

國立清華大學 電機工程學系

實作專題研究成果摘要

Implementation of self-driving car
techniques

智能車之自動駕駛技術

專題領域：系統領域

組別：B269

指導教授：朱家齊

組員姓名：葉柏佑、張華泰、黃品崙

研究期間：2022年2月14日至 2022年11月14日止，共9個月

Abstract

近年來自動駕駛汽車的技術正在蓬勃發展，也有許多科技公司正在研發相關技術，或是在私有路段真正的實驗無人駕駛，雖然安全性備受質疑，而且因為專利問題，部分先進技術無法投入商用，但現今仍已有許多汽車廠商應用在駕駛輔助系統，因此我們希望能從建構車子硬體開始，以 raspberry pi 為控制器，從物理結構的建立到軟體控制，實作自走車之物件追蹤與道路線追蹤。

此次實作專題使用的嵌入式開發版為 raspberry pi，並以 MATLAB 的 Simulink 作為開發平台，感測器有500萬像素的 Rev 1.3相機和超音波感測器 SR04，用來檢測物體位置，除此之外，還有兩個光電開加上測速碼盤，用來實現閉合迴路的速度控制。我們主要實作的技術有三大部分，從底層的馬達速度量測與控制、物體相對位置的估計，到結合兩者結果，設計一套追蹤物體的流程，然後運用類似的邏輯實作道路線追蹤。

Introduction

此次實作專題使用的嵌入式開發版為 raspberry pi，車身為網路上常用的電動車模板，搭載了 L298N 馬達驅動板、兩個減速直流馬達、橡膠輪胎，以3顆3.7 V 的可充電鋰電池串聯供電，raspberry pi 則另外使用直流降壓模組供電，感測器有500萬像素的 Rev 1.3相機和超音波感測器 SR04，用來檢測物體位置，除此之外，還有兩個光開加上測速碼盤，用來實現閉合迴路的速度控制，車身的正反面如圖1所示。

開發環境選用 MATLAB 的 Simulink，其內部有對 raspberry pi 支援的附加組件，分為機組馬達速度與方向控制、感測器檢測物體位置與追蹤物體算法三大部分，下面將針對這三大部分進行簡單的介紹。

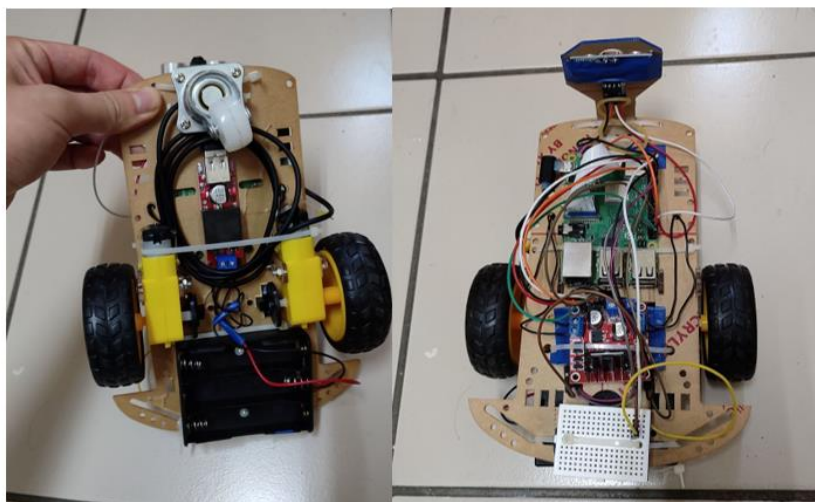


圖1: 使用之車體正反面

1. 馬達速度測量與控制

1.1. 馬達速度量測

測量速度需要用光電閘搭配測速碼盤，將測速碼盤裝在馬達內側，使用光電閘去計算經過的格子數，藉由一段間內的格子數量回推平均轉速，然後再過一個 low pass filter 降低 noise，最後再取加權平均便可得到與實際速度誤差小於3%的結果，如圖1所示。

