

2023 年臺灣國際科學展覽會 優勝作品專輯

作品編號 100020

參展科別 工程學

作品名稱 超音波自動跟車系統

得獎獎項

就讀學校 國立臺灣師範大學附屬高級中學

指導教師 蘇偉雋、戴仁欽

作者姓名 陳玠安

關鍵詞 Arduino、PID、自動跟車

作者簡介



我是陳玠安，目前就讀師大附中科學班高三，對於機械與控制系統十分有興趣。在研究過程中，對於各種控制方法有更深入的了解，也獲得了更好的資訊能力。希望未來能朝相關的領域持續精進。

摘要

在本研究中，希望能由與前車距離、相對速度等數據，設計符合自動跟車功能的系統。能與前車保持固定距離，並追蹤相對速度，在前車煞車或加速時，使後車即時反應。

分別使用軟體模擬及 Arduino 控制車輛，測試 PID 控制在各情況的可行性。前車行駛模式包含等速、等加速度等，測試後車的 PID 組合，以設計出最佳的直線跟車系統。在軟體模擬中，經過變換車道、圓型軌跡、轉向等環境測試，分析出 P 值在各個情境的作用，及其大小的優劣或限制。利用第二測距感測器，以前車的轉向幅度，控制後車兩輪速度差，使系統具備轉向跟隨的功能。

Abstract

The research aims to develop an ACC (Adaptive Cruise Control) system based on distance measuring and PID control. The system was first simulated under VPython environment before being applied to the Arduino vehicle. Ultrasonic distance sensors were installed on both sides of the vehicle to detect the distance or direction change.

In various scenes, different PID combinations were recorded and tracked to analyze the sensitivity and functionality. The car in front follows possible driving conditions including constant velocity, constant acceleration, lane switching, and turning so as to examine the performance and reaction of the system in real-life scenarios.

壹、前言

一、研究動機

自動跟車系統 (Adaptive Cruise Control) 汽車現今發展迅速，可以廣泛的在各個廠牌和種類的汽車看到這樣的功能，其中包含許多複雜的辨識與運算過程。本研究中嘗試以測距方式，達到追蹤同步前車速度，並以兩組感測器，判斷前車的轉向，設計能跟隨前車速度與轉向的自動跟車系統，使用成本較低的超音波系統，可以運用在速度較低，精確度要求較低的行進系統。

二、文獻回顧

(一) PID 控制

兩車在同一條筆直的車道上前後行駛，前車預先寫入行駛模式，過程中模擬真實駕駛情形，包含起步、加速、等速行駛、煞車等。並測試調整，嘗試以兩車相對速度、距離為基礎，使用 PID 方法設計多種速度的控制函式組合。

使用 PID 控制作為速度的函式，使用感測器的測量數據，應由不同的控制項組合，以達到最佳的跟車系統。PID 控制中， $u(t)$ 為控制輸出，由以下方程式：

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t)$$

K_p : 比例控制項

K_i : 積分控制項

K_d : 微分控制項

讀取感測器數據後，加上微分與積分響應，組成 PID 控制輸出。在本研究中，分別嘗試單獨比例控制(Proportional Control) 及比例-積分控制 (Proportional-Integral Control)，加入積分控制消除穩態誤差(steady-state error)。測試 K_p 、 K_i 、 K_d 組合，以距離作為比例控制基礎，加入相對速度作為積分控制。

貳、研究方法或過程

一、直線行駛跟車系統

(一) 軟體模擬

兩車在同一條筆直的車道上前後行駛，前車預先寫入行駛模式，過程中模擬真實駕駛情形，包含起步、加速、等速行駛、煞車等。並測試調整，嘗試以兩車相對速度、距離為基礎，使用 PID 方法設計多種速度的控制函式組合。

依據動畫模擬及繪製速度、距離對時間圖表，觀察討論 PID 控制組合對跟車反映靈敏度、最大加速度、車速平穩度等面向的影響。

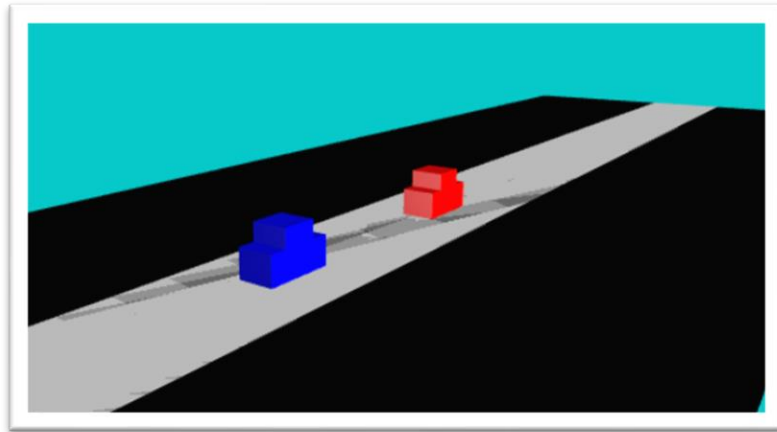


圖 1、動畫模擬

(二) Arduino 車輛測試

使用 Arduino 系統，以 PWM 控制馬達轉速，並安裝超音波距離偵測器，以距離及相對速度等資訊，設計 PID 控制程式，調整各個控制項，找出最佳的控制系統。

符合跟隨前車起步、保持等距行駛、跟隨牽車煞車等功能，並在行駛中保持平穩，防止產生過大的加速度，同時提高跟車的靈敏度。

二、轉向跟車系統

(一) 軟體模擬

使用與上述直線行進直線相同的設置，在車道方面，使前車走圓形、曲線、直角轉彎等模擬真實道路情況。藉由車身兩側的距離差，觀測前車的轉向狀況，在保持速率相同的同時，實現跟隨前車轉向。

