

國立交通大學

電機與控制工程學系

碩士論文

彎道之影像偵測與自動駕駛之實現



Image Detection of the Curved Road and
Realization of the Autonomous Vehicle

研究生：許瑞原

指導教授：徐保羅 博士

中華民國九十六年七月

彎道之影像偵測與自動駕駛之實現

Image Detection of the Curved Road and
Realization of the Autonomous Vehicle

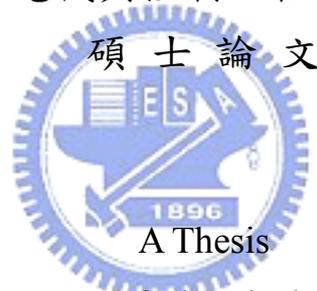
研究生：許瑞原

Student: Ruei-Yuan Shiu

指導教授：徐保羅 博士

Advisor: Prof. Pau-Lo Hsu

國立交通大學
電機與控制工程學系



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

National Chiao-Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

July 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

彎道之影像偵測與自動駕駛之實現

研究生：許瑞原

指導教授：徐保羅 博士

國立交通大學電機與控制工程學系

摘要

自動駕駛是 ITS 近幾年來主要的研究方向，現今也已發展出不同的方式來完成這項工作。本研究選擇以 CCD 為感測器來擷取外界環境，其中，對我們最主要的特徵就是車道線。在直線道路上，沿用 IPM based 之車道線偵測方法，找出車道線偵測視窗之定位，再由動態視窗進行對車道線追跡，此方式能兼具辨識率及即時性的優點，並估測出虛擬的車道中心線作為導航的依據。但在彎道上，存在紅色連續的右車道標記及不連續的左車道標記，造成原偵測方法因不連續及紅色標記而失敗，故在彎道上，為了突顯紅色標記，以色度座標轉換及條件式二值化取代原先的二值化步驟，使原偵測法則亦能有效偵測彎道上的右車道線，並做為導航的依據。

從車道線可進一步測出車輛在道路上的相關狀態：(1) 前視偏移距離，(2) 偏轉角度，及(3) 道路曲率。在直線道路上，使用 P 控制器對車輛進行控制，根據實驗結果顯示，當給予 $K_p=5\sim 11$ 時，能獲得穩定及準確的結果。但在彎道上，由於曲率的存在容易影響車子偏離車道，故除了 P 控制器外，本文提出多加入一以曲率為輸入之模糊推論器建立對彎道之偏移量補償器，以改善行駛彎道時偏離車道情形。綜合以上之對直線及彎道之控制補償設計，本研究整合影像與控制法則，完成能在校園環校道路上直線及彎道的自動駕駛。

Image Detection of the Curved Road and Realization of the Autonomous Vehicle

Student : Ruei-Yuan Shiu

Advisor : Dr. Pau-Lo Hsu

Department of Electrical and Control Engineering

National Chiao-Tung University

ABSTRACT

This thesis applies a CCD sensor to detect the lane marks for the present autonomous driving system. As the road is straight, the IPM (inverse perspective mapping)-based detection method is adopted to locate the suitable windows position properly. Then, by using the on-line dynamic detection windows approach, the road lanes can be thus efficiently tracked and the estimated virtual center-lane can be further obtained for the present autonomous driving. However, as on the curved road, some discontinuous lane marks cause the developed lane detection method invalid. Thus, the image of the red lane marks alone the road side can be processed by using the color-space transform with the conditional threshold to result in more accurate detection results in the case of curved roads.

The vehicle states can be further estimated from the processed image as (1) the look-ahead offset, (2) the yaw angle, and (3) the road curvature. In the case of a straight road, the P controller is suitable for the autonomous vehicle and experimental results indicate that K_p between 5 to 11 leads to satisfactory performance by concerning the system stability and control accuracy. On the curved road, a curved road offset compensator is processed by applying the fuzzy inference, with the input of curvature, to significantly improve vehicle driving performance for the present autonomous system.

致謝

回想起研究生涯，首先要感謝的是指導教授徐保羅博士對我的熱心指導以及照顧，讓我有充分的發展空間，使我能順利的完成碩士學位，在此表達我最深誠的敬意與感謝。另外感謝口試委員王伯群博士、洪哲文博士與李俊賢博士對於本論文的指正，並給予寶貴的建議，使得本論文更加完善。

感謝實驗室 816 的夥伴們：博士班謝鎮洲學長、幸琮政學長、王安平學長、張昭琳學長、賴建良學長、黃煒生學長和已畢業的李俊賢學長，平時給我的意見、指導與幫助，以及實驗室學長景文、學姊尚玲、同學元銘、興漢、宗翰、孝麟，學弟雨坤、林億、宗勝、也強、建龍、醇偉在學業上的相互切磋與指教，並在實驗階段給予適時的幫助，也讓實驗室裡增添了不少的歡樂與笑聲，還有新進實驗室來的學弟們，在最後口試階段的幫忙。

最後，將此論文獻給我最敬愛的父母與家人，因為有你們在身旁的支持與關懷，使我能夠心無旁騖地完成學業，並且順利地克服我人生中的每一個重大考驗。

感謝所有曾經幫助過我與默默祝福我的朋友，謝謝你們。

目錄

中文摘要	i
英文摘要	ii
致謝	iii
目錄	iv
表目錄	vii
圖目錄	viii
第 1 章 緒論	1
1-1 研究動機與目的	1
1-2 研究背景與發展概況	2
1-3 問題界定	4
1-4 研究方法與步驟	6
1-5 論文架構	7
第 2 章 影像之車道線偵測	8
2-1 IPM(inverse perspective mapping) based之車道線偵測	8
2-1-1 Inverse Perspective Mapping	8
2-1-2 車道線擷取	10
2-1-3 定義車道線及偵測結果	13
2-2 動態視窗之車道線偵測方法	17
2-3 整合型車道線偵測方法	21
2-4 夜間車道線偵測方法	23
2-5 彎道車道線偵測方法	25

第 3 章	導航資訊與前輪轉動角度	29
3-1	車輛轉向運動分析	29
3-2	導航資訊擷取	33
3-2-1	車道標記點(Lane Marking)的選取	33
3-2-2	車道中心線(Center Lane)	34
3-2-3	座標系統的轉換	36
3-2-4	實際車道模型的建立	37
3-2-5	建立參考點資訊	38
3-3	導航資訊測試	39
3-3-1	近區間影像估測	39
3-3-2	近區間結合遠區間影像之估測	43
3-3-3	討論	44
3-4	前輪轉動角度	47
3-4-1	控制法則	47
3-4-2	前輪轉動角度誤差	49
3-4-3	相關角度轉換	49
第 4 章	視覺導航車實驗結果	50
4-1	視覺導航車實驗介紹	50
4-1-1	系統設備與使用者介面	50
4-1-2	實驗路段	54
4-1-3	油門值與實際車速關係	54
4-1-4	實驗架構圖	55
4-1-5	USB-CAN	56
4-1-6	DSP端的工作情形	61
4-2	直線道路實驗結果與分析	63
4-2-1	P控制器偏移距離實驗結果	63
4-3	彎曲道路實驗結果與分析	65

4-3-1	P控制器偏移距離實驗結果.....	65
4-3-2	(P控制器+彎道偏移量)實驗結果.....	68
4-3-3	實驗分析.....	74
第 5 章	結論與未來發展.....	77
5-1	結論.....	77
5-2	未來發展.....	78
參考文獻	80



表目錄

表 2-1 色度座標轉換RGB→YCrCb	26
表 3-1 參數符號及解釋	29
表 4-1 油門值與實際速度關係表	55



圖目錄

圖 1.1 自動導航車研究部份.....	2
圖 2.1 兩個座標系(W and I)的關係	9
圖 2.2 xy plan in the W	9
圖 2.3 $z\eta$ plane in the W	9
圖 2.4 原始圖及IPM後的結果.....	10
圖 2.5 利用灰階特徵所擷取出的車道線.....	11
圖 2.6 利用GMD加強車道線影像	11
圖 2.7 GMD處理過程	12
圖 2.8 二值化後的影像.....	13
圖 2.9 細化處理後的影像.....	13
圖 2.10 以特定點 S 為軸之示意圖	14
圖 2.11 所找出的sample lane.....	15
圖 2.12 所找出的車道線.....	15
圖 2.13 將車道線對應回原始圖上.....	16
圖 2.14 所找出的偵測視窗	16
圖 2.15 IPM based 之車道線偵測方法其程式流程	16
圖 2.16 IPM based之車道線偵測方法其車道線偵測結果	17
圖 2.17 在影像上取兩個小視窗.....	17
圖 2.18 利用灰階特徵所擷取出的車道線	17
圖 2.19 二值化後的影像.....	18
圖 2.20 形態擴張後的影像.....	18
圖 2.21 細化處理後的影像.....	19
圖 2.22 定義車道線資料取樣的方法.....	19

圖 2.23 動態視窗之車道線偵測方法所定義出的車道線影像	20
圖 2.24 動態視窗之車道線偵測方法其程式流程	20
圖 2.25 整合型車道線偵測方法其程式流程	22
圖 2.26 整合型車道線偵測方法其偵測結果	22
圖 2.27 不同的閾值所得之車道線偵測結果	23
圖 2.28 夜間及黃昏路況之車道線偵測結果	24
圖 2.29 左車道線不連續發生偵測錯誤.....	25
圖 2.30 右車道線為具紅色成份之車道線	25
圖 2.31 彩色影像經灰階、IPM、二值化處理後的偵測結果	26
圖 2.32 彩色影像經處理後的偵測結果(其中閾值為 $Y=140$ ， $Cr=135$).....	27
圖 2.33 彎道偵測方法程式流程修正.....	28
圖 3.1 bicycle model	30
圖 3.2 視覺系統在參考點處所估測出的相關資訊	32
圖 3.3 駕駛資訊選取流程.....	33
圖 3.4 車道標記點的選取.....	34
圖 3.5 直線道路上不同位置所找出的車道中心線	35
圖 3.6 彎道上利用右車道線找出的車道中心線	36
圖 3.7 近區間影像估測方式.....	40
圖 3.8 前視偏移距離估測結果.....	41
圖 3.9 偏轉角度估測結果.....	42
圖 3.10 道路曲率估測結果.....	42
圖 3.11 直線道路估測道路曲率方式.....	43
圖 3.12 直線道路曲率估測結果.....	43
圖 3.13 彎道估測道路曲率方式.....	44
圖 3.14 彎道曲率估測結果.....	44

圖 3.15 車輛行駛直線道路在不同前視距離處所量測到的前視偏移距離	45
圖 3.16 比較前視偏移距離有無濾波結果	46
圖 3.17 曲率濾波後估測結果.....	47
圖 3.18 控制法則與考慮導航資訊情形.....	48
圖 4.1 實驗用電動高爾夫球車.....	50
圖 4.2 實驗目標.....	50
圖 4.3 USB埠CCD.....	51
圖 4.4 UPS	51
圖 4.5 硬體架構圖.....	51
圖 4.6 系統設備配置圖.....	52
圖 4.7 影像程式介面.....	53
圖 4.8 實驗路段.....	54
圖 4.9 馬達透過皮帶連結方向盤.....	55
圖 4.10 實驗架構圖.....	56
圖 4.11 USBCANII外觀.....	57
圖 4.12 網路監控程式主畫面.....	58
圖 4.13 USB CAN系統參數設定畫面	59
圖 4.14 傳輸畫面擷取及說明.....	61
圖 4.15 馬達位置的命令與實際量測.....	62
圖 4.16 車輛於油門值=390 行駛直線道路實驗結果	64
圖 4.17 顯示車輛於 $K_p=7$ 行駛直線道路實際駕駛情形	64
圖 4.18 車輛於油門值=650 時行駛直線道路實驗結果	65
圖 4.19 車輛於 $K_p=7$ 行駛彎道時實驗結果.....	66
圖 4.20 車輛於 $K_p=9$ 行駛彎道時實驗結果.....	67
圖 4.21 車輛於 $K_p=7$ 行駛彎道時實驗結果.....	68

圖 4.22 車輛於 $Kp=9$ 行駛彎道時實驗結果.....	68
圖 4.23 車輛使用(P+固定補償偏移量)行駛彎道實驗結果.....	69
圖 4.24 模糊推論系統流程.....	70
圖 4.25 曲率與偏移補償量歸屬函數.....	71
圖 4.26 模糊規則庫.....	72
圖 4.27 道路曲率與馬達偏移補償量關係.....	72
圖 4.28 車輛使用(P+模糊偏移補償量)行駛彎道實驗結果.....	73
圖 4.29 彎道實驗PC部份架構圖修正.....	74
圖 4.30 彎道車道線受到陽光影響偵測結果.....	75
圖 4.31 車子利用右車道線於油門值=390 時行駛直線道路結果.....	76



第一章 緒論

1-1 研究動機與目的

智慧型運輸系統 ITS (intelligent transportation system)係藉由先進之電腦、資訊、電子、通訊與感測等科技的應用，透過所提供即時資訊的溝通與連結，以改善人、車、路等運輸次系統間的互動關係，進而增進運輸系統之安全、效率與舒適，同時減少交通環境衝擊之有效整合型運輸系統。以美國 ITS America 之分類方式，ITS 大體包含七個子系統，其中一項便是先進車輛控制及安全系統 AVCSS (advanced vehicle control and safety systems)的發展。自 90 年代以後，汽車已成為路上的主要交通工具，也由於車子的普及率越來越高，使得交通日漸壅塞，隨之而來的就是交通意外事件的發生，汽車的安全性受到了嚴苛的考驗，因此，各先進國家：歐洲、美國、日本，都投注了相當大的心力在 AVCSS 的發展。AVCSS [1] 之主要特色係利用感測器協助人類感官功能之不足，減少危險之發生；提高自動控制之程度，從事更安全、準確、可靠之控制，彌補駕駛人因判斷錯誤及技術不佳所造成的疏失與危險，其相關的技術包含：防撞警示系統、自動停放車輛、車間與車-路間通訊、自動車輛診斷、自動橫向/縱向控制。而近年來 ITS 主要在發展自動導航車 AGV (automated guided vehicle)方面。

在自動導航車方面，許多的感測器被用來擷取外界的環境，諸如：光學式雷射掃瞄器(laser scanner)，CCD (charge coupled device)攝影機，超音波(ultrasonic)，全球定位系統 GPS (global position system)。但這些感測器在 AGV 中均有較佳的使用地方，如：雷射掃瞄器一般用來做障礙物偵測，藉由障礙物位置及方向資訊實現出車輛防撞功能；CCD 一般用來做道路線及障礙物的偵測，藉由相關資訊以實現車輛防偏及防撞功能、超音波所掃射的距離無法像雷射掃瞄器一樣遠，但仍可將其做防撞功能用，如側向防撞；GPS 一般會搭配電子地圖(e-map)實現出車輛定位，但誤差方面稍來的大，約 10-20 公尺，所以只能實現出大致上的定位。另外，

也有較特別的如[11]，利用磁鐵的裝置來當作感測器，但此缺點為須在路面埋設磁鐵裝置，成本也較高，對於一般道路並不適用。而近年來，由於計算機的處理速度不斷提昇及成本的考量下，使得影像已被應用在許多不同的領域中，尤其在自動導航的領域中，因為影像可提供較豐富的資訊。不少的研究就是利用影像的方式來得知外界的環境，進一步達到自動導航車。

綜合以上，可知影像已廣泛的被使用在實現自動導航車方面，畢竟得到的結果是比較接近人類視覺感官。而本研究著重在影像部份，在不增加計算資源及成本考量下，利用單顆 CCD 攝影機為系統擷取道路環境的感測器，將道路環境中的特徵擷取出來，進一步達成自動導航車，其目標為使車輛能安全且平穩的沿著道路中央行駛，即使面對不同的道路，存在不確定性因素等等，達到如同人類駕駛一般，最後將其實現於高爾夫球車上。

1-2 研究背景與發展概況

自動導航車已被研究了多年，至今也已經發展出不同的方法實現，但大致上要實現出自動導航車可分成三個研究部份：(1) 環境擷取方式 (2) 道路特徵擷取 (3) 導航控制器，其中，環境擷取方式如前面所述，本研究選擇用影像的方式擷取環境。

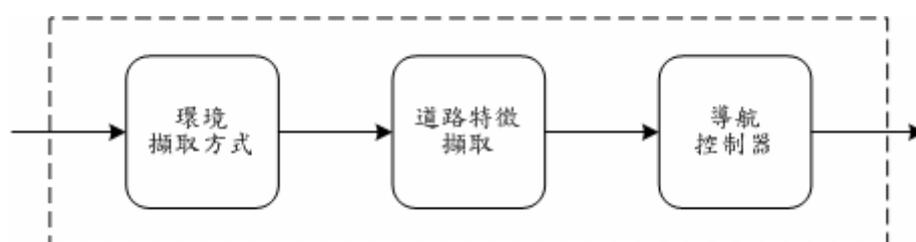


圖 1.1 自動導航車研究部份

就道路特徵方面，本研究選擇車道線為主要特徵。車道線偵測方面，Bertozzi 與 Broggi [2] 提出一個利用 IPM (inverse perspective mapping) 法則，將影像轉成俯視圖並移除不必要的影像，能有效的偵測在不同環境情況下的車道線。Kluge 與 Lakshmanan [9] 提出一個偵測車道線的方法，對付因陰影、坑洞、油污等情況

導致車道線不易辨識的情況，其利用 LOIS (likelihood of image shape) 偵測演算法，使用一個可變形的模版來解決上述的種種問題。Wang et al. [3] 提出 B-snake based 車道偵測演算法，其最大特性為適用在不同的道路結構，如直線或曲線，乃因車道線是由控制點的任意組合而成，並使用 CHEVP (canny/hough estimate of vanishing points)及 MMSE (minimum mean square energy)來決定初始及追蹤控制點的位置，實驗結果顯示對付雜訊，陰影，亮度變化，連續或不連續的道路線，或無標記的道路，均有不錯的效果。Jung 與 Kelber [4]結合了 EDF (edge distribution function)及 Hough transform 運算的方式找出車道邊界，並利用 linear-parabolic model 來追蹤之後的車道線。

對自動導航車來說，除了能有一套有效的擷取道路特徵外，更需要知道如何從特徵中擷取出所需的資訊，以提供導航控制器使用。在導航資訊方面，RALPH (rapidly adapting lateral position handler)系統在影像中定出一梯形區域[5]，將此區域轉成俯視圖，在所假設的五種道路曲率中：右彎，微右彎，垂直，微左彎，左彎，判定出最佳的影像以求得到路變化的趨勢，另外，再將俯視圖進行亮度上的分析統計，從最大，最小位置找出車道線的位置，與車子在車道中心時的車道線位置比較，判斷出車輛是否有偏移。Redmill [6]利用視覺系統選取出在左右兩旁的車道標記點，進而估測出影像中的車道中心線，透過幾何轉換，最後再找出通過轉換後的車道標記點的曲線方程式，從此曲線可得知車子在任何的前視距離與車道中心的側向偏移及道路的曲率。Gangyi et al. [10] 將車道線影像分為 LLM (lower lane markings)和 ULM (upper lane markings) 兩部份，利用 LLM 求得車輛的偏轉角度及車輛重心處與車道線的側向偏移量，而道路曲率主要從 LLM 所建立的消失點(左、右邊車道線在影像平面的交點)與 ULM 建立的消失點，此兩消失點的位置變化求得，而作者更進一步指出準確的側向偏移量與道路的坡度有關。

導航控制器方面，目的主要將導航資訊轉換成前輪轉動角度，並將車子控制在車道中心上。Wu et al. [7] 提出以模糊集合理論為基礎的模糊增益規劃，來補償迴授控制使能夠適度的調整駕駛命令，以改善因車子重量、轉角係數等因素對側向控制所造成的影響，其系統已在實驗道路 ARTC (automotive research and

testing center) 及高速公路上測試，在實驗道路 ARTC 部份，車輛在直線上可加速至 145km/hr，在較不平的路面上可達 120km/hr；而在高速公路上，速度約可達 80km/hr，均可穩定及準確的行使在車道中心，成功實現於 TAIWAN ITS-1 上。Unyelioglu et al. [12] 設計了一個固定型式的駕駛控制器，其中只量測前視距離處的側向距離的資訊，而此控制器是車子的縱向速度、重心的位置、重量，輪胎的剛性係數的函數。Kosecka et al. [8] 分別比較了 lead-lag control、full state feedback、input-output linearization，從實驗結果顯示出在實驗道路 (含直線及彎道，曲率半徑約 1200m) 上，使用 lead-lag control 有最好的表現，另外，設計了一觀察器來估測道路的曲率，再搭配前饋控制(feedforward control)，使車子行駛在彎道時有較小的誤差及改善暫態行為，實驗車速約可達 75mph，成功實現於 Honda Accord LX sedan 上。Hessburg 與 Tomizuka [11] 設計了一個模糊控制器，包含 feedback、preview 及 gain scheduling 模組，來做為車輛側向導航的控制器。Feedback 模組主要對感測器量測到狀態的誤差對駕駛角度做調整；preview 模組主要對曲率因素；gain scheduling 模組主要對速度因素，實驗結果顯示所使用的模糊控制器在實驗道路上能得到準確及穩定的結果，實驗用車為 Toyota Celica，車速約 50km/hr。

1-3 問題界定

1、影像系統方面

(1)實際車道線偵測方面

本研究主要是以道路上的車道線為基礎來實現出視覺導航車，故除了車道線外，其他所有的影像皆可視為雜訊。在彎道的車道線方面，有別於直線之道路線，而是紅色標記的車道線，所以我們必須找出一些影像加強的過程來突顯目標物，以及一些基本的二值化過程來有效的濾除雜訊，以確實的偵測出車道線，避免影響著之後選取導航資訊及控制部分。

(2) 導航資訊方面

從車道線建立出實際的車道模型，並從中得到導航資訊，但是得到的導航資訊是否符合實際的駕駛情形以及是否含有雜訊成份，都是必須考量在內。

2、控制系統方面

(1) 穩定性與準確度

控制器存在的目的主要將影像資訊轉換成前輪轉動角度，利用此資訊來操控車子的行進模式，故如何選擇一適當的控制器使本系統離目標位置較小的誤差，且穩定的行駛在車道上相當重要的。

(2) 不確定性(uncertainty)因素

對本研究來說，車子在航行的過程中，會遇到許多不確定性的因素：路面不平、濕滑、風力、道路黏滯係數、車子載重等等，這些因素都有可能讓車子產生不好的表現。在面對這些的不確定性因素，觀察是否對車子航行有太大的影響及如何修改控制器來克服這些不確定性因素。

3、即時性問題

為了實現自動導航車，即時性的問題是須考量的。在馬達對車子進行控制前，所花費的時間包含了 PC 部份及網路傳輸部分。PC 部份處理時間除了抓取影像後到分析出導航資訊外，也包含了控制器運算的時間，該考慮此部份是否花費太多的時間；而網路傳輸部份該考慮傳送訊息至 DSP 端時是否有延遲時間存在。這兩者的搭配是否會造成控制車輛有不即時現象。

1-4 研究方法與步驟

本論文研究的方法與步驟如下所述:

1、影像系統方面

- (1) 針對直線車道線方面，沿用 IPM based 之車道線偵測方法，找出車道線偵測視窗之定位，再由動態視窗進行對車道線追跡，此方式能兼具辨識率及即時性的優點。由於希望車子最終能準確及穩定的行駛在車道中心上，所以在兩旁車道線可利用的情況下，利用車道線的偵測結果估測出虛擬的車道中心線做為導航的依據。
- (2) 在彎道上的車道線方面，存在紅色連續的右車道標記及不連續的左車道標記，造成原偵測方法因不連續及無法找出紅色標記而失敗，故在彎道上，為了突顯紅色標記，修改原處理方法，使原偵測法則亦能有效偵測彎道上的右車道線。然而，由於左車道為不連續，並無法順利向直線車道估測出車道中心線，所以為了避免麻煩增加太多的計算量，直接先利用右車道線做為導航的依據，在缺乏對稱性的情形下，觀察利用右車道線是否會有太大的誤差。
- (3) 最後，利用車道線建立出 3D 車道模型，紀錄從車道模型所得到的導航資訊，並觀察車子的變動情形是否有與影片內容吻合並修正之。

2、網路傳輸方面

本研究選擇 USB-CAN 做為傳輸的工具，透過 USB-CAN 的使用介面，觀察 PC 端是否能即時且準確的傳送要轉動馬達的量及大小。

3、控制系統方面

完成影像部分後，要實際驗證其方法的可行性，將導航資訊搭配控制器來得到前輪轉動角度，先以最基本的 P 控制器分別對直線及彎道上進行控制觀察控制車輛結果的好壞，並做適度的修改，做細部調整，或演算方式的修改，達到能在實際狀況下，正確的操控方向盤，使車子能準確且穩定的行駛在車道中心上。

1-5 論文架構

本論文共分為五章，首先第一章中旨在說明研究的動機與目的、研究的背景與發展、界定出所要研究的問題、研究方法與步驟以及本論文貢獻。第二章為說明車道線的偵測方法及目前在白天及夜間的車道線部份已有的研究成果，將所使用的方法做一詳細的介紹。第三章說明利用車道線建立車道模型，並擷取出導航資訊，驗證資訊是否正確及說明使用的控制法則。第四章為視覺導航車系統的實車實驗，分別在直線及彎道上對實車進行測試並分析實驗結果。最後第五章中會對本論文做一個結論，並提出未來可再改進發展之方向。

