim v_{4k}$ ：

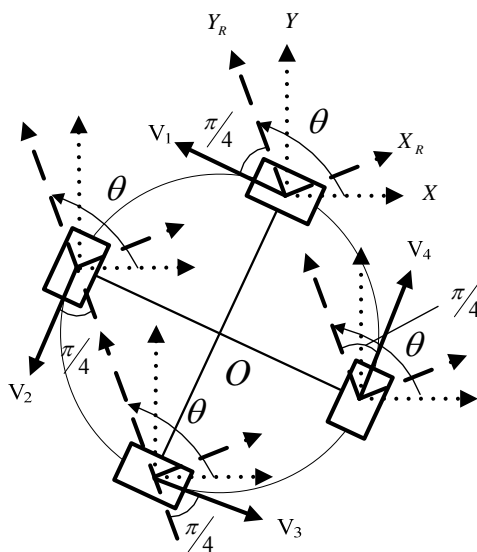


圖 2-7 平台座標旋轉示意圖

$$\begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} = \begin{bmatrix} -\sin(-\frac{\pi}{4} + \theta) & \cos(-\frac{\pi}{4} + \theta) & L \\ -\sin(\frac{3\pi}{4} - \theta) & -\cos(\frac{3\pi}{4} - \theta) & L \\ \sin(-\frac{\pi}{4} + \theta) & -\cos(-\frac{\pi}{4} + \theta) & L \\ \sin(\frac{3\pi}{4} - \theta) & \cos(\frac{3\pi}{4} - \theta) & L \end{bmatrix} \begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} = T(\theta) \begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} \quad (2-5)$$

令速度轉換矩陣為 $T(\theta)$ ，為了估測平台於 k 時刻的姿態，需由四輪速度 $v_{1k} \sim v_{4k}$

先推得平台本身速度 $[\dot{x}_k \quad \dot{y}_k \quad \dot{\theta}_k]^T$ ，將 2-5 式之 $T(\theta)$ 作 Jacobian 反矩陣運算，得

到以 $v_{1k} \sim v_{4k}$ 為輸入的運動模型：

$$\begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} -0.5\sin(-\frac{\pi}{4} + \theta) & -0.5\sin(\frac{3\pi}{4} - \theta) & 0.5\sin(-\frac{\pi}{4} + \theta) & 0.5\sin(\frac{3\pi}{4} - \theta) \\ 0.5\cos(-\frac{\pi}{4} + \theta) & -0.5\cos(\frac{3\pi}{4} - \theta) & -0.5\cos(-\frac{\pi}{4} + \theta) & 0.5\cos(\frac{3\pi}{4} - \theta) \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix} \begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} \quad (2-6)$$

2.2.1 姿態估測 (Odometry)

得到 2-6 式後，便可由此基於平台四個馬達之速度估測，預測機器人姿態。如圖 2-8 所示， (x, y) 為平台所在座標位置，定義機器人於世界座標之朝向角 θ 右手算起為 0 度，前方為 90 度。假設機器人在 Δt 時間內是以固定速度朝 θ 方向沿一直線軌跡移動，若機器人在 $k-1$ 時刻之姿態為 $[x_{k-1} \quad y_{k-1} \quad \theta_{k-1}]^T$ ，且保持移動速度經過 Δt 時間後，其 k 時刻姿態 $[x_k \quad y_k \quad \theta_k]^T$ 可由 $T(\theta)$ 作 Jacobian 反矩陣運算得到並表示成如 2-7 式：

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} -0.5\sin(-\frac{\pi}{4} + \theta) & -0.5\sin(\frac{3\pi}{4} - \theta) & 0.5\sin(-\frac{\pi}{4} + \theta) & 0.5\sin(\frac{3\pi}{4} - \theta) \\ 0.5\cos(-\frac{\pi}{4} + \theta) & -0.5\cos(\frac{3\pi}{4} - \theta) & -0.5\cos(-\frac{\pi}{4} + \theta) & 0.5\cos(\frac{3\pi}{4} - \theta) \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix} \begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} \Delta t$$

$$= \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} 0.5[-\sin(-\frac{\pi}{4} + \theta)v_{1k} - \sin(\frac{3\pi}{4} - \theta)v_{2k} + \sin(-\frac{\pi}{4} + \theta)v_{3k} + \sin(\frac{3\pi}{4} - \theta)v_{4k}]\Delta t \\ 0.5[\cos(-\frac{\pi}{4} + \theta)v_{1k} - \cos(\frac{3\pi}{4} - \theta)v_{2k} - \cos(-\frac{\pi}{4} + \theta)v_{3k} + \cos(\frac{3\pi}{4} - \theta)v_{4k}]\Delta t \\ \frac{1}{4L}(v_{1k} + v_{2k} + v_{3k} + v_{4k})\Delta t \end{bmatrix} \quad (2-7)$$

其中 Δt 為 Sampling time，當機器人移動時，藉由馬達上之 Encoder 回傳四輪 Pulse 數，得到機器人四輪之速度命令 $v_{1k} \sim v_{4k}$ ，進而推算機器人於 Δt 時間內的位移。此即為全向式機器人於世界座標上的姿態估測。

2.3 平台之運動控制

有了機器人姿態估測後，就可對其進行運動控制。其中最簡單的設計是利用一 P 控制器對機器人四輪下控制命令，圖 2-9 為控制方塊圖。設定目標點為 $[x_d \ y_d \ \theta_d]^T$ ，Odometer 得到機器人於 k 時刻的姿態為 $[x_k \ y_k \ \theta_k]^T$ ，將兩者相減得到機器人姿態誤差：

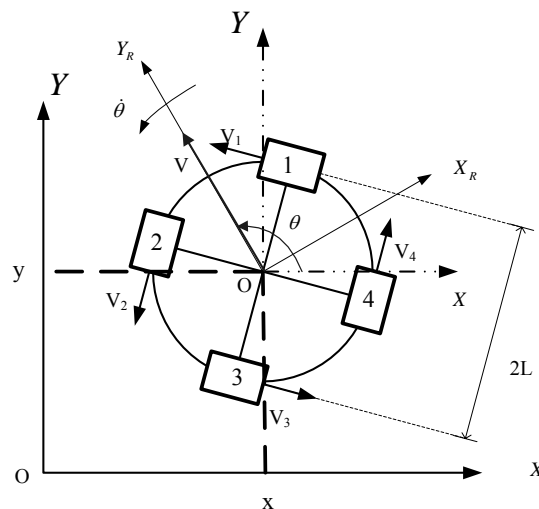


圖 2-8 平台運動模型圖

$$\begin{bmatrix} x_{ek} \\ y_{ek} \\ \theta_{ek} \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix} - \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad (2-8)$$

透過控制器(Controller)將姿態誤差 x_{ek} 、 y_{ek} 、 θ_{ek} 各乘上 K_x 、 K_y 、 K_θ ，得到

控制命令 V_{cmd_k} ：

$$V_{cmd_k} = \begin{bmatrix} \dot{x}_{cmd_k} \\ \dot{y}_{cmd_k} \\ \dot{\theta}_{cmd_k} \end{bmatrix} = \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_\theta \end{bmatrix} \begin{bmatrix} x_{ek} \\ y_{ek} \\ \theta_{ek} \end{bmatrix}, \text{ 且 } K_x, K_y, K_\theta > 0 \quad (2-9)$$

控制命令 V_{cmd_k} 經由四輪轉換函數 $T(\theta)$ 可得對機器人控制之四輪速度命令：

$$\begin{bmatrix} v_{1cmd_k} \\ v_{2cmd_k} \\ v_{3cmd_k} \\ v_{4cmd_k} \end{bmatrix} = \begin{bmatrix} -\sin(-\frac{\pi}{4} + \theta) & \cos(-\frac{\pi}{4} + \theta) & L \\ -\sin(\frac{3\pi}{4} - \theta) & -\cos(\frac{3\pi}{4} - \theta) & L \\ \sin(-\frac{\pi}{4} + \theta) & -\cos(-\frac{\pi}{4} + \theta) & L \\ \sin(\frac{3\pi}{4} - \theta) & \cos(\frac{3\pi}{4} - \theta) & L \end{bmatrix} \begin{bmatrix} \dot{x}_{cmd_k} \\ \dot{y}_{cmd_k} \\ \dot{\theta}_{cmd_k} \end{bmatrix} \quad (2-10)$$

機器人移動後，接收到馬達上之 Encoder 回傳值 $v_{1_pulse} \sim v_{4_pulse}$ ，經由 2-11 式轉

換成速度 $v_{1_actual} \sim v_{4_actual}$ ：

$$v_{i_actual} = (v_{i_pulse} / 2000) \cdot (0.12\pi) / 14 / \Delta t, \quad i = 1 \sim 4 \quad (2-11)$$

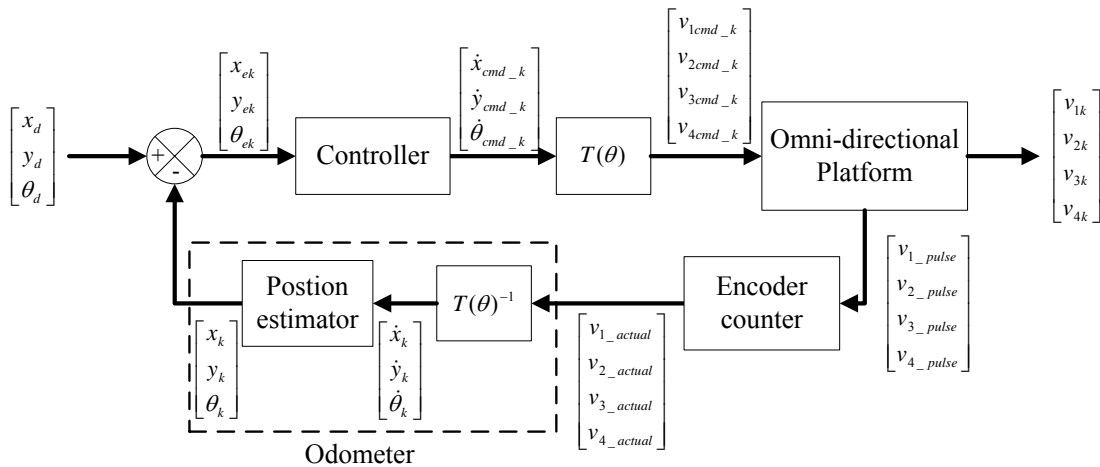


圖 2-9 平台控制方塊圖

其中馬達轉一圈所需的 Pulse 數為 2000，故 $v_{i_pulse} / 2000$ 即為輪子所轉的圈數。

輪子之半徑為 0.06m， 0.12π 即為輪子週長。齒輪減速比為 14， Δt 為 Sampling

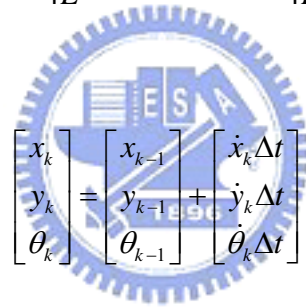
time。如 2-12 式，接著將四輪實際速度 $v_{1_actual} \sim v_{4_actual}$ 經由 $T(\theta)^{-1}$ 反求出機器人

於 k 時刻之速度 $[\dot{x}_k \quad \dot{y}_k \quad \dot{\theta}_k]^T$ ，透過 2-13 式，將 $[\dot{x}_k \quad \dot{y}_k \quad \dot{\theta}_k]^T$ 乘上 Δt ，加上機器

人於 k-1 時刻的姿態 $[x_{k-1} \quad y_{k-1} \quad \theta_{k-1}]^T$ 後，便得到機器人於 k 時刻之姿態

$[x_k \quad y_k \quad \theta_k]^T$ 作為速度控制的輸入，完成一輪控制迴圈。

$$\begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} -0.5 \sin(-\frac{\pi}{4} + \theta) & -0.5 \sin(\frac{3\pi}{4} - \theta) & 0.5 \sin(-\frac{\pi}{4} + \theta) & 0.5 \sin(\frac{3\pi}{4} - \theta) \\ 0.5 \cos(-\frac{\pi}{4} + \theta) & -0.5 \cos(\frac{3\pi}{4} - \theta) & -0.5 \cos(-\frac{\pi}{4} + \theta) & 0.5 \cos(\frac{3\pi}{4} - \theta) \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix} \begin{bmatrix} v_{1_actual} \\ v_{2_actual} \\ v_{3_actual} \\ v_{4_actual} \end{bmatrix} \quad (2-12)$$



$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \dot{x}_k \Delta t \\ \dot{y}_k \Delta t \\ \dot{\theta}_k \Delta t \end{bmatrix} \quad (2-13)$$

2.4 方形軌跡實驗

為了驗證前述所推導之全向式移動機器人速度運動模型，使用上一節所述之控制方法，實驗環境為實驗室 622 門前走廊，如圖 2-10 所示，讓機器人採繞行 3m x 1.5m 的方形軌跡實驗。

實驗過程中設定機器人朝 y 方向前進，以(0,0)為起點，先到達座標點(0,3)，而後橫向移動到(1.5,3)，再往後移到(1.5,0)，最後回到起點(0,0)，以此路徑繞行 2 圈及 3 圈，分別實驗 5 次後量測機器人移動終點與起點的距離誤差，將結果列於表 2-1 及 2-2。

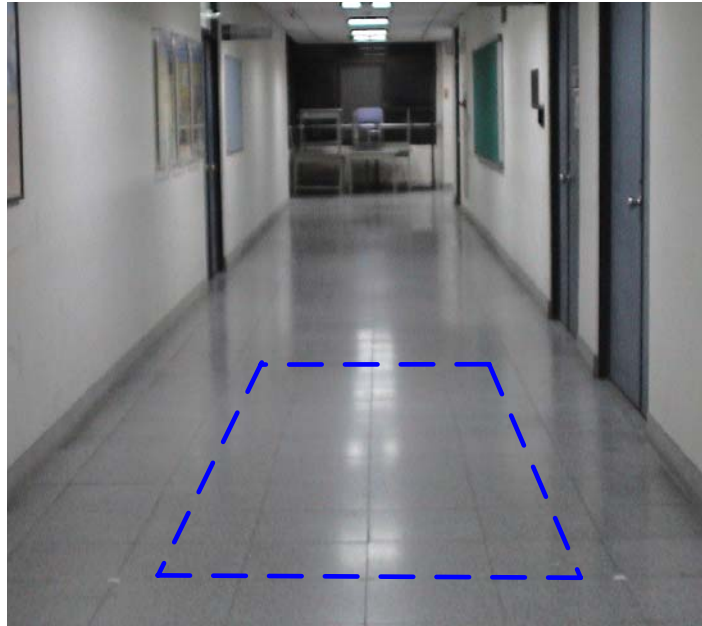


圖 2-10 走廊方形軌跡實驗場景

由實驗結果可知，單純以全向式移動平台的 Odometer 資訊作為定位控制，其誤差是很大的，且此種累積誤差會隨著圈數的增加不斷變大，因此勢必需要依賴其他感測器，以減少累積誤差的問題。

表 2-1 繞行方形 2 圈實驗結果

Error	x (m)	y (m)	θ (°)
Experiment 1	0.11	0.12	10
Experiment 2	0.17	0.11	5
Experiment 3	0.05	0.07	6
Experiment 4	0.12	0.02	3
Experiment 5	0.28	0.23	5
Average error	0.15	0.11	5.8