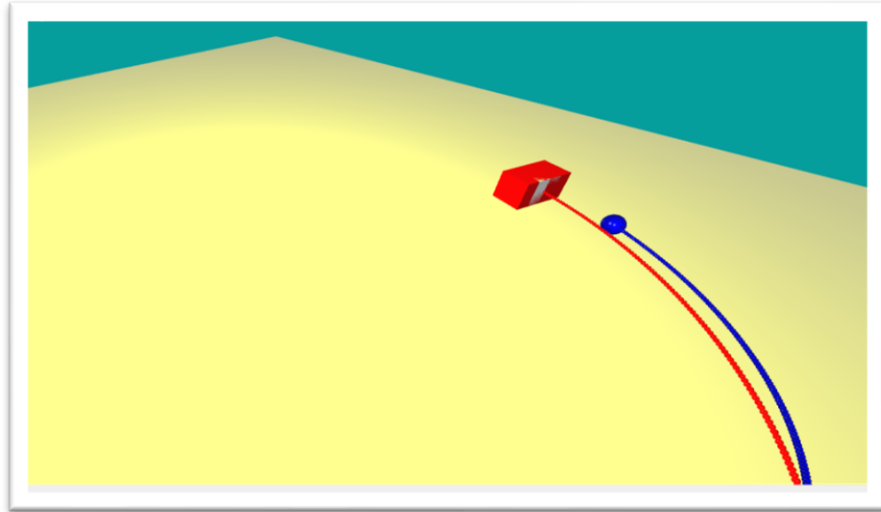


圖 2、圓形軌跡動畫模擬

(二) Arduino 車輛測試

使用與上述直線行進直線相同的 Arduino 車輛，並增加一個超音波距離感測器，使左右兩個感測器同時取得距離資訊，藉此判斷前車的轉向，並調整兩輪速度差，以跟隨前車轉向。

三、研究裝置

(一) 車輛模型

以 Arduino Uno 搭配驅動版，使用 PWM 控制連接到 6V 直流電壓的左右輪馬達，車輛前方架設超音波測距儀，Arduino 接收感測數據，運算預先寫入的 PID 控制，並分別輸出類比訊號至兩馬達。達到偵測前車距離、速度，並且控制後車速度的目標，以進行跟車。

1. PWM 和速率關係

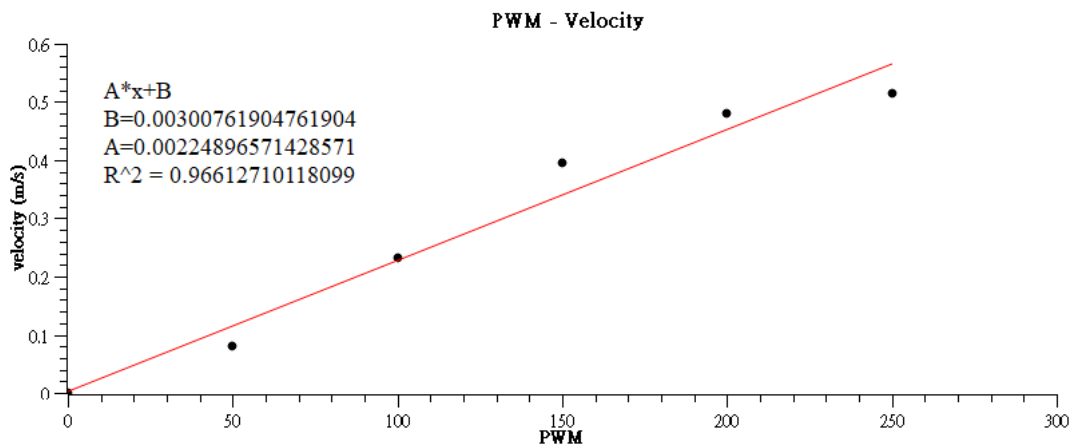


圖 3、PWM 和速度關係

PWM 輸出訊號大小(0-255)和 Arduino 車輛行駛速度呈正比，因此可以

藉由調整輸出大小，控制車輛的速度。

(二) 位置追蹤

架設相機從側向錄製兩車行駛過程，使用 Tracker 追蹤前後車位置。

(三) 軟體

動畫模擬使用 Glowscript 的 VPython 3.0 環境。車輛測試中，前車的行駛模式程式，後車的反應系統，皆使用 Arduino IDE。

四、直線行駛跟車系統

(一) 軟體模擬

在 VPython 環境中設計筆直的車道，並有兩個獨立的前、後車，在前車行進時，即時取得兩車的距離、相對速度，在經過各種組合的 PID 控制，輸出後車的速度或加速度。

測試過程繪製兩車的 x-t 圖、v-t 圖，再以此分析每一組測式的 PID 性能，檢查是否有破壞安全距離的情況，並且比較不同組合的優劣。

(二) Arduino 車輛

車輛包含前方超音波感測器，Arduino、馬達驅動版、左右兩馬達，

前車有多種行進方式，包含等速前進、起步、等加速度、煞車等。後車依據測距距離，輸入 Arduino 寫入的 PID 控制，輸出 PWM 至驅動版，控制馬達的轉速。

架設相機從側向錄製兩車的行進狀況，使用電腦軟體 Tracker 追蹤兩車位置、速度，畫出兩車 v-t 圖、x-t 圖、相對速度圖。分析 PID 的啟動時機，及作用情況、震盪情形等，以找出最佳的 PID 控制組合。