第二章 影像之車道線偵測

本章節主要介紹如何從影像中在不同的環境中擷取出外界環境中的特徵-車道線。不只是白天，彎道，透過調整閾值的方式使其在夜間也能有不錯的偵測效果。

2-1 IPM(inverse perspective mapping) based 之車道線偵測[14]

本研究在車道線擷取方面，主要是以經過 IPM 後的影像為基礎，經過適當影像處理及定義，所發展出的車道線偵測處理，以下將逐一介紹。

2-1-1 Inverse Perspective Mapping

由於透視效果(perspective effect)關係，一般從攝影機所獲取的道路影像，其車道線寬度會隨著與攝影機的距離而有所改變，而每個像素所代表的實際大小也不同，因此造成原本平行的車道線，在圖像上會形成非平行而相交在遠處兩直線，造成不容易從圖片中找出車道線，使用 IPM 方法，將透視效果的影響從圖片中移除，形成等比例的俯視圖(top view)，讓之後車道線偵測更容易，先定義所需物理參數如下：

$W = \{(x, y, z)\} \in E^3$ ，表示 3-D world space。

$I = \{(u, v)\} \in E^2$ ，表示 2-D image space(如圖 2.1 所示)。

(1) Viewpoint：攝影機的位置 $C = (l, d, h) \in W$ 。

(2) Viewing direction：optical axis \hat{o} 由下述兩個角度來決定

$\bar{\gamma}$ ：為 optical axis \hat{o} 投影到 $z=0$ 平面(定義為向量 $\hat{\eta}$)，與 x 軸之夾角(如圖 2.2 所示)。

$\bar{\theta}$: 為optical axis \hat{o} 與向量 $\hat{\eta}$ 之夾角(如圖 2.3 所示)。

(3) Aperture : camera angular aperture is 2α 。

(4) Resolution : camera resolution is $n \times n$ 。

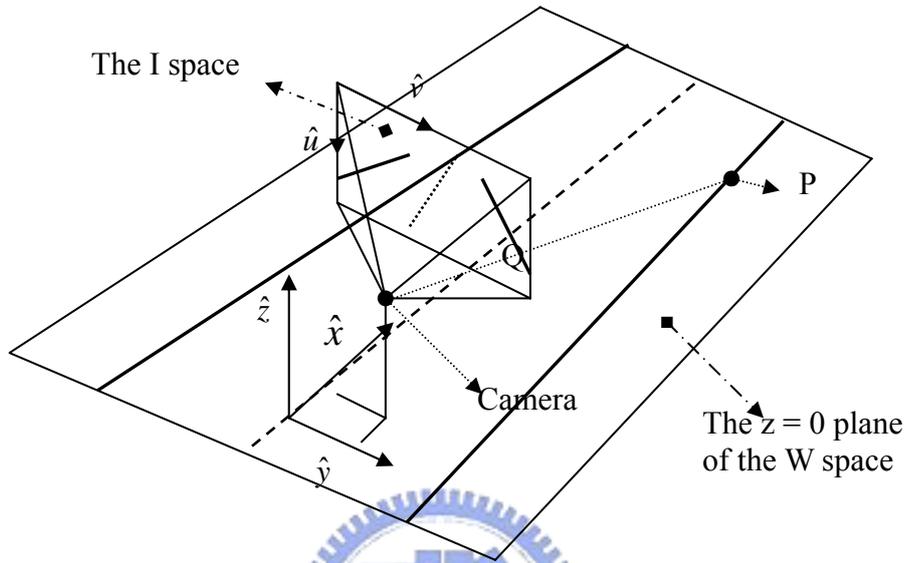


圖 2.1 兩個座標系(W and I)的關係

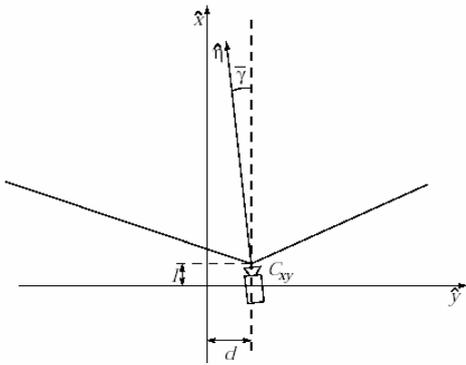


圖 2.2 xy plan in the W

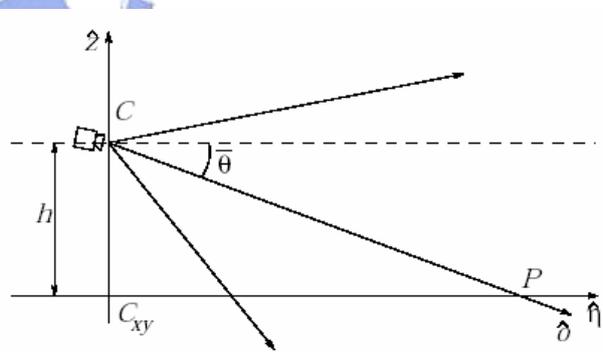
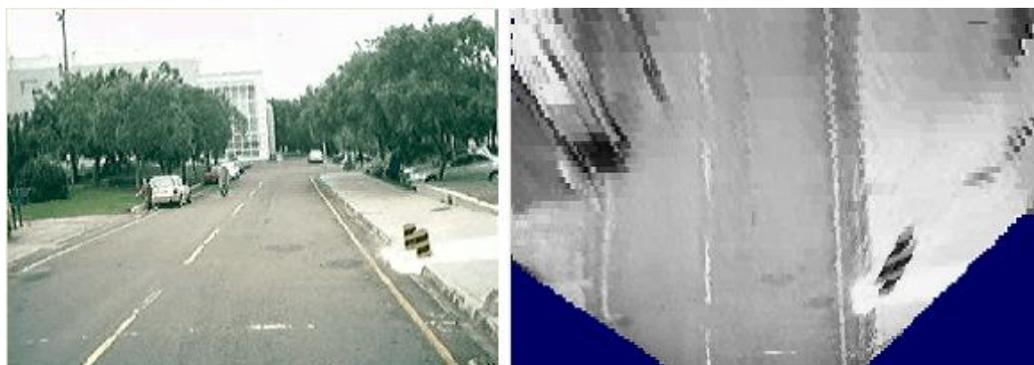


圖 2.3 $z\eta$ plane in the W

利用 (2-1)式及 (2-2)式($z=0$)對應到世界座標上實際的位置(在此將原點定在車子裝設雷射掃描器的地方),再依世界座標上實際的位置,由俯視方向等比例重新應對回圖像座標,重新分配其對應的灰階值,將透視影響移除,其結果如圖 2.4 所示。

$$x(u, v) = h \times \cot[(\bar{\theta} - \alpha) + u \frac{2\alpha}{n-1}] \times \cos[(\bar{r} - \alpha) + v \frac{2\alpha}{n-1}] + l \quad (2-1)$$

$$y(u, v) = h \times \cot[(\bar{\theta} - \alpha) + u \frac{2\alpha}{n-1}] \times \sin[(\bar{r} - \alpha) + v \frac{2\alpha}{n-1}] + d \quad (2-2)$$



(a)原始圖

(b) IPM 後的結果

圖 2.4 原始圖及 IPM 後的結果

此方法除了可將透視影響移除，形成俯視圖，並可將部份不屬於車道的影像移除，讓影像集中在車道上，減少在車道線擷取時受到周圍影像影響。



2-1-2 車道線擷取

由於經過IPM的轉換後，車道線形成相互平行且其寬度較為一致，並且在一定m pixel寬度下，車道線的水平灰階值形成『低灰階值—高灰階值—低灰階值』如此分布，因此，我們就利用此特徵，依(2-3)式，將車道線從圖像中擷取出來，其結果如圖2.5所示。

$$r(x, y) = \begin{cases} d_{+m}(x, y) + d_{-m}(x, y), & \text{if } (d_{+m}(x, y) > 0 \wedge d_{-m}(x, y) > 0) \\ 0, & \text{otherwise} \end{cases} \quad (2-3)$$

且 $d_{+m}(x, y) = b(x, y) - b(x, y + m)$

$d_{-m}(x, y) = b(x, y) - b(x, y - m)$ 而 $b(x, y)$ 為圖像上 (x, y) 位置的灰階值

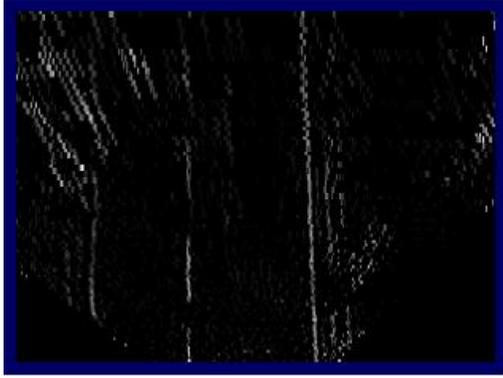


圖 2.5 利用灰階特徵所擷取出的車道線 圖 2.6 利用 GMD 加強車道線影像

從圖 2.5 我們可以看出取抓出的車道線並不是很清楚，這是由於其灰階值是由車道線與車道線 $\pm m$ pixel 的灰階值差值的和決定，所以當原始圖像的車道線不是得清楚時，其所得的灰階值也較小，造成部份車道線較為模糊，因此，我們採用 GMD (geodesic morphological dilation) 的方法來[13]加強車道線的影像。

GMD 方法主要是能將有相互連通的區塊，依其原有的灰階值為基礎，來提升整區域灰階值，方法敘述如下：

- (1) 首先由上而下，由左而右掃瞄所輸入的影像，遇到灰階值不為零的位置，以其為九公格的中心。
- (2) 比較九公格的中心與四連通方向的灰階值，找出最大的灰階值為所得的灰階值，代替原灰階值。
- (3) 繼續掃瞄圖像，直到掃完圖像所有的位置，為完成一次動作。
- (4) 重複上述動作，直到所有灰階值不再改變。

對圖 2.5 作 GMD 的結果如圖 2.6 所示。GMD 的詳細過程如圖 2.7 所示。

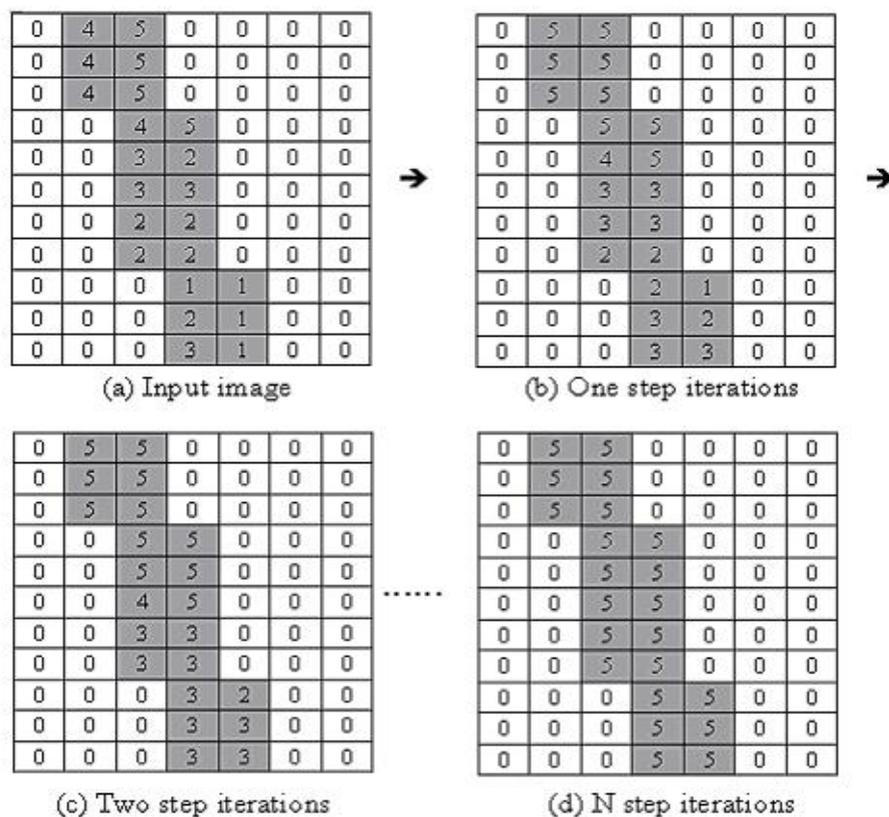


圖 2.7 GMD 處理過程

由圖 2.5 來看一般車道線區域其灰階值偏高，而其它雜訊區域灰階值都偏低，因此，利用此方法除了提升車道線區域的灰階值，其雜訊區域的灰階值也會有增加但不影響之後的處理。但由於此方法執行到最後需要較長的時間，所以我們並不執行到最後，只需執行數次動作，將部份車道線影像加強，達到我們所需求即可，以縮短處理時間。

我們將對圖像作進一步二值化的處理，將車道線與其它雜訊分離，而最簡單的二值化方法是選取一固定閾值，作為二分法的標準，但由於所要分離出來的車道線其灰階值在整張圖上並不一致，若只使用一固定閾值，可能造成部份車道線被捨去或可能得到大量的雜訊點，因此，我們在此採用局部區域二值化方法，利用局部區域的灰階值，定出局部區域的二值化閾值，如 (2-4) 式所示，來避免或減少上述情形的發生，其結果如圖 2.8 所示。

$$t(x, y) = \begin{cases} 255, & \text{if } e(x, y) \geq \frac{m(x, y)}{k} \\ 0, & \text{otherwise} \end{cases} \quad (2-4)$$

$e(x, y)$:圖上的灰階值， $m(x, y)$:在 $c \times c$ 區域上最大灰階值， k :常數。 $[c=5, k=2]$

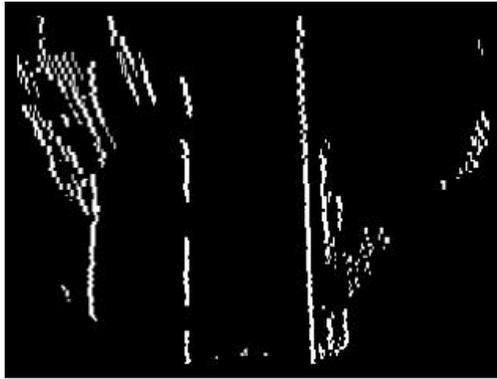


圖 2.8 二值化後的影像

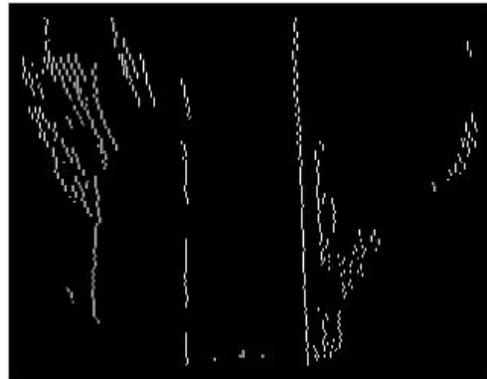


圖 2.9 細化處理後的影像

最後，為了以後定義車道線處理上的方便、及減少處理量和錯誤的發生，我們將所得的車道線影像，計算從水平方向其灰階值連續為 1(255)的個數，來獲得線段寬度，再取其寬度的中心點，做細化處理，如圖 2.9 所示。

2-1-3 定義車道線及偵測結果

在定義車道線方面，首先由影像上定義出最可能的車道寬度，由於一般來說車道內的雜訊是最少的，所以我們定義特定點 S (即CCD的位置)為軸，如圖 2.10，統計軸的左、右兩邊，在影像上其灰階值不為零到軸的最短距離，其和為車道寬度，利用其找出最有可能的車道寬度。但仍有可能找出不正確的車道寬度，所以我們統計圖像上各點與各點間的距離，設定其車道寬度的條件，如(2-5)式所示，找出另一可能的車道寬度，利用其所獲得的兩個可能的車道寬度，相互的輔助、驗證，來提升其正確性。

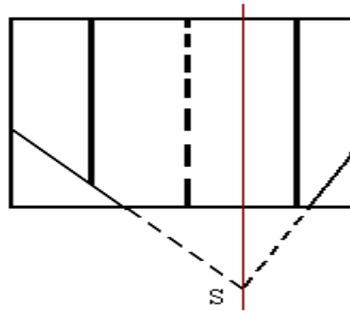


圖 2.10 以特定點 S 為軸之示意圖

$$\frac{N}{8} < w_i < \frac{3}{4}N \quad \frac{N}{16} < c_i < \frac{15}{16}N \quad c_i - w_i > \frac{3}{4}N \quad c_i + w_i > \frac{N}{4} \quad (2-5)$$

w_i : 車道寬度, c_i : 車道中線位置

由於一般來說在影像底部雜訊較少，且車道線在小區域中大約為垂直的直線，所以我們將由圖片的底部依車道線的垂直關係，先定幾個小區域計算其灰階值總和，找出其值最大者取其中央位置為一車道線的代表點，再由代表點依此向上再定出幾個小區域找出其它代表點，一直向上尋找直到圖像邊緣或所定的區域其灰階值皆為零為止，利用此方法定出車道線，並找出所有可能的車道線，從所有可能的車道線中，將其代表點上下相連，取左右 5 個 pixel 為寬度，計算其所占的灰階值總和，利用其所占灰階值最高者，我們定義為 sample lane，如圖 2.11 所示，再利用 sample lane 及所找出的車道線寬度，由 sample lane 以車道線寬度為單位向左、右位移，重建出所有的車道線，如圖 2.12 所示，利用此方法，可以彌補部份車道線殘缺的部份，獲較佳的車道線。

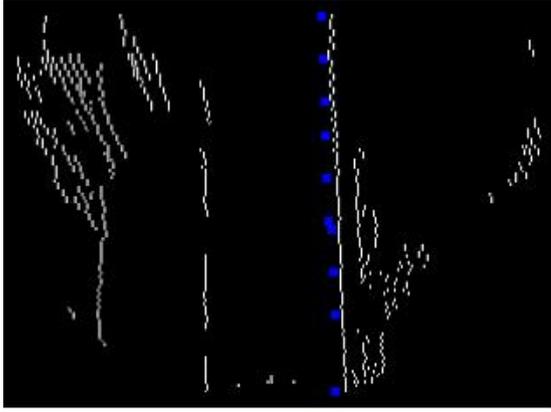


圖 2.11 所找出的 sample lane

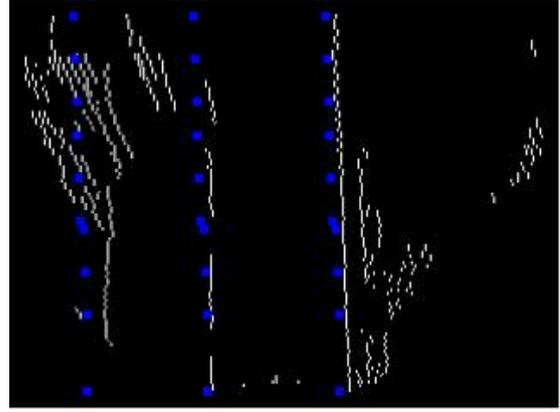


圖 2.12 所找出的車道線

之後，要將所找出的車道線重新對應回原始圖上，所以先由影像座標等比例應對到將世界座標，再將世界座標依 (2-6) 式對應回影像座標，如圖 2.13 所示。最後，我們以特定點 S 為軸，找出最靠近左、右兩條車道線，並以所得的車道線為基礎定義出偵測視窗，並對視窗內的影像做灰階、特徵抽取(同上述抽取車道線的方法)，局部區域二值化(同上述二值化之方法)，利用所得的影像，找出灰階值分布最集中的區域，將其設在視窗的中間區域，來微調偵測視窗，如圖 2.14 所示，其程式流程如圖 2.15 所示，最後所得的偵測視窗，將提供動態視窗之車道線偵測方法其偵測視窗。

$$\begin{aligned}
 u(x, y, 0) &= \frac{\arctan\left[\frac{h \sin \gamma(x, y, 0)}{y - d}\right] - (\bar{\theta} - \alpha)}{\frac{2\alpha}{n - 1}} \\
 v(x, y, 0) &= \frac{\arctan\left[\frac{y - d}{x - l}\right] - (\bar{\gamma} - \alpha)}{\frac{2\alpha}{n - 1}} \\
 \gamma(x, y, 0) &= \arctan\left(\frac{y - d}{x - l}\right)
 \end{aligned} \tag{2-6}$$

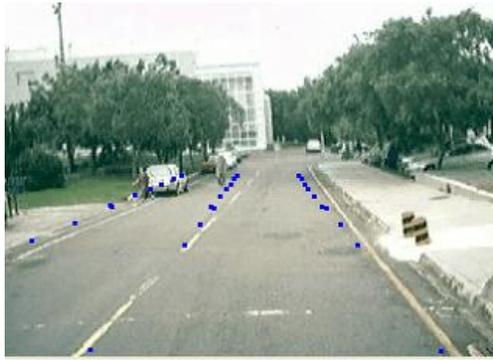


圖 2.13 將車道線對應回原始圖上

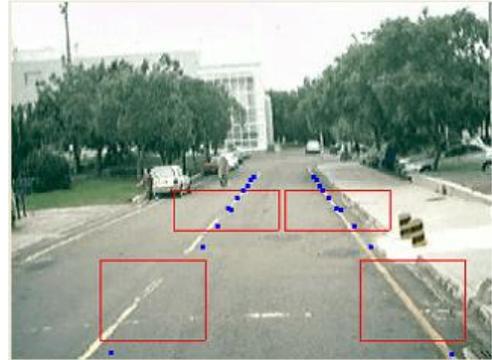


圖 2.14 所找出的偵測視窗

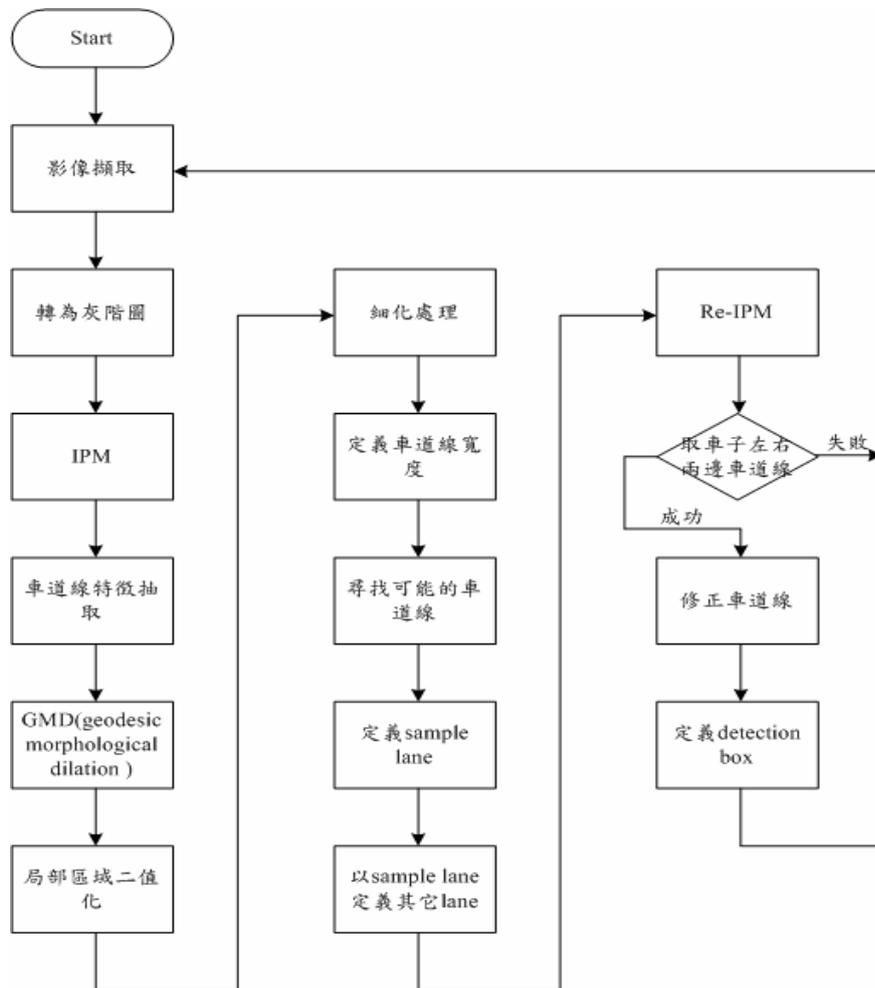


圖 2.15 IPM based 之車道線偵測方法其程式流程

IPM based 之車道線偵測方法，定義偵測視窗雖可達到不錯的效果，但所花的時間較長約需 500 ~ 600 ms(以 320*240 pixel 在AMD CPU 1.1GHZ為例)，若以 PC based 來說是無法達到即時的要求，需與其它方法配合改善之，因此提出動態