表 2-2 繞行方形 3 圈實驗結果

Error	x (m)	y (m)	θ ($^{\circ}$)
Experiment 1	0.51	0.06	2
Experiment 2	0.41	0.27	11
Experiment 3	0.29	0.28	5
Experiment 4	0.30	0.07	10
Experiment 5	0.27	0.06	15
Average error	0.36	0.15	8.6



第三章 全向式移動平台之定位設計

本章描述利用攝影機擷取環境中的 Landmark 作為定位演算法的設計。所採用之特徵點擷取及 EKF 定位演算法之主要內容為參考本實驗室先前完成之的 vSLAM 設計[19]。

3.1 全向式攝影機

一般的單眼攝影機所能拍攝到的影像僅限於攝影機前方的視角範圍，用於機器人導航時，會因其狹窄的視野產生限制，例如當機器人移動時，對於同一個 Landmark 無法做長時間追蹤；由於機器人場景的辨識是依據特徵點位置，當視角改變時，會有同一場景因視角的不同而需重新辨識的情形，導致運算量的增加。而相較於一般單眼式攝影機[12]，全向式影像系統能夠即時取得周圍環境 360°之影像資訊，即使因為機器人的移動使得一部分的影像變得不易辨識，讓特徵點的 Scale 改變了，但其他方向的影像改變卻不會很大。如圖 3-1，為全向式攝影機拍攝的兩張影像，左圖為機器人在走廊中心所拍攝，右圖為向影像上方的門移動一段距離後所拍攝。可發現上方的門(上方圈出部份)因為機器人的移動使得形狀大小有明顯變化，特徵點因此不易辨識。下方的佈告欄(下方圈出部份)因為和機器人仍有一段距離，故影像中大小無明顯變化。因此使用全向式攝影機，即使因機器人的移動使一部分特徵點不易辨識，仍有其他方向的特徵點可以清楚辨識，讓定位演算法繼續進行，並增加特徵點的追蹤時間，運用於機器人的環境辨識與導航會更有幫助。

目前常用的全向式視覺系統有如使用雙曲面反射鏡與攝影機所組成的全向式攝影機系統，因其機構簡單，故常見於機器人系統上。如圖 3-2 之類型的攝影機，其可視角度範圍大致在水平零度以下，除了攝影機本身，整個地板平面都是可視範圍，[20]就是利用此種攝影機之影像來判斷牆角，使機器人能在走廊環境中行走。

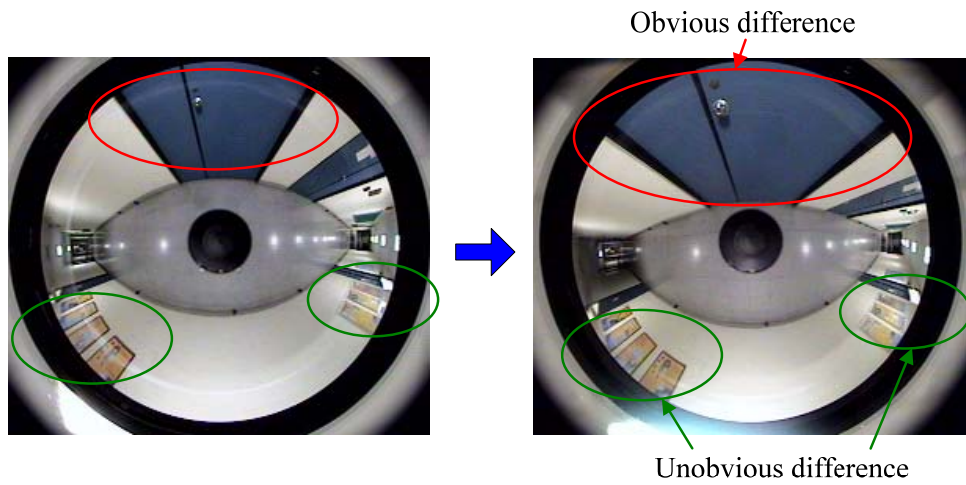


圖 3-1 相鄰影像中之物體大小變化比較

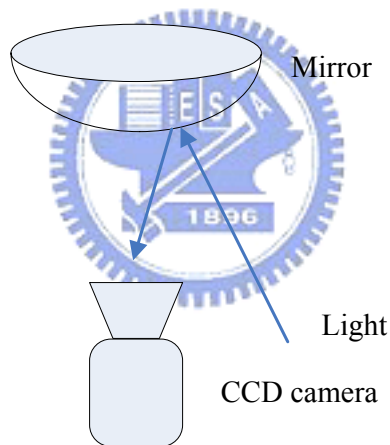


圖 3-2 全向式攝影機示意圖

本論文中所使用的全向式攝影機即是圖 3-2 的類型，為機器人上方之全向式攝影機，為日本 Vstone 公司所生產，型號為 VS-C450-TR，主要規格如下：尺寸為 59(Φ)x199(H) mm，可視範圍：水平以上 15 度以下 60 度，有效影像最大像素 768*494，重量 228g。圖 3-3 為攝影機的照片，圖 3-4 為一張此攝影機所擷取到的影像。

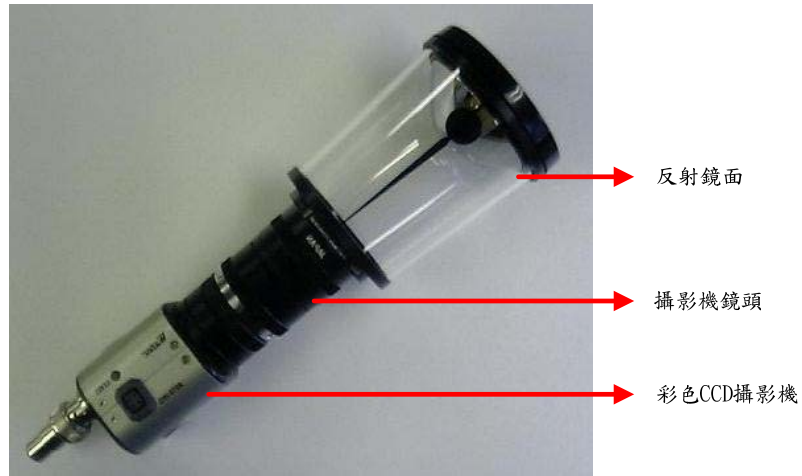


圖 3-3 Vstone VS-C450N-TR 全向式攝影機



圖 3-4 全向式攝影機所擷取之影像

機器人上另外裝置了 Imagination 公司的影像擷取卡 PXC-200，用以擷取全向式攝影機系統之 CCD 所拍攝到的畫面，此影像擷取卡每秒可以擷取 30 張 $640*480 \text{ Pixel}^2$ 大小的影像。

3.2 vSLAM 系統架構

圖 3-5 為 vSLAM 整體系統架構圖[19]，首先將兩張相鄰影像作 Modified SIFT[23]特徵點比對，接著從影像擷取出來的特徵點作距離與位置的估測，作為

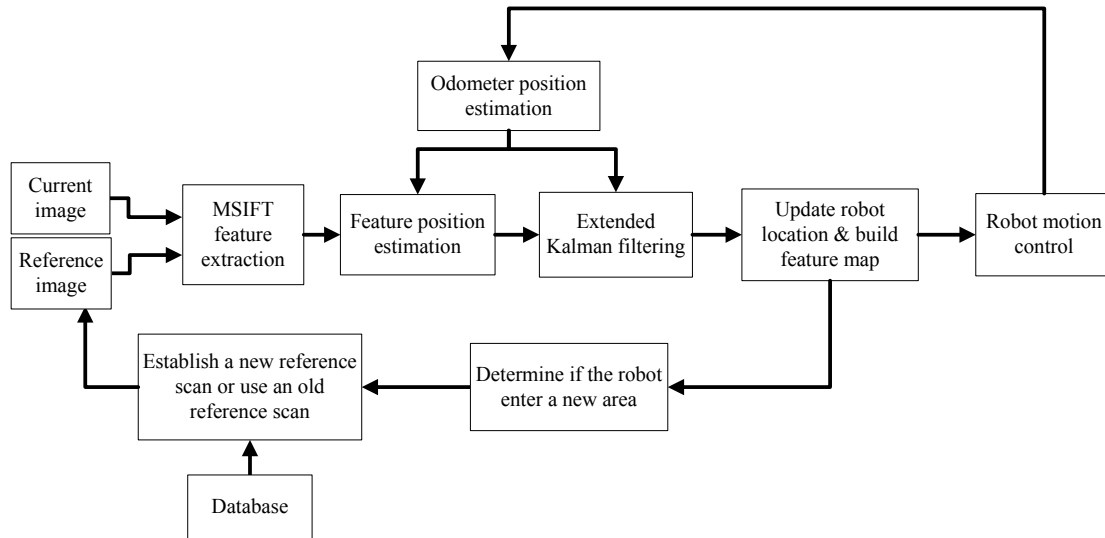


圖 3-5 vSLAM 系統架構圖[19]

全向式攝影機的觀測資訊，並利用基於全向式機器人之速度模型作為系統之預測模型估測下一時刻之系統狀態。將兩者資訊透過 Extended Kalman filter 作系統最佳狀態的估測，更新機器人位置及環境特徵點地圖。最後為延長機器人定位的距離範圍，加上一參考點建立及轉換的流程，當機器人移動到新地點則建立此處的參考點影像，回程時則從資料庫中取回之前已存入的參考點資訊，讓機器人能在大範圍環境中持續重複行走，達成機器人長距離導航的功能。演算法的詳細內容將在接下來的幾個小節中說明。

3.2.1 基於全向式攝影機之特徵點比對演算法

使用環境中自然存在的 Natural Landmark，為現今 SLAM 設計的趨勢。取環境中的特徵點像是物體的邊緣、角落、或是色彩對比明顯的地方作為 Landmark，讓機器人不管移動到任何地方，都能直接從環境中擷取特徵點，是最具一般性的作法。

本論文中所採用的特徵點擷取及辨識方法為 Modified SIFT 演算法(MSIFT) [22][23]，由此方法所擷取出的特徵點，具有“特徵點描述有抵抗旋轉的能力”以及“對於視角的改變有一定的容忍能力”的特性，因此對在移動中之機器人的影

像特徵點辨識上具有優勢。相較於原始 SIFT 演算法[24]，此演算法為了減少特徵點擷取之運算時間，其特徵擷取不使用對不同尺度大小的影像作 DoG 來擷取特徵點，使用較為簡易之角點偵測[23]來擷取環境中特徵點，影像方面運算量較為降低。由於是由 SIFT 演算法延伸而來，仍保有了原始 SIFT 演算法的部份特性，故稱之為 Modified SIFT。圖 3-6 為本論文中所採用基於全向式攝影機之特徵點擷取與比對之演算法流程圖[19]，其作法是先對連續兩張相鄰之影像做角點偵測，找出影像中的特徵點，接著將特徵點及其周圍區域旋轉至同一方向後，對特徵點作描述向量(Descriptor vector)；為了增加特徵點比對的正確性，使用最小距離法(Nearest-neighbor)比對出兩張影像中最相似的點，最後再用 Outlier detection，刪除在已 Match 的特徵點中明顯比對錯誤的點，完成環境中的特徵點擷取[19]。

3.2.2 水平及垂直位置估測

在本論文 vSLAM 中，使用的感測器為單一攝影機，因此無法由單一影像平面得知特徵點的景深，只能得到其相對於機器人的視角，必須藉由對同一特徵點兩次以上的觀測才能得到特徵點的景深資訊。使用三角定位法，圖 3-7 為 Landmark 深度估測方法之示意圖。由機器人在 t 時刻和 $t+\Delta t$ 時刻相對於 Landmark

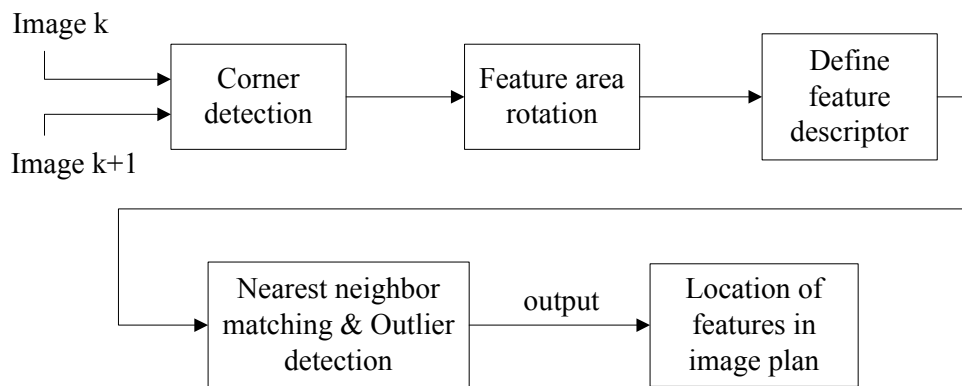


圖 3-6 基於全向式攝影機之特徵點擷取與比對演算法流程圖[19]

的視角 Φ_1 、 Φ_2 ，及機器人在兩個時間點的直線距離 ΔS ，以此三角關係經由(3-1)求得機器人相對於 Landmark 的距離 d_z ：

$$\begin{aligned}
 h \cot(\phi_1) + h \cot(\phi_2) &= \Delta S \\
 h &= \frac{\Delta S}{\cot(\phi_1) + \cot(\phi_2)} \\
 \Rightarrow d_z &= \frac{\Delta S}{\cot(\phi_1) + \cot(\phi_2)} \cdot \frac{1}{\sin(\phi_2)} \quad (3-1)
 \end{aligned}$$

由(3-1)式可推得 Landmark 的座標位置：

$$\begin{cases}
 f_x = x_{t+\Delta t} + d_z \cos(\phi_2) \\
 f_y = y_{t+\Delta t} + d_z \sin(\phi_2)
 \end{cases} \quad (3-2)$$

由以上式子可知，直線距離 ΔS 直接的影響 d_z 的準確性。 ΔS 為機器人的 Odometer 資訊，因此當 Odometer 誤差過大， d_z 的資訊也就變得不可靠。為求得 ΔS 的容許誤差範圍，讓機器人於實驗室走廊前行 2 公尺作多次實驗，測試不同誤差的 ΔS 所觀測到的 Landmark 與機器人相對距離 d_z 及 x, y, θ 之標準差。表 3-1 為實驗結果，可發現百分比誤差越大， d_z 標準差也越大，不確定性也就越高。由於 d_z 為全向式攝影機之觀測值，故 d_z 的量測也間接影響 EKF-SLAM 的定位結果，當百分比誤差到了 5%，其定位效果已經很差，故 ΔS 可接受的誤差範圍約在 2%~3% 左右。