除了使用兩車距離做 P 控制，預計使用超音波感測器量測速度，以胞率(Baud)及資料密度進行差分取得兩車的相對速度。

五、轉向跟車系統

(一) 軟體模擬

在平面上讓前車走出圓形、橢圓等各種曲線的軌跡，後車追蹤前車左右兩側到後車左右兩側的距離，以此調整後車的轉向角度、轉彎速度等。

1. 前車行駛路徑

為創造各種的行駛狀況，測試後車在前車轉向、加速行駛中的跟車性能，設計不同的行駛情境。

(1) 前車以 sin 路徑行進

前車以 sin 波左右行進，以前後兩車的行進方位(以 θ 表示)隨時間作圖，觀察 PID 控制中的變因對兩者的波型的影響。

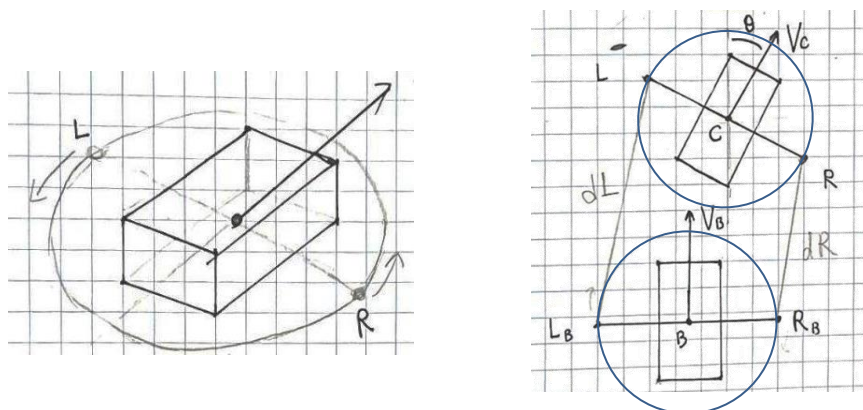
(2) 前車手動操控

以鍵盤左右按鍵控制前車方向。按下鍵盤時，產生垂直行進方向的加速度，改變方向而不影響速率。亦可直接設定行進方位或角加速度，以不同的取率半徑行駛。

(3) 前車變換車道

前車左右變換車道，觀察後車的跟車軌跡及延遲情形。

2. 車輛模型



$$\begin{cases} V_x = V_C \sin \theta \\ V_y = V_C \cos \theta \end{cases}$$

$$\begin{cases} V_{BX} = V_B \sin \theta_B \\ V_{BY} = V_B \cos \theta_B \end{cases}$$

$$\vec{CL}(-\cos \theta, \sin \theta)$$

$$\vec{BL}_B(-\cos \theta_B, \sin \theta_B)$$

$$\vec{CR}(\cos \theta, -\sin \theta)$$

$$\vec{BR}_B \cdot (\cos \theta_B, -\sin \theta_B)$$

$$w_B = (dL - dR) \cdot P_1$$

$$V_B = \left(\frac{dL + dR}{2} \right) \cdot P_2$$

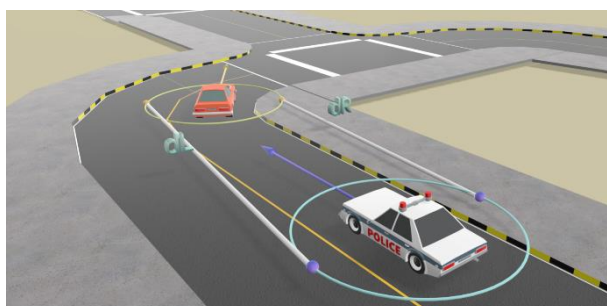


圖 4、車輛模型

垂直於車輛模型的兩側有兩個小球，以此代表車輛左右兩側的距離 (dL, dR)。角度 θ 代表車輛的行進方向，後車方向 θ_B 由 P_1 (第一組 PID) 控制，速度 V_B 由 P_2 (第二組 PID) 控制。

(二) Arduino 車輛測試

在前車預先寫入各種圓形軌跡、轉彎等行駛路徑。後車的兩個感測器分別記錄數據，將數據的差作為判斷前車轉向幅度的標準，以 PID 控制輸出為兩輪的速度差，轉向至與前車方向一致。

參、研究結果與討論

一、直線行駛情境

(一) 軟體模擬

1. $K_p = 20$, $K_i = 10$, $K_d = 0$ (PI 控制)

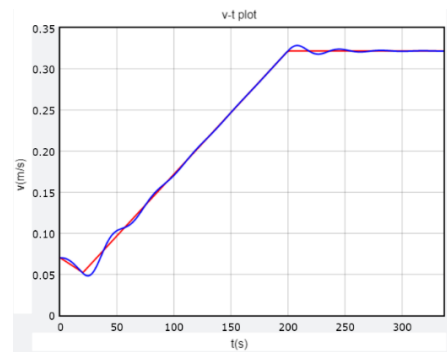
```

36 while(vb>=0 and va>0):
37     #print(va,vb)
38     cnt += 1
39     ss=cube.pos.x
40     cube1.pos.x=cube.pos.x
41     cube3.pos.x=cube2.pos.x
42     rate(300)
43     vb += -(((cube.pos.x-cube2.pos.x)-4*size)+(va-vb)*2)*10*a*dt
44
45
46     if(cnt<2000):
47         va += 0.3*a*dt
48     if(cnt>=2000 and cnt<6000):
49         va += -0.5*a*dt
50     if(cnt>=6000):
51         va += 0
52

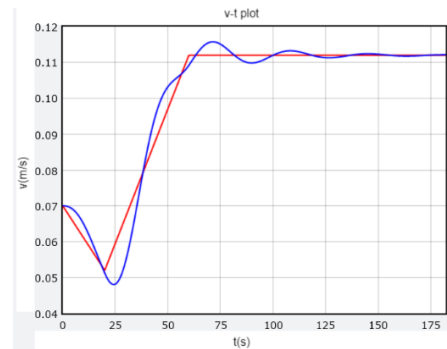
```