視窗之車道線偵測方法與之配合。圖 2.16 是偵測結果。

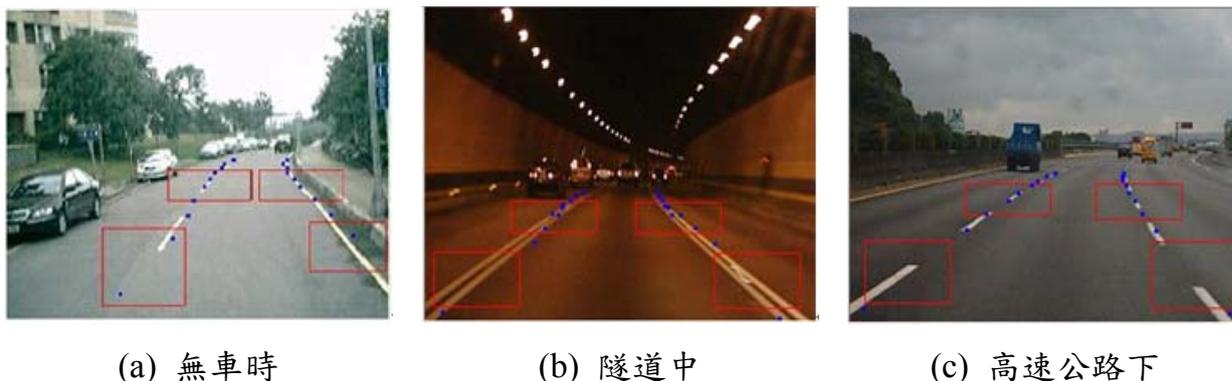


圖 2.16 IPM based 之車道線偵測方法其車道線偵測結果

2-2 動態視窗之車道線偵測方法

本方法我們所關注的影像只有車道線的部分，而其餘部分的影像皆可視為是雜訊，故我們應盡可能地將其濾除掉，以使我們所要處理的影像能單純化，如此不僅能夠提昇判斷的正確性，且能增加處理的速度。在此我們採用的方法是在影像上取左右四個小視窗，如圖 2.17 所示，而我們只對進入這四個小視窗內的影像作處理，這種作法不僅去除了絕大多數我們不需要的影像，且減少了處理的資料量，但其大小及位置並不是任意地選取皆可，必須能夠讓車道線能夠進入視窗內才行，在此我們採取動態視窗方法，是由前一張影像所得的車道線，來定義視窗的位置及大小。

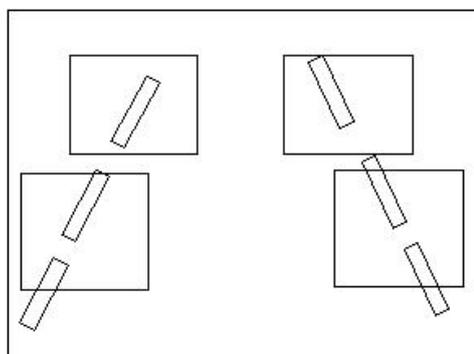


圖 2.17 在影像上取兩個小視窗

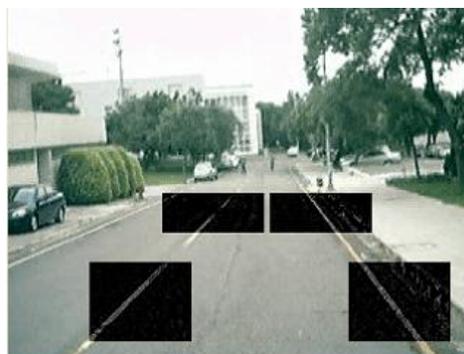


圖 2.18 利用灰階特徵所擷取出的車道線

首先，先將車道線抽取出來，由於車道線在所選取的視窗中，其寬度大小大約一致，所以仍然會存在，在一定寬度(m pixel)下，車道線的水平灰階值形成『低灰階值—高灰階值—低灰階值』如此分布，因此，我們仍利用此特徵，依 (2-3) 式，將車道線從圖像中擷取出來，其結果如圖 2.18 所示。

由於所抓出的車道線，在視窗中其所得的灰階值大約一致，因此，我們將選取一固定的閾值，作二值化的處理，但因天氣條件的不同，其灰階值也有所不同，在天氣暗時，灰階值偏低，在天氣亮時，灰階值偏高，所以單取一固定閾值，可能無法適應各種狀況，因此，我們在作二值化前，要先作一些前置處理。一般而言，在視窗中所抓取到的車道線其灰階值，比一般雜訊點高得多，所以我們採取將其灰階值作線性分配，對應到灰階值 0 ~ 255 區間，使得車道線的灰階值落在較高的區段，雜訊點的灰階值落在較低的區段，經過如此處理後，可以減少固定閾值之二值化無法分離出車道線的狀況，圖 2.19 最後二值化後的影像。

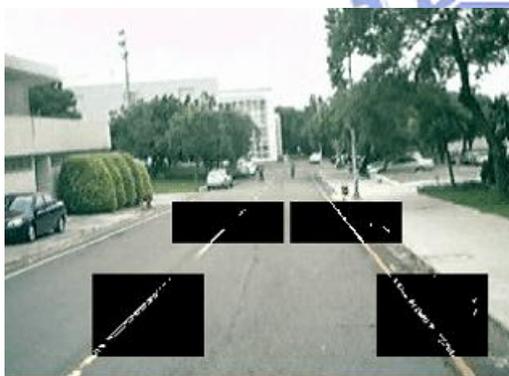


圖 2.19 二值化後的影像

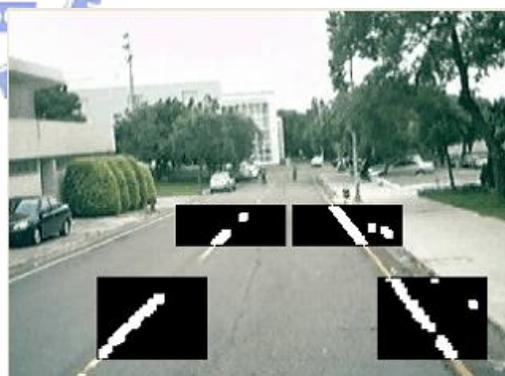


圖 2.20 形態擴張後的影像

接著，我們使用 (2-7) 式形態擴張的方式，將破碎區塊的車道線接合在一起，如圖 2.20 所示，而為了以後定義車道線處理上的方便、及減少處理量和錯誤的發生，我們將所得的車道線影像做細化處理，如圖 2.21 所示。

$$t(u, v) = x \cup (x_0 \cup x_1 \cup x_2 \cup x_3 \cup x_4 \cup x_5 \cup x_6 \cup x_7) \quad (2-7)$$

where	x_0	x_1	x_2	說明：形態擴張處理是藉由 3x3 的罩遮， 比對 x 周圍的點，只要其中有一點 是 1 (255) 則輸出為 1 (255)。
	x_3	x	x_4	
	x_5	x_6	x_7	

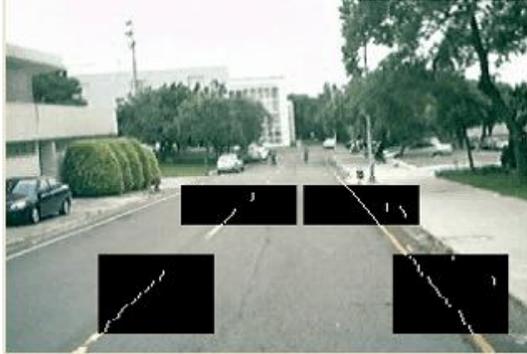


圖 2.21 細化處理後的影像

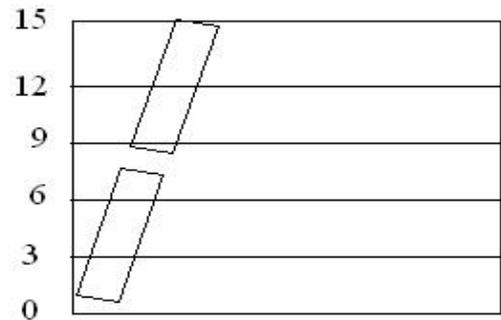


圖 2.22 定義車道線資料取樣的方法

最後，我們將定義出車道線，一般而言，若想獲得一直線線段時，只需取出線段上的任兩點，計算其斜率、截距即可得知，但若我們取到的不是車道線，而是雜訊點的位置時，將會造成錯誤的結果，故在此我們希望不只取一個數據就依此決定車道線，因此，我們在計算車道線的斜率時，所採用的量測方法為在所取的兩個影像視窗中，每隔 3 行取 1 行的值，如圖 2.22 所示。

我們車道線定義方式是，先計算相鄰的兩個取值間的斜率，再由我們所得的這些斜率值去求平均，然後再把跟平均值差異過大的值剔除掉，再把剩餘的值做平均，再把跟平均值差異過大的值剔除。我們經由這種平均、去除差異過大的值、再平均的步驟重複數次，並把每一次與平均的差異範圍逐步地縮小，如此將可以減少雜訊對斜率計算的影響。最後我們利用所得到的斜率，和車道線上的一點，計算出其截距，即獲車道線直線方程式，再由所得的車道線，調整視窗的位置及大小，讓車道線位於視窗的中央，提供下張影像視窗的位置，其程式流程，如圖 2.24 所示，上圖所偵測之車道線及其它偵測車道線的結果，如圖 2.23 所示。

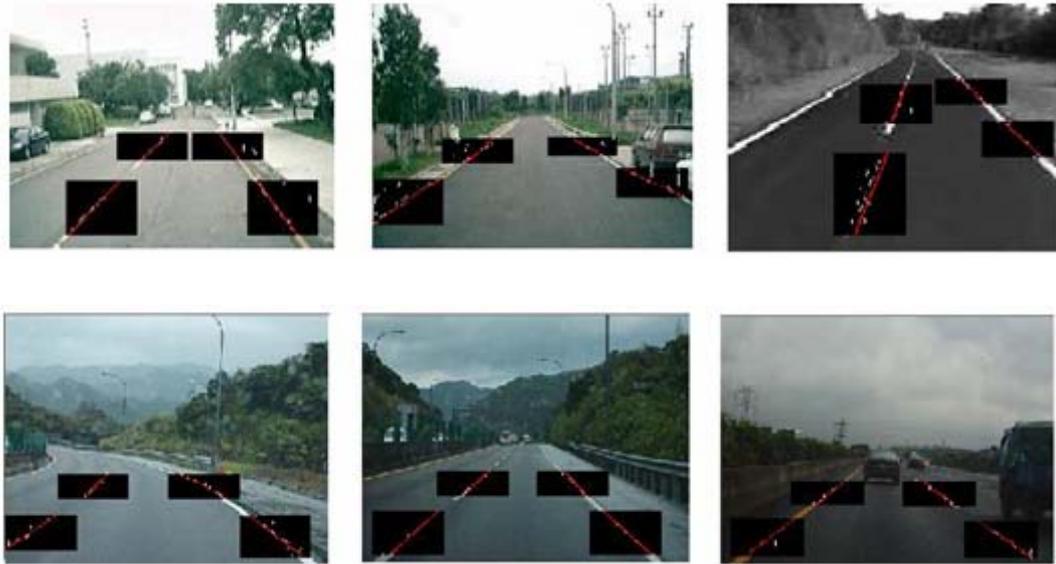


圖 2.23 動態視窗之車道線偵測方法所定義出的車道線影像

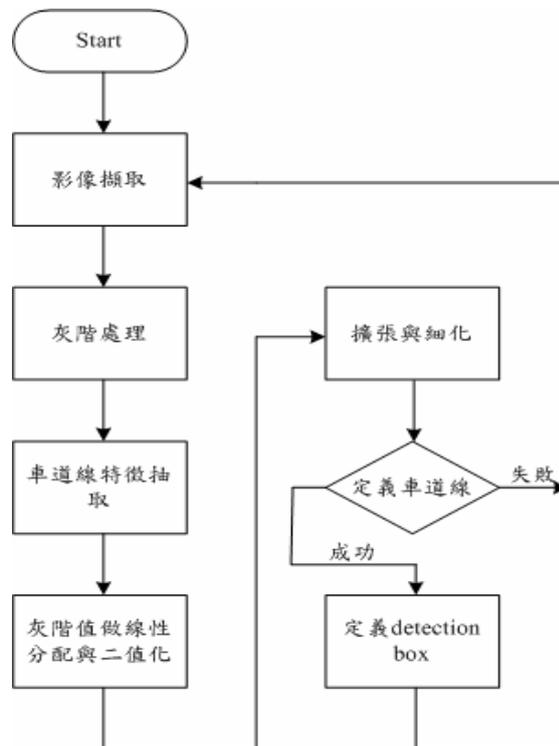


圖 2.24 動態視窗之車道線偵測方法其程式流程

在這裡我們所取樣的這些行當中，我們選擇最靠車道內側值為 1(255)的點，視為車道線所在的位置。我們所使用這個計算車道線的方法，有幾個值得注意的特點：

- 1、當我們每隔 3 行取 1 行時，可確保當車道線進入視窗內時，不會因為路

上的車道線為不連續的線段，而有偵測不到的現象。

2、重複平均、剔除誤差較大的值再平均這樣的步驟，將可降地雜訊的影響，並提高計算的準確率。

3、另外，我們取每行中最靠車道內側的點，亦可減少雜訊的影響，因我們可從道路影像中察覺到，車道內側的雜訊明顯比車道外側少很多，故最靠車道內側有值的像素，剛好即是車道線位置的可能性明顯較高，故這種取法可降低錯誤的發生。

此方法由於處理量較少，因此處理速度較快大約 60 ms 以下 (以 AMD CPU 1.1GHZ 為例)，可達到即時的要求，而其辨識率在略差的影像下還有不錯的效果，但若太過模糊時仍無法辨識，且其動態的偵測視窗在車子變動太大時，會無法跟上，造成無法辨識的結果，因此也要配合其它方法改善之。由於 IPM based 之車道線偵測方法和動態視窗之車道線偵測方法，各有其優缺點，所以我們提出整合型車道線偵測方式。

2-3 整合型車道線偵測方法

整合型車道線偵測方法，主是先利用 IPM based 之車道線偵測方法，找出車道線偵測視窗，再由動態視窗之車道線偵測方法，從所獲得偵測視窗對車道線作追蹤動作，但若一直無法抓取到車道線，再切換回 IPM based 之車道線偵測方法。此作法是在取 IPM based 之車道線偵測方法其辨識率高，並較能適應多種狀況之優點，與動態視窗之車道線偵測方法其處理速度快的優點，來相互配合，達到我們的需求。其程式流程如圖 2.25 所示。

圖 2.26 呈現整合行車道線偵測結果，其中對於彎曲的車道線也很不錯的效果，如圖 2.26(c)所示，實驗發現，IPM based 的方法雖不能很準確的抓到車道線，但只要車體不搖晃過大，可透過動態視窗的偵測方法微調後抓到。陰影路段如圖 2.26(d)容易造成相對應的雜訊變多，影響車道線的定義而造成辨識率下降，但影響不會很嚴重，仍有相當的辨識率。其它天候影像中，雨天路況，如圖 2.26(e)

所示，也是我們嘗試狀況之一，當路面的雨水過多，其水漬的反光容易造成偵測失敗，而目前還沒有一個好的解決方式，故此偵測法則仍有改善的空間。

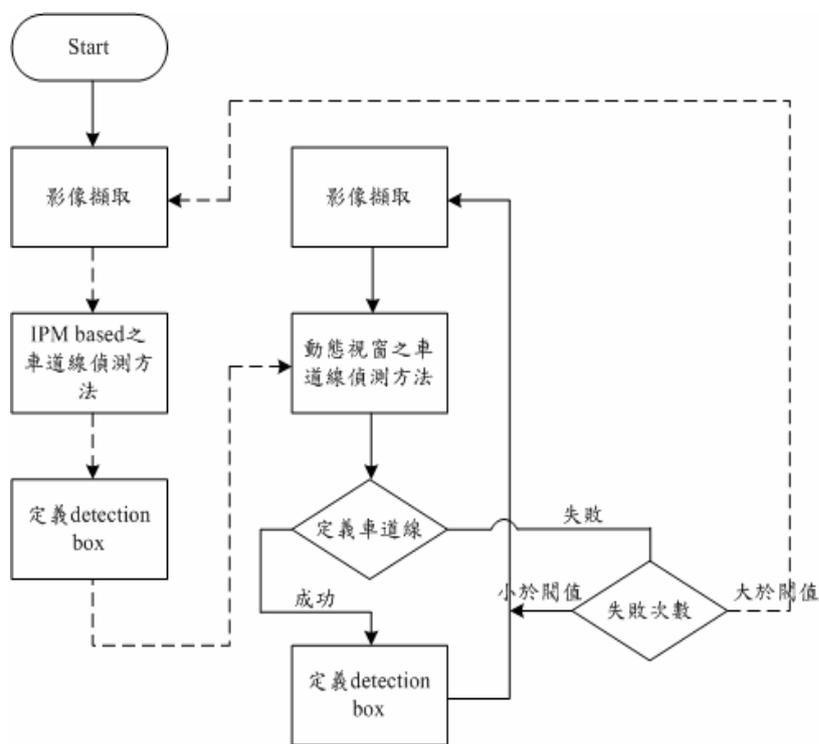


圖 2.25 整合型車道線偵測方法其程式流程



(a) 無車



(b) 有車



(c) 彎道



(d) 陰影



(e) 雨後路面

圖 2.26 整合型車道線偵測方法其偵測結果

整合型車道線偵測方法，從我們所使用的狀況來看，已有不錯的辨識率及即時性(平均處理速度約 80 ms -- 以 AMD CPU 1.1GHZ 為例)，可達到我們的需求。一般在 PC based 架構下，在影像辨識方面要有較高的辨識率，通常就需要較長的處理時間，而要有較快的處理速度，一般而言其辨識率會大幅下降，因此整合型車道線偵測方法在這方面提供了一相對應的解決方案，讓在 PC based 下也可以擁有較高的辨識率與即時性。

2-4 夜間車道線偵測方法[15]

在夜間車道線偵測方面，我們主要還是採用以IPM 為基礎的整合型車道線偵測法。然而，由於影像明亮度與白天有差，所以我們必須作些微的修改以適用於夜間情況。在車道線截取步驟中，我們增加特徵抽取的閾值，這是因為夜間影像中，路面與車道線顏色較接近，且路面因路燈、車前燈等的照射下造成顏色不均，此時必須將閾值調大才能去除路面雜訊而留下正確的車道線(如圖 2.27 所示)。因此程式中，白天的閾值為 3，而夜間的閾值調為 6。

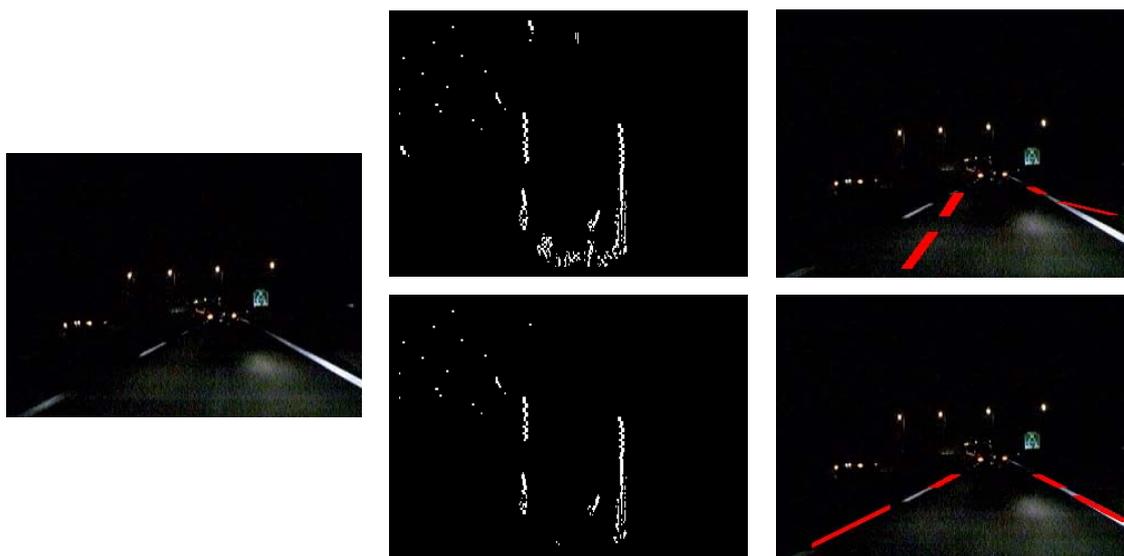


圖 2.27 不同的閾值所得之車道線偵測結果

圖 2.28 顯示一些夜間其它狀況及黃昏影像之車道線偵測結果，在一般夜間影像中，只要車道線標示明確，即使有對方來車車頭燈的影響，皆有不錯的偵測結果，如圖 圖 2.28(a)(b)，而對於 圖 2.28(c)來說，路面上的標示，不論是白天或夜晚，仍是需待解決的狀況。另外，我們發現黃昏狀況(圖(d))的偵測最容易受到駕駛人車頭燈的影響，當天色還不夠黑，車道線特徵擷取的閾值太小會造成錯誤的偵測。



(a)有車



(b)彎道



(c)路面上有路標



(d)黃昏



圖 2.28 夜間及黃昏路況之車道線偵測結果

2-5 彎道車道線偵測方法

對於一般的彎道，在本系統仍可偵測的到，但在此要說明的彎道部分跟一般不一樣，對於左車道線部份，是屬於不連續的情形，如圖 2.29 所示，而右車道線部份為一具有紅色成份及連續的線，如圖 2.30 所示。經由測試結果顯示出，由於左車道線存在著不連續的情形，所以容易發生偵測錯誤。而右車道部分，由於是連續的，故試著將其當作車道線以作為導航的依據。而為了使原系統能偵測到右車道線，所以最直覺的想法就是修改原本的前處理部份使得紅色部分能跟其他顏色分離。



圖 2.29 左車道線不連續發生偵測錯誤



圖 2.30 右車道線為具紅色成份之車道線

根據我們的目的-尋找右車道線，故影像中非右車道線之物皆屬於雜訊。於是，在此步驟中，我們必須使用一些影像處理的方法將雜訊去除。一般去除雜訊的方式多用二值化方法，也就是先將彩色影像取灰階，再取一固定閾值進行二值化處理，此二值化處理會將閾值以下的圖素設為黑色，閾值以上的設為白色，最後得到的是一黑白影像，而此黑白影像的好壞容易影響偵測結果。如圖 2.31 所示，雖然已經能將大部份雜訊濾除，但對於亮度高的車道線雜訊卻不盡理想，一再的調高二值化閾值並不是好的解決方法，這是由於灰階影像是明亮度的表現，當雜訊的明亮度相等於車道線的明亮度時，此種方法便會失效而影響之後的控制部份，所以原本的偵測方式並不適用。因此，我們必須更改去除雜訊的程序。

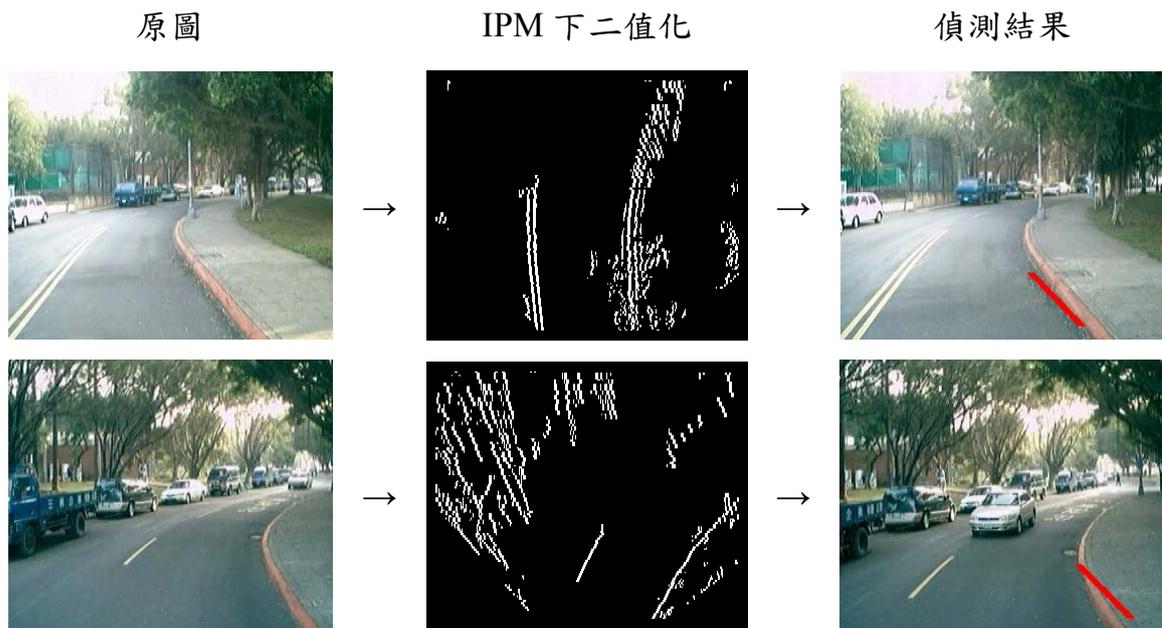


圖 2.31 彩色影像經灰階、IPM、二值化處理後的偵測結果

前處理的程序更改為色度座標轉換(color-space transform)以及條件式二值化。在此我們所進行的色度座標轉換是將原彩色影像RGB座標依(2-8)的式子轉換至YCrCb座標。



$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ 0.439 & -0.368 & -0.071 \\ -0.148 & -0.291 & 0.439 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (2-8)$$

表 2-1 色度座標轉換 RGB→YCrCb

顏色名稱	(R G B)	→	(Y Cr Cb)
紅(red)	(191 27 75)		(86 196 125)
白(white)	(255 255 255)		(235 128 128)
黑(black)	(0 0 0)		(16 128 128)