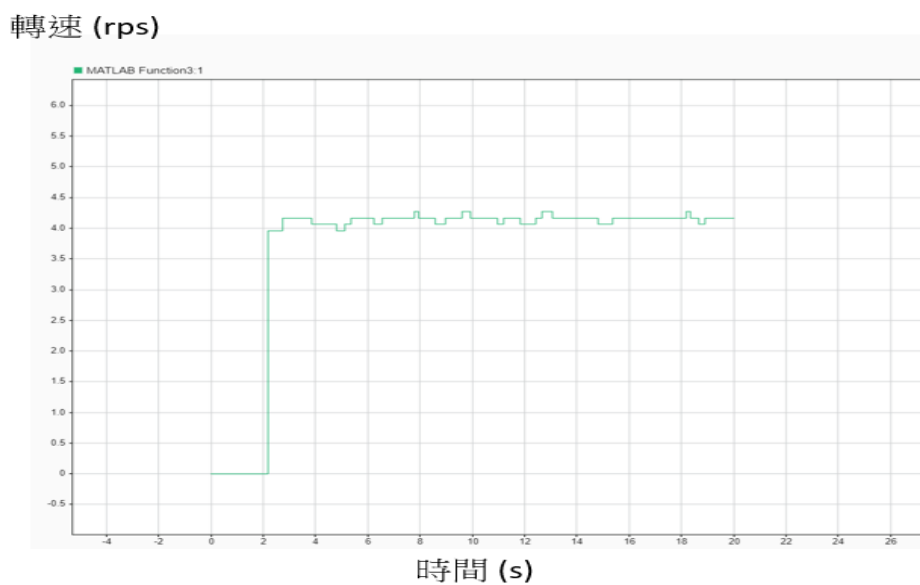


圖2: 測速結果

1.2. 馬達控制

我們控制馬達的核心原理是 PID 控制，圖3是 PID 控制的方塊圖，Set speed 是我們想要的速度，方塊 Measure speed 是輸出量測到的速度，利用兩個的誤差項 $e(t)$ ，經過比例、積分、微分三個項相加後，用來調整輸出的 PWM 值，然後接到馬達驅動板上作控制，除了量測到的速度，我們也將部分狀態下的控制改為用角度誤差與距離誤差納入 $e(t)$ ，作進一步的調整，另外因為微分項容易造成系統不穩定，所以此時實作並沒有使用。