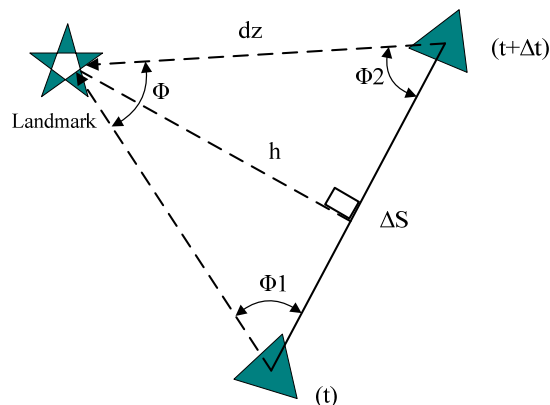


圖 3-7 Landmark 水平位置估測示意圖

表 3-1 不同誤差 ΔS 與 dz 及定位之相對關係

ΔS 百分比誤差	2%	3%	5%	6%
dz 標準差	0.108m	0.174m	0.188m	0.195m
x 標準差	0.09m	0.13m	0.17m	0.18m
y 標準差	0.02m	0.05m	0.07m	0.09m
θ 標準差	0.5°	0.7°	1.3°	1.7°

計算出特徵點的水平位置後，特徵點高度也可藉由水平位置資訊估算出來。由圖 3-8 可知，在機器人高度 h 已知的情形下，可透過機器人與 Landmark 之距離 r 和機器人與 Landmark 之垂直視角 ϕ 藉由(3-3)式求得 Landmark 高度 z 。

$$(h - z)\tan\phi = r \Rightarrow z = \frac{h \tan\phi - r}{\tan\phi} \quad (3-3)$$

為了找出攝影機垂直視角與影像平面上 pixel 數間的數學關係，本論文參考 [19] 之校正實驗，在 120cm x 60cm 的紙板上貼滿 15 x 8 個等間距的格子點並置於牆面(如圖 3-9)，讓全向式移動平台在與牆面已知距離的情形下，從全向式攝影機觀察各格子點距離鏡心的 pixel 值並紀錄，算出各 pixel 值對應的垂直視角 ϕ 。根據此關係求得一近似二次曲線作為估測 Landmark 高度的公式，如(3-4)式。

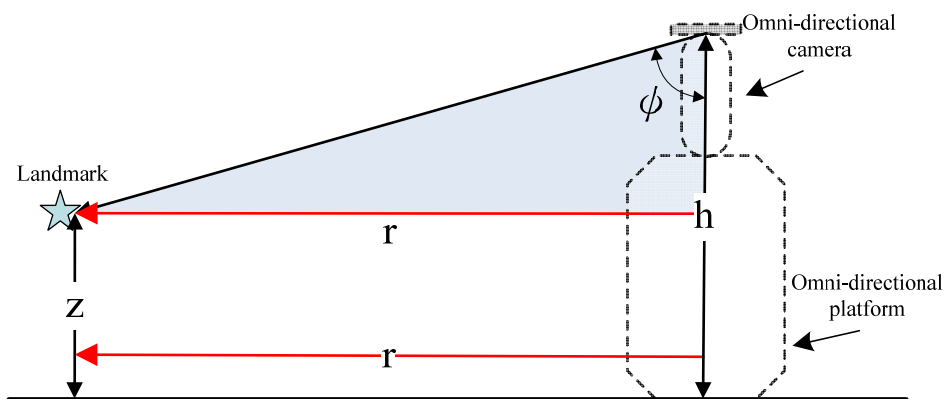


圖 3-8 Landmark 垂直估測示意圖

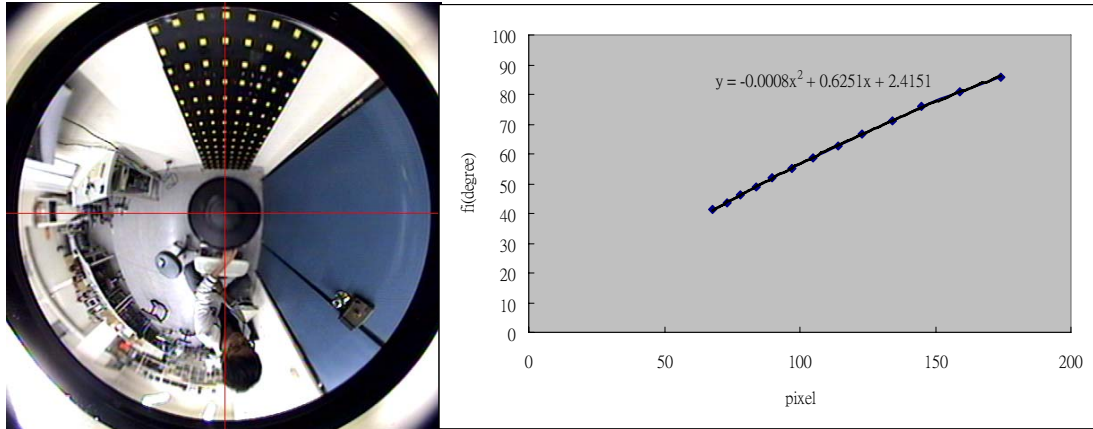


圖 3-9 視角校正及其關係式[19]

其中 ϕ 為垂直視角($^{\circ}$)， x 為影像平面上鏡心距離特徵點的 pixel 值。

$$\phi = -0.0008x^2 + 0.6251x + 2.4151 \quad (3-4)$$

3.2.3 基於 Extended Kalman filter 之 SLAM 演算法

在 SLAM 中，機器人位置與地圖建立之不確定性問題常是主要解決的議題之一[10]。Kalman filter 是一種利用系統模型與系統狀態量測值，對系統狀態作最佳估測的方法。利用系統狀態方程式預測當時的狀態值，感測器測量系統狀態之量測值，兩者都有其不確定性存在，Kalman filter 即是依據兩者資訊之可靠程度，以最佳比例融合兩者數值，得到系統之最佳化狀態估測值。因本系統之模型為非線性系統，為了達成對全向式機器人姿態及環境地圖[25]的狀態作最佳估測，並使用全向式攝影機作為感測裝置，因此採用 Extended Kalman filter 作資訊的融合。本估測系統之系統狀態包含了機器人本身姿態及環境地圖的位置資訊，狀態向量以 Y_k 表示，如 3-5 式，其中 X_k 代表機器人姿態，以水平位置座標(x, y)及朝向角 θ 表示， \mathbf{m} 則代表環境地圖資訊，也就是環境中所有 Landmark 位置，分別以 $(m_{n,x}, m_{n,y}, m_{n,z})$ 表示，下標 n 代表第 n 個 Landmark，下標 x, y, z 則為平面座標(x, y)及高度 z 。

$$\text{State vector : } Y_k = \begin{pmatrix} X_k \\ \mathbf{m} \end{pmatrix}^T$$

$$= (x \quad y \quad \theta \quad m_{1,x} \quad m_{1,y} \quad m_{1,z} \quad \dots \quad m_{n,x} \quad m_{n,y} \quad m_{n,z}) \quad (3-5)$$

圖 3-10 為 EKF 設計概念圖，主要分成兩個階段。在 Prediction step 利用機器人的 Motion model 預測出系統姿態 $\bar{y}_k, \bar{\Sigma}_k$ ，接著在 Update step 整合系統狀態及感測器量測出的測量值 z_k^i ，計算出 Filter gain 以更新對系統狀態的估測，更新完後的狀態資訊再回傳到 Prediction step，作為下一時刻狀態預測依據 y_{k-1}, Σ_{k-1} 。

3.3 平台之運動模型

本實驗室過去已完成之方法[19]中，所使用之平台為二獨立驅動輪式機構，本論文所使用之平台為全向式移動平台，除了移動性高，運動模型也有所差異。使用於 2.2 節所推導之基於全向式機器人之速度移動模型(Velocity motion model)。為了將其運動模型使用於 EKF-SLAM 演算法，接下來將推導[26]平台包含 Noise 之實際運動模型及 Jacobian matrix G_t 。

全向式機器人的速度命令主要由四輪所組成，如 2.2 節所述，速度控制模型

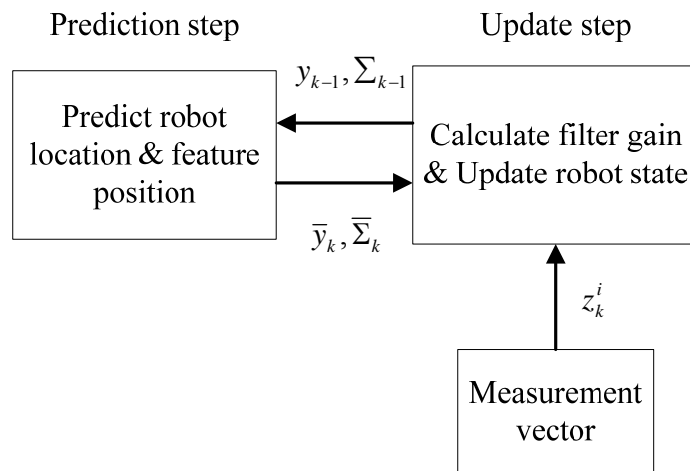


圖 3-10 EKF 演算法概念圖

即以 v_{1k} 、 v_{2k} 、 v_{3k} 、 v_{4k} 為輸入，假設機器人在 Δt 時間內是以一固定速度移動一段直線距離：

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} 0.5[-\sin(-\frac{\pi}{4} + \theta)v_{1k} - \sin(\frac{3\pi}{4} - \theta)v_{2k} + \sin(-\frac{\pi}{4} + \theta)v_{3k} + \sin(\frac{3\pi}{4} - \theta)v_{4k}] \Delta t \\ 0.5[\cos(-\frac{\pi}{4} + \theta)v_{1k} - \cos(\frac{3\pi}{4} - \theta)v_{2k} - \cos(-\frac{\pi}{4} + \theta)v_{3k} + \cos(\frac{3\pi}{4} - \theta)v_{4k}] \Delta t \\ \frac{1}{4L}(v_{1k} + v_{2k} + v_{3k} + v_{4k}) \Delta t \end{bmatrix} \quad (3-6)$$

不過實際上機器人移動時並非如運動模型般理想，必定有誤差存在，故以 (3-7) 式表示全向式機器人四輪實際移動速度[26]：

$$u_t = \begin{bmatrix} \hat{v}_{1k} \\ \hat{v}_{2k} \\ \hat{v}_{3k} \\ \hat{v}_{4k} \end{bmatrix} = \begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} + \begin{bmatrix} \mathcal{E}_{(\alpha_{11}|v_1| + \alpha_{12}|v_2| + \alpha_{13}|v_3| + \alpha_{14}|v_4|)^2} \\ \mathcal{E}_{(\alpha_{21}|v_1| + \alpha_{22}|v_2| + \alpha_{23}|v_3| + \alpha_{24}|v_4|)^2} \\ \mathcal{E}_{(\alpha_{31}|v_1| + \alpha_{32}|v_2| + \alpha_{33}|v_3| + \alpha_{34}|v_4|)^2} \\ \mathcal{E}_{(\alpha_{41}|v_1| + \alpha_{42}|v_2| + \alpha_{43}|v_3| + \alpha_{44}|v_4|)^2} \end{bmatrix} \quad (3-7)$$

$\hat{v}_{1k} \sim \hat{v}_{4k}$ 代表四輪實際的速度， $v_{1k} \sim v_{4k}$ 為四輪理想值， \mathcal{E} 為雜訊(在 EKF 中均假設為高斯雜訊)，其中參數 α_{11} 為 1 號輪影響本身的雜訊比例， α_{12} 為 2 號輪影響 1 號輪的雜訊比例， α_{13} 為 3 號輪影響 1 號輪的雜訊比例， α_{14} 為 4 號輪影響 1 號輪的雜訊比例。同理可推得 $\alpha_{21} \sim \alpha_{24}$ 、 $\alpha_{31} \sim \alpha_{34}$ 、 $\alpha_{41} \sim \alpha_{44}$ 所代表的意義。移動情況越不準確， α 值越高。在 EKF 中，假設速度控制雜訊之 Covariance matrix 設定為 M_t ：

$$M_t = \begin{bmatrix} \mathcal{E}_{(\alpha_{11}|v_1| + \alpha_{12}|v_2| + \alpha_{13}|v_3| + \alpha_{14}|v_4|)^2} & 0 & 0 & 0 \\ 0 & \mathcal{E}_{(\alpha_{21}|v_1| + \alpha_{22}|v_2| + \alpha_{23}|v_3| + \alpha_{24}|v_4|)^2} & 0 & 0 \\ 0 & 0 & \mathcal{E}_{(\alpha_{31}|v_1| + \alpha_{32}|v_2| + \alpha_{33}|v_3| + \alpha_{34}|v_4|)^2} & 0 \\ 0 & 0 & 0 & \mathcal{E}_{(\alpha_{41}|v_1| + \alpha_{42}|v_2| + \alpha_{43}|v_3| + \alpha_{44}|v_4|)^2} \end{bmatrix} \quad (3-8)$$

本論文中以實驗結果設定參數 α 的值。給全向式機器人其中一輪設定好之速度命令，使其移動一段距離，量測移動距離與時間，算出實際速度，比較速度

命令與實際命令之誤差後，設定速度誤差的百分比。經數次實驗得到速度誤差約在控制命令的 4% 左右，故往後的實驗中即以此值來設定運動模型的雜訊。

3.3.1 Prediction model

此步驟為當機器人移動時，藉由機器人之運動模型預測下一時刻的機器人姿態，而在 SLAM 中，狀態預測還包含了 Landmark 之位置估測，預測模型如下：

$$Y_k = Y_{k-1} + \begin{bmatrix} 0.5[-\sin(-\frac{\pi}{4} + \theta)v_{1k} - \sin(\frac{3\pi}{4} - \theta)v_{2k} + \sin(-\frac{\pi}{4} + \theta)v_{3k} + \sin(\frac{3\pi}{4} - \theta)v_{4k}]\Delta t \\ 0.5[\cos(-\frac{\pi}{4} + \theta)v_{1k} - \cos(\frac{3\pi}{4} - \theta)v_{2k} - \cos(-\frac{\pi}{4} + \theta)v_{3k} + \cos(\frac{3\pi}{4} - \theta)v_{4k}]\Delta t \\ \frac{1}{4L}(v_{1k} + v_{2k} + v_{3k} + v_{4k})\Delta t \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (3-9)$$

其中右式之前三項為機器人之姿態預測，即(3-6)式之全向式機器人運動模型，代表機器人姿態隨模型的預測改變；其他元素則為零，代表機器人的狀態不會影響 Landmark 的狀態，與前一刻相同。接著將(3-9)式簡化：

$$Y_k = Y_{k-1} + F_x^T \begin{bmatrix} 0.5[-\sin(-\frac{\pi}{4} + \theta)v_{1k} - \sin(\frac{3\pi}{4} - \theta)v_{2k} + \sin(-\frac{\pi}{4} + \theta)v_{3k} + \sin(\frac{3\pi}{4} - \theta)v_{4k}]\Delta t \\ 0.5[\cos(-\frac{\pi}{4} + \theta)v_{1k} - \cos(\frac{3\pi}{4} - \theta)v_{2k} - \cos(-\frac{\pi}{4} + \theta)v_{3k} + \cos(\frac{3\pi}{4} - \theta)v_{4k}]\Delta t \\ \frac{1}{4L}(v_{1k} + v_{2k} + v_{3k} + v_{4k})\Delta t \end{bmatrix} \quad (3-10)$$

$$F_x = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ & & & \underbrace{\hspace{2cm}}_{3N \text{ columns}} & & \end{bmatrix} \quad (3-11)$$