表 2-1 顯示數個我們較關心的代表色之轉換結果。明顯地，相對於白色與黑色，紅色具有較高的Cr值，因此，我們可單就Cr值進行二值化的處理，以期將雜訊(白

色車道線)濾除。經由實驗結果，由於車道線並非只是單純的紅色圖素，而是摻有亮度的成份，經轉換後發現Y值會偏高，所以如再將Y值加入二值化條件效果會更好。最後，我們採用的條件式二值化如 (2-9)式所描述，並由實驗中調整其閾值為Y=135，Cr=140。

$$r(i, j) = \begin{cases} 255, & \text{if } r_Y(i, j) \geq 140 \wedge r_{Cr}(i, j) \geq 135 \\ 0, & \text{otherwise} \end{cases} \quad (2-9)$$

經由色度座標轉換以及條件式二值化處理後的結果如圖 2.32 所示，程式修正後流程如圖 2.33。由圖中可清楚發現，地面上的除了右車道線外，其它幾乎完全被濾除，效果非常不錯。

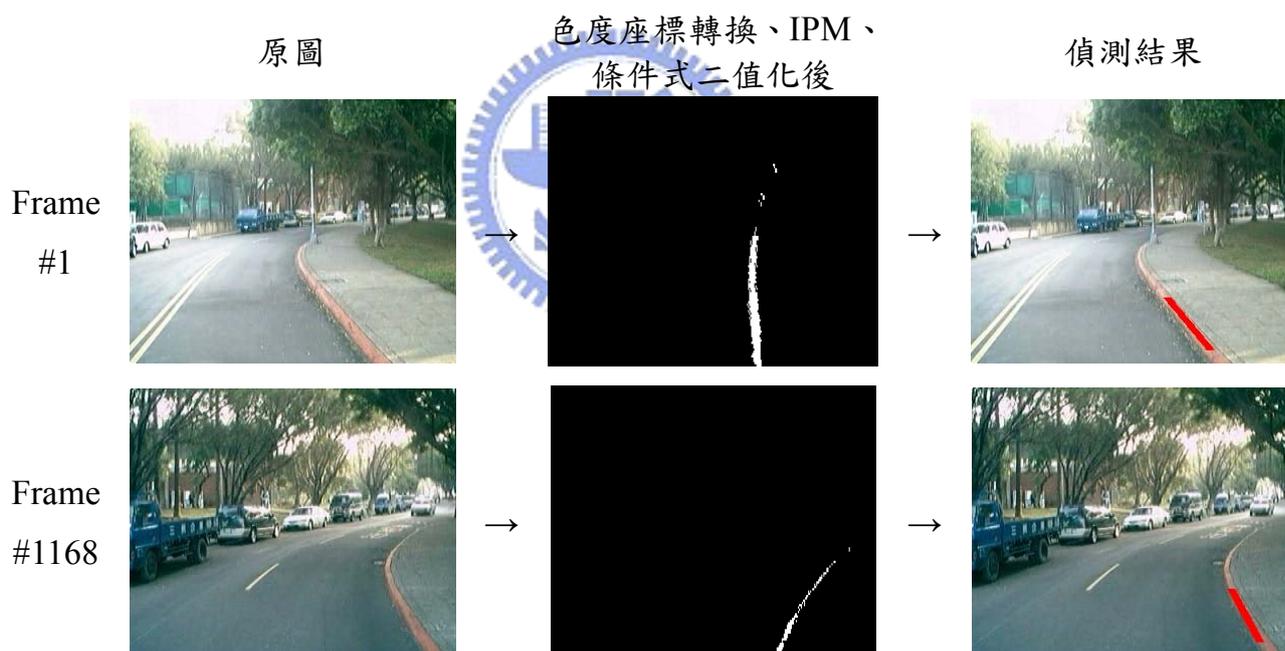
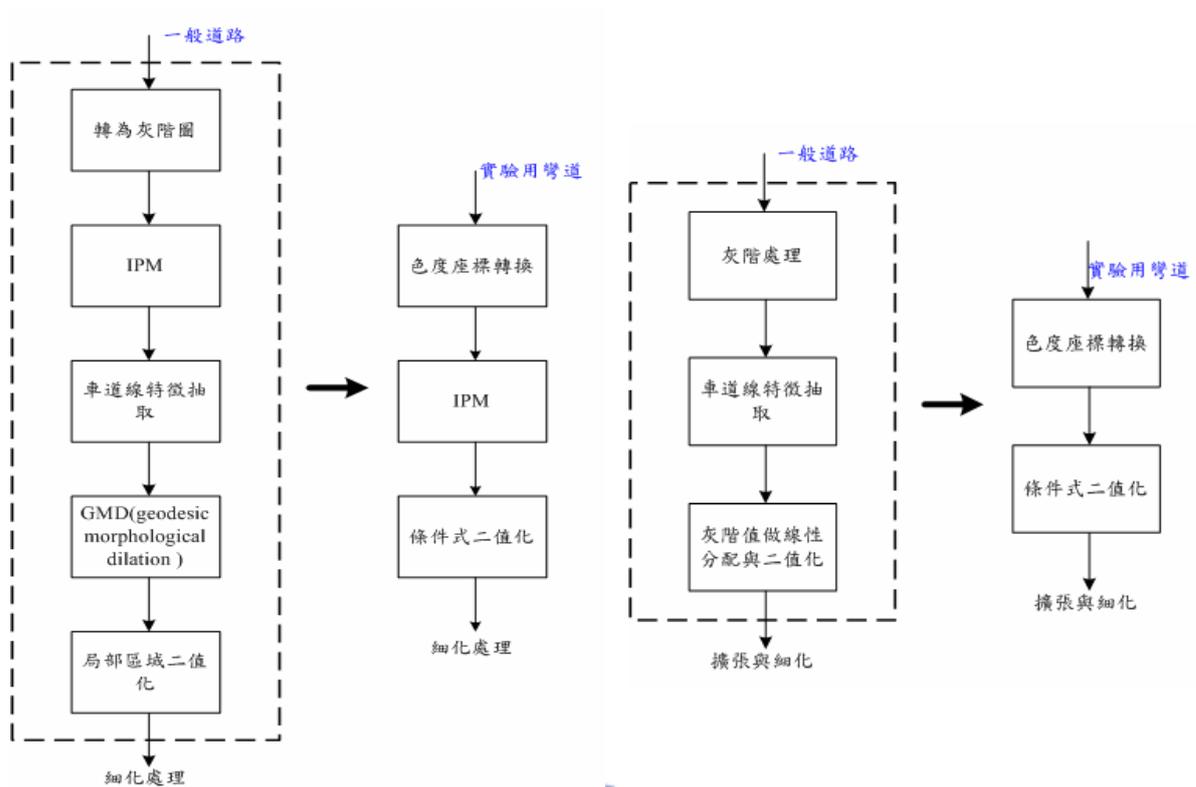


圖 2.32 彩色影像經處理後的偵測結果(其中閾值為 Y=140，Cr=135)



(a) IPM-based 偵測流程修正 (b) 動態視窗偵測流程修正

圖 2.33 彎道偵測方法程式流程修正

另外，由圖 2.32 的偵測結果來看，在小幅度轉彎的曲線路段時，由於在車前的車道線其實是相當接近直線的，因而我們是可以利用直線的方式來代表偵測結果。就一條曲線而言，是可以視為直線的組合，所以在彎道上，是可以利用片段的直線來近似整條彎曲道路。

第三章 導航資訊與前輪轉動角度

前一章節已詳細介紹本系統如何將影像中的特徵(車道線)擷取出來，包含了彎道部份，也從結果顯示出目前的車道線偵測能力不管是在白天或夜晚都有足夠的準確性，而本章節主要利用車道線成果去發展視覺導航車，先了解車輛的轉向運動分析，之後介紹如何利用車道線此特徵來獲得相關的導航資訊，並驗證資訊的可靠性，最後介紹如何本系統得到前輪轉動角度的方式。

3-1 車輛轉向運動分析

本小節主要藉由簡化後的模型bicycle model (或single track model)(圖 3.1)來了解車輛轉向運動的基本特性。一般車子的動態系統是屬於六維的非線性模型，但是過於繁雜，對於分析方面並不容易，也造成處理時間增加，無法達到即時性。所以透過一些假設，將其分解成縱向(longitudinal)和側向(lateral) 模型，使其成為線性系統。另外，有不少的研究及實驗證明顯示，使用此線性模型來分析車輛的運動行為是可行的，因為使用此線性模型也可得到準確的結果，而本研究將著重在側向模型。

表 3-1 參數符號及解釋

Symbol	Description
m	車輛重量 (kg)
γ	偏轉速率 (rad/s)
α_f, α_r	前後輪的側滑角 (rad)
c_f, c_r	前後輪的剛性係數 (Nt/rad)
I_z	車輛重心(CG)處的轉動慣量 ($kg \cdot m^2$)
l_f, l_r	前後輪離車輛重心的距離 (m)

δ_f	前輪轉動角度 (degree)
v	車輛速度 (m/s)
y_L	前視距離處的偏移距離 (m)
φ_L	前視距離處車子縱軸與道路切線的夾角 (degree)
ρ_L	道路曲率 (1/m)

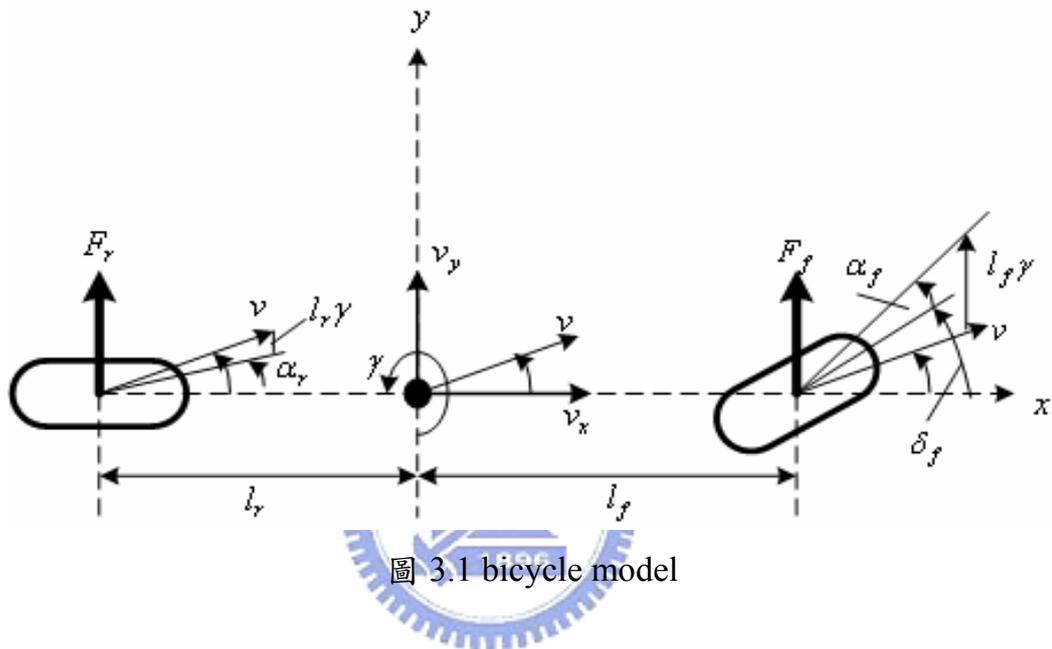


圖 3.1 bicycle model

轉向運動方程式的獲得是根據 Newton-Euler 等式，計算所有作用在車子上的側向力(lateral force)及力矩(torque)，分別代表了側向運動與偏轉運動，在假設小的駕駛角度及線性的輪胎模型之下，轉向動態方程式如下所示：

(a)側向運動

$$F = F_f + F_r = ma = m(\dot{v}_y + v_x \gamma) \quad (3-1)$$

(b)偏轉運動

$$\tau = F_f l_f - F_r l_r = I_z \dot{\gamma} \quad (3-2)$$

若角度以逆時針方向為正，則前後輪受到的側向力作用在負的 y 方向，分別為

$$\begin{aligned} F_f &= -c_f \alpha_f \\ F_r &= -c_r \alpha_r \end{aligned} \quad (3-3)$$

輪胎的側滑角定義為輪胎的前進方向與輪胎所朝方向的夾角，而前後輪側滑角分別為

$$\begin{aligned} \alpha_f &= \arctan\left(\frac{v_y + l_f \gamma}{v_x}\right) - \delta_f \approx \frac{v_y + l_f \gamma}{v_x} - \delta_f \\ \alpha_r &= \arctan\left(\frac{v_y - l_r \gamma}{v_x}\right) \approx \frac{v_y - l_r \gamma}{v_x} \end{aligned} \quad (3-4)$$

將 (3-3)、(3-4) 代入 (3-1) 與 (3-2)，選擇以 γ 及 v_y 為狀態變數，二維線性的狀態方程

式呈現如 (3-5)，而 δ_f 為系統的主要輸入，輸出分別為質心處的側向加速度及偏轉角加速度。

$$\begin{bmatrix} \dot{v}_y \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} -\frac{c_f + c_r}{mv_x} & \frac{c_r l_r - c_f l_f}{mv_x} - v_x \\ -\frac{l_f c_f + l_r c_r}{I_z v_x} & -\frac{l_f^2 c_f + l_r^2 c_r}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ \gamma \end{bmatrix} + \begin{bmatrix} \frac{c_f}{m} \\ \frac{l_f c_f}{I_z} \end{bmatrix} \delta_f \quad (3-5)$$

另外，與視覺系統結合來看，由於視覺系統可量測在參考點處的 y_L 及 φ_L ，故觀察車子在道路上行駛時 y_L 及 φ_L 的變化情形，如圖 3.2。首先，把前視距離 L 處與車道中心線的交點定為參考點：

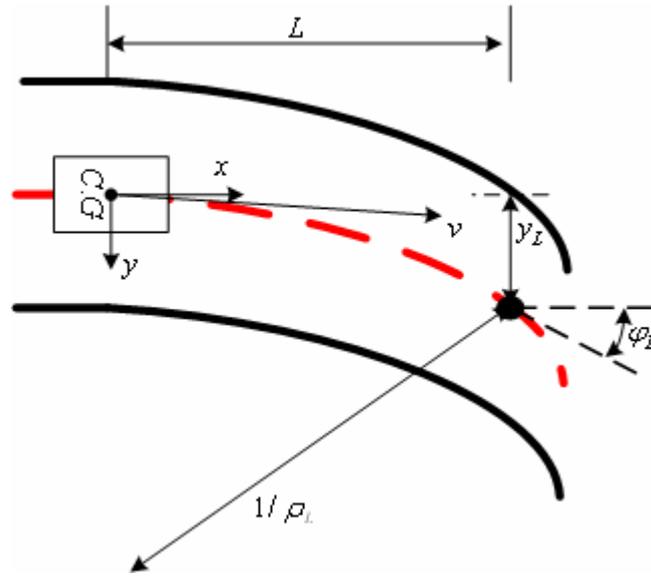


圖 3.2 視覺系統在參考點處所估測出的相關資訊

視覺系統的動態特性：

$$\begin{aligned} \dot{y}_L &= v_x \phi_L - v_y - \gamma L \\ \dot{\phi}_L &= v_x \rho_L - \gamma \end{aligned} \quad (3-6)$$

將側向動態 (3-5) 與視覺動態 (3-6) 結合之後的動態特性如下：

$$\begin{bmatrix} \dot{v}_y \\ \dot{\gamma} \\ \dot{y}_L \\ \dot{\phi}_L \end{bmatrix} = \begin{bmatrix} -\frac{c_f + c_r}{mv_x} & -v_x + \frac{c_r l_r - c_f l_f}{mv_x} & 0 & 0 \\ -\frac{l_f c_f + l_r c_r}{I_z v_x} & -\frac{l_f^2 c_f + l_r^2 c_r}{I_z v_x} & 0 & 0 \\ I_z v_x & -L & 0 & v_x \\ -1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_y \\ \gamma \\ y_L \\ \phi_L \end{bmatrix} + \begin{bmatrix} \frac{c_f}{m} \\ \frac{l_f c_f}{I_z} \\ 0 \\ 0 \end{bmatrix} \delta_f + \begin{bmatrix} 0 \\ 0 \\ 0 \\ v_x \end{bmatrix} \rho_L \quad (3-7)$$

從 (3-7) 看來，雖然好像有兩個輸入，但是最主要的輸入還是 δ_f 。而 ρ_L 部份，可將其視為外界干擾，畢竟曲率只有行駛在彎道時才存在，同時，也不難看出， ρ_L 的存在會影響著前視偏移距離。

由此可知，藉由視覺系統不斷的更新車輛的橫向狀態與道路的彎曲變化，獲得 δ_f 來操控車輛的運動行為，來達成車輛的即時導航控制，故接著要介紹如何從影像資訊得到 δ_f 。

3-2 導航資訊擷取

一般來說，人為駕駛可以知道前方道路的變化情形，知道該怎樣對車子進行控制。同樣的，對自動駕駛來說，也是要能不斷的知道前方道路的變化，知道車子目前本身的狀態，才能對車子做出適當的控制。本小節主要介紹利用車道線特徵來擷取出自動導航系統所需的資訊，如在參考點處的側向偏移距離、偏轉角度及道路曲率資訊等等，圖 3.3 為流程圖，以下將逐一介紹。

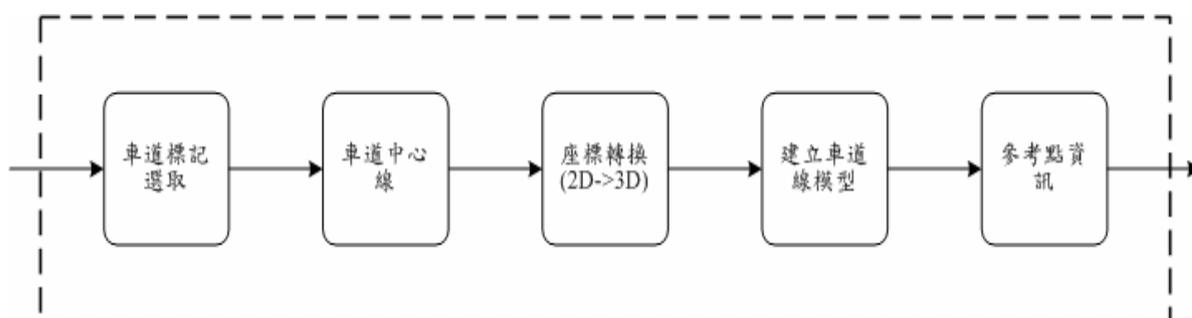
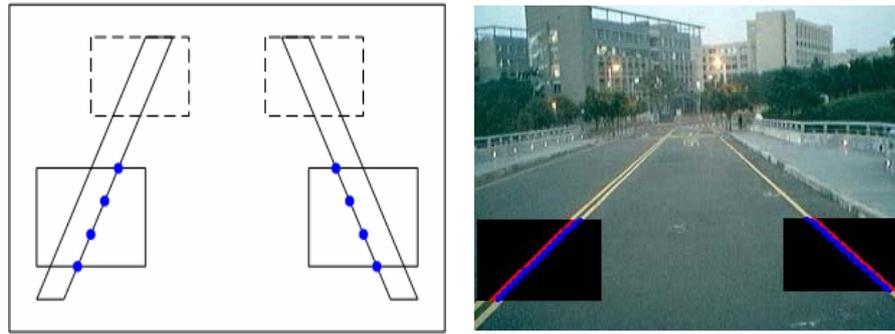


圖 3.3 駕駛資訊選取流程



3-2-1 車道標記點(Lane Marking)的選取

首先，我們關心的只有在動態偵測視窗內所偵測到的車道線且靠近車道內側的部分，只要能從中獲得數點，便足以讓我們建立起實際的車道模型，而前一章的車道線偵測幫助我們準確的擷取出車道點，建立更準確的車道模型。如圖 3.4 所示，雖然在圖 2.17 中已能找出四個動態視窗，但為了減少計算量，在此主要擷取靠近車子的動態視窗內的車道點且每行取一點，如有需要，再另擷取出上方的車道點。



(a) 示意圖

(b) 實際情形

圖 3.4 車道標記點的選取

3-2-2 車道中心線(Center Lane)

擷取完兩旁車道線邊緣上的點後，接著要說明如何在影像中得到車道中心線。對於車輛自動導航而言，其實最主要的車道線是車道中心線，因為最終是希望車子能回到車道中心，找出此線，就能知道目前車子相對於車道中心的狀態。

對於在同一高度而言，車道中心線組成方式如 (3-8) 所：

$$x_{Ck} = (x_{Lk} + x_{Rk}) / 2 \quad (3-8)$$

x_{Lk} ， x_{Rk} 分別表示左、右車道線在 k 高度的水平位置。

但從車道線偵測結果來看，有時兩邊的車道線未必同長，也就是動態偵測視窗的大小必非一樣，所以左(或右)邊的車道點可能在右(或左)邊的車道線找不到同高度的點，所以在此利用一個簡單的方式找出車道中心線。假設

左車道點為 $(x_{Li}, y_{Li}), i=1 \dots n$ ，車道方程式為 $y_L = m_L \times x_L + b_L$

右車道點為 $(x_{Rj}, y_{Rj}), j=1 \dots m$ ，車道方程式為 $y_R = m_R \times x_R + b_R$

m_L 、 m_R ：左、右車道線斜率， b_L 、 b_R ：左、右車道線截距

中線車道點獲得如下:

第一組為 $(x_{CLi}, y_{CLi}), i=1 \dots n$ ，由左車道點所建立，其中

$$\begin{aligned} y_{CLi} &= y_{Li} \\ x_{CLi} &= \frac{y_{Li} - b_R}{m_R} \end{aligned} \quad (3-9)$$

第二組為 $(x_{CRj}, y_{CRj}), j=1 \dots m$ ，由右車道點所建立，其中

$$\begin{aligned} y_{CRj} &= y_{Rj} \\ x_{CRj} &= \frac{y_{Rj} - b_L}{m_L} \end{aligned} \quad (3-10)$$

若遇到同一高度有兩個點 $y_{CLi} = y_{CRj}$ ，則取平均來當做中線車道點 $\frac{(x_{CLi} + x_{CRj})}{2}$ ，所以中線車道點最多有 $(m+n)$ 個點，結果如圖 3.5 中的黑線所示。

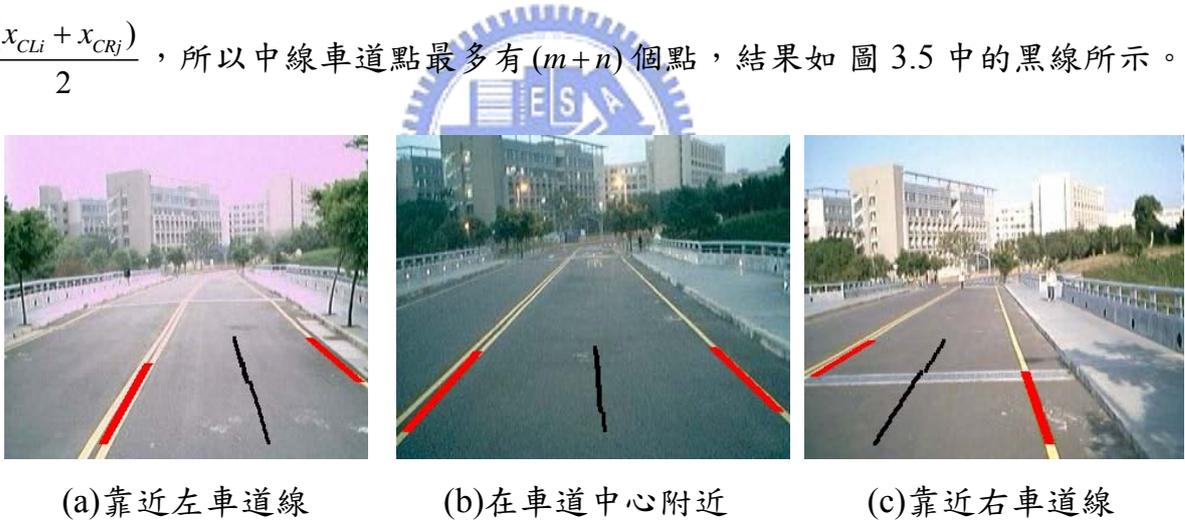


圖 3.5 直線道路上不同位置所找出的車道中心線

從圖 3.5 中的結果看來，雖然結果有時候沒有呈現出很理想的直線，有些許的誤差，但只要誤差不大的話還是足以利用此線作為導航系統的依據。

對於在彎道上時的右車道線，要直接找出車道中心線並不容易，但仍可找出車道中心線。其做法為將影像上所選取到的右車道點，經過座標轉換轉為 3D 的

座標點，之後，再將這些座標點扣掉一半的路寬，最後，再經過一逆座標轉換(3D→2D)，就可找出車道中心線，其結果如圖 3.6。圖 3.6 顯示在彎道時用這樣的方式所得到的資訊無太大的誤差，所以為了避免麻煩，多餘的計算量，在彎道上就不必再刻意先從影像上找出車道中心線，直接將 3D 的座標點扣掉一半的路寬來當作 3D 的車道中心點。



圖 3.6 彎道上利用右車道線找出的車道中心線

將圖 3.5 與圖 3.6 做個比較，雖然圖 3.6 的方式也可以得到導航資訊，但實際上用圖 3.5 的方式會得到較準確的導航資訊。理由為假如現在分別有一組左右車道線在世界座標中的車道點，而所要的車道中心點，對右車道點來說，是必須扣掉路寬，對左車道點來說，是必須加上路寬，然而，這些組合起來的車道中心點容易顯的較不匹配(亦即由左車道線建立的車道中心點，不容易對稱於左右車道線；反之亦然)，得到的導航資訊也會較不準確。