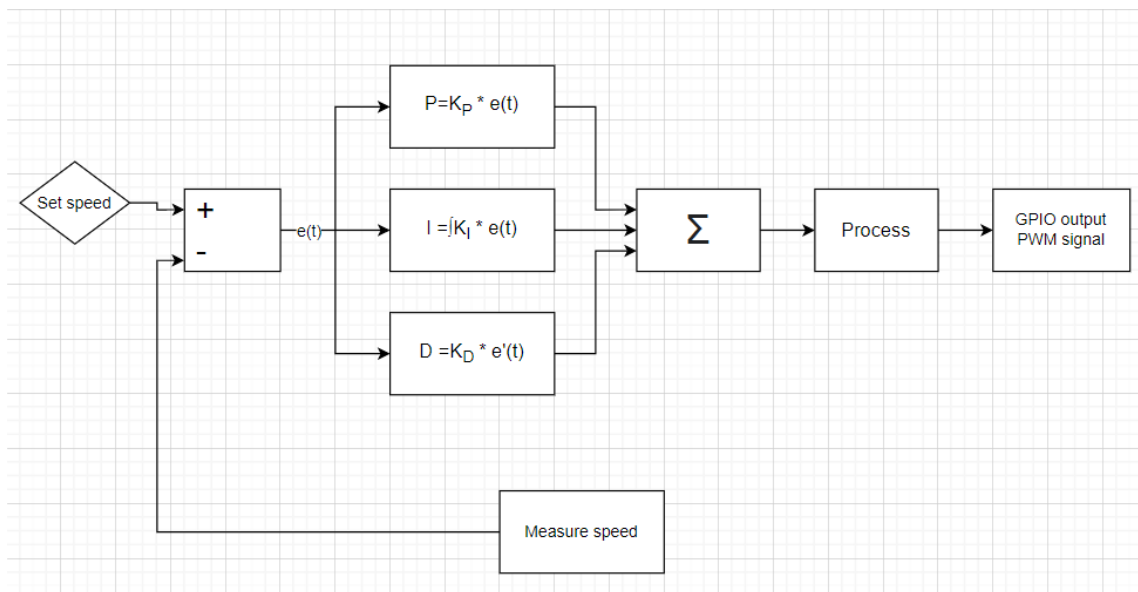


圖3: PID 控制

2. 感測器檢測物體

2.1. 偵測物體與道路線

我們是將圖片中最大的紅色區塊作為目標並追蹤中心點，因此物體與道路線必須是紅色。我們先把圖片 RGB 轉成 HSV，接著利用 H、S、V 三個值做 threshold，取 H 值介於 $0 \sim 10$ 或 $160 \sim 180$ 、S 值介於 $50 \sim 255$ 、V 值介於 $20 \sim 255$ 的 pixel，也就是把所有紅色的 pixel 取出來設為 1 其餘設為 0 的點陣圖，之後將點陣圖經過 erosion 把一些雜訊過濾掉，erosion 作法是將 image 經過 filter，將 $5 * 5$ 範圍內的 local minima 標示出來，因此小於 $5 * 5$ 的紅色區塊都會被過濾掉，再利用 blob analysis 找出所有紅色區塊的中心點，最後再將面積大於 100 的區塊中最大面積的中心點取出，即為物體或道路線的中心點，如圖 2-1-5 所示，在經過影像處理後，我們可以得到面積最大的紅色物體中心點。

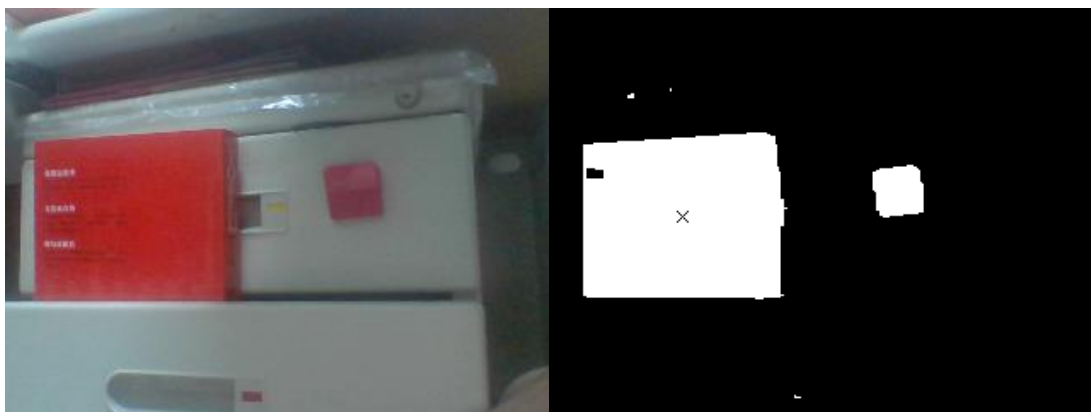


圖4: 左為實際原圖影像，右為處理過後的點陣圖

2.2. 角度估計

在針孔成像模型中，成像平面與相機的距離是固定的，所以物體與穿過相機的成像平面法向量夾角固定，那麼物體在成像於平面的位置也會固定，所以在經過校正後，我們可以利用上面求得的物體中心位置，透過反三角函數，求得物體角度。

2.3. 距離估計

利用超音波測距器測量距離，原理是發射一個10 us 超音波脈衝，藉由發射與接受到的時間差，除以當時音速除以2，便可得到物體距離，從 datasheet 與實測結果得到有效測量角度為 15° 。