F 為同一維度的轉換矩陣，將 3 維矩陣擴展成 3N+3 維。

接著將(3-6)式全向式機器人姿態向量 $X_k = g(u_k, X_{k-1})$ 在 μ_{k-1} 處以一階

Taylor 展開式作線性近似：

$$g(u_k, X_{k-1}) \approx g(u_k, \mu_{k-1}) + G_k (X_{k-1} - \mu_{k-1}) \quad (3-12)$$

其中 μ_{k-1} 為 EKF 在 k-1 時刻所估測出的狀態，Jacobian matrix G_k 即為 $g(u_k, X_{k-1})$

對 X_{k-1} 之偏導數：

$$G_k = \frac{\partial g(u_k, \mu_{k-1})}{\partial X_{k-1}} = \begin{bmatrix} \frac{\partial x'}{\partial \mu_{k-1,x}} & \frac{\partial x'}{\partial \mu_{k-1,y}} & \frac{\partial x'}{\partial \mu_{k-1,\theta}} \\ \frac{\partial y'}{\partial \mu_{k-1,x}} & \frac{\partial y'}{\partial \mu_{k-1,y}} & \frac{\partial y'}{\partial \mu_{k-1,\theta}} \\ \frac{\partial \theta'}{\partial \mu_{k-1,x}} & \frac{\partial \theta'}{\partial \mu_{k-1,y}} & \frac{\partial \theta'}{\partial \mu_{k-1,\theta}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -0.5 \cos(-\frac{\pi}{4} + \theta)v_{1k} + 0.5 \cos(\frac{3\pi}{4} - \theta)v_{2k} + 0.5 \cos(-\frac{\pi}{4} + \theta)v_{3k} - 0.5 \cos(\frac{3\pi}{4} - \theta)v_{4k} \\ 0 & 1 & -0.5 \sin(-\frac{\pi}{4} + \theta)v_{1k} - 0.5 \sin(\frac{3\pi}{4} - \theta)v_{2k} + 0.5 \sin(-\frac{\pi}{4} + \theta)v_{3k} + 0.5 \sin(\frac{3\pi}{4} - \theta)v_{4k} \\ 0 & 0 & 1 \end{bmatrix} \quad (3-13)$$

另求出 Jacobian matrix V_k ， $g(u_k, X_{k-1})$ 對 u_k 之偏導數：

$$V_k = \frac{\partial g(u_k, \mu_{k-1})}{\partial u_k} = \begin{bmatrix} \frac{\partial x_k}{\partial v_{1k}} & \frac{\partial x_k}{\partial v_{2k}} & \frac{\partial x_k}{\partial v_{3k}} & \frac{\partial x_k}{\partial v_{4k}} \\ \frac{\partial y_k}{\partial v_{1k}} & \frac{\partial y_k}{\partial v_{2k}} & \frac{\partial y_k}{\partial v_{3k}} & \frac{\partial y_k}{\partial v_{4k}} \\ \frac{\partial \theta_k}{\partial v_{1k}} & \frac{\partial \theta_k}{\partial v_{2k}} & \frac{\partial \theta_k}{\partial v_{3k}} & \frac{\partial \theta_k}{\partial v_{4k}} \end{bmatrix}$$

$$= \begin{bmatrix} -0.5 \sin(-\frac{\pi}{4} + \theta) & -0.5 \sin(\frac{3\pi}{4} - \theta) & 0.5 \sin(-\frac{\pi}{4} + \theta) & 0.5 \sin(\frac{3\pi}{4} - \theta) \\ 0.5 \cos(-\frac{\pi}{4} + \theta) & -0.5 \cos(\frac{3\pi}{4} - \theta) & -0.5 \cos(-\frac{\pi}{4} + \theta) & 0.5 \cos(\frac{3\pi}{4} - \theta) \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix} \quad (3-14)$$

V_k 為控制空間雜訊與狀態空間雜訊的線性轉換，由(3-8)式之 Covariance matrix

M_k ，該時刻的移動雜訊之 Covariance matrix R_k 可由下式轉換得到：

$$R_k = V_k \cdot M_k \cdot V_k^T \quad (3-15)$$

最後，將包含 Noise 之完整的運動模型表示如下：

$$Y_k = Y_{k-1} + F_x^T \underbrace{\begin{bmatrix} 0.5[-\sin(-\frac{\pi}{4} + \theta)v_{1k} - \sin(\frac{3\pi}{4} - \theta)v_{2k} + \sin(-\frac{\pi}{4} + \theta)v_{3k} + \sin(\frac{3\pi}{4} - \theta)v_{4k}]\Delta t \\ 0.5[\cos(-\frac{\pi}{4} + \theta)v_{1k} - \cos(\frac{3\pi}{4} - \theta)v_{2k} - \cos(-\frac{\pi}{4} + \theta)v_{3k} + \cos(\frac{3\pi}{4} - \theta)v_{4k}]\Delta t \\ \frac{1}{4L}(v_{1k} + v_{2k} + v_{3k} + v_{4k})\Delta t \end{bmatrix}}_{g(u_k, Y_{k-1})} + N(0, F_x^T R_k F_x) \quad (3-16)$$

與 EKF 同理，將 $g(u_k, X_{k-1})$ 以一階 Taylor 線性展開近似可得 Jacobian matrix G_k ：

$$G_k = I + F_x^T \begin{bmatrix} 0 & 0 & -0.5 \cos(-\frac{\pi}{4} + \theta)v_{1k} + 0.5 \cos(\frac{3\pi}{4} - \theta)v_{2k} + 0.5 \cos(-\frac{\pi}{4} + \theta)v_{3k} - 0.5 \cos(\frac{3\pi}{4} - \theta)v_{4k} \\ 0 & 0 & -0.5 \sin(-\frac{\pi}{4} + \theta)v_{1k} - 0.5 \sin(\frac{3\pi}{4} - \theta)v_{2k} + 0.5 \sin(-\frac{\pi}{4} + \theta)v_{3k} + 0.5 \sin(\frac{3\pi}{4} - \theta)v_{4k} \\ 0 & 0 & 0 \end{bmatrix} F_x \quad (3-17)$$

新增的移動雜訊 Covariance R_k 則以 F_x 擴展成： $F_x^T R_k F_x$ (3-18)

從(3-15)、(3-16)、(3-18)可整合出系統狀態之 Covariance matrix $\bar{\Sigma}_k$ ：

$$\bar{\Sigma}_k = G_k \Sigma_{k-1} G_k^T + F_x^T R_k F_x \quad (3-19)$$

其中 Σ_{k-1} 為前一時刻所估測出之系統 Covariance。

3.3.2 Update model

本論文中主要所使用的感測器為全向式攝影機，故量測模型(Measurement model)即是利用攝影機的觀測值為依據作設定，此 Model 參考本實驗室[19]先前的設定，Observed measurement vector 即為感測器量測到之狀態向量，將其定義

為 z_k^i :

$$\text{Observed measurement vector : } z_k^i = \begin{bmatrix} r_k^i \\ \theta_k^i \\ \phi_k^i \end{bmatrix} \quad (3-20)$$

其中 r_k^i 代表在 t 時刻對第 i 個 Landmark 所觀測到的相對距離，量測方法如之前 3.1.2 節所提，利用一非即時性方法求得； θ_k^i 代表對 Landmark 的水平視角，在全向式影像平面上，Landmark 所出現的方向即是實際相對於機器人所在的方向，因此計算 Landmark 在影像平面上對於鏡心的角度即可得知 θ_k^i ； ϕ_k^i 為相對於 Landmark 的垂直視角，使用影像平面中 Landmark 相距鏡心的 Pixel 值代入 3-4 式之 Pixel 與垂直視角之轉換公式即可求得。

將觀測 Model 以包含 Noise 之完整 Measurement model 表示如下：

$$\hat{z}_k^i = \begin{bmatrix} \sqrt{(m_{i,x} - x)^2 + (m_{i,y} - y)^2} \\ a \tan 2(m_{i,y} - y, m_{i,x} - x) - \theta \\ a \tan 2(\underbrace{\sqrt{(m_{i,x} - x)^2 + (m_{i,y} - y)^2}, h - m_{i,z}}_{h(Y_k, j)}) \end{bmatrix} + N \left[\begin{array}{ccc} \sigma_r & 0 & 0 \\ 0 & \sigma_\theta & 0 \\ 0 & 0 & \sigma_\phi \end{array} \right] \quad (3-21)$$

此式代表對第 j 個 Landmark 之觀測模型，其中 Q_k 為 Observation noise 之 Covariance matrix， σ_r 、 σ_θ 、 σ_ϕ 分別代表感測器對 Landmark 的觀測值 r 、 θ 、 ϕ 之標準差，讓機器人在環境中行走一段距離後，將對環境中特徵點的估測值紀錄下來，並計算其標準差，經數次實驗後得到 $\sigma_r = 0.6m$ 、 $\sigma_\theta = 3.5^\circ$ 、 $\sigma_\phi = 5.6^\circ$ ，此數值即代表感測器的 Noise。將 Observation function $h(Y_k, j)$ 近似為線性方程式：

$$h(Y_k, j) \approx h(\bar{\mu}_k, j) + H_k^j (X_k - \bar{\mu}_k) \quad (3-22)$$

其中 H_k^j 為 $h(Y_k, j)$ 對於系統狀態 Y_k 之偏導數，此時 $h(Y_k, j)$ 僅對機器人的姿態

X_k 以及第 j 個 Landmark 的資訊 m_j 偏微有意義，故將 H_k^j 改寫成：

$$H_k^j = h_k^j F_{x,j} \quad (3-23)$$

h_k^j 代表對 X_k 及 m_j 的偏導數，而 $F_{x,j}$ 則是將 3×6 維之 h_k^j 擴展成 $3 \times (3N+3)$ 之轉換矩陣。

$$h_k^j = \frac{\partial h(\bar{\mu}_k, j)}{\partial Y_k} = \begin{bmatrix} \frac{\partial r_k^j}{\partial \bar{\mu}_{k,x}} & \frac{\partial r_k^j}{\partial \bar{\mu}_{k,y}} & \frac{\partial r_k^j}{\partial \bar{\mu}_{k,\theta}} & \frac{\partial r_k^j}{\partial \bar{\mu}_{k,m_{jx}}} & \frac{\partial r_k^j}{\partial \bar{\mu}_{k,m_{jy}}} & \frac{\partial r_k^j}{\partial \bar{\mu}_{k,m_{jz}}} \\ \frac{\partial \theta_k^j}{\partial \bar{\mu}_{k,x}} & \frac{\partial \theta_k^j}{\partial \bar{\mu}_{k,y}} & \frac{\partial \theta_k^j}{\partial \bar{\mu}_{k,\theta}} & \frac{\partial \theta_k^j}{\partial \bar{\mu}_{k,m_{jx}}} & \frac{\partial \theta_k^j}{\partial \bar{\mu}_{k,m_{jy}}} & \frac{\partial \theta_k^j}{\partial \bar{\mu}_{k,m_{jz}}} \\ \frac{\partial \phi_k^j}{\partial \bar{\mu}_{k,x}} & \frac{\partial \phi_k^j}{\partial \bar{\mu}_{k,y}} & \frac{\partial \phi_k^j}{\partial \bar{\mu}_{k,\theta}} & \frac{\partial \phi_k^j}{\partial \bar{\mu}_{k,m_{jx}}} & \frac{\partial \phi_k^j}{\partial \bar{\mu}_{k,m_{jy}}} & \frac{\partial \phi_k^j}{\partial \bar{\mu}_{k,m_{jz}}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{m_{j,x} - \bar{\mu}_{k,x}}{\sqrt{q_k}} & \frac{m_{j,y} - \bar{\mu}_{k,y}}{\sqrt{q_k}} & 0 & \frac{m_{j,x} - \bar{\mu}_{k,x}}{\sqrt{q_k}} & \frac{m_{j,y} - \bar{\mu}_{k,y}}{\sqrt{q_k}} & 0 \\ \frac{m_{j,y} - \bar{\mu}_{k,y}}{q_k} & \frac{m_{j,x} - \bar{\mu}_{k,x}}{q_k} & -1 & \frac{m_{j,y} - \bar{\mu}_{k,y}}{q_k} & \frac{m_{j,x} - \bar{\mu}_{k,x}}{q_k} & 0 \\ \frac{(m_{j,x} - \bar{\mu}_{k,x})(h - m_{j,z})}{(q_k + (h - m_{j,z})^2) \cdot \sqrt{q_k}} & \frac{(m_{j,y} - \bar{\mu}_{k,y})(h - m_{j,z})}{(q_k + (h - m_{j,z})^2) \cdot \sqrt{q_k}} & 0 & \frac{(m_{j,x} - \bar{\mu}_{k,x})(h - m_{j,z})}{(q_k + (h - m_{j,z})^2) \cdot \sqrt{q_k}} & \frac{(m_{j,y} - \bar{\mu}_{k,y})(h - m_{j,z})}{(q_k + (h - m_{j,z})^2) \cdot \sqrt{q_k}} & \frac{\sqrt{q_k}}{q_k + (h - m_{j,z})^2} \end{bmatrix} \quad (3-24)$$

其中 $q_k = (m_{j,x} - \bar{\mu}_{k,x})^2 + (m_{j,y} - \bar{\mu}_{k,y})^2$

$$F_{x,j} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \quad (3-25)$$

由上述之推導結果，可求得

$$\text{Filter gain : } K_k^j = \bar{\Sigma}_k H_k^{jT} (H_k^j \bar{\Sigma}_k H_k^{jT} + Q_k)^{-1} \quad (3-26)$$

利用 Filter gain 及 Innovation vector 更新系統之狀態估測向量 $\bar{\mu}_k$:

$$\bar{\mu}_k = \bar{\mu}_k + K_k^j (z_k^j - \hat{z}_k^j) \quad (3-27)$$

最後更新系統之 Covariance matrix $\bar{\Sigma}_k$:

$$\bar{\Sigma}_k = (I - K_k^j H_k^j) \cdot \bar{\Sigma}_k \quad (3-28)$$

EKF-SLAM 至此完成一輪計算，接著重複此流程以完成機器人之同時定位與建立環境地圖的功能。

3.4 參考點轉換

本研究之主要目的就是希望藉由全向式攝影機豐富的影像資訊及廣闊的視野，幫助全向式移動機器人達成在室內環境中的自我定位及導航的功能。前一節中說明了利用 EKF-SLAM 演算法達成全向式移動機器人自我定位的方法，由於使用的感測器為攝影機，對於偵測辨識環境中的 Landmark 需要兩張影像才可比對出相同的特徵點。本論文中所採用的演算法是以起始點影像為參考點，機器人移動後所擷取到的畫面均與起始點影像作比較，以此方法定位出機器人自身的位置並可建立出起始點週遭環境特徵點之地圖。

但是此方法並不能讓機器人走出較長遠的距離，因為機器人移動越遠，與起始點影像所能比對到的特徵點也就越少。因此為了讓機器人能走出較長的距離，且能比對到足夠的特徵點供自身定位，本論文採用文獻[19]所使用之參考點建立及轉換的流程：當機器人移動到新地點時則建立此處的參考點影像，使接下來機器人移動所擷取到的畫面能與參考點影像比對特徵點。每到一個新的區域，重新設定參考點影像，並將之前的參考點畫面存入資料庫，讓各區域都各有作為基準的參考點影像，機器人便可藉此行走更長的距離。判斷參考點建立的時機決定於移動的距離及特徵點的數目，距離為主要因素，當機器人移動距離超過原先起始的參考點畫面，所擷取到的特徵點必然無法與起始影像比對。因此若機器人