3-2-3 座標系統的轉換

在座標轉換方面，主要將 2D 上的車道中心點轉為 3D 的車道點，提供車子與實際道路間的關係。在此直接利用 IPM 的座標轉換公式，如 (2-1)，(2-2) 式所示，參數方面如前一章所述，因此不必另外測量 CCD 的狀態，因為一開始在調整 IPM 的參數時就已經知道 CCD 的狀態。

$$\text{其中 } Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1 \end{pmatrix}, \quad C = \begin{pmatrix} C_2 \\ C_1 \\ C_0 \end{pmatrix}$$

最後，由 (3-15) 式可求出方程式的最佳係數，得到實際的車道模型。

$$C = (X^T X)^{-1} (X^T Y) \quad (3-15)$$

3-2-5 建立參考點資訊

當獲得實際的車道模型後，接著就是要知道怎樣利用所建立的車道模型來取得想要的資訊。如圖 3.2 所示，在任何前視距離處定為參考點，擷取出未來在設計前輪轉動角度的資訊，例如：前視偏移距離(lateral offset)，偏轉角度(yaw angle)，實際道路的曲率(road curvature)等等。



前視偏移距離:

$$\text{LateralOffset}(L) = y_L = y(L) = C_2 L^2 + C_1 L + C_0 \quad (3-16)$$

其中當 $y_L > 0$ ，表示參考點在車子縱軸方向的右方；反之，當 $y_L < 0$ ，表示參考點在車子縱軸方向的左方。

偏轉角度:

$$\text{Yawangle}(L) = \varphi_L = y'(L) = 2C_2 L + C_1 \quad (3-17)$$

其中當車子順時針方向轉動時， φ_L 會往正值方向變動；反之，當車子逆時針方向轉動時， φ_L 會往負值方向變動。

道路曲率:

由微積分對曲率的定義為 $\frac{f''(x)}{(1+(f'(x))^2)^{3/2}}$

故在參考點處道路的曲率為

$$\text{Curvature}(L) = \rho_L = \frac{y''(L)}{(1+(y'(L))^2)^{3/2}} = \frac{2C_2}{(1+(2C_2L+C_1)^2)^{3/2}} \quad (3-18)$$

若在直線道路上，曲率半徑 $1/\rho_L$ 趨近於無限大，所以此數值理應接近於零。

3-3 導航資訊測試

本小節主要呈現利用上述的概念對影片來進行測試，包含了直線與彎道，藉由測試的結果觀察前述的方法中是否有錯誤，並加以改進，另外，由於影片是屬於人為駕駛，故可以觀察人為駕駛於不同道路的情況。影片路段如同實驗路段，可參考 4-1-2。



3-3-1 近區間影像估測

近區間影像估測方式如圖 3.7 所示，利用靠近車子的車道線對駕駛資訊做一分析，並用二次曲線逼近導航車道線。在直線道路上，以車道中心線為主，而彎道上，以右車道線為主，但為了得到在車道中心的資訊，多加入路寬資訊來幫助估測。另外，均取前視距離 $L=10\text{m}$ 的參考點來做為估測結果。



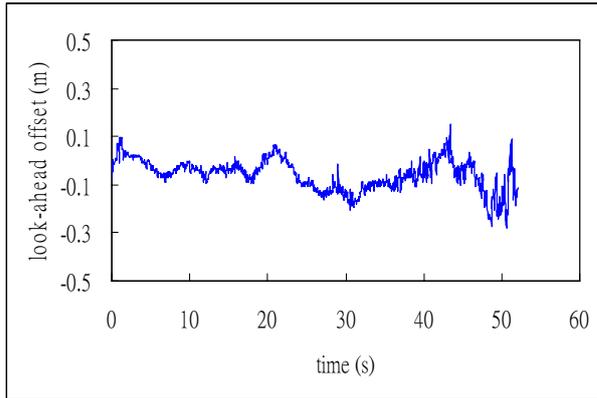
(a) 直線道路

(b) 彎道

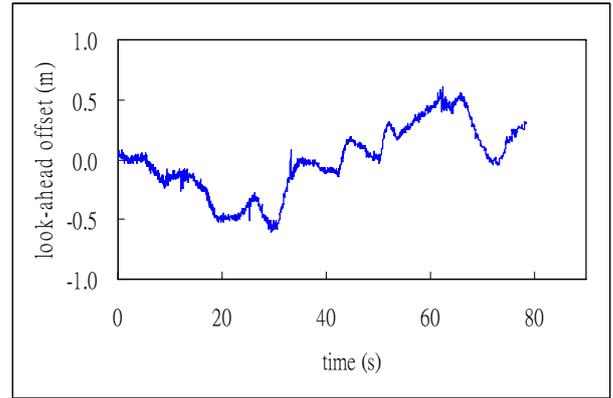
圖 3.7 近區間影像估測方式

(1) 前視偏移距離

前視偏移距離估測結果如圖 3.8。圖 3.8(a)是在直線道路上測試結果，由於車子是直線行進，所以結果上並無明顯的遞增或遞減的現象。但從圖 3.8(b)彎道上變化趨勢來看，當道路是屬於左彎時，車子尚無明顯的轉向時，從影像所擷取的參考點會隨道路變化而往左變動，所以前視偏移距離會遞減，與圖形一開始的趨勢符合。之後，當車子有轉向時，前視偏移距離會遞增，但從圖形看來，轉的幅度跟不上彎道變化的幅度，約 20s-30s，所以車子有稍微偏離的情形。之後，當車輛有轉向及彎道幅度變緩接連直線時，前視偏移距離會遞增，有較明顯的變化，約在 30s 後。最後，行駛在直線時，理應所有資訊都保持在零，但從圖形看來，車子並未保持平行，乃駕駛人駕駛造成的誤差；從圖 3.8(b)中外觀上來看，圖中會有上下起伏的情形乃車輛轉向對彎道反應所造成，屬正常現象。另外值得注意的就是圖 3.8 所量測到的資訊含有高頻的成份。大致上看來，圖 3.8 符合駕駛內容。



(a) 直線道路

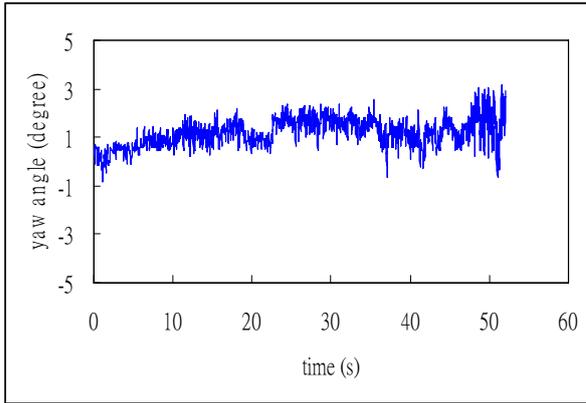


(b) 彎道

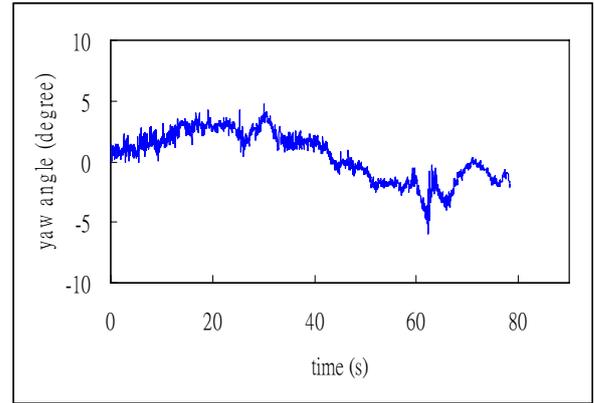
圖 3.8 前視偏移距離估測結果

(2) 偏轉角度

偏轉角度估測結果如圖 3.9。圖 3.9(a)由於車子是直線行進，所以結果上同樣不會有明顯的遞增或遞減的現象。從圖 3.9(b)來看，當道路是屬於左彎時，車子尚無明顯的轉向時，以俯視的角度來看車道線，車道線是逆時針方向轉動，根據偏轉角度方向上的定義(3-2-5 節)，其轉動方向如同車輛順時針轉動，所以偏轉角度遞增變動的現象，與圖形一開始的趨勢符合。之後，當彎道幅度變緩接連直線時，車道線是順時針變動，偏轉角度又會慢慢的遞減。大致上看來，圖 3.9 符合駕駛內容，另外值得注意的就是圖 3.9 所量測到的資訊同樣含有高頻的成份。從圖 3.8 與圖 3.9 結果來看，可發現前視偏移距離與偏轉角度的變動方向是相反的。



(a) 直線道路

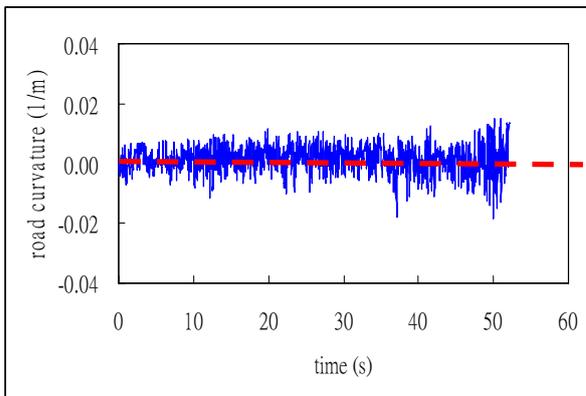


(b) 彎道

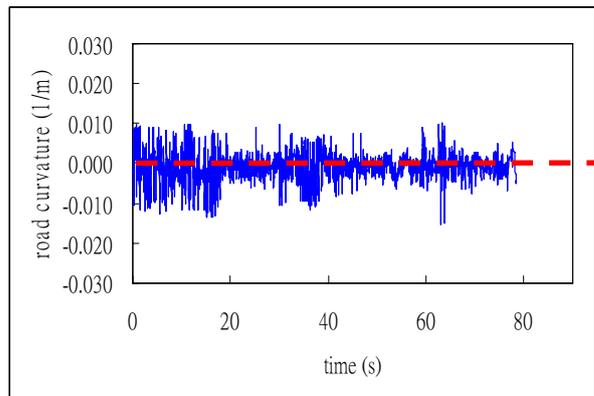
圖 3.9 偏轉角度估測結果

(3) 道路曲率

道路曲率估測結果如圖 3.10，不管是直線或彎道，從圖形均可看出在零附近上下變動。對直線來說，理論上應該等於零，雖然在零附近變動，但得到的數值以曲率的觀點來看是屬於偏高的，而對彎道來說，其結果不但數值偏高，更應有一趨勢存在才對，所以估測出的曲率資訊並不是很正確，最主要的原因就是在影像上是用單直線來做為導航資訊，非用曲線來描述，而實際上是二次曲線來逼近，所以造成多項式的二次係數錯誤，所以若要考量到曲率資訊，則應使用別的方式會比較好。



(a) 直線道路



(b) 彎道

圖 3.10 道路曲率估測結果

3-3-2 近區間結合遠區間影像之估測

為了完整的得到導航資訊，在此要述說如何得到道路曲率，其方式為多加入一遠方的車道線來估測道路曲率，如圖 3.11、圖 3.13，理由為多加入的車道線與原本的车道線搭配起來近似於在影像上用二次曲線去找出車道線。在前視距離方面同樣取 $L=10m$ 情形下，直線與彎道曲率估測結果分別如圖 3.12，圖 3.14 所示，直線道路曲率幾近於零，符合直線道路的特性，而在彎道上也顯示出一開始在彎道上該有的曲率，從圖 3.12、圖 3.14 估測結果可發現利用這樣的方式已能大致上把道路的曲率的趨勢估測出來，也符合道的變化情形，但唯一的缺點就是所量測到的資訊同樣含有些微的高頻成份。另外值得注意的就是經由測試後的結果，發現加入遠區間的影像對於前視偏移距離及偏轉角度的估測影響不大，最受影響的其實只有道路曲率。

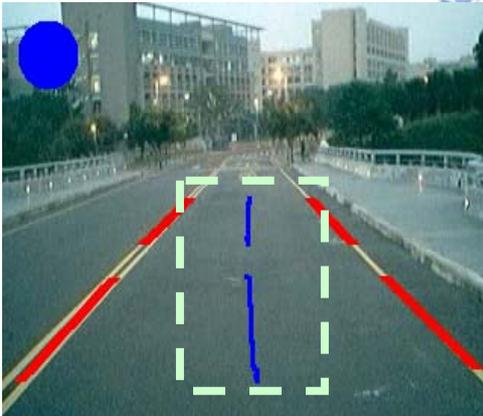


圖 3.11 直線道路估測道路曲率方式

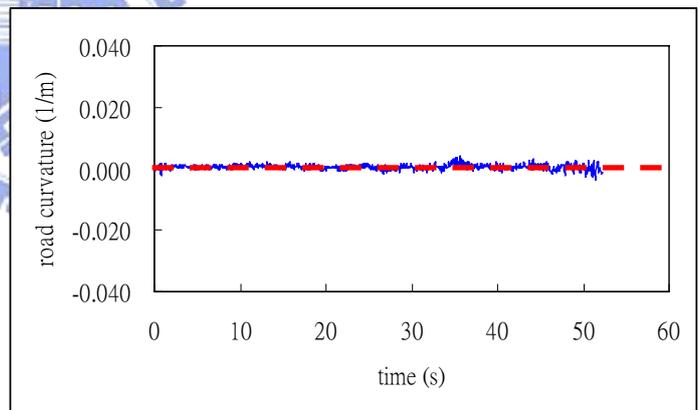


圖 3.12 直線道路曲率估測結果

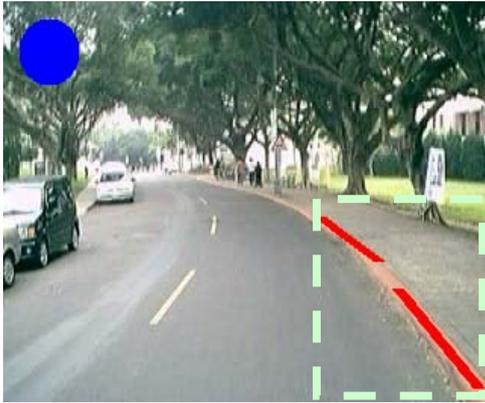


圖 3.13 彎道估測道路曲率方式
(因為左車道不連續)

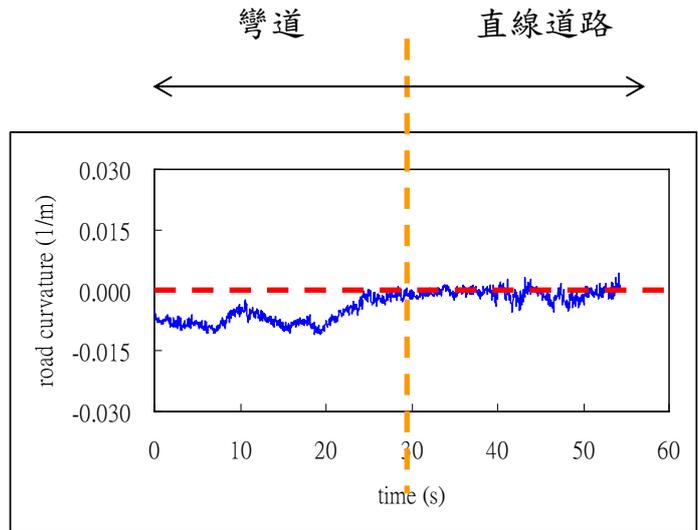
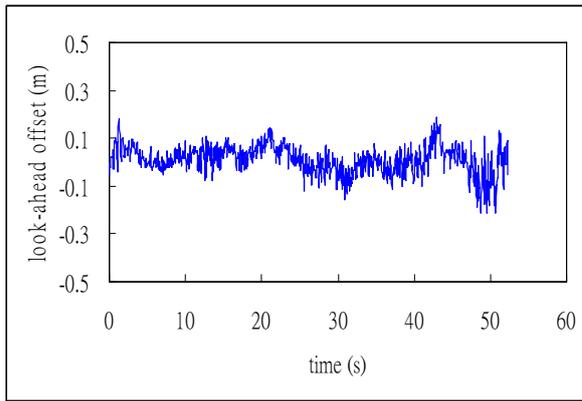


圖 3.14 彎道曲率估測結果

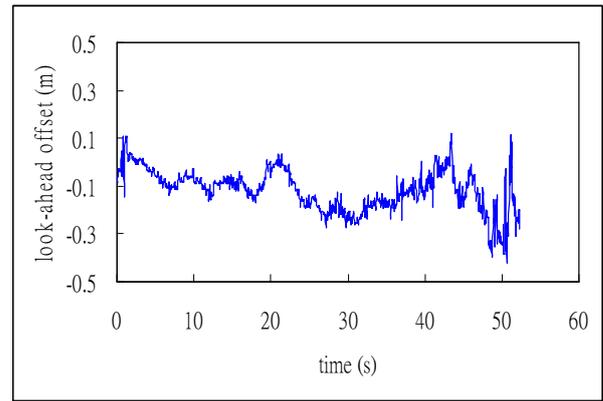
3-3-3 討論

1、前視距離的選擇

圖 3.8、圖 3.9、圖 3.12 與 圖 3.14 為當 $L=10\text{m}$ 所量測到的導航資訊，選擇 $L=10\text{m}$ 的原因主要是靠近車子的車道點，轉換後約在車子前方 7-13 公尺處(CCD 高度約 200cm)，所以取中間值來測試，而在此要探討不同的前視距離對前視偏移距離的影響。圖 3.15 為不同前視距離處得到的前視偏移距離結果。從 圖 3.15 與 圖 3.8(a)比較可知，縮短前視距離 $L=7\text{m}$ 後所受到的資訊變動情形比 $L=10\text{m}$ 來的嚴重，但未必會對系統穩定度造成影響，只是影響系統的暫態響應方面；而當增加前視距離 $L=13\text{m}$ 後時，結果與 $L=10\text{m}$ 並無太大的差異，故以資訊變動的程度來看，在此先選擇 $L=10\text{m}$ 。另外要注意的是，適當的前視距離雖然可提供較好的穩定度當參數變動時，但是，對於有較大曲率的道路，前視距離應考慮是否縮短以避免有過多的偏移距離產生，而本研究在彎道實驗前視距離方面先選 $L=10\text{m}$ 測試之。



(a) $L=7\text{m}$



(b) $L=13\text{m}$

圖 3.15 車輛行駛直線道路在不同前視距離處所量測到的前視偏移距離

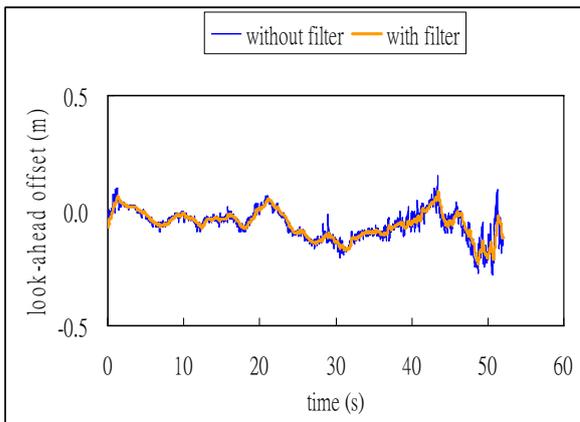
2、綜合近區間及結合遠區間估測的結果，可發現前視偏移距離和偏轉角度資訊和近區間的影像有較直接的關係，而曲率的資訊須依賴著遠方的車道線，所以若要得到完整的資訊，在車道線方面必須使用近區間結合遠區間的車道線的方式會比較好，即使在直線上得到的曲率也較使用單直線來的準確(比較圖 3.10(a)與圖 3.12)。另外，從估測結果均可發現得到的資訊夾雜著高頻雜訊，故選擇一低通濾波器來改善。

3、低通濾波器(Low-Pass Filter, LPF)

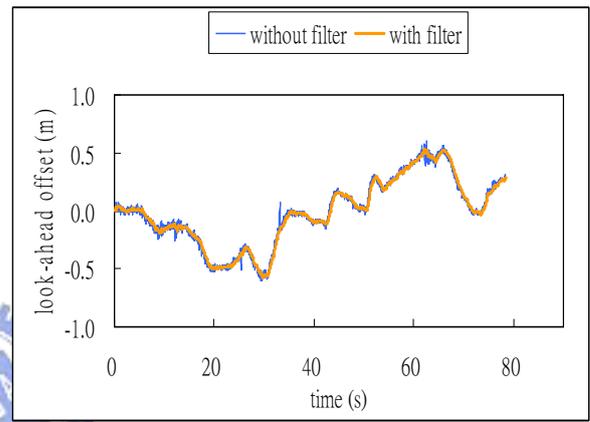
從以上估測結果看來，所量到的資訊均含有高頻成份，而資訊變動劇烈的原因推估為車子所行駛的路面不是呈現很理想的平坦路面，導致 CCD 的狀態改變(例如:CCD 的高度)，使得 2D->3D 轉換中發生了誤差。所以理想上而言，若要獲得更精確的轉換，必須隨時知道會影響轉換中的相關參數，如 CCD 的高度，道路坡度等等，並正確的求得，然而這並非容易的事，故在本系統加入一低通濾波器改善之。

低通濾波器主要目的就是抑制高頻的成分，防止瞬間的變動太快，破壞此時的訊號。本研究選擇一階的低通濾波器，不論直線或彎道，前視偏移距離與偏轉角度截止頻率選擇 $w_c = 3\text{Hz}$ ，道路曲率選擇 $w_c = 1\text{Hz}$ 。由於較關心的

就是前視偏移距離(3-4 節)與道路曲率(4-3-2 節)，故觀察前視偏移距離與道路曲率的變化。圖 3.16 為前視偏移距離有無低通濾波器後的結果比較。直線與彎道曲率濾波後的結果如 圖 3.17(a)(b)所示，圖 3.17(c)(d)(e)也顯示其它估測不同道路(右彎、直線、左彎)的曲率結果，就結果而言，右彎的話，曲率會是一個正值，左彎的話，會是一個負值。可發現加入濾波後大致上仍維持住變化的趨勢，而原本訊號變的較平滑。

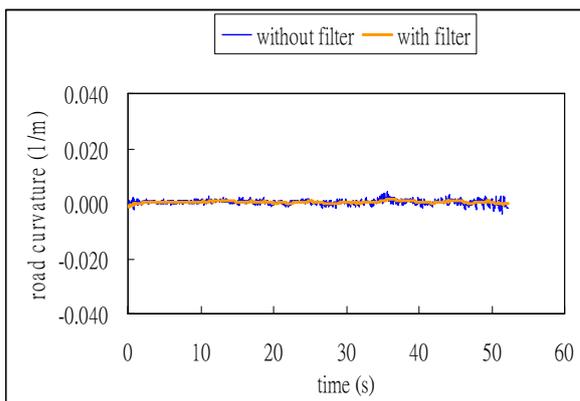


(a) 直線道路

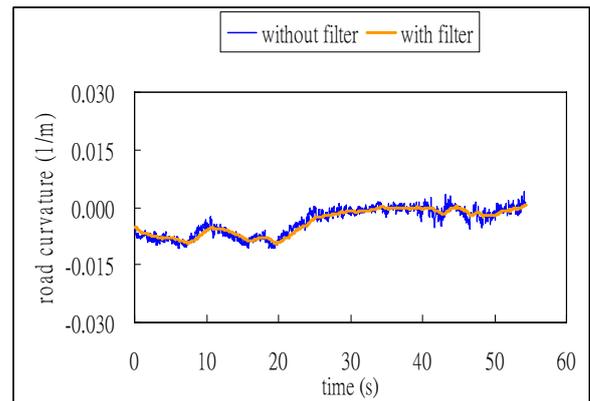


(b) 彎道

圖 3.16 比較前視偏移距離有無濾波結果



(a)比較直線道路曲率有無濾波結果



(b)比較彎道曲率有無濾波結果

右彎道

直線道路

左彎道



(c) 曲率=0.02229 (1/m) (d) 曲率=-0.00078 (1/m) (e) 曲率=-0.00824 (1/m)

圖 3.17 曲率濾波後估測結果

由 3-3-1、3-3-2 看來，大致上已經敘述出如何得到完整的導航資訊，也已無太大的問題存在，接著就是要敘述利用導航資訊來得到前輪轉動角度，對車輛進行操控的動作。

3-4 前輪轉動角度

當獲得參考點資訊後，接著就是在自動導航中的另一個重點：車輛的前輪轉動角度，因為只要能獲得此角度，就能操控車輛的轉動行為。而本小節除了了解如何得到前輪轉動角度外，並進一步的表示在實際上能操控的是車子的方向盤，描述其間的關係，以下將逐一介紹。

3-4-1 控制法則

控制器存在的目的主要將影像資訊轉換成前輪轉動角度 δ_f ，其目標為希望控制車輛能達到前視偏移距離為零，也能達到準確度及穩定度的需求，故如何選擇一適當的控制器成為本系統能否即時地控制車輛自動導航的關鍵。