前車行駛模式：

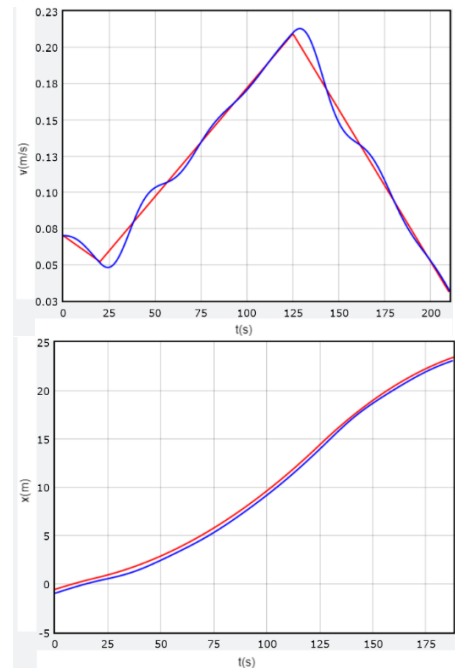
時間	前車加速度
0~2000	0.3 a
2000~6000	-0.5 a
>6000	0



時間	前車加速度
0~2000	0.3 a
2000~20000	-0.5 a
>20000	0



時間	前車加速度
0~2000	0.3 a
2000~12500	- 0.5 a
>12500	0.7 a



後車控制程式：

$$a(x) = 20 e(x) + 10 K_i \int_0^t e(x) dt$$

$$a(x) = 20 (v_{前} - v_{後}) + 10 (x_{前} - x_{後})$$

2. $K_p = 1, K_i = 0, K_d = 0$

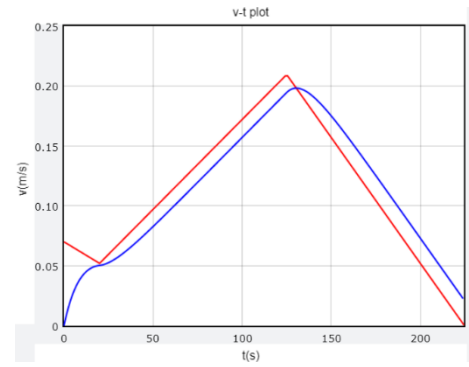
```

36 while(vb>=0 and va>0):
37     #print(va,vb)
38     cnt += 1
39     ss=cube.pos.x
40     cube1.pos.x=cube.pos.x
41     cube3.pos.x=cube2.pos.x
42     rate(300)
43     vb = ((cube.pos.x-cube2.pos.x)-4*size)*0.1
44
45
46     if(cnt<2000):
47         va += 0.3*a*dt
48     if(cnt>=2000 and cnt<12500):
49         va += -0.5*a*dt
50     if(cnt>=12500):
51         va += 0.7*a*dt
52

```

前車行駛模式：

時間	前車加速度
0~2000	0.3 a
2000~12500	- 0.5 a
>12500	0.7 a



後車控制程式：

$$v(x) = e(x)$$

$$v(x) = (x_{前} - x_{後})$$

(二) Arduino 車輛測試

1. 前車等速行駛

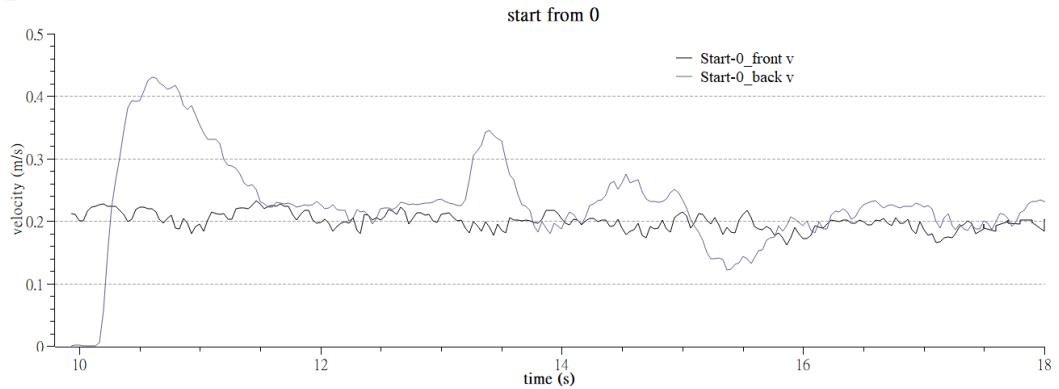


圖 5、前車等速行駛

見上圖 4，前車固定以 0.2m/s 行駛，後車初速度為零，當兩車距離超過設定的安全距離後，後車的跟車系統啟動，加速起步至約 0.4m/s，在依照距離減速(P 控制)，最後達到與前車相同的 0.2m/s，相對速度為 0，此時兩車保持固定距離行駛。