移動超過設定距離時，則需再次建立此時位置的參考點影像；另外當特徵點比對成功的數目越來越少時，即表示當前的參考點影像已經不敷使用，也是替換參考點的時機之一。圖 3-11 為參考點轉換的流程：

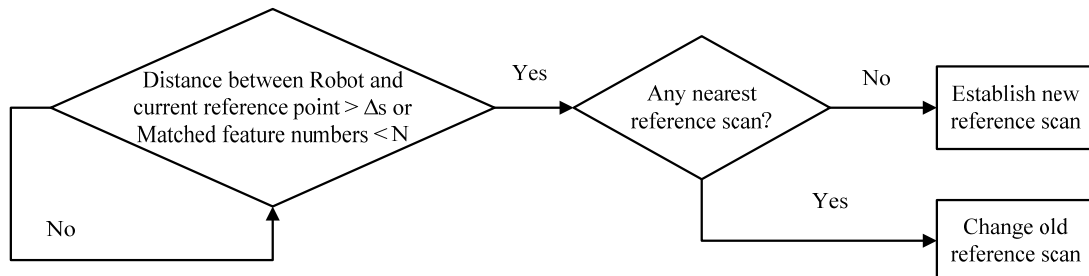


圖 3-11 參考點轉換流程圖

替換參考點的同時，要先將原本的參考點區域相關資訊存入資料庫，如參考點影像，特徵點資訊，環境地圖(特徵點位置)、各個特徵點之 Covariance... 等，並將新舊兩個參考點影像作比較，將共有的特徵點以及其 Covariance 等相關資料留下，作為新參考點區域的已知環境特徵點，讓新舊兩區域有了關連，環境資訊能銜接上。

在[13]中，所使用的攝影機為雙眼攝影機，因其視角有限，故當旋轉至一定角度時，會有無法與原參考點影像成功比對的情形而需建立新的參考點。若是再次走向相同的地點，相同的特徵點有可能因為視角的不同而被定義成不同的點。而本論文因為使用的是全向式攝影機，即使機器人旋轉了 180 度，仍能在影像平面中辨識出是同一區域，且走向原來地點時，攝影機仍可在影像中辨識出之前估測過的特徵點，此種情況下不需重新執行建立參考點步驟，而可直接取出先前的參考點資訊。如圖 3-12，各參考點影像都有其負責的區域範圍，機器人在移動到新地點時所建立各區域(Ref01、Ref02、Ref03、Ref04)參考點影像，其所擷取的即時影像和區域參考點比對來定位出機器人的位置。而回到了走過的地方，則取舊有的參考點影像資訊幫助機器人定位。

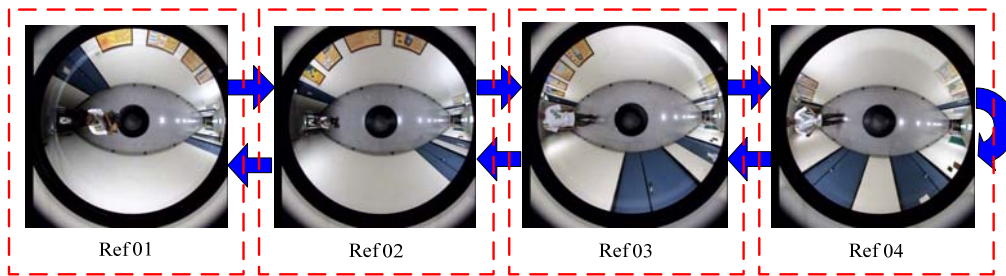


圖 3-12 參考點影像轉換

完成了參考點轉換流程後，機器人便可在移動長距離的情況下，逐一建立參考點影像，可將環境中不同地點的影像特徵整合，且可重複行走，達成機器人長距離導航的功能。

Step1：

即時影像與參考點比對，完成對機器人位置與特徵點的估測後，判斷機器人此刻與參考點距離是否超過最大半徑。若否，則繼續下一個影像 Frame 的運算；若是，代表機器人離開參考點區域，則開始參考點轉換的步驟。

Step2：

搜尋 Database 中區域半徑內是否有其他參考點，若否，則將原參考點影像存入並更新 Database，此刻的即時影像視為一新的參考點，並在 Database 中標記。若是，同樣將原參考點影像中資訊存入並更新 Database，並轉換為舊有的參考點影像。參考點轉換完畢繼續 Step1 的運算。

第四章 實驗結果

本章設計了幾個實驗用以驗證所發展演算法之性能，最後並以機器人之長距離導航情境展示本論文所提出方法之可行性。

4.1 節為環境特徵點偵測之實驗，測試應用於全向式攝影機之 MSIFT 演算法擷取環境特徵點效能，在不同位置，不同角度所拍攝到的畫面，以確定機器人在移動或是旋轉時，是否仍能擷取足夠的特徵點提供機器人定位使用。4.2 節為 EKF-SLAM 以及參考點轉換實驗，此實驗主要是驗證第三章 EKF-SLAM 演算法之正確性，過程中以手動遙控機器人繞行實驗室走廊，比較機器人以 Odometer 和 EKF-SLAM 估測出來的移動軌跡。4.3 節及 4.4 節以方形及 8 字形實驗確認機器人軌跡以及環境地圖是否已經被正確紀錄下來，並測試長距離情況下參考點是否能正確轉換。4.5 節展示機器人長距離導航之情境。



4.1 環境特徵點辨識實驗

本實驗於走廊環境中測試全向式影像中所擷取出的環境特徵點及其比對結果。實驗中移動機器人全向式影像在不同位置、不同角度下進行特徵點比對，以驗證演算法的正確性與穩定性。實驗是取 $480 \times 480 \text{ pixel}^2$ 大小的影像畫面作處理，圖 4-1 為兩張相距 1 公尺所拍攝到的全向式影像，圖 4-2 為機器人旋轉 45° 後的特徵點比對結果，圖中的方點代表所偵測到的特徵點位置，以線條連接所比對到的特徵點。可發現大部分特徵點都對應到另一張影像中相同的環境位置，驗證了在機器人移動或旋轉後特徵點比對之演算法仍有一定的正確率。

表 4-1 及 4-2 是針對機器人移動不同距離、旋轉不同角度後與原始影像的特徵點比對結果。其中 Matched point 為此影像有幾個比對成功的特徵點，Ground truth 為比對正確無誤的特徵點，Matching rate 為比對成功率。可以發現當兩張影像相距到 1 公尺後，比對正確率有顯著的下降，所能比對到的特徵點也大幅減少

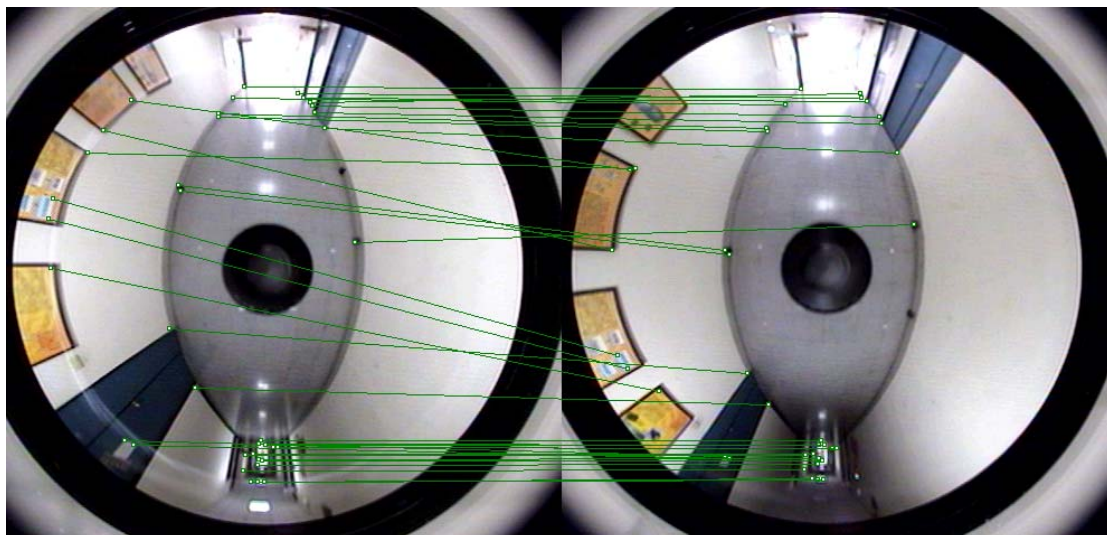


圖 4-1 移動 1m 的特徵點比對結果

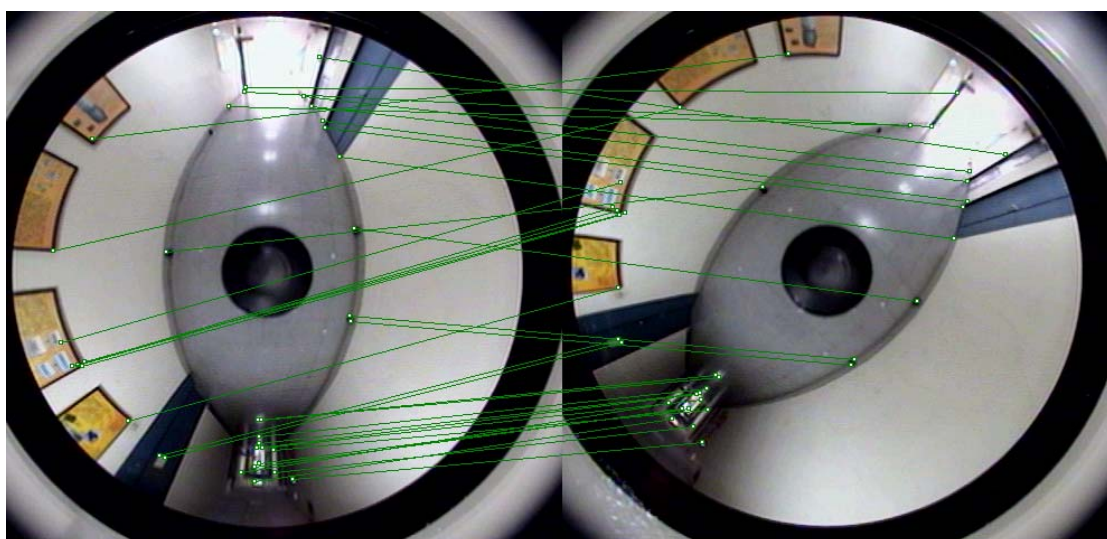


圖 4-2 旋轉影像 45° 後的特徵點比對結果

，但平均正確率仍有 90%；而影像旋轉後，平均正確率為 97%，旋轉角度間的比對正確率差異不大。經此實驗驗證，本特徵點比對演算法在機器人移動一小段距離或是旋轉的情況下，均能有效辨識出環境特徵點，且正確率達 93% 以上。這些特徵點即為機器人定位所需要的 Landmark，以此結果作為定位演算法的輸入，幫助達成機器人定位的功能。

表 4-1 特徵點比對實驗結果(平移)

Moved distance	0.5m	1m	1.5m	2m	Average matching rate
Matched point	44	37	18	15	
Ground truth	44	35	16	12	
Matching rate	100%	94%	88%	80%	90%

表 4-2 特徵點比對實驗結果(旋轉)

Rotated angle	45°	90°	135°	180°	225°	270°	315°	Average matching rate
Matched point	36	40	29	30	30	29	38	
Ground truth	34	40	29	30	30	27	37	
Matching rate	94%	100%	100%	100%	100%	93%	97%	97%
Total rate								93%

4.2 EKF-SLAM 定位實驗

為了驗證機器人於走廊環境中移動時的定位功能，本論文分別設計兩種實驗，使用以第三章為基礎之 EKF-SLAM 演算法，比較 EKF 與 Odometer 的量測誤差，其中 x, y 分別為水平位置座標， θ 為機器人朝向角。

4.2.1 Ground truth 實驗(直行)

此實驗過程中，手動遙控機器人於實驗室走廊沿設定之路徑運動，以(0,0)為起點，分別朝 y 方向走 2 公尺、4 公尺、6 公尺、8 公尺及 10 公尺，每種距離各作 3 次，目的為得知在機器人直行的情況下， x, y 及 θ 的誤差。圖 4-3~4-7 為

比較 EKF 與 Odometer 所紀錄的軌跡差異，其中 X 線為 Odometer 所紀錄的軌跡，--線為 EKF 估算出來的軌跡，o 點則為實際機器人到達的目標點位置。將每種距離各作 3 次的實驗數據列於表 4-3~4-7。可發現 EKF 較 Odometer 準確，移動 10 公尺後橫向 x 方向平均誤差為 14 公分、朝向角 θ 平均誤差為 2 度。

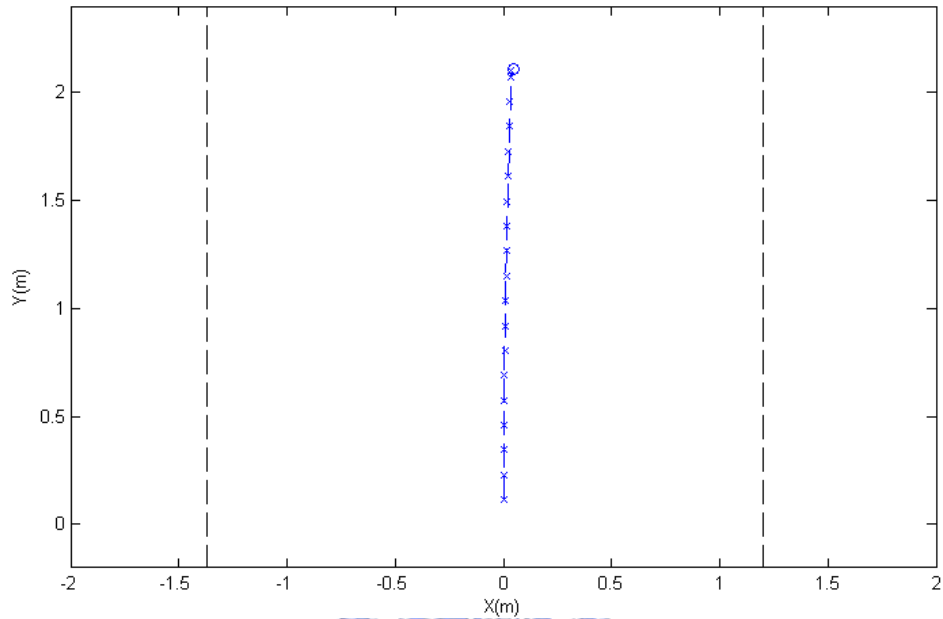


圖 4-3 Odometer 與 EKF 紀錄軌跡比較(2m)

表 4-3 Odometer 與 EKF 量測結果比較(2m)

	2m								
	Ground truth			Odometer estimation			EKF estimation		
	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)
1	0.01	2.00	88	0.04	2.00	87.33	0.02	2.00	87.88
2	0.05	2.10	88	0.04	2.07	87.72	0.04	2.07	87.74
3	0.04	2.01	88	0.04	2.00	87.71	0.04	2.00	87.87
Average error				0.01	0.01	0.41	0.007	0.01	0.17