1、P 控制器

前輪轉動角度的獲得如圖 3.18(a)所示，但首先要考慮的就是導航資訊，

就導航資訊而言，考慮以下情形：

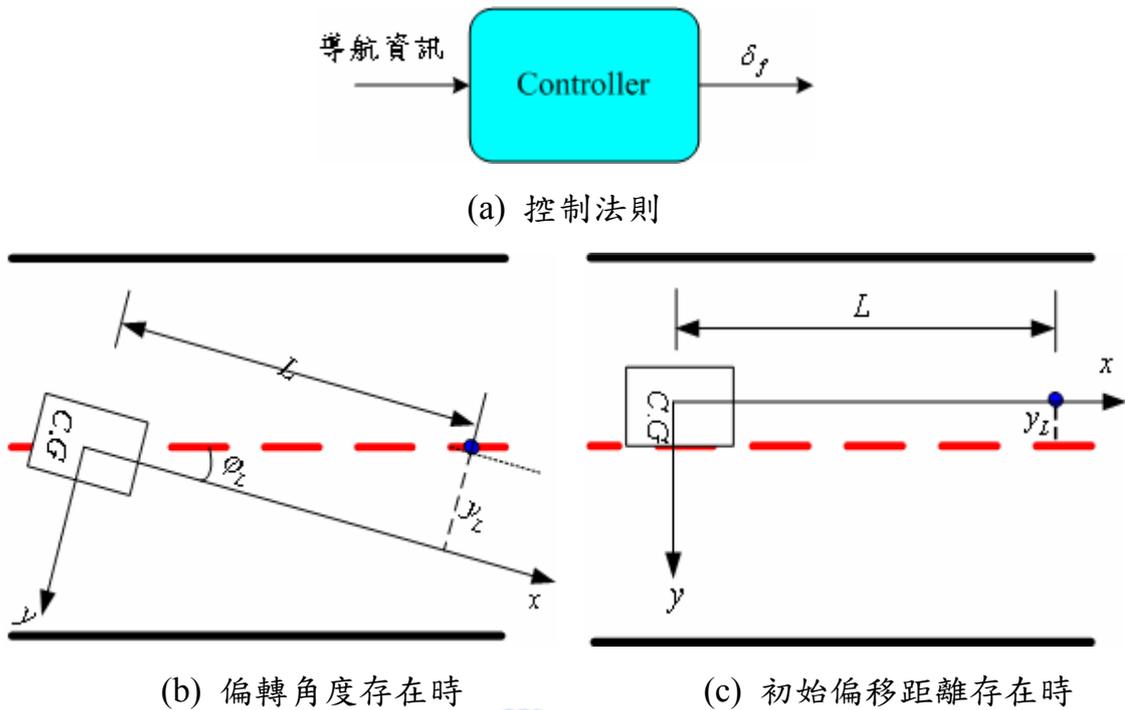


圖 3.18 控制法則與考慮導航資訊情形

- (1) 當車子在車道中心上，假設車子偏轉了 n 度($n > 0$ ，表示在本系統車子順時針轉動 n 度)，雖然從角度資訊 ϕ_L 可知道偏轉情形，但從 y_L 資訊亦可知道車子有偏轉現象，如圖 3.18(b)。
- (2) 當車子離車道中心有些微的差距且平行於車道時，只有從 y_L 資訊可知要如何轉動，如圖 3.18(c)。

綜合以上，以P控制器為主要控制器， y_L 為主要的導航資訊，前輪轉動角度的獲得如 (3-19):

$$\delta_f = K_p \times y_L \quad (3-19)$$

3-4-2 前輪轉動角度誤差

為了使方向盤的轉動達到人類駕駛一般，故所得到的前輪轉動角度會與前一筆資料作相比的動作，從上一個操作點進行調整：

$$\Delta\delta_f(k) = \delta_f(k) - \delta_f(k-1) \quad (3-20)$$

$\delta_f(k)$ ：此刻的前輪轉動角度

$\delta_f(k-1)$ ：前一刻的前輪轉動角度

$\Delta\delta_f(k)$ 是此刻的前輪轉動角度誤差。

3-4-3 相關角度轉換

雖然得到前輪轉動角度，但實際上我們能操控的是方向盤，而方向盤又是由馬達的轉動來帶動，所以最後在本系統要得到的數值是馬達要轉的圈數。故在此介紹得到前輪轉動角度後如何從前輪轉動角度換算到馬達要轉的圈數，如 (3-21) 式。

$$Command = (\Delta\delta_f \times 20) \times 5 / 45 \quad (3-21)$$

其中當 $Command > 0$ ，表示馬達順時針方向轉動；反之，當 $Command < 0$ ，表示馬達逆時針方向轉動。

$$\delta_w(\text{方向盤角度}) = \Delta\delta_f \times 20 (\text{輪胎和方向盤的比}) \quad (3-22)$$

$$\delta m(\text{馬達轉動角度}) = \delta_w \times 5 (\text{方向盤和馬達的比}) \quad (3-23)$$

$$Command (/8:\text{馬達圈數}) = \delta m / 45 (\text{馬達劃分成 8 個刻度})$$

ex: $Command=1 \rightarrow$ 馬達轉動 1/8 圈

$Command=2 \rightarrow$ 馬達轉動 2/8 圈

以此類推.....

第四章 視覺導航車實驗結果

從前兩章來看，已大致上敘述如何從CCD camera影像擷取出道路特徵所使用的方法，並進一步的利用車道線獲得導航系統所需的資訊，最後再設計一控制器搭配以獲得前輪轉動角度來操縱車輛。而本章節主要利用前面所提的方法，帶到實際的直線與彎曲的道路上進行實驗。本實驗用車為電動高爾夫球車，如圖 4.1 所示。而實驗目標如圖 4.2，安全、穩定且準確的在車道上行駛，即使是直線道路或是彎道。



圖 4.1 實驗用電動高爾夫球車

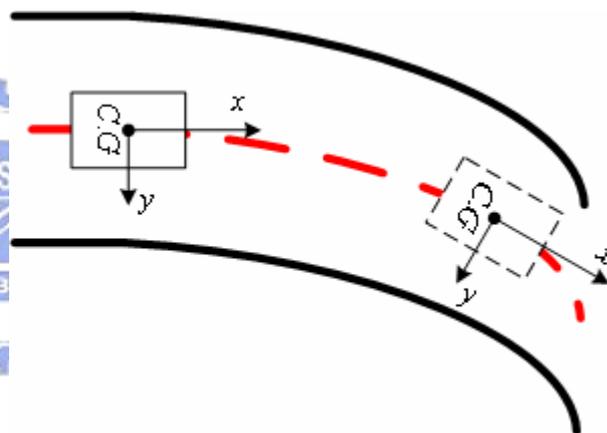


圖 4.2 實驗目標

4-1 視覺導航車實驗介紹

4-1-1 系統設備與使用者介面

就一個視覺導航車而言，因我們必須將其實現在車輛行駛中的狀況下，即時地處理。在即時處理與判斷方面，我們並沒有以硬體的方式來實現，而是以軟體來做處理與判斷，故在本節中概述我們的系統設備及使用者介面。

1、系統設備

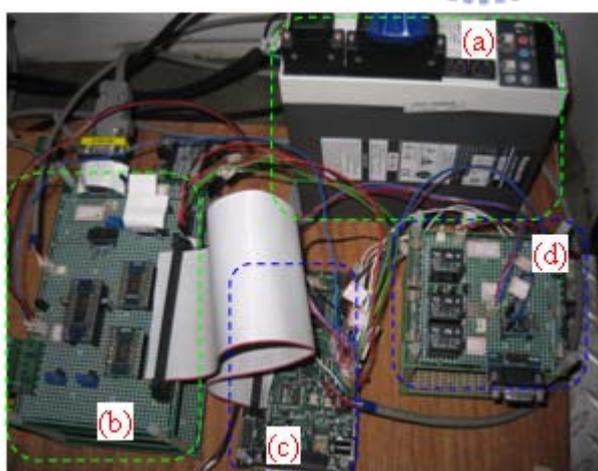
整個系統所需的硬體包括一台USB埠CCD、一部筆記型電腦、一台USB-CAN、二台UPS、及控制方向盤的一些硬體電路。我們所使用的影像擷取設備是一台簡單的視訊會議USB埠CCD，如圖 4.3 所示，將其連接到PC上後，完成PC-based之影像處理系統。而這部CCD其解析度最高為 640×480 ，擷取速度最高可達每秒 30 個張影像(frame)。而UPS(圖 4.4)主要提供硬體電路部份所需的電源。硬體電路功能說明如圖 4.5。



圖 4.3 USB 埠 CCD

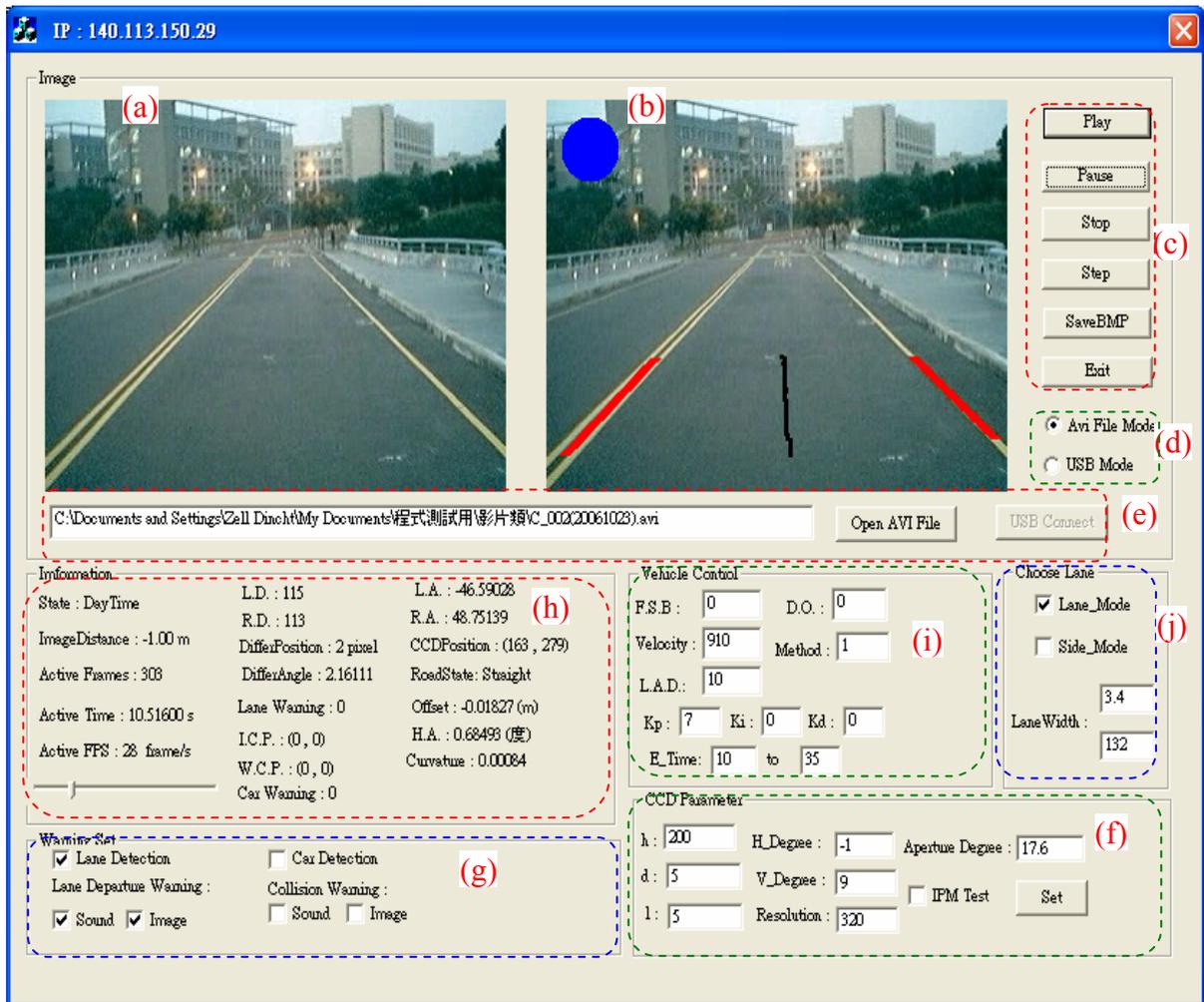


圖 4.4 UPS



- (a)馬達驅動器
- (b)車輛控制相關電路
- (c)DSP2812 控制板
- (d)CAN 電路-負責接收由 PC 傳至 CAN 上的訊息

圖 4.5 硬體架構圖



- (a): 來自 CCD 所接收的或影像檔的影像
 (b): 偵測後的影像
 (c): 相關操作
 (d): 兩種影像選擇模式
 (e): 不同模式的相關操作
 (f): IPM 相關物理參數的輸入
 (g): 警告訊息的開關設定
 (h): 偵測後的相關結果
 (i): 控制車輛操作(含控制器)
 (j): 單雙車道線模式選擇

圖 4.7 影像程式介面

此介面清楚的顯示駕駛人所需的資訊以及可操作的功能，駕駛人經由圖中的(b)顯示目前車前的偵測狀況(車道線被偵測出)，左上角圖示為防偏警告圖像；(g)為影像的警告功能開關，包括車道線和前方車輛偵測功能、圖像警示、聲音警示，可依使用者需要自行調整；因此透過此介面，駕駛人不僅可即時的得知系統的偵測狀況，而且操作方式也非常的簡易，這便是我們系統特別所要求的互動關係。

為了方便設計者進行程式演算法的測試，此介面還保留了某些功能鍵，(a)是原始的輸入影像，可用以與(b)作比對；(c)程式演算法的操作按鈕，有關於執行(Play)、暫停(Pause)...等功能；(d)選擇影像輸入模式，影像檔(avi 檔)或 CCD 畫面；(e)開啟某一影像檔或是連接 CCD；(f)車道線偵測法則(IPM)的相關物理參數的輸入，因為目前還未能做到參數的自動調整，須靠人工輸入；(h)可即時提供設計者有關目前偵測狀況的文字資訊。

4-1-2 實驗路段

本實驗路段如圖 4.8 所示，包含了直線道路與彎曲道路。圖 4.8(a)為直線道路，兩旁為連續的車道線，路寬為 4.25m，而車道線厚度約為 10cm。圖 4.8(b)為左彎曲道路，左車道線有不連續現象，右邊則有紅色標記，實際路寬約為 3.3m，紅色標記厚度約為 10cm，此道路一開始屬於彎道部份，之後是屬於直線道路。



(a) 直線道路

(b) 彎曲道路

圖 4.8 實驗路段

4-1-3 油門值與實際車速關係

由於目前系統無法藉由感測器取得速度資訊，加上實驗地形的因素，容易造成給予同樣的油門值在不同道路呈現不同的速度，所以在此要先做個測試，測試所給予的油門值對應到實際車速的關係。由於一開始車子會有加速的現象，故測試方式為先訂出測試區段，讓車子行駛一段距離後進入此測試區段，記錄行經此區段的時間，對應關係如表 4-1。

表 4-1 油門值與實際速度關係表

	油門值	量測距離(m)	時間(s)	實際車速(m/s)
直線 道路	390	18	8.74	2.06
	520	18	7.41	2.43
	650	18	6.57	2.74
彎曲 道路	910	18	14.89	1.21
	1040	18	11.08	1.62
	1170	18	8.57	2.10

4-1-4 實驗架構圖

圖 4.10 為實驗架構圖，大致上分為三個部份：PC、網路、及車輛控制部份。PC 部份是本研究最主要的地方，其最終目的主要藉由影像的資訊再搭配控制器來得到前輪轉動角度，因為只要能夠得到這個資訊，我們便能操控車輛的行進方向，達到視覺導航的目的。網路部份則負責將PC端所得到的前輪轉動角度傳給DSP，在此選擇USB-CAN當作傳輸的工具。最後，當接收到命令後，馬達便會透過一有彈性的皮帶來帶動車輛的方向盤(圖 4.9)，對車輛進行控制，此彈性皮帶可吸收些微因震動所產生的誤差。



圖 4.9 馬達透過皮帶連結方向盤

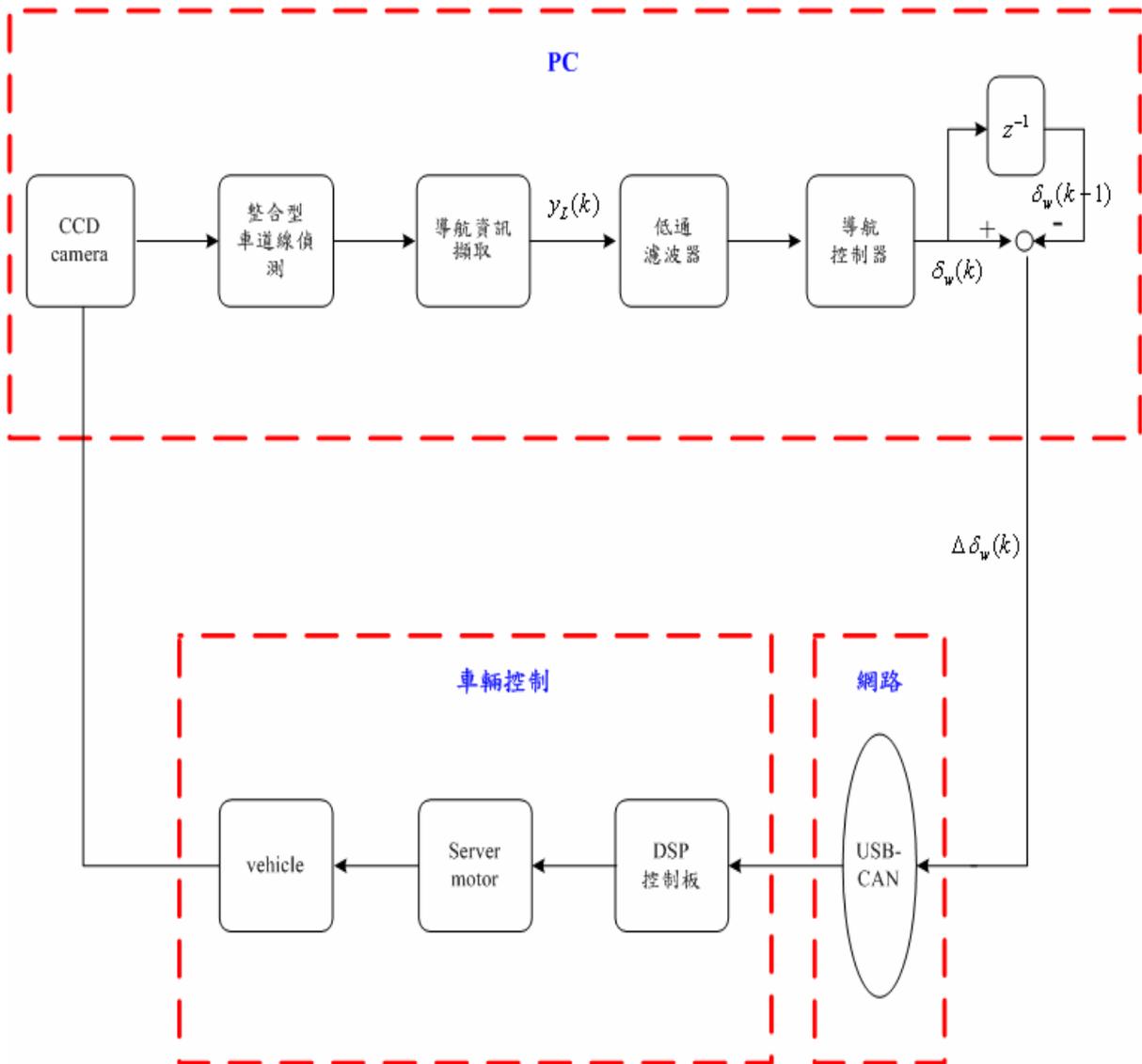


圖 4.10 實驗架構圖

4-1-5 USB-CAN[18]

1、簡介 USB-CAN

USB CAN可以讓PC透過USB介面連接到CAN網路上面，使得PC也變成CAN網路上面的一個節點，透過USB CAN方便讓電腦收集和分析CAN網路上面的訊息，或是透過USB CAN傳達訊息到CAN網路上各節點。本研究採用的USB CAN為周立功單片機發展有限公司所開發出來的USBCANII智能CAN接口卡，USBCANII有兩個CAN通道，可讓使用者同時連到兩個不同的

CAN網路上面，並且附有ZLGVCI接口函式庫，方便使用者搭配不同開發程式開發符合自己需求的產品，以下圖 4.11 為USBCANII外觀和硬體規格：



圖 4.11 USBCANII 外觀

硬體規格：

- PC 接口：USB1.1
- CAN 控制器：PHILIPS SJA1000
- CAN 收發器：PHILIPS PCA82C250
- CAN 網路傳輸速率：5Kbps~1Mbps
- CAN 通訊接口：DB9，符合 DeviceNET 和 CANOpen 標準
- 支援 CAN 網路協定：支援 CAN 2.0B(兼容 CAN2.0A 協定)，符合 ISO/IS 11898
- 最高 Frame 流量：每個通道 5000 Frame/sec
- 供電方式：USB 供電或是外接+9V~+25V，400mA 的電源

2、網路監控程式

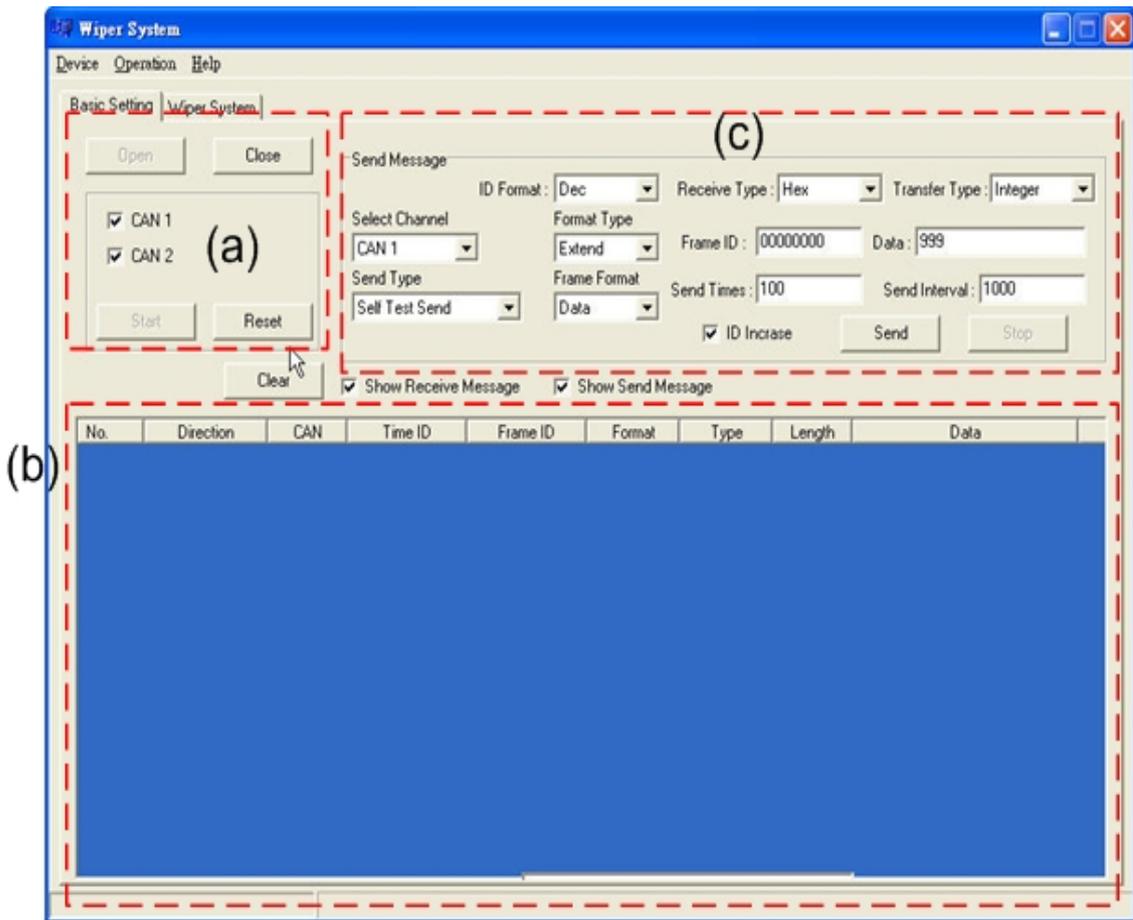


圖 4.12 網路監控程式主畫面

圖 4.12 為網路監控程式的主要畫面，總共分成三個部份：

(a) 控制 USBCAN 節點屬性：

當要開始使用監控程式的時候，需先設定 USBCAN 相關參數好。首先，先點下 Open 按鈕後會跳出參數設定視窗如圖，此畫面可以讓我們設定 ID 遮罩，濾波器形式，工作模式還有傳輸速度，由於本 USBCAN 使用的 CAN Controller 為 PHILIPS SJA-1000，因此參數的詳細設定，可以直接參考 SJA-1000 的說明文件。此 USBCAN 有兩組 CAN 節點，所以有兩組參數可供設定，Device Index 是當 PC 不只接上一台 USBCAN 時，可以讓我們選擇想要控制的裝置。當設定好參數之後，選擇 OK 跳回主畫面，此時再按 Start 按鈕即可開始監聽 CAN 網路上面的訊息。