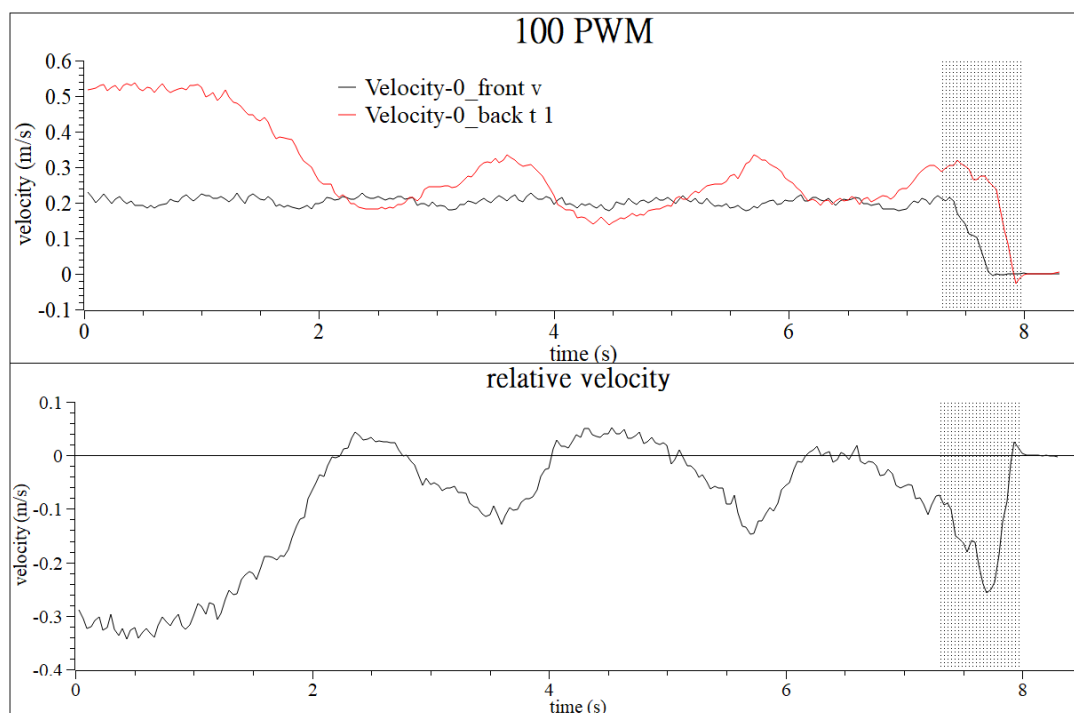


圖 6、前車等速行駛

見上圖 5，前車固定以 0.2m/s 行駛，後車初速度 0.5m/s 從後方接近，後車偵測到前方車輛後，依據 P 控制，減速至 0.2m/s，兩車以固定距離行駛。

網點部分為前車(黑色)煞車，後車經過反應時間後跟著減速。相對速度圖中，由負值修正後，在相對速度為 0 處有震盪現象，前車煞車時可以觀察到相對速度大幅降低。

藉由調整 P 控制的比例控制項，或改進為 PI 控制改變積分控制項，希望減少或消除圖 5 中所見的，後車速度、相對速度的震盪情形，同時增進系統在應對煞車、起步的靈敏度。

2. 前車等加速度行駛

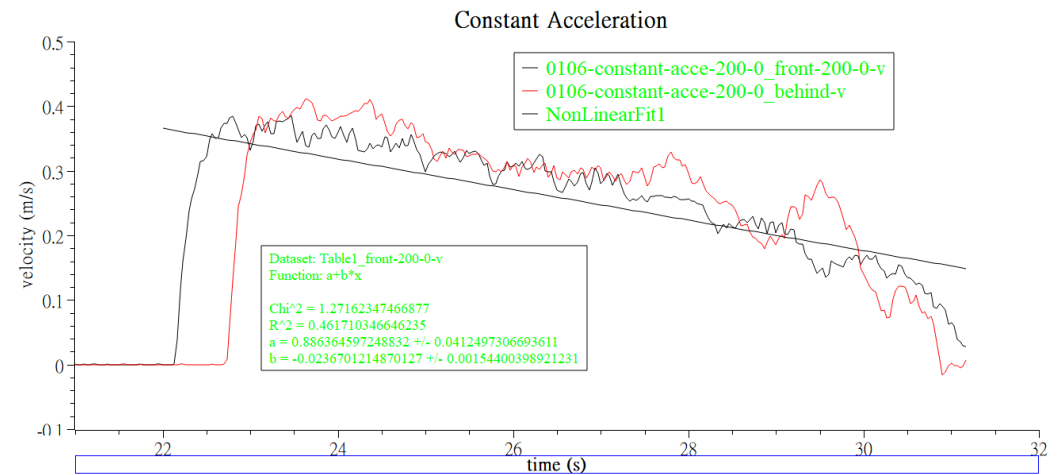


圖 7、前車等加速度行駛

上圖 6 為前車等加速度，逐漸減速至零，而後車跟隨，同樣以等加速行駛，最後兩車皆停止。過程中兩車保持固定距離。等加速度系統可以看出，每一種 PID 控制組合，在前車速度不停變化的情況下，保持相同速度的難易。

3. 前車轉向

前車轉向過程中，後車的左右兩個感測器分別取得距離，計算距離差 ($x_{左} - x_{右}$)，經過 PID 控制決定兩輪的速度差。

$$v(t) = K_p (v_{左} - v_{右}) + K_i (x_{左} - x_{右})$$

藉由調整比例控制項(K_p)，找出能順利跟隨前車轉彎的程式。並且記錄轉彎過程中，兩感測器距離差、兩馬達輸出 PWM 等數據，繪製關係圖，觀察轉彎的速度、準確性，檢驗系統是否能修正轉向過度、轉向不足的情況，最終穩定方向。