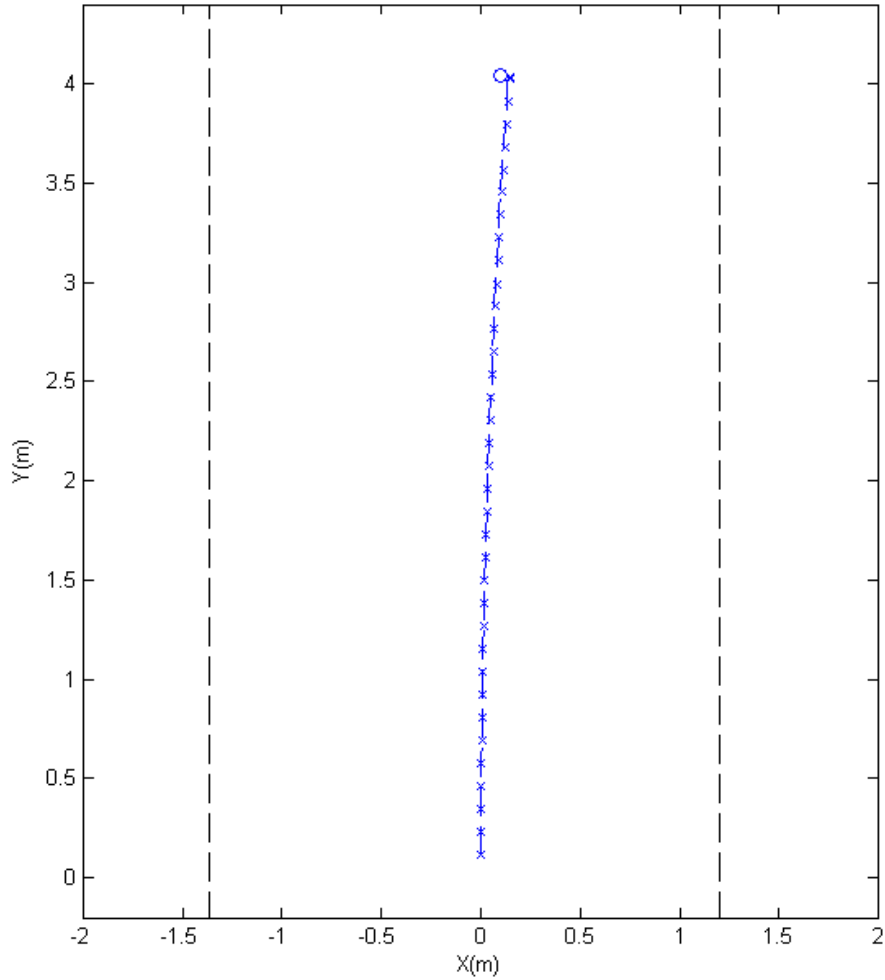


圖 4-4 Odometer 與 EKF 紀錄軌跡比較(4m)

表 4-4 Odometer 與 EKF 量測結果比較(4m)

	4m								
	Ground truth			Odometer estimation			EKF estimation		
	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)
1	0.10	4.02	87	0.19	4.00	84.34	0.14	4.01	85.15
2	0.11	4.10	86	0.16	4.07	87.45	0.13	4.07	85.88
3	0.14	4.08	87	0.17	4.06	85.11	0.12	4.06	85.94
Average error				0.06	0.02	2.00	0.03	0.02	0.96

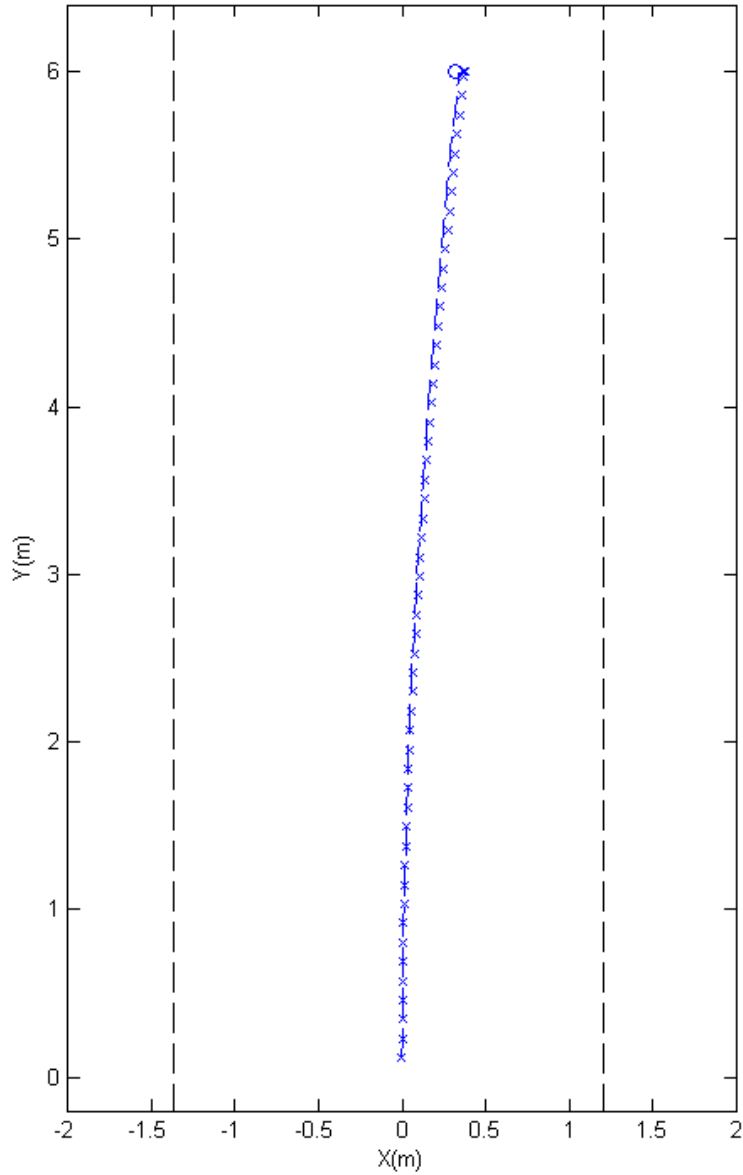


圖 4-5 Odometer 與 EKF 紀錄軌跡比較(6m)

表 4-5 Odometer 與 EKF 量測結果比較(6m)

	6m								
	Ground truth			Odometer estimation			EKF estimation		
	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)
1	0.27	6.17	85	0.45	6.14	81.77	0.40	6.14	82.31
2	0.32	6.04	84	0.36	6.01	83.04	0.31	6.02	83.64
3	0.32	6.08	84	0.38	6.04	82.89	0.25	6.05	84.68
Average error				0.09	0.03	1.77	0.07	0.03	1.24

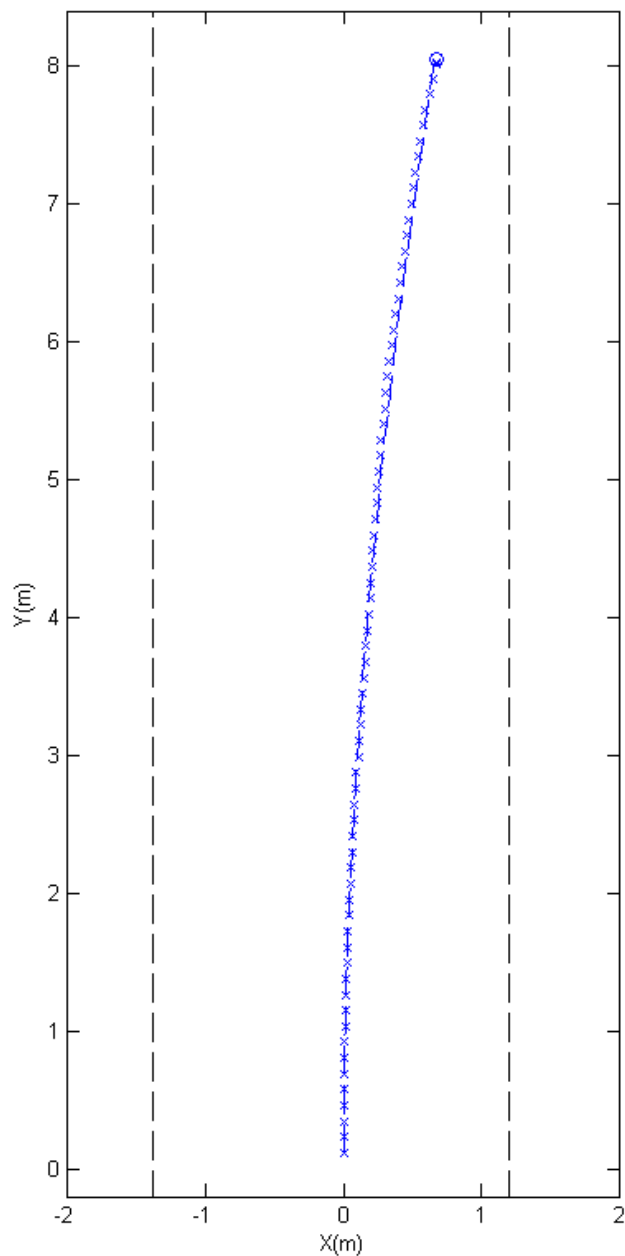


圖 4-6 Odometer 與 EKF 紀錄軌跡比較(8m)

表 4-6 Odometer 與 EKF 量測結果比較(8m)

	8m								
	Ground truth			Odometer estimation			EKF estimation		
	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)
1	0.48	8.02	82	0.75	7.98	79.64	0.68	7.99	80.18
2	0.60	8.18	81	0.67	8.13	80.08	0.60	8.14	80.70
3	0.57	8.08	84	0.66	8.01	80.87	0.47	8.04	82.67
Average error				0.14	0.05	2.14	0.10	0.04	1.15

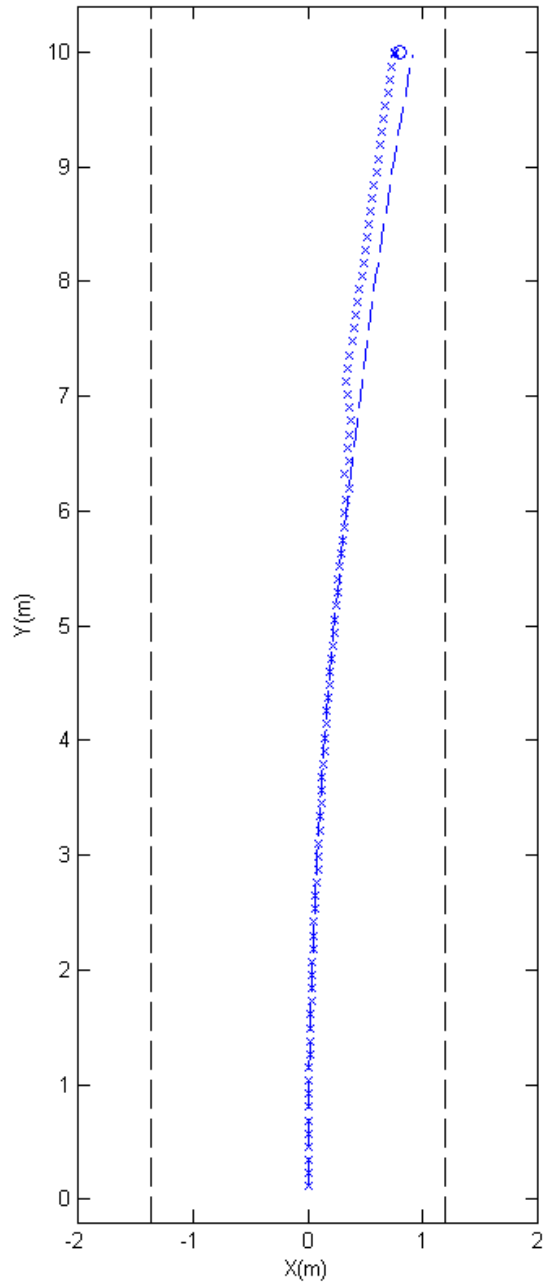


圖 4-7 Odometer 與 EKF 紀錄軌跡比較(10m)

表 4-7 Odometer 與 EKF 量測結果比較(10m)

	10m								
	Ground truth			Odometer estimation			EKF estimation		
	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)
1	0.8	10.00	80	1.17	9.94	76.56	1.08	9.96	77.07
2	0.96	10.10	79	1.06	10.06	77.09	0.97	10.07	77.60
3	0.91	10.06	78	1.03	10.01	77.90	0.78	10.04	79.69
Average error				0.20	0.05	1.82	0.14	0.03	2.01

4.2.2 Ground truth 實驗(直行加旋轉)

圖 4-8 為在實驗室外走廊，以手動遙控機器人往前走 6 公尺，旋轉 180 度，返回起點，比較 Odometer 以及 EKF 估算出來的結果。X 線為 Odometer 所紀錄的軌跡，--線為 EKF 估算出來的軌跡。此實驗之目的為評估機器人在旋轉後，角度的誤差大小。其中(a)為 x、y 方向 Odometer 與 EKF 的紀錄軌跡差異，兩側虛線代表牆面，o 點表示起點和轉折點。(b) 則為角度 θ 的變化紀錄，為機器人從 90 度轉向 270 度的過程。此實驗分別做 3 次，數據紀錄列於表 4-8。實驗結果可發現，Odometer 終點與起點平均誤差為 0.65m，EKF 終點與起點平均誤差為 0.07m，證明定位演算法是有效的，且有一定程度修正 Odometer 所造成的累積誤差。然而機器人在旋轉 180 度之後朝向角 θ 的誤差有 5 度左右，仍有改進的空間。