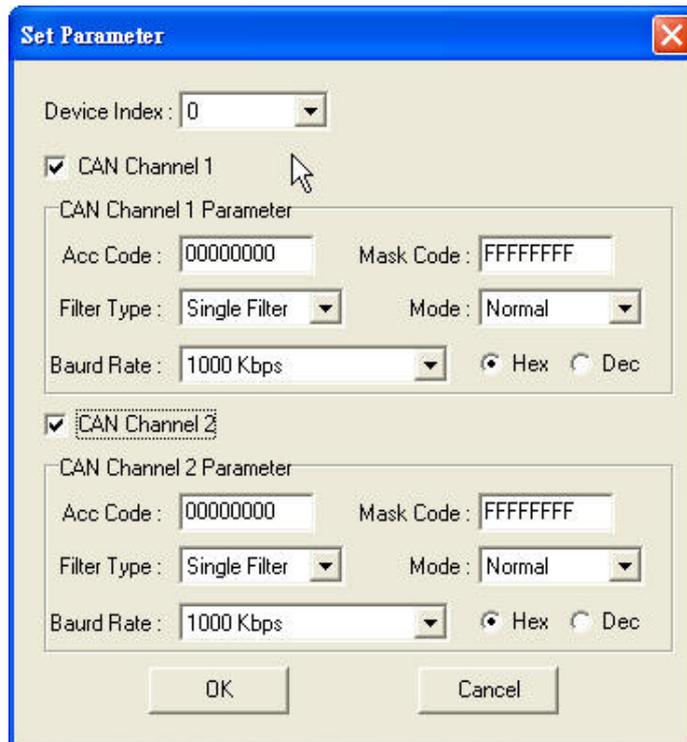


圖 4.13 USB CAN 系統參數設定畫面

(b) 監聽 CAN BUS 訊息的顯示部份：

監控程式將在 CAN Bus 上面監聽到的訊息會在此部份顯示出來，在此我們可以看到訊息的

- (1) direction：傳遞方向，讓我們知道該訊息是由 PC 端接收或是發送。
- (2) CAN：接收或傳送節點，由於 USBCANII 可以讓 PC 擁有兩個 CAN 節點，此部份可以顯示出是哪個節點收發該訊息。
- (3) Time ID：顯示時間 Tag
- (4) Frame ID：顯示該訊息的識別碼，此 ID 是將接收此訊息的識別 ID，可以藉由 Frame ID 來斷該訊息的去處。
- (5) Format：顯示該訊息是資料欄框 (Data Frame) 或是遙控欄框 (Remote Frame)

- (6) Type : 資料欄框的訊息格式，該看訊息是 Extend 或是 Standard 的格式
- (7) Length : 該訊息所帶的資料長度
- (8) Data : 該訊息所攜帶的資料內容，可搭配(c)部份的 Receive Type 來顯示該資料對應的格式。

(c) 傳送訊息部份：

傳送部份可以讓 PC 節點不只扮演監聽的角色，也可讓 PC 主動發出訊息，藉由 PC 發出訊息來看對應的節點是否有對該訊息做出正常的反應。在此部份可以設定訊息傳送的種類，由哪個節點發出，等相關設定。同時也可以指定該節點，每隔多久時間所送多少筆訊息。



3、USB-CAN 傳輸測試

介紹完程式各部份之後，我們來看看實際使用的狀況。圖 4.14(a)為PC端傳送命令的傳輸畫面，檢查是否真的有把駕駛命令傳送出去。此外，觀察傳輸的過程，發現並無delay的現象存在。圖 4.14 中顯示出Data經由第二個節點接收到PC端所傳的命令，該命令的ID為 1，長度為 2Bytes。命令包含了兩個部份:(1)前 13bits代表馬達要轉的量 (2)後 3bits代表方向盤順/逆時針轉動，方向上的定義為:+(順時針) → 001；-(逆時針) → 010。

No.	Direction	CAN	Time ID	Frame ID	Format	Type	Length	Data
1	Receive	2	01:27.834.5	1	Data	Extend	2	10
2	Receive	2	01:27.908.4	1	Data	Extend	2	1

(a) PC 傳送訊息至 USB CAN 畫面擷取

No	PC 端的命令 (十進制)	PC 端的命令 (二進制)		Data	意義
		馬達轉動的量 (前 13 bits)	馬達轉動的方向 (後 3 bits)		
1	-1	00000000000001	010	10	馬達逆時針轉動 1/8 圈
2	+0	00000000000000	001	1	馬達順時針但不轉動

(b) 訊息說明

圖 4.14 傳輸畫面擷取及說明

4-1-6 DSP 端的工作情形

1、工作原理

當 PC 端送出命令時，會透過 USB-CAN 傳送至 CAN 的節點上，之後由負責接收/傳送 CAN 節點上的訊息的電路抓取訊息，最後在經由此電路傳到 DSP。

DSP 端的控制器部分利用很簡單的 PI 控制器，當訊息到達時，經過控制器後的電壓輸出，會對應到相對的電壓，使馬達轉動。

$$P_error(k) = P_command(k) - P_now$$

$$V_command(k) = V_command(k-1) + (K_p + K_i) \times P_error(k) + (-K_p) \times P_error(k-1)$$

$V_command(k)$: 電壓輸出

PI 控制器的參數: $K_p = 0.001$, $K_i = 0.00000005$

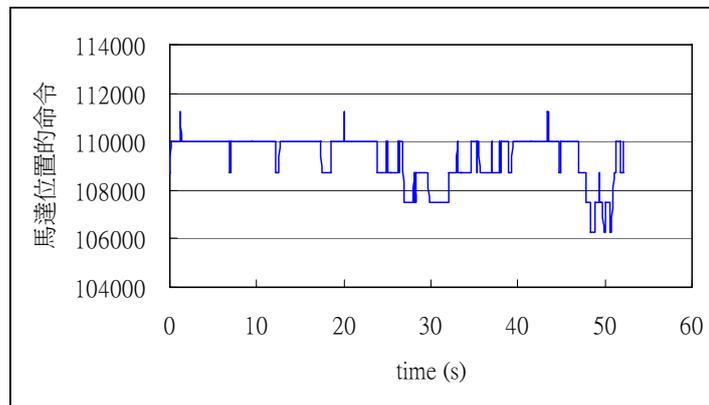
2、測試馬達的轉動位置

此目的除了檢查在 DSP 端的馬達位置是否有轉到最後 PC 端所給的命令，以確保 PC 端給的命令馬達皆會工作正常，所以 PC 端紀錄的馬達位置理應和 DSP 端所記錄的位置的圖形相似；另外，也可看得到命令是否有問

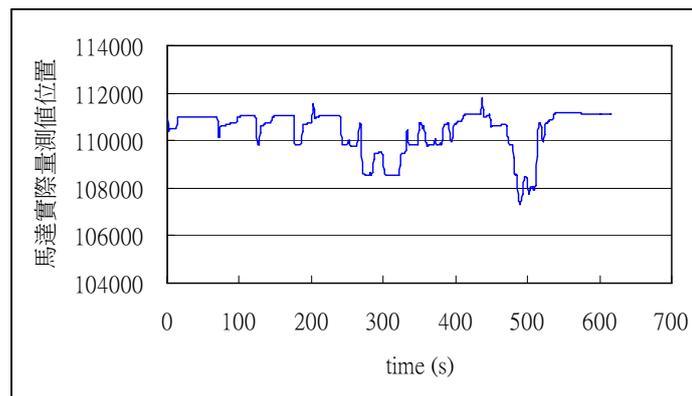
題，是否會有應該要左轉卻發生右轉的現象發生或是會以某個基準值轉動。

控制器方面，首先先選擇較基本的 P 控制器來進行測試，其中 $K_p = 7$ 。

結果如圖 4.15 所示，基準值設在 110000。而縱軸若是小於基準值的話代表方向盤逆時針轉動，大於基準值的話代表順時針轉動，而馬達每順/逆時間轉動一圈，其數值會加/減 1250。從圖 4.15(a)與圖 3.16(a)看來，當圖 3.16(a)中發生偏移距離是負/正的時候，圖 4.15(a)馬達所呈現的轉動方向也是逆/順，所以沒有太大的問題。圖 4.15(a)與圖 4.15(b)雖有些微的差距，但在實際上影響很小，而從整體變化趨勢來看，兩者的變化趨勢是相同的，故 DSP 的控制器方面也無問題。



(a) 馬達位置的命令



(b) 馬達實際量測值位置

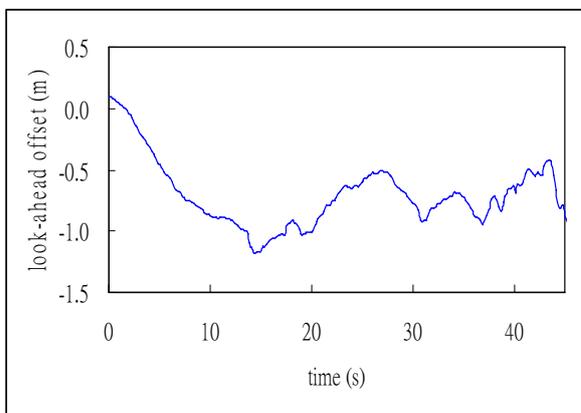
圖 4.15 馬達位置的命令與實際量測

4-2 直線道路實驗結果與分析

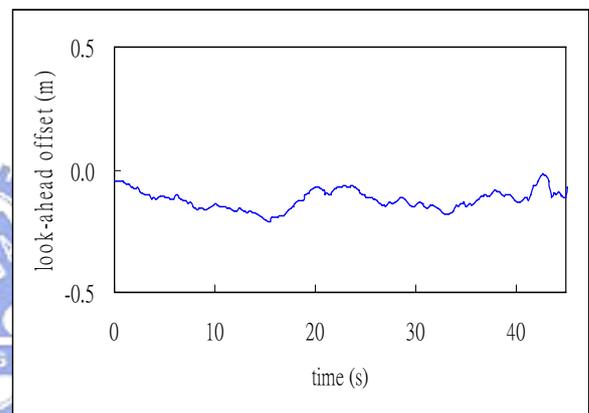
本小節主要呈現在直線道路實車測試的結果，由於未使用到曲率資訊，所以車道線方面取靠近車子端的車道中心線來當作導航的依據，並從中擷取所需的導航資訊:前視偏移距離，其中，前視距離方面不論油門值=390 或油門值=650，均選擇 $L=10\text{m}$ 。

4-2-1 P 控制器偏移距離實驗結果

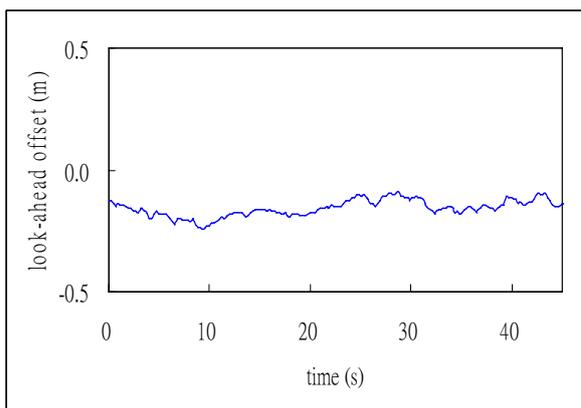
(1)油門值=390 (2m/s)



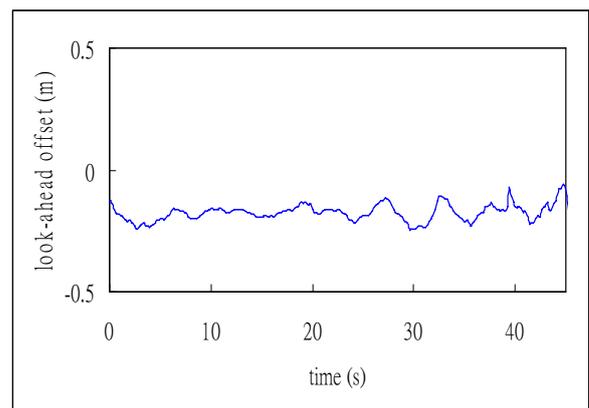
(a) $K_p=3$



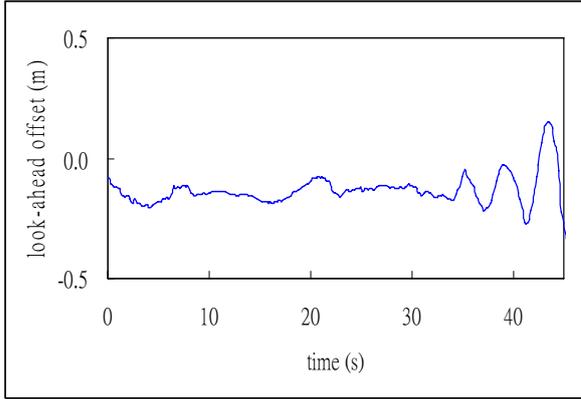
(b) $K_p=5$



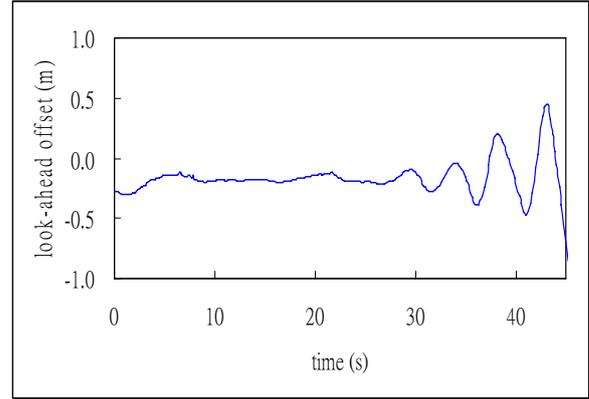
(c) $K_p=7$



(d) $K_p=9$



(e) $K_p=11$



(f) $K_p=13$

圖 4.16 車輛於油門值=390 行駛直線道路實驗結果

圖 4.16 為當油門值=390 時，使用P控制器在直線道路上量測到車子的側向距離。從圖形可明顯的看出，當 K_p 值小時，主要的缺點就是有較大的前視偏移距離誤差，容易偏離車道，如圖 4.16(a)-(b)；當 K_p 值慢慢增大時，有較小的前視偏移距離誤差，但明顯的會開始有不穩定的情形，如圖 4.16(e)-(f)，此結果相當符合P控制器的特性。最後，可以發現要使車子有較好的追蹤表現，從準確度及穩定度的觀點來看， K_p 較適合在 5~11，如圖 4.16(c)-(d)。圖 4.17 顯示車子於 $K_p=7$ 行駛直線道路實際駕駛情形。



(a) 車輛行駛中

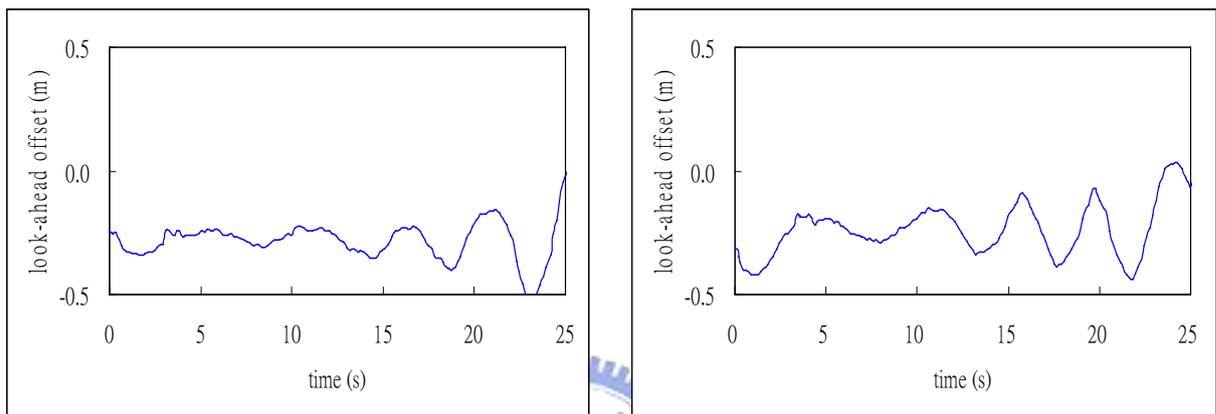


(b) 進入末段道路後

圖 4.17 顯示車輛於 $K_p=7$ 行駛直線道路實際駕駛情形

(2)油門值=650 (2.7m/s)

圖 4.16 為當油門值=390 時的駕駛情形，而目前影像的處理速度約 4-5 frames/s，所以在此要觀察不同的油門值(車速)對系統所產生的影響。圖 4.18 為當油門值=650 時，可看出圖 4.18 中，車子容易出現不穩定的情形。由圖 4.16(c)(d)與圖 4.18 比較可知，當增加車速後，由於相鄰的圖片間有比較大的差距，改變的幅度也會變大，造成本導航系統的穩定性降低。



(a) $K_p=7$

(b) $K_p=9$

圖 4.18 車輛於油門值=650 時行駛直線道路實驗結果

4-3 彎曲道路實驗結果與分析

本小節主要呈現在彎道實車測試的結果，觀察結果並加以分析與改進。實驗車速均保持在油門值=910(約 1.2m/s)。

4-3-1 P 控制器偏移距離實驗結果

車道線方面主要以右車道線為主，如圖 3.7(b)，導航資訊方面取前視偏移距離來進行測試。

(1) 前視距離 $L=10m$

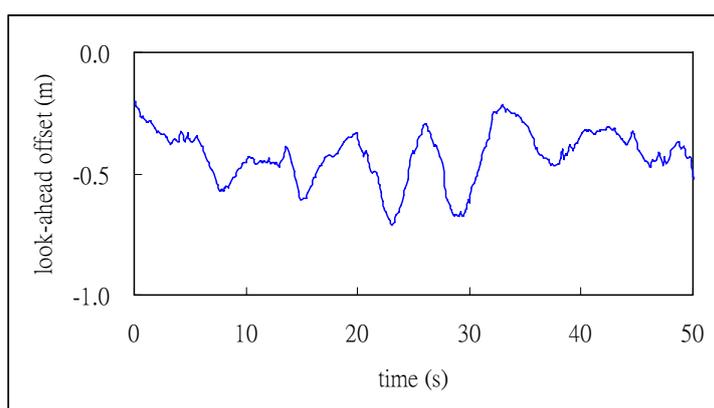
圖 4.19 顯示原系統在 $K_p=7$ ， $L=10m$ 下，使用 P 控制器行駛彎道的結果。初

步從外觀分析結果，有較大的偏移距離產生及上下變動的現象，而此現象即使調整 K_p 大小也會存在同樣的問題，如圖 4.20。接著就要逐一分析結果。

- ◆ 由圖 4.19(a)與圖 4.20(a)結果來看，會有上下起伏的現象乃系統對彎道變化的反應所造成。
- ◆ 車子行駛彎道有較大的偏移距離產生

根據圖 4.19(b)與圖 4.20(b)圖片內容可觀察出在實際上並無明顯的偏離車道，車子幾乎是行走在道路中心上，但從前視偏移距離結果來看卻顯示出有較大的偏移距離產生，與實際駕駛情形不相符，會有此現象乃因所選的前視距離過長所導致。因為在彎道上，過長的前視距離容易產生有過多的偏移距離，而產生的偏移距離卻足夠對彎道曲率做出部份補償，也就是讓車子沒有產生明顯偏離車道的情形。

另一方面，我們希望能將前視偏移距離保持在零附近，所以若要達到此目的，需給系統一補償量，但在實際駕駛上卻會使車輛較靠近左車道(以本實驗道路而言)，並非我們所要的。

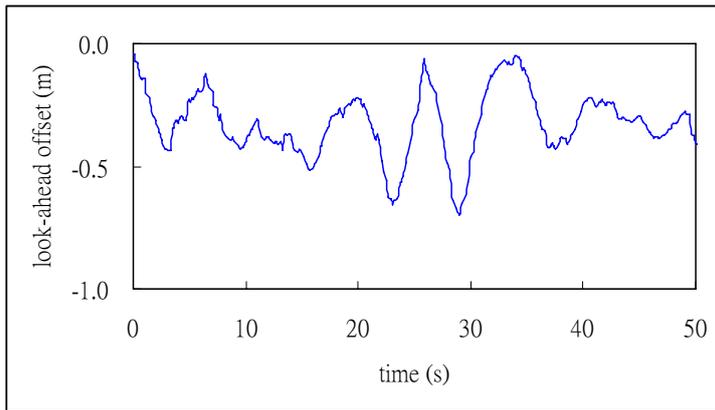


(a) 量測到的前視偏移距離



(b) 顯示行駛彎道中情形

圖 4.19 車輛於 $K_p = 7$ 行駛彎道時實驗結果



(a) 量測到的前視偏移距離

(b) 行駛彎道中

圖 4.20 車輛於 $K_p = 9$ 行駛彎道時實驗結果

(2) 前視距離 $L=5m$

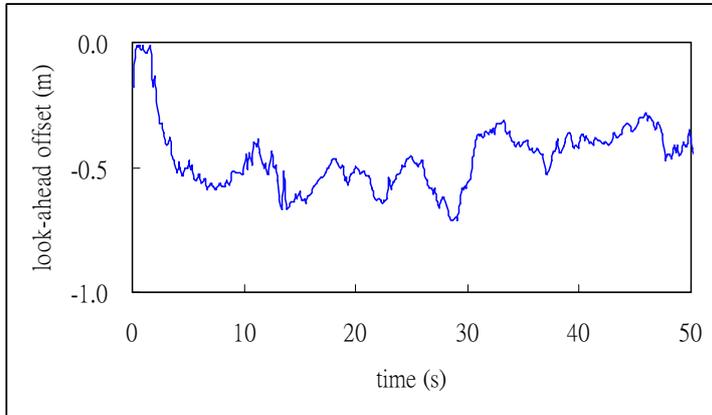
由於 $L=10m$ 的結果容易產生過多的前視偏移距離，造成與實際駕駛不符，故縮短前視距離觀察駕駛情形，如圖 4.21、圖 4.22 為實驗結果，接著就逐一分析結果：

- ◆ 同樣由圖 4.21(a)與圖 4.22(a)結果來看，會有上下起伏的現象乃系統對彎道變化的反應所造成。
- ◆ 車子行駛彎道有較大的偏移距離產生

根據圖 4.21(b)與圖 4.22(b)圖片內容可觀察出車子駕駛上出現了偏離車道的情形，而圖 4.21(a)與圖 4.22(a)也同樣有較大的偏移距離產生，所以此偏移距離可代表車子有無偏離車道的現象，與實際駕駛情形相符，所以若將圖 4.21(a)與圖 4.22(a)前視偏移距離結果控制在零附近，也可保證車輛並無出現偏離車道的情形。偏離車道乃因系統反應跟不上道路的變化，符合如 3-1 節中的動態系統方程式所示，曲率被視為一雜訊，容易破壞系統行走在彎道上的表現。

從圖 4.21(b)與圖 4.22(b)比較而言，圖 4.21(b)的偏離車道情形比圖 4.22(b)來的嚴重，其中原因跟 K_p 值大小有關，當 K_p 值大時，偏離車道

情形較不嚴重。由於希望能將前視偏移距離保持在零附近，但是考量到 K_p 值太大會造成系統不穩定，故不以調整 K_p 方式來改善，改用補償的方式。

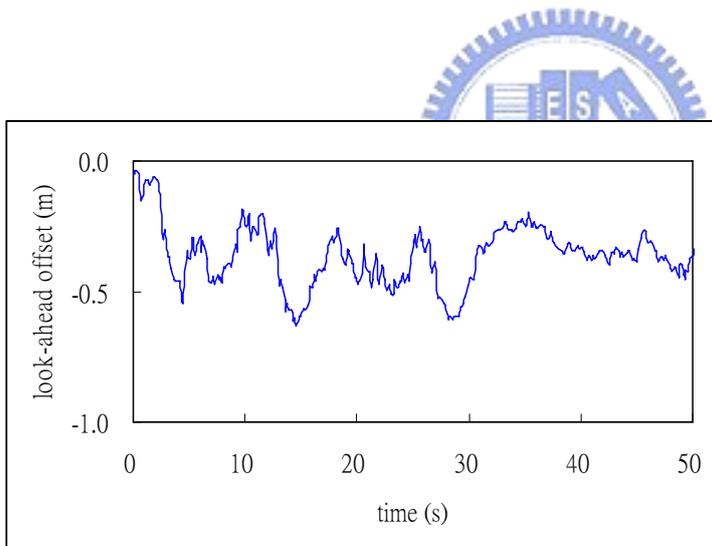


(a) 量測到的前視偏移距離



(b) 行駛彎道中

圖 4.21 車輛於 $K_p=7$ 行駛彎道時實驗結果



(a) 量測到的前視偏移距離



(b) 行駛彎道中

圖 4.22 車輛於 $K_p=9$ 行駛彎道時實驗結果

4-3-2 (P 控制器+彎道偏移量)實驗結果

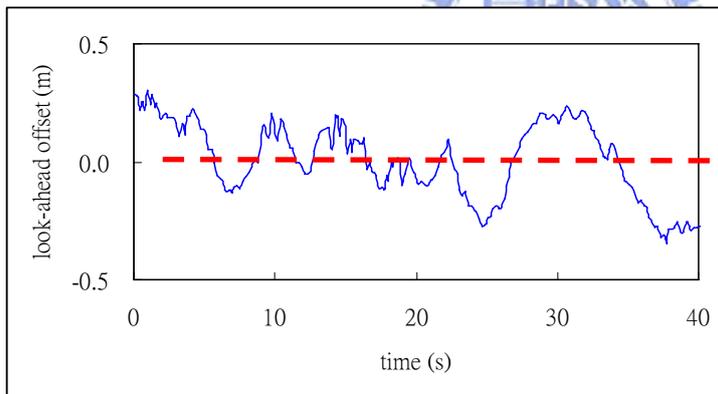
本小節的目的主要針對當車子出現偏離車道時，提出不同方式對車子做出補償，來抵抗曲率對系統的影響，使車子亦能準確的行駛在道路中心。由於當 $L=5m$ 時有較明顯的偏離車道情形，故針對 $L=5m$ 的情況來設計，而設計的準則以達到