二、轉向行駛情境

(一) 軟體模擬

1. 前車變換車道

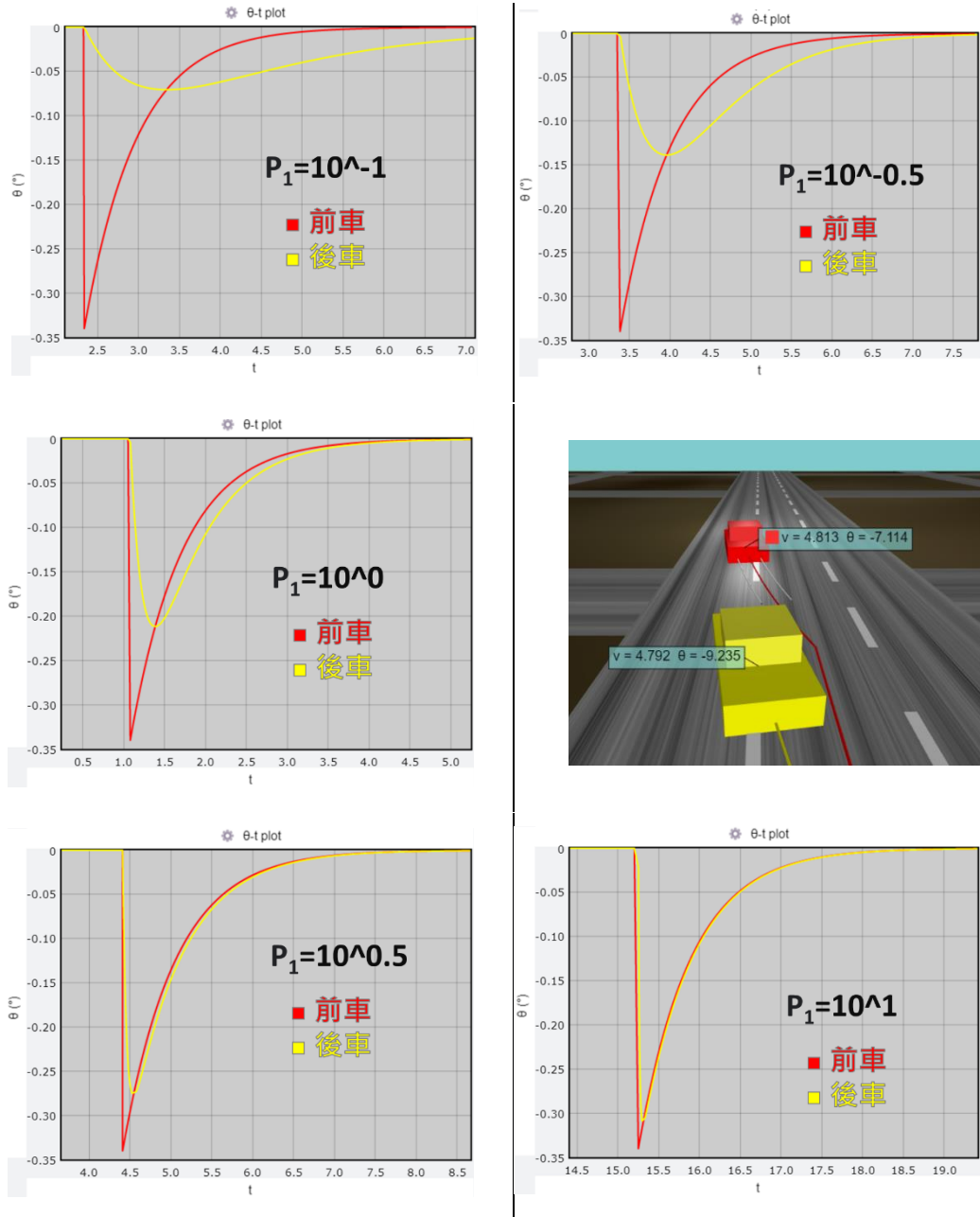


圖 8、前車變換車道

上圖為在 P_1 從 10^{-1} 到 10^1 的測試組中，前後兩車方位(θ)對時間作圖，觀察兩者的波型，能判斷跟車系統的反應速度、精確度等。後車的角速度函式如下：

$$\omega(t) = P_1 \cdot (dR - dL)$$

發現 P_1 值越大時，兩者的波型越接近，到 10^1 時兩者接近重合。較高的 P_1 有較佳的靈敏度，可以即時跟上前車的方向。較低的 P_1 則能避免過大的轉向加速度，提高行駛的穩定度。選擇適中的 P_1 值，可以避免頻繁轉向，同時跟上前車的整體行進方向。

2. 前車沿圓形軌跡行駛

在直線行駛一段距離，且兩車速度達到一致後，前車以不同半徑的圓形為軌跡行駛，轉向加速度(ω)恆定，測試後車在不同曲率半徑下的跟車情形。

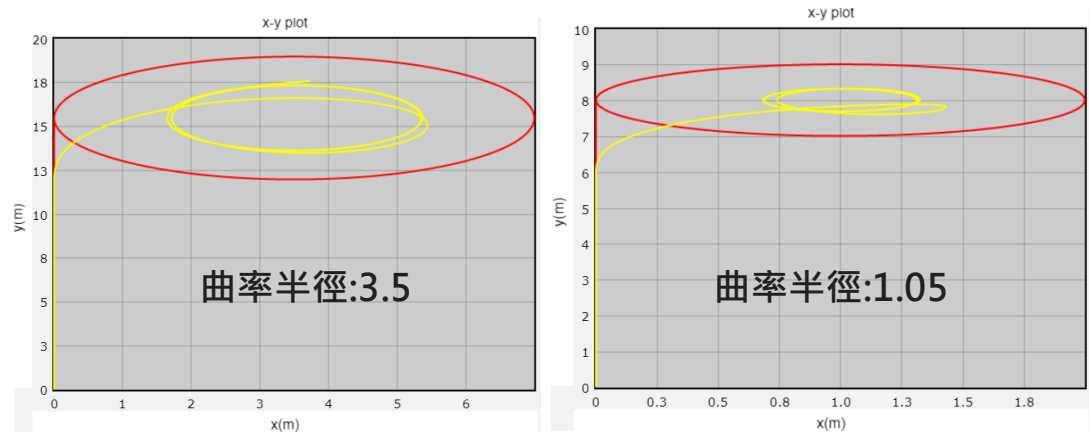


圖 9、固定 P_1 ，改變前車 ω

當前車(紅)轉彎時的曲率半徑越大，轉向加速度(ω)越小時，後車的軌跡(黃)越接近前車，前後車的軌跡半徑比越接近 1，可見前車轉佳速度較小時，後車的跟車軌跡較接近前車軌跡。