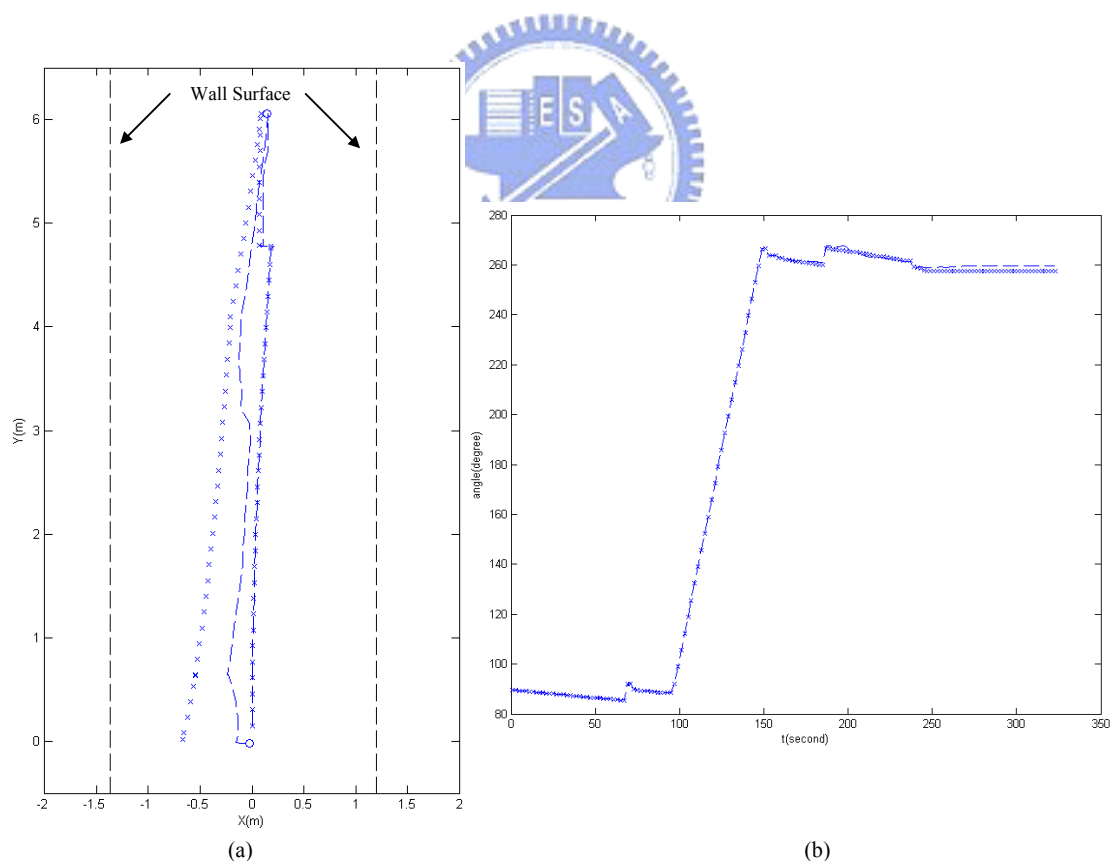


圖 4-8 6 公尺來回 EKF 及 Odometer 比較

表 4-8 來回行走實驗誤差

	Ground truth			Odometer estimation			EKF estimation		
	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)	x (m)	y (m)	θ ($^{\circ}$)
1	0.02	0.01	268	0.68	0.04	261.82	0.11	0.05	261.74
2	0.02	0.01	266	0.67	0.01	257.63	0.04	0.02	259.61
3	0.05	0	266	0.68	0.07	261.90	0.15	0.007	263.51
Average error				0.65	0.03	6.22	0.07	0.02	5.05

4.2.3 參考點轉換實驗

有鑑於機器人移動一段距離後，如 1 公尺或 1.5 公尺，兩張影像的特徵點比對數目將大幅減少，本論文以實驗驗證參考點影像在平移多少距離後的轉換能比對出足夠的特徵點供影像定位，並正確估測出機器人的位置。將參考點影像轉換的距離在分別設定為 1 公尺及 1.5 公尺的情形下，控制機器人前行 5 公尺，量測終點與 Ground truth 的距離誤差。表 4-9 及 4-10 為實驗結果，在 Reference 影像取 1 公尺時，機器人移動至終點的平均誤差在 x 方向為 0.08m，y 方向為 0.02m，朝向角 θ 為 3 度；在 Reference 影像取 1.5 公尺時，機器人移動至終點的平均誤差在 x 方向為 0.12m，y 方向為 0.01m，朝向角 θ 為 3.33 度。可發現在 Reference 影像範圍為 1 公尺時，能提供機器人較好的定位效果，因此本論文即是以 1 公尺作為機器人取參考點轉換影像的距離。

表 4-9 Reference 影像取 1 公尺實驗結果

Error	x (m)	y (m)	θ ($^{\circ}$)
Experiment 1	0.07	0	1
Experiment 2	0.15	0.02	5
Experiment 3	0.03	0.04	3
Average error	0.08	0.02	3

表 4-10 Reference 影像取 1.5 公尺實驗結果

Error	x (m)	y (m)	θ (°)
Experiment 1	0.10	0	2
Experiment 2	0	0.01	2
Experiment 3	0.25	0.02	6
Average error	0.12	0.01	3.33

4.3 沿方形移動實驗

為驗證機器人於室內的定位功能，實驗場景設定於實驗室 622，如圖 4-9，讓機器人持續繞行一 3m x 1.5m 的方形軌跡 3 圈，移動距離達 27 公尺，分別作 5 次，繞行結束後量測機器人起點與結束點的距離誤差。將 5 次的實驗數據列於表 4-11。圖 4-10 為 EKF 估測之機器人軌跡和建立的室內環境特徵點地圖，此實驗共建立了 7 個參考點，共 89 個特徵點被建立在地圖上。

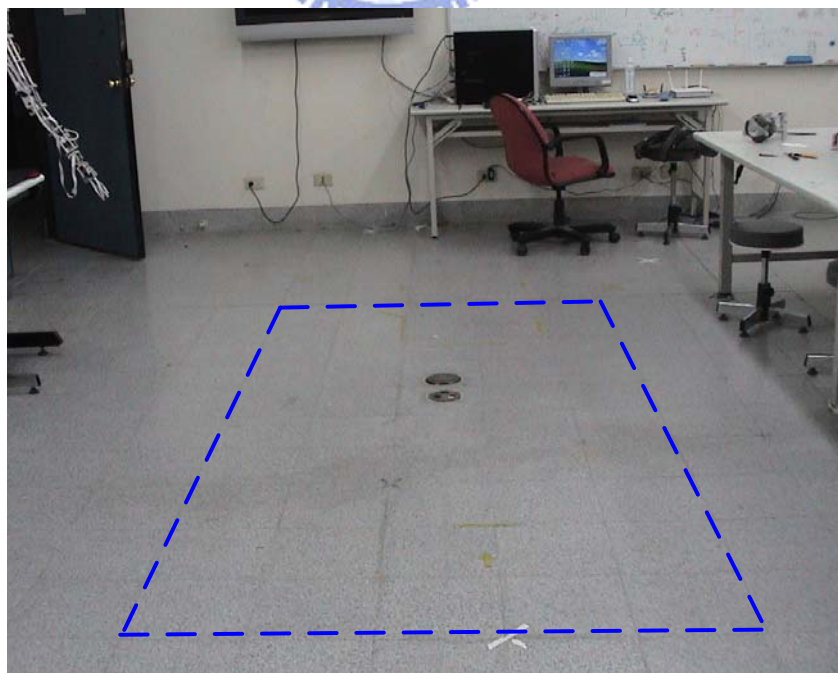


圖 4-9 室內沿方形移動實驗場景

表 4-11 室內方形軌跡實驗結果

Error	x (m)	y (m)	θ ($^{\circ}$)
Experiment 1	0.07	0.03	0
Experiment 2	0.09	0.08	3
Experiment 3	0	0.03	0
Experiment 4	0.06	0.07	1
Experiment 5	0.06	0.04	2
Average error	0.05	0.05	1.2

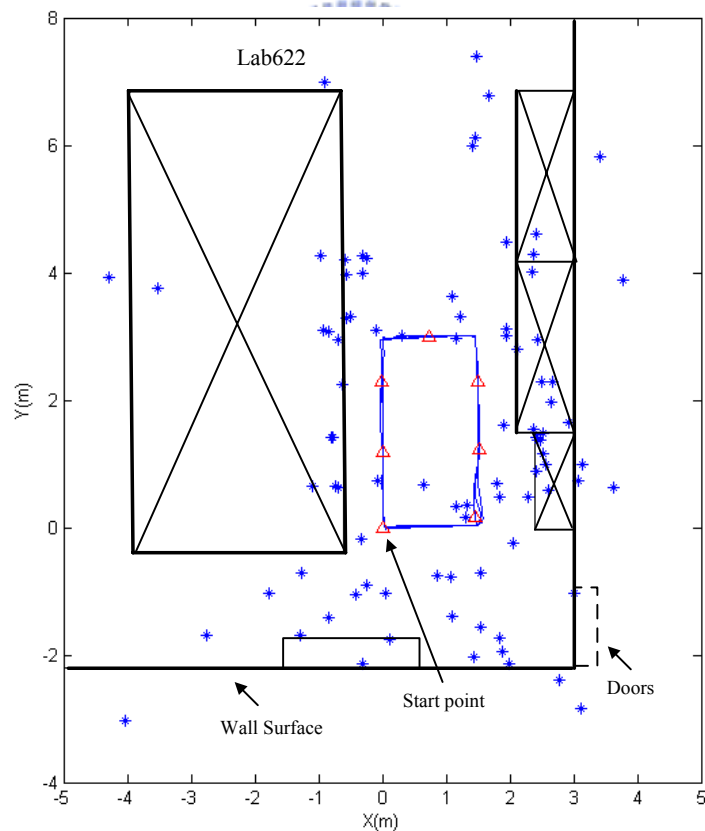


圖 4-10 室內方形移動實驗結果

實驗結果可發現，與第二章純以 Odometer 作定位控制走方形的結果相較，誤差於 x 方向及 y 方向都減少許多，均為 0.05m。由於是全向性移動平台，x、

y 方向誤差沒明顯的區別。可見此演算法確實能在一定程度上修正 Odometer 所帶來的打滑累積誤差，且達成室內複雜環境的定位功能。

4.4 8 字形移動實驗

圖 4-11 為機器人移動 8 字形軌跡的實驗結果，實驗環境位於一 8m x 10m 大小的環境空間，粗黑實線代表實際牆壁位置，右方虛線為門的位置，左上角則為樓梯。讓機器人以一 8 字形的封閉軌跡行走，用以證實機器人對環境地圖的建立，從起點出發繞行完路徑後，能回到起點附近且能銜接上。此實驗共建立了 9 個參考點，共 102 個特徵點被建立在地圖上。

機器人以(0,0)為起點，繞行完 8 字形後回到起點附近，量測與原點的距離誤差。作了 5 次實驗的結果列於表 4-12，其中 x 方向平均誤差為 0.09m，y 方向平均誤差為 0.09m，朝向角 θ 的平均誤差則為 1.6 度。

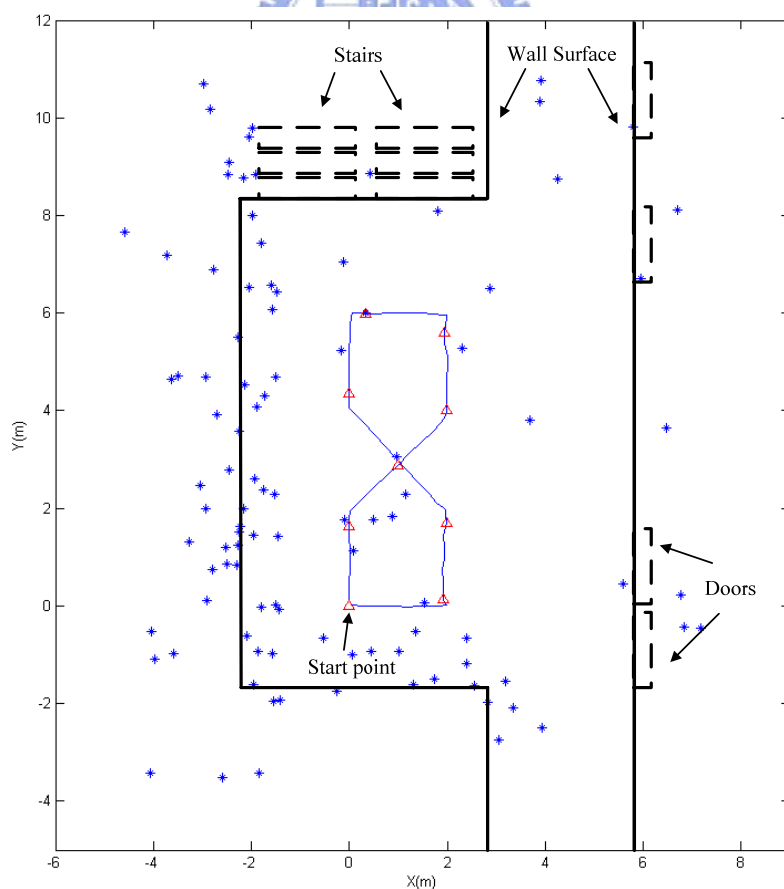


圖 4-11 8 字形移動軌跡結果

表 4-12 機器人 8 字形移動誤差

Error	x(m)	y(m)	$\theta(^{\circ})$
Experiment 1	0.08	0.07	1
Experiment 2	0.07	0.10	5
Experiment 3	0.00	0.12	2
Experiment 4	0.19	0.15	0
Experiment 5	0.15	0.00	0
Average error	0.09	0.09	1.6

4.5 室內導航之情境展示實驗

4.5.1 室內長距離移動實驗

本實驗之目的為驗證機器人能在室內進行長距離自主導航移動，且能同時定位出機器人本身的位置，建立環境地圖。並證實機器人不單能在走廊，也能在複雜環境下擷取出特徵點幫助定位，達成自主式導航的能力。此實驗以機器人從實驗室 622 為起點出發，繞行一長方形軌跡後走出 622 門口，移動約 11 公尺進入 621 門內，再出 621 沿路回到 622 門內停下。

圖 4-12 為 EKF 軌跡之紀錄，其中*代表特徵點位置，藍線則為定位系統所紀錄下機器人的軌跡，紅色三角形為機器人在移動過程中所建立的參考點位置，粗黑線則為實際環境走廊與房間的牆壁位置。

先設定好機器人的移動座標： $(0,0)$ 為 622 內起始點座標，朝 $(0,3)$ 座標前進，接著橫向移動至 $(-1,3)$ 座標，再往後至 $(-1,0)$ 座標；繞完一個矩形後，走出 622 門口到目標點 $(3,0)$ ，並接著往 621 方向的目標點 $(3,11)$ 前進，到了 621 門口後便橫

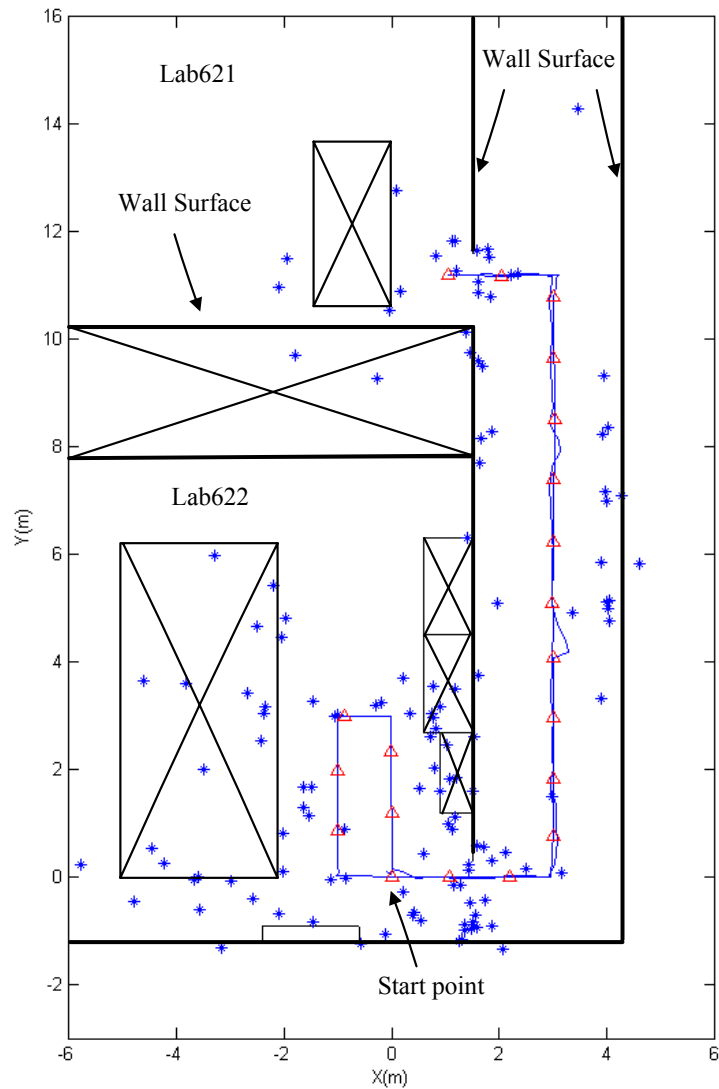


圖 4-12 機器人導航實驗結果

向移動至(0.8,11)的 621 門內，最後沿原路回到 622 內的(0,0)位置。全程除了預先設定好的目標點座標外，其餘均為機器人本身的自我定位，自主式的完成一段路徑迴圈。