3. 追蹤物體算法

因為感測器有不同的有效測量範圍，所以我們分三個不同的狀態去做追蹤物體的控制

(1) 超音波量不到距離，物體在相機視野範圍內:

此時會做單純的角度追蹤，嘗試縮小角度差距，進入測距範圍內，不過也有可能物體距離超出可測最大距離，但依然能測得不精確但保證大於最大可測距離的數據。

(2) 超音波量到距離，物體在相機正負 7.5° 內:

同時利用角度差與距離差跟隨物體，嘗試縮小距離差距，當距離小於5 cm，超音波測距儀可以量到小於5 cm 但不準確的數據。

(3) 超音波量到距離小於5 cm:

此時也是做角度追蹤，然而與第一種狀態不同的是，距離太近角速度太大容易丟失物體視野，所以會以低速去調整角度，最後誤差小於 3° 時煞車停止。

在三個狀態內，追蹤物體的核心原理是 PID 控制，並前置二階以下的 low pass filter 過濾雜訊，根據角度誤差與距離誤差，生成對應的左右輪速度，每一個狀態所用的係數不同，然後使用速度計算所需的 PWM 值作為初始值，再根據量到的速度做第二層 PID 控制。

除此之外，在靜止狀態下直接給過大 PWM 初始值會導致馬達暴衝，有 overshoot 的現象，造成不必要的初始位移，還有可能使物體視野丟失，另一方面，過低的初始 PWM 值則會導致無法克服馬達內阻和地面的靜摩擦力而無法移動，因此在步時會逐漸增加 PWM 值，然後在克服阻力後再回到設定的 PWM 值，但可能會由於地形關係又停下來了，所以設定的低速 PWM 值會根據突破阻力時的 PWM 值作動態調整。

4. 實驗結果

物件追蹤的最後結果是在相機視角 22° 以內可以進行有效的角度跟隨，而在超音波感測器的可視角度 15° 內與距離2.3 m內可以進行有效的距離跟隨，如果目標物皆在兩個感測器的範圍內，且物體移動過程中沒有離開範圍，則可以進行動態追蹤，路線追蹤的結果是可追蹤小於 90° 彎的實驗道路。

5. 結論

目前大致可以追蹤板子，理論上只要將板子黏到另一台車子的後面就可以實現跟車，但還是存在些許問題，主要是相機視角仍然過小，車子只能跟隨看的見的目標，可能更換成廣角相機能得到有效改善，但角度量測需要將影像做額外轉換，此外超音波測距有所侷限，需要等車子轉到一定角度內才能量到距離，並追蹤目標，所以容易丟失目標仍然是一大問題，跟隨移動速度快的物體仍然不可行，日後若能添加物體移動速度與方向作為控制參數，或是選擇更廣的距離感測器，應該能有效改善。

6. 參考文獻

- [1] Agus Kurniawan, *Getting Started with Matlab Simulink and Raspberry Pi*, 1st ed. PE Press, 2013
- [2] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 13th edition, Pearson Education Limited, 2017.
- [3] Dr. Kaputa. 2017. Raspberrypi mex. <https://github.com/hoanhhan101/ultimate-go>

Review and reflections

這次專題實作隊我們來說算是一個新的挑戰，由於指導教授的碩博士生們沒有相關經驗，所以大部分時候需要我們自己摸索，所以做了很多最後沒用到的技術，例如: LoRa 射頻通訊技術、path planning (RRT*) 的模擬、path tracking (pure pursuit、Stanley)的模擬、EKF-SLAM 的模擬還有深度學習中的影像辨識，因為買不到 Jetson Nano，而 Raspberry Pi 沒有 cuda 而放棄，但過程中學到了很多新知識，解決問題的過程中也充滿了成就感，除此之外，我們還有去參加 AWS DeepRacer 的比賽，是一個線上賽車比賽但控制器是 DRL (deep reinforce learning) agent 而不是傳統的 model based 的方法，不過打入複賽後沒有拿到前三名，只能算是增添一點經驗，最後感謝教授的支持與幫助，讓我們能夠完成這次實作專題。