後車在起初轉向時，有較大的修正幅度，接著才會達到穩定狀態，跟前車同步繞圓。當曲率半徑較小(ω 較大)時，尚未達到穩定狀態前，偏離圓形的現象較明顯。

3. 前車沿 sin 軌跡行駛

前車以 sin 軌跡左右行駛，以前後兩車方位(θ)對時間作圖，觀察週期、振幅、波型。前車行駛方式：

$$\omega(t) = \sin t$$

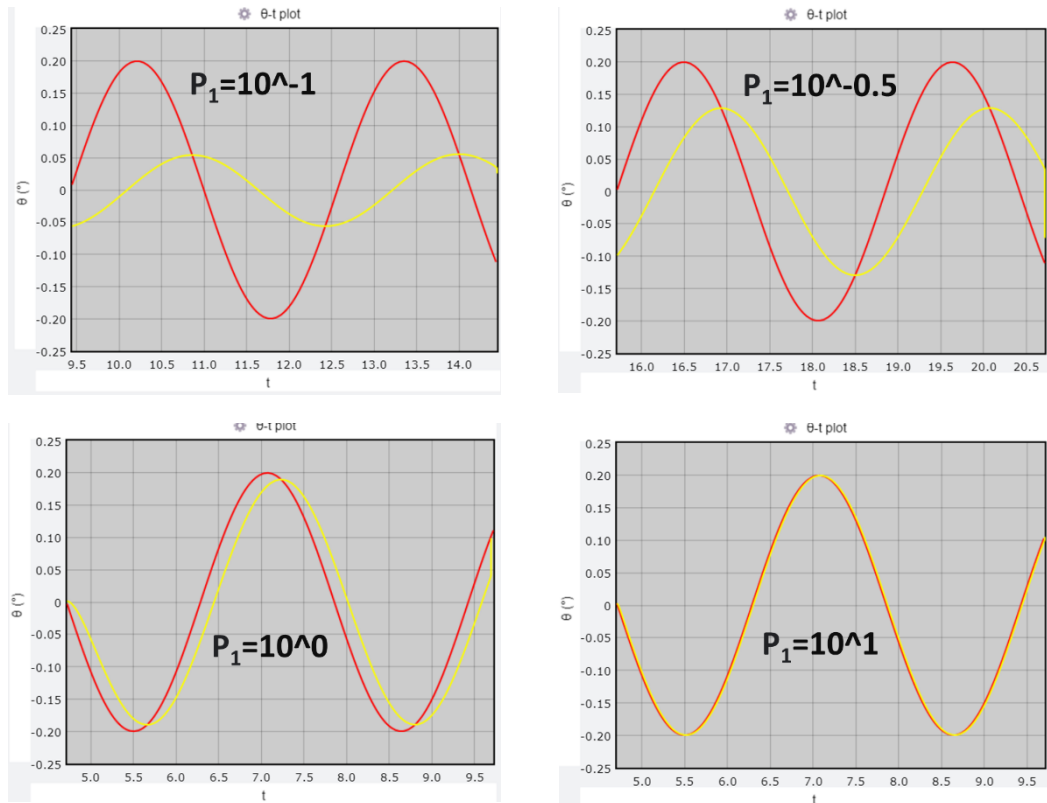


圖 10、前車沿 sin 軌跡行駛

P_1 值越小時，後車有振幅較小、波峰落後的現象，需要更多時間跟上前車狀態，但同時也減少了多餘轉向，能以更小的轉向幅度跟上前車的行駛趨勢。同時必須考慮兩者當下的行進速度，速度較慢時，可使用較低的 P_1 值以避免過多轉向，速度較快時，則必須以較大的 P_1 值，以即時跟上前車角度，避免過多的軌跡偏移。