本次實驗中，機器人從室內開始移動後，即一直重覆的運行 SLAM 演算法，即時的定位出機器人所在座標，讓機器人可得知如何移動到目標點，使其能成功走出房門並到達遠處的目標點位置，且環境特徵點地圖亦能同時建立出來。本次實驗總共建立了 136 個特徵點，20 個參考點。由圖 4-12 可發現，當機器人於室內移動時，特徵點多分布於周圍環境物體如桌子、門...等，走廊上的特徵點則多位於兩側的牆上；可證明於移動過程中，依照參考點建立及轉換的策略，可成功

延長機器人的行走及定位距離。圖 4-13 顯示實驗過程中所擷取的照片，其中(a)為機器人從起點出發，(b)(c)為機器人於實驗室 622 繞行矩形的情形，(d)為機器人正走出 622 的門，(e)為走向實驗室 621 的途中，(f)為走進 621，(g)(h)為回程途中，(i)為回到 622 門口，(j)為回到起點，完成整個路徑迴圈。表 4-13 為實驗 3 次後終點與起點的平均誤差，x 方向平均為 0.137m，y 方向平均為 0.059m，朝向角 θ 平均為 0.667 度。

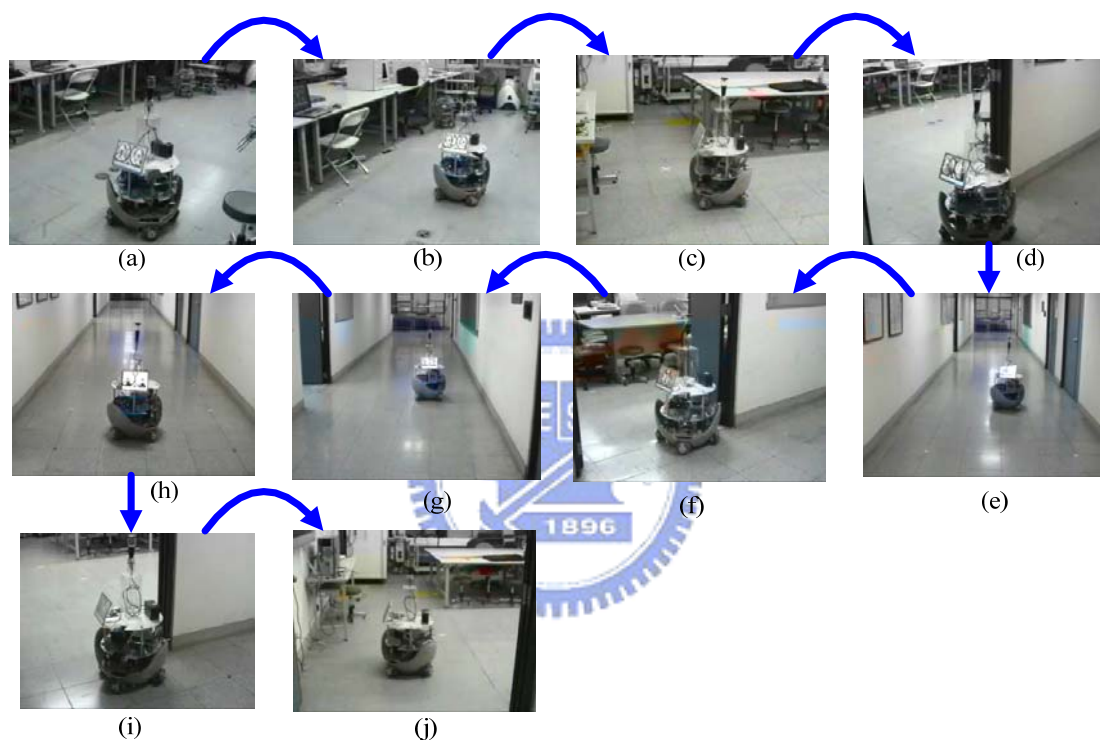


圖 4-13 機器人導航實驗過程

表 4-13 機器人導航實驗誤差

Error	x(m)	y(m)	$\theta(^{\circ})$
Experiment 1	0.13	0.02	1
Experiment 2	0.13	0.09	0
Experiment 3	0.15	0.05	1
Average error	0.137	0.053	0.667

圖上。

圖 4-15 顯示機器人於長距離轉向實驗過程中所擷取的照片，證明於移動過程中，使用 EKF-SLAM 定位演算法，可成功達成全向式機器人於行走的同時轉向並定位。其中(a)為機器人從起點(0,0)出發至(0,4)的位置，(b)(c)為機器人橫向移動至(-4,4)並轉向的情形，(d)為機器人往前走至(-4,10)的位置，(e)為橫向移動至長廊(0,10)並轉向的途中，(f)為前行至(0,14)，(g)(h)(i)為回程途中，(j)為回到起點，完成整個路徑迴圈。

此實驗共進行 3 次，表 4-14 為實驗結果，紀錄機器人出發點與完成整個路徑迴圈後回到原來位置的距離誤差。x 方向 1 平均誤差為 0.073m，y 方向平均誤差為 0.13m，朝向角 θ 平均誤差為 1 度。

4.6 討論

本論文主要是使用基於全向式影像及參考點轉換於全向式移動平台的定位方法：利用全向式影像特性的特徵點辨識方法，擷取環境中的特徵；並以設定一



圖 4-15 機器人轉向移動實驗過程

表 4-14 機器人長距離轉向移動實驗誤差

Error	x(m)	y(m)	$\theta(^{\circ})$
Experiment 1	0.09	0.21	2
Experiment 2	0.12	0.02	0
Experiment 3	0.01	0.16	1
Average error	0.073	0.13	1

參考點影像為基準，機器人移動時擷取到的即時畫面與參考點影像比較，藉此定位出機器人的自身位置及建立環境特徵點地圖。並藉由參考點轉換讓機器人定位距離延長。

然而於 Ground truth 直行及旋轉的實驗中，可發現 EKF 在移動距離越長的情況下，x 方向的估測誤差也隨之增大，不確定性增加，雖整體效能優於 Odometer，證實了影像處理以及定位演算法的可行性，但於影像的距離及角度估測仍應作改進。最後以情境模擬實驗證實本論文所提供的機器人定位方法可幫助全方位移動機器人達成室內長距離導航的功能。於第一個實驗中，由於機器人全程是以同一朝向角作移動，中間並無經過旋轉，因此最後機器人回到原點時，朝向角 θ 平均誤差能控制在 1 度以內。第二個實驗則在機器人直行時加入轉向的移動控制，發揮了全向式機器人靈活移動的特性，並完成了全向式移動機器人導航的基本情境。將設定好的目的地讓機器人依序前進，且同時建立環境地圖，使機器人即使在未知的環境，也能定位出自身的位置，達成了全向式移動機器人的 vSLAM 展示。

第五章 結論與未來工作

5.1 結論

本論文提出了一個全向式移動平台結合全向式攝影機的定位系統設計，以基於EKF-SLAM之演算法，和全向式攝影機對環境做觀測，修正全向式機器人的打滑誤差，並在全向式機器人移動的同時，能夠建立出環境地圖且定位出機器人的位置。

以所推導出之全向式機器人運動模型，及不同地點對特徵點的觀測視角，推算出特徵點相對於機器人本身的距離關係，以此資訊做為定位系統的輸入。使用SLAM演算法解決機器人之定位及地圖建立的問題，達成EKF-based SLAM演算法。結合影像處理與SLAM演算法，實現在全向式機器人平台上。

論文中以實驗驗證定位演算法：手動遙控機器人來回行走約12公尺後起點與終點誤差平均為0.07公尺，於室內繞方形3圈及走8字形軌跡起點與終點的x及y方向誤差均在0.10公尺以內。並以情境模擬實驗結果證實機器人藉由此演算法的幫助，為在實驗室622繞行一圈後走出房門，沿走廊移動約11公尺進621門，之後再沿原路回到622。另加入了機器人於長距離移動中同時轉向的實驗，發揮了全向式機器人靈活移動的特性，並同時建立其沿途路徑的特徵點環境地圖，完成全向式移動機器人的室內導航功能。

5.2 未來工作

在本論文中之定位演算法約需耗時2秒左右，此運算時間在較為慢速的測試實驗中雖然可行，但在實際的應用中若是到了更複雜的環境，改變就不能有及時的反應，且若在機器人的移動速度加快的情形下，影像運算也不能作及時的反

應，影響定位結果的正確性。因此在影像處理及定位演算法的設計上可以再加以改善，提升整體定位系統的效能。

全向式攝影機常被用於多機器人系統上，而本論文發展出的基於全向攝影機之機器人定位方法亦可與之整合，並藉由全向式移動平台的高機動性，讓機器人不僅能夠觀測機器人隊伍中的隊友，也讓機器人能對周遭整體的環境做估測，更迅速地移動至目標點，增加多機器人隊伍的實用與功能。



參考文獻

- [1] M. Wada and H.H. Asada, "Design and Control of a Variable Footpoint Mechanism for Holonomic Omni-directional Vehicles and its Application to Wheelchairs," in *IEEE Transactions on Robotics and Automation*, Vol.15 No.6, pp. 978-989, 1999.
- [2] S. Ziaie-Rad, F. Janabi-Sharifi, M.M. Danesh-Panah, A. Abdollahi, H. Ostadi and H. Samani., "A Practical Approach to Control and Self-Localization of Persia Omni-Directional Mobile Robot," in *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 3473-3479.
- [3] H.R. Moballeg, P. Amini, Y. Pakzad, M. Hashemi and M. Narmiani, "An Improvement of Self-localization for Omni-directional Mobile Robots Using a New Odometry Sensor and Omni-directional Vision," in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, Niagara Falls, 2004, pp. 2337-2340.
- [4] Y. Liu, R.L. Williams and J.J. Zhu, "Integrated Control and Navigation for Omni-directional Mobile Robot Based on Trajectory," in *Proceedings of American Control Conference*, New York City, USA, 2007, pp. 2153-2158.
- [5] K. Watanabe, "Control of an Omni-directional Mobile Robot," in *Proceedings of Second International Conference on Knowledge-Based Intelligent Electronic Systems*, Adelaide, Australia, 1998, pp. 51-60.
- [6] K. Watanabe, K. Izumi and F. Han, "Stochastic Fuzzy Servo Control Using Multiple Linear Dynamic Models," in *Proceedings of KES '98. 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems*, Adelaide, Australia, 1998, pp. 474-482.
- [7] C.C. Tsai, L.B. Jiang, T.Y. Wang and T.S. Wang, "Kinematics Control of an Omni-directional Mobile Robot, " in *Proceedings of 2005 CACS Automatic Control Conference*, Taiwan, 2005, pp. 18-19.
- [8] E.J. Jung, H.Y. Lee and J.H. Lee, "Navigation of an Omni-directional Mobile Robot with Active Caster Wheels," in *Proceedings of IEEE International Conference on Robotics and Automation*, Pasadena, USA, 2008, pp. 1659-1665.
- [9] J.S. Hu, J.H. Cheng and Y.J. Chang, "Spatial Trajectory Tracking Control of Omni-directional Wheeled Robot Using Optical Flow," in *Proceedings of IEEE 22nd International Symposium on Intelligent Control*, Singapore, 2007, pp. 1462-1467.
- [10]H. Durrant-Whyte and T. Bailey, "TUTORIAL : Simultaneous Localization and Mapping: Part I", *IEEE Robotics & Automation Magazine*, Vol. 13, No. 2,

- pp.99-108, 2006.
- [11]H. Durrant-Whyte and T. Bailey, "TUTORIAL : Simultaneous Localization and Mapping (SLAM) : Part II", *IEEE Robotics &Automation Magazine*, Vol. 13, No. 2, pp. 108-117, 2006.
- [12]J.M.M. Montiel, J. Civera and A. J. Davison, "Unified Inverse Depth Parametrization for Monocular SLAM," in *Proceedings of Robotics Science and Systems Conference*, Philadelphia, 2006.
- [13]F. Bertolli, P. Jensfelt and H. I. Chistensen, "SLAM Using Visual Scan-Matching with Distinguishable 3D Points," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, 2006, pp.4042-4047.
- [14]Y. Maeda and D. Idou, "Multiple Omni-directional Vision System and Its Self-Localization Experiment," in *Proceedings of IEEE International Fuzzy Systems Conference*, London, UK, 2007, pp. 1-6.
- [15]J.H. Kim and M.J. Chung, "SLAM with Omni-directional Stereo Vision Sensor," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, 2003, pp. 442-447.
- [16]M. Saedan, C. W. Lim and M.H. Aug, "Appearance-based SLAM with Map Loop Closing Using an Omni-directional Camera," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Switzerland, 2007, pp. 1-6.
- [17]G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, SIGGRAPH 2001 Course.
- [18]T. B. Lauwers, G. A. Kantor and R. L. Hollis, "A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive," in *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, 2006, pp. 2884-2889.
- [19]F.S. Huang and K.T. Song, "Vision SLAM Using Omni-Directional Visual Scan Matching," Accepted for Publication in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE/RSJ IROS 2008)*, Nice, France, 2008.
- [20]N. Winters, J. Gaspar, G. Lacey and J. Santos-Victor, "Omni-directional Vision for Robot Navigation," in *Proceedings of IEEE Workshop on Omni-directional Vision*, South Carolina, 2000, pp.21-28.
- [21]S. Panzieri, F. Pascucci, R. Setola and G. Ulivi, "A Low Cost Vision Based Localization System for Mobile Robots," in *Proceedings of 9th Mediterranean Conference on Control and Automation*, Dubrovnik, Croatia, 2001, pp. 1-6.
- [22]H. Andreasson, A. Treptow and T. Duckett, "Localization for Mobile Robots using Panoramic Vision, Local Features and Particle Filter," in *Proceedings of*

- IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005, pp. 3348-3353.
- [23]J. Shi and C. Tomasi, “Good Features to Track.” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 1994, pp. 593-600.
- [24]David G. Lowe, “Distinctive Image Feature from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91-110.
- [25]S. Thrun, “Learning Metric-Topological Maps for Indoor Mobile Robot Navigation,” in *Journal of Artificial Intelligence*, vol. 99, pp. 21-71, 1998.
- [26]S. Thrun, D. Fox and W. Burgard, *Probabilistic Robotics*, MIT Press, 2005.
- [27]S. Thompson and A. Zelinsky, “Accurate Local Positioning Using Visual Landmark from a Panoramic Sensor,” in *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, 2002, pp. 2656-2661.
- [28]J. Kim, K.J. Yoon, J.S. Kim and I. Kweon, “Visual SLAM by Single-Camera Catadioptric Stereo,” in *Proceedings of SICE-ICASE International Joint Conference*, Busan, Korea, 2006, pp. 2005-2009.