前視偏移距離為最小誤差為主，在實際駕駛上亦保持在車道中心。原先的 (3-19) 式修改成 (4-1) 式，其中 δ_{ref} 表示在彎道中所須的補償偏移量，而 $\delta_{feedback} = Kp \times y_L$ 代表原系統的前輪轉動角度。

$$\delta_f = \delta_{feedback} + \delta_{ref} \quad (4-1)$$

1、固定補償偏移量

由圖 4.21(a)與圖 4.22(a)結果可知，車子行駛彎道會有偏離車道的情形，所以為了改善偏離車道情形，在車子行駛彎道時，先給予一固定的補償偏移量，測試是否能改善在彎道行駛的情形。經實驗結果顯示，當車子當給予馬達逆時針轉動 4 至 6 刻度時，時間約在 30 秒內，的確改善了車子行駛彎道偏離車道的情形及較原本不加補償量的系統產生較少的前視偏移距離。圖 4.23 顯示給予馬達逆時針轉動 5 個刻度的實驗結果。



(a) 量測到的前視偏移距離



(b) 行駛彎道中

圖 4.23 車輛使用(P+固定補償偏移量)行駛彎道實驗結果

2、模糊偏移補償量產生器

在此考慮使用較彈性的方法來抵抗曲率對系統的影像，主要原因就是當在彎道上可能存在非固定曲率的情形，避免有補償過多或過少的現象，由於曲率出現破壞了原系統，所以選擇以曲率為輸入，利用模糊推論的方式來設計偏移補償量，而根據經驗，在行經彎道時需給予馬達多轉動 4 至 6 個刻度，

故觀察單只有曲率資訊時，馬達是否也逆時針轉動 4 至 6 個刻度。接著說明如何用曲率設計出彎道偏移量產生器。

(1) 模糊推論概述：

模糊推論系統流程如圖 4.24，主要包含一些步驟：模糊化 (fuzzification)、推論引擎(inference engine)、規則庫(fuzzy rule base)、解模糊化(defuzzification)。

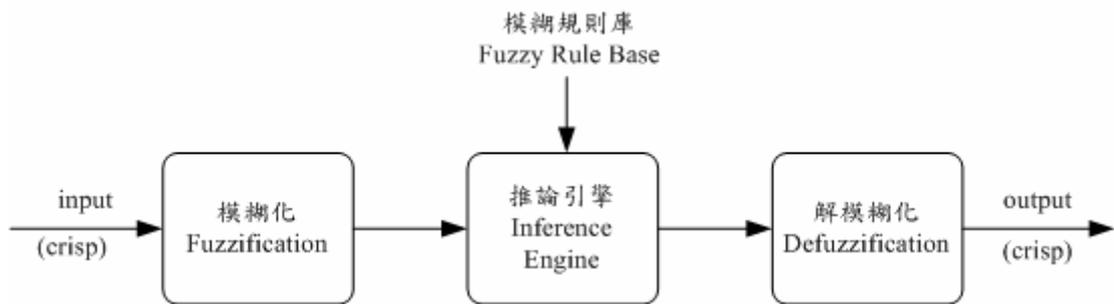


圖 4.24 模糊推論系統流程

- (i) 模糊化: 根據輸入 x_0 ，找相對應的模糊集合，再從模糊集合的定義找出相對的歸屬函數 $\mu_{A_i}(x_0)$ 。
- (ii) 模糊規則庫: 根據有經驗的操作者或專家知識制定模糊控制規則，通常是 *IF-THEN* 的條件敘述句。
- (iii) 推論引擎: 將包含於規則庫中的事實和規則輸入推理引擎(*max-min*)以得出一個結論，也就是輸出的模糊集合 $\mu_B(y)$ 。

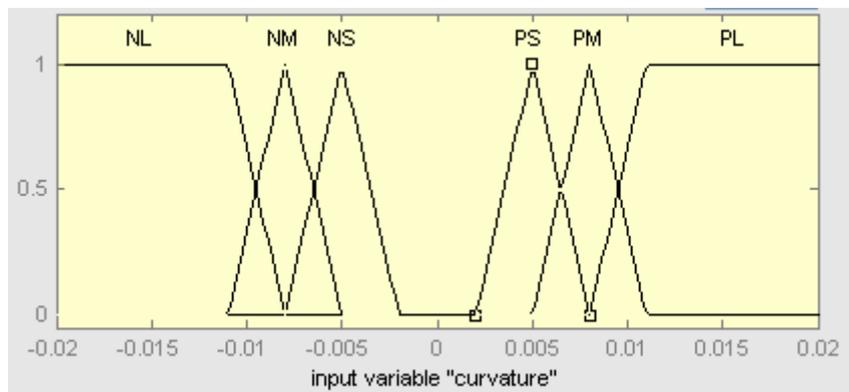
$$\mu_B(y) = \max_{\forall i} \{ \min \{ \mu_{A_i}(x_0), \mu_{A_i \rightarrow B_i}(x, y) \} \} \quad (4-2)$$

- (iv) 解模糊化: 根據推論引擎所找出來的輸出模糊集合，利用重心法 (center of area, COA,) 或最大值平均法 (mean of maximum, MOM)，將輸出模糊集合轉成明確的輸出值 y_0 。在此採用重心法，式子如 (4-3)。

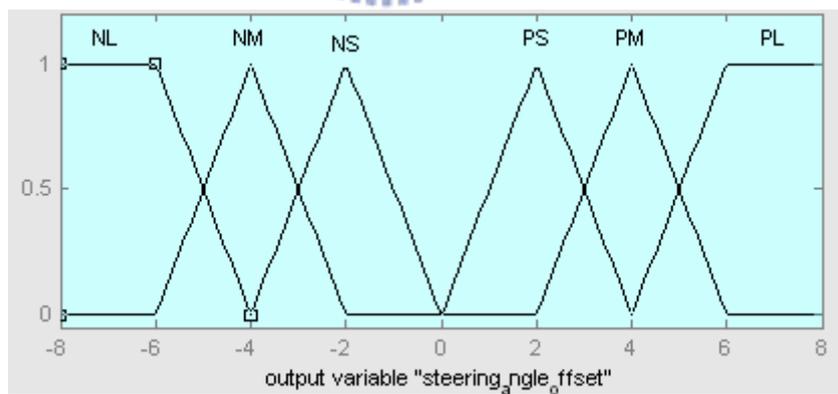
$$y_0 = \frac{\sum_{y \in Y} y \times \mu_B(y)}{\sum_{y \in Y} \mu_B(y)} \quad (4-3)$$

(2) 輸入與輸出模糊集合及規則庫的建立

圖 4.25(a)為曲率的歸屬函數集合，圖 4.25(b)為偏移補償量的歸屬函數集合。而規則庫的建立如圖 4.26，基本原則就是當遇到較大的曲率時，便給予較多的偏移補償量。

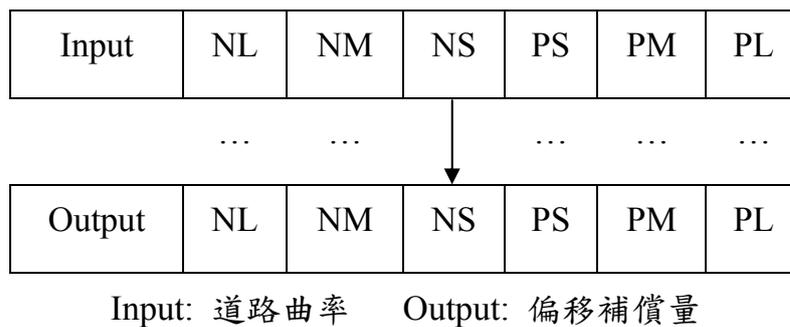


(a) 曲率歸屬函數



(b) 偏移補償量歸屬函數

圖 4.25 曲率與偏移補償量歸屬函數

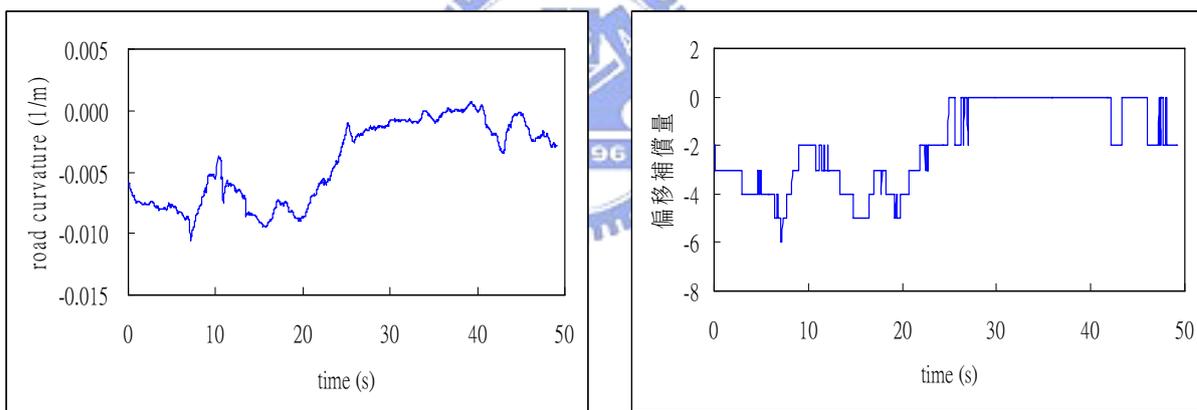


P: Positive N: Negative S: Small M: Medium L: Large

圖 4.26 模糊規則庫

(3) 曲率與補償量測試結果

圖 4.27 為驗證曲率與馬達偏移量之間的關係，大致上可以看出當車子處於彎道時，馬達偏移量約逆時針偏轉 4 至 6 刻度，所以並無太大的問題存在。



(a) 道路曲率

(b) 馬達偏移補償量

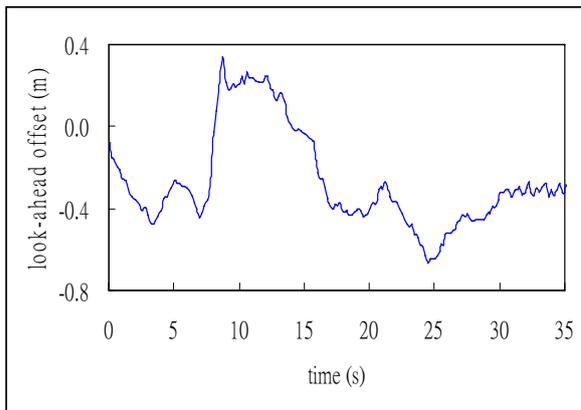
圖 4.27 道路曲率與馬達偏移補償量關係

(4) 實驗結果

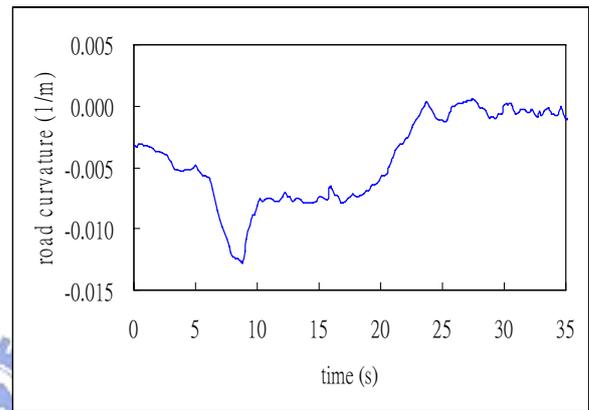
圖 4.28 為使用(P+模糊偏移補償量)實驗結果，其中 $K_p=7$ 。從圖 4.28 來看，當車子在彎道時，約 5-10 秒，影像偵測方面出了問題，導致曲率在短時間內有較大的誤差。而觀察 0-5 秒及 15 秒附近，可看出系統有點跟不上道路的變化，代表所給予的轉向不夠。而圖形在 25 秒時，有一個較大的偏移距離產

生，與道路曲率搭配來看，曲率顯示結果為直線狀態，加上影像主要擷取的就是前方資訊，所以推估此原因乃因車子要從彎道進入直線所產生的，當足夠大的偏移距離產生時便足以把車子轉正以進入直線道路。

圖 4.28(a)雖然稍比圖 4.23 要來的差，但在實際上結果證明了使用模糊的方式也是可行的，如圖 4.28(c)所示，並無出現明顯偏離車道的情形。



(a) 前視偏移距離



(b) 道路曲率



(c) 顯示行駛彎道中情形

圖 4.28 車輛使用(P+模糊偏移補償量)行駛彎道實驗結果

使用模糊補償的方式，由於多考量曲率資訊，故將PC端架構圖修正，如圖 4.29 所示。

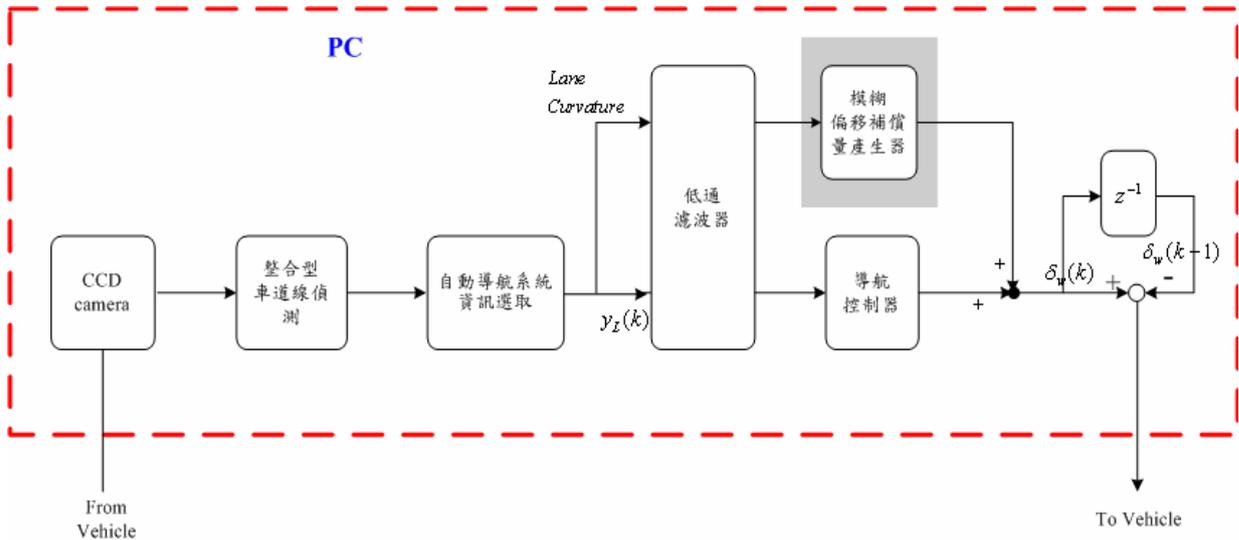


圖 4.29 彎道實驗 PC 部份架構圖修正



4-3-3 實驗分析

1、經由 4-3-1、4-3-2 節的實驗顯示，以 $L=5\text{m}$ 為例，當給予一補償量後，的確可達到較少的前視偏移距離產生，及沒有偏離車道的情形，不論是使用固定補償或模糊補償的方式。就補償方面而言，雖然目前使用固定補償的方式較使用模糊補償的方式還要好，但是使用模糊補償的方式仍是可行的，實驗也已證明此點。而未來可針對模糊補償的方式加以發展，理由為使用固定補償的方式在一條非固定曲率的道路上，容易有補償過多或過少的現象，而使用模糊偏移補償的方式可改善此缺點。

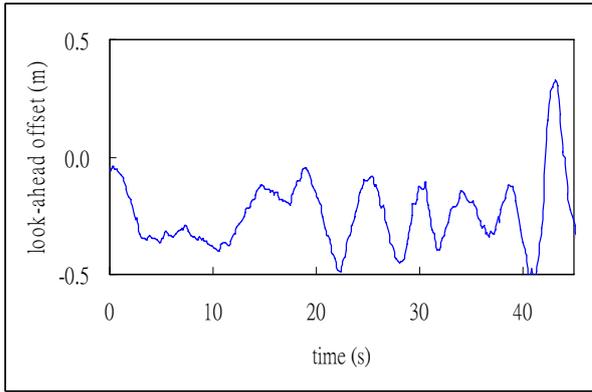
2、本系統在彎道車道線偵測方面容易受到一些外在因素影響，如：陽光，或路面上的一些雜物，如圖 4.30 所示，造成辨識方面出了問題，影響到駕駛行為。



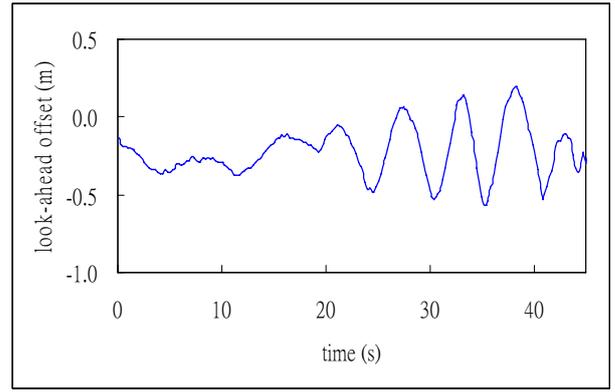
圖 4.30 彎道車道線受到陽光影響偵測結果

3、經由實驗測試，雖然改善了車子行駛彎道時偏離車道的情形，但從實驗中仍可發現，當車子從彎道要進入直線道路時，轉換道路瞬間會產生較大偏移距離，將車子的偏轉角度修正。進入直線道路後車子容易出現了方向或位置上的誤差，理論上應該等於零，初步分析結果，有可能是因為轉角係數或其他原因等等對控制方面造成的影響，使得需比原先更多的轉向角度，故未來再設計控制法則方面可多考量其它的資訊。

4、由於在直線及彎道部分用不同的車道線做為依據，所以在此主要探討若只利用右車道線在直線道路上駕駛，觀察其結果的好壞，圖 4.31 顯示使用右車道線在直線道路行駛情形。由圖 4.16(c)(d)與圖 4.31 比較可知，明顯的可以看出在相同的速度下，利用車道中心線的方式會比使用右車道線的方式得到更穩定的結果，主要原因是導航資訊在實驗道路上受變動的程度較小，經過P控制器後改變的幅度也較小，所以影響車子駕駛情形的程度較小。另外，使用單一車道線做為導航依據容易造成在不同道路上找出的道路中心與實際的道路中心有較大的誤差，所以車子容易與車道中心有一偏移距離。



(a) $K_p=7$



(b) $K_p=9$

圖 4.31 車子利用右車道線於油門值=390 時行駛直線道路結果



第五章 結論與未來發展

5-1 結論

本研究最終整合了影像與控制法則，將 PC 端訊息透過 USB-CAN 網路傳輸到 DSP 端，搭配 DSP2812 控制板及週邊電路部份來對車輛進行控制，發展一個以視覺為主自動導航車。在此對本論文提出以下幾點做為結論：

1、影像系統方面

- (1) 在直線道路上，沿用 IPM based 之車道線偵測方法，找出車道線偵測視窗之定位，再由動態視窗進行對車道線追跡，此方式能兼具辨識率及即時性的優點，在左右車道線可利用的情況下，估測出虛擬的車道中心線作為導航的依據。
- (2) 在彎道上，存在紅色連續的右車道標記及不連續的左車道標記，不連續現象容易造成偵測錯誤且一般處理不容易突顯紅色標記，故在彎道上，將原影像經過色度座標轉換(RGB->YCrCb)，再利用條件式二值化，以去除大部份的雜訊，使原偵測方法亦能有效偵測彎道上的右車道線，並做為導航的依據。
- (3) 利用車道線建立起 3D 車道模型，從中得知車輛在道路上的相關狀態：(1) 前視偏移距離 (2) 偏轉角度 (3) 道路曲率等，以提供控制器所使用。另外，由於得到的資訊含有高頻成分，故加入低通濾波器使其變的較平滑，以避免資訊突然變動太大而造成車輛的不穩定。

2、控制系統方面

- (1) 在直線道路上，本研究以 P 控制器為主要的車輛控制器。
- (2) 經由彎道實驗後，結果顯示只使用 P 控制器仍會造成車子行駛彎道時有偏離車道的情形，故除了 P 控制器外，多加入一以曲率為輸入之模糊推論器，以改善行駛彎道時偏離車道情形。

3、整合方面

- (1) 在直線道路上，以近區間車道線做為主要依據，前視偏移距離為主要導航資訊，搭配 P 控制器，根據實驗結果顯示，當給予 $K_p=5\sim 11$ 時，並根據準確度及穩定度的觀點來看，能獲得穩定及準確的結果，實驗車速約在 2m/s。
- (2) 在彎道上，以近區間及遠區間車道線做為依據，以前視偏移距離與道路曲率為主要導航資訊，用(P 控制器+模糊偏移量補償器)來控制車輛，實驗顯示出加入補償的方式的確可改善在彎道上 偏離車道的情形，實驗車速約在 1.2m/s。
- (3) 綜合以上，本研究已完成能在校園環校道路上直線及彎道的自動駕駛，成功實現在電動高爾夫球車上。

5-2 未來發展

- 1、目前在影像的處理方面，我們是以 PC based 來實現，處理速度約 4-5frames/s，但在未來發展可以將其移植到微電腦處理器上，如 DSP 等，這更能確保其即時性，也較節省成本且較實用，另外亦能解決在 Windows 下，因系統有時使用部份資源去執行其它程式，造成處理速度並不是都維持一定。

- 2、在針對實驗道路-彎道部份，可發展更精確的車道線偵測法則，以解決因週遭環境而影響偵測結果，如陽光因素等等，或一套可自動調整二值化閾值的機制，以提供更好的偵測影像。另外，對於影像所獲得的導航資訊方面，容易因車子行駛在高低起伏的道路或上下坡而造成誤差，未來可估測出坡度的變化，以獲得更準確的導航資訊。

- 3、本論文在定速且低速下已完成初步的視覺導航車，在未來可針對車子行駛遇到的一些不確定性因素(ex:轉角係數、載重等等)設計出適當控制器，亦可加入其它的感測器，考量更多的導航資訊，以提高系統的穩定度及準確度。另外，可多加入感測器取得速度資訊，進一步控制車速、煞車等等，再從已有的前方車輛偵測的資訊，加入相關的控制，實現出高表現的自動駕駛。



參考文獻

- [1] <http://traffic.tycg.gov.tw/service/genius/upt.asp?p0=20&uid=&con=&seccid=14&cid=138&year=&month=&day=&key=>
- [2] M. Bertozzi and A. Broggi, "GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection", *IEEE Transactions on Image Processing*, Vol. 7, Issue 1, pp. 62 - 81, Jan. 1998.
- [3] Y. Wang, E. K. Teoh, and D. Shen, "Lane Detection Using B-Snake", *Proceedings of International Conference on Information Intelligence and Systems*, pp. 438-443, 31 Oct.-3 Nov. 1999.
- [4] C.R. Jung and C.R. Kelber, "A Robust Linear-Parabolic Model for Lane Following", *Proceedings of 17th Brazilian Symposium on Computer Graphics and Image Processing*, pp. 72-79, 17-20 Oct. 2004.
- [5] D. Pomerleau and T. Jochem, "Rapidly Adapting Machine Vision for Automated Vehicle Steering", *IEEE Intelligent Systems and Their Applications*, Vol. 11, Issue 2, pp. 19 - 27, April 1996.
- [6] K. A. Redmill, "A Simple Vision System for Lane Keeping", *IEEE Conference on Intelligent Transportation System*, pp. 212-217, 1997.
- [7] S. J. Wu, H. H. Chiang, and J. W. Perng, T. T. Lee, C. J. Chen, "The Automated Lane-Keeping Design for An Intelligent Vehicle", *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 508 – 513, 6-8 June 2005.
- [8] J. Kosecka, R. Blasi, C.J. Taylor, and J. Malik, "A Comparative Study of Vision-Based Lateral Control Strategies for Autonomous Highway Driving", *Proceeding of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 1903 – 1908, 16-20 May 1998.
- [9] K. Kluge and S. Lakshmanan, "A Deformable-Template Approach to Lane Detection", *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 54-59, 1996.

- [10] J. Gangyi, C. Yanhua, Y. Mei, and Z. Yi, “Approach to Lane Departure Detection”, *International Conference on Signal Processing*, Vol. 2, pp. 971-974, 2000.
- [11] T. Hessburg and M. Tomizuka, “Fuzzy Logic Control for Lateral Vehicle Guidance” *IEEE Control Systems Magazine*, Vol.14, Issue 4, pp. 55 – 63, Aug. 1994.
- [12] K.A. Unyelioglu, C. Hatipoglu, and U. Ozguner, “Design and Stability Analysis of A Lane Following Controller”, *IEEE Transactions on Control Systems Technology*, Vol. 5, Issue 1, pp. 127 – 134, Jan. 1997.
- [13] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982
- [14] 吳政衍, Multi-Agent 駕駛輔助系統, 國立交通大學 電機與控制工程學系 碩士論文, 2004.
- [15] 李尚玲, Multi-Agent 全天駕駛輔助系統, 國立交通大學 電機與控制工程學系 碩士論文, 2005.
- [16] 林重甫, 智慧型車輛之自動視覺導航系統, 國立交通大學 電機與控制工程學系 碩士論文, 2001.
- [17] 蔡尚錚, 以視覺為主的智慧型車輛導航之研究, 國立東華大學 電機工程學系 碩士論文, 2001.
- [18] 鄭景文, 動態網路控制系統之時間延遲分析, 國立交通大學 電機與控制工程學系 碩士論文, 2006.