4. 前車直角轉彎

模擬前車直線行進時轉向 90 度，觀察後車的軌跡是否能保持在道路範圍，並且轉向後仍保持在前車正後方。

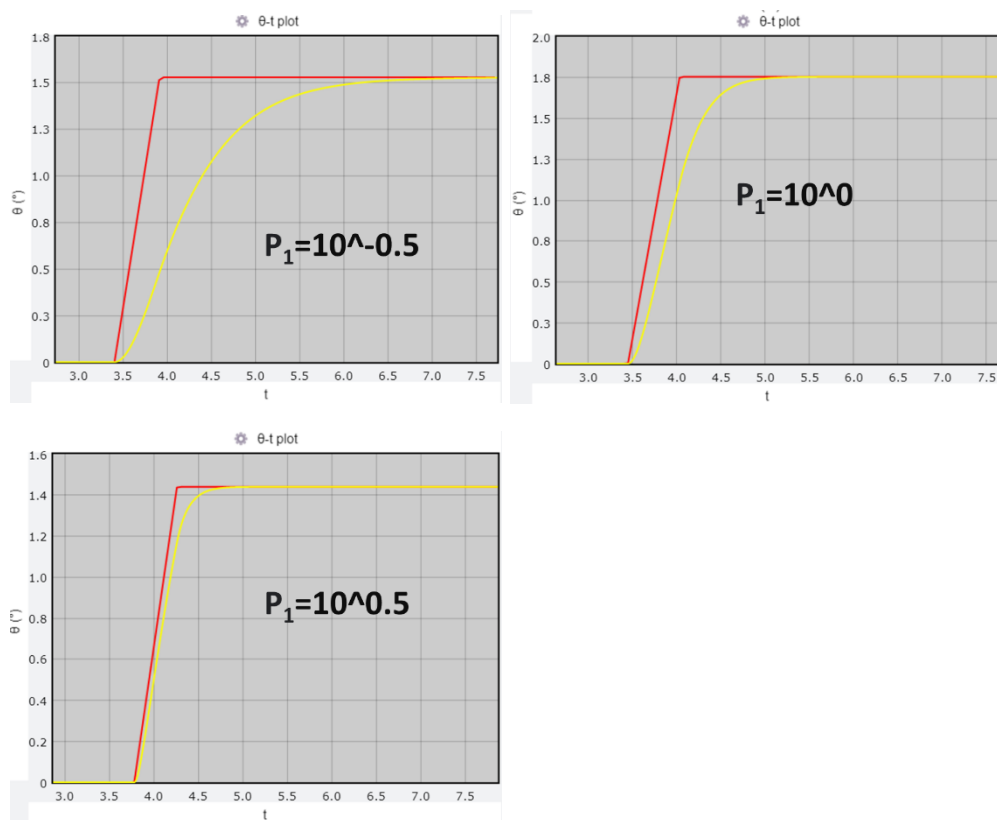


圖 11，前車 90 度轉向

前車轉向中， θ 穩定上升時， $\theta - t$ 圖中後車會在一段時間後平行前車， P_1 值越小，平行線間距越大， θ 穩定落後越多，但 ω 皆與前車一致。當前車完成轉向後， ω 回復為 0，後車 θ 漸漸接近前車，最後重合， P_1 值越小所需的時間越長， ω 變化較緩。

(二) Arduino 車輛測試

1. 後車的轉向程式

延續先前軟體模擬的控制方式，分別安裝左右兩個超音波測距儀，以此計算左右距離差，並經由 PID 控制輸出為左右兩輪的輪速差，使後車轉向，跟隨前車的行進方向。

```

void loop() {
  if((distance2-distance)*3<=50){
    rotSpeed=(distance-distance2)*3;
  }
  else if((distance-distance2)*3>50){
    rotSpeed=50;
  }
}

```

圖 12、後車的 PWM 輸出控制

2. 後車轉向性能初步測試

將車輛以斜向角度面對牆面，模擬前車角度，造成左右距離差，後車依據測量結果轉向，最後面向牆面，左右距離差歸零，不再轉向，以直線垂直向牆面駛去。

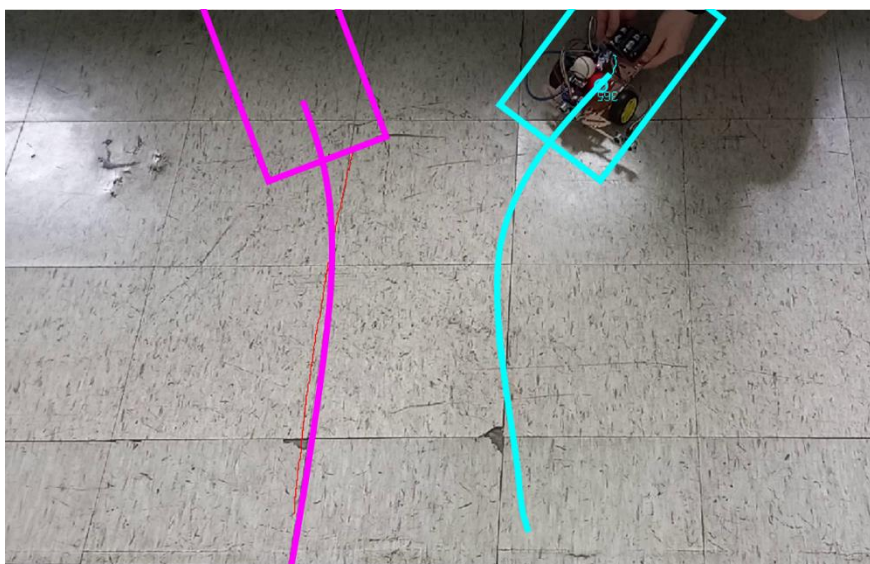


圖 13、車輛初始位置與行駛軌跡



圖 14、左側車輛 x 方向 v-t 圖

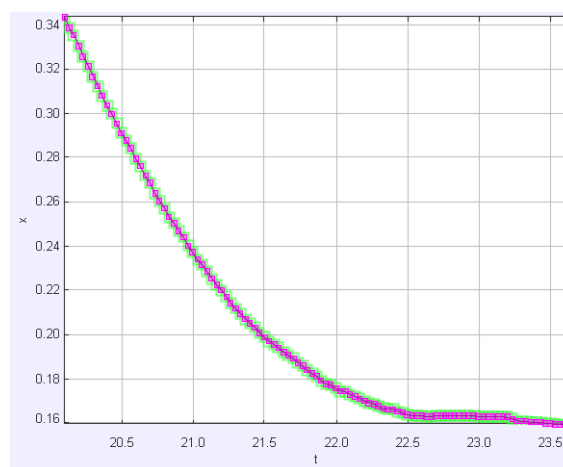


圖 15、左側車輛 x 方向 x-t 圖

左側車輛原先 x 方向速度為負值，經 PID 控制後轉向，x 方向速度漸漸接近零，x 方向速度對時間變化接近直線，x-t 圖則為凹向上的曲線，顯示 ω 隨著左右距離差漸小，最後 x-t 圖保持 0 值，在 y 方向直線行進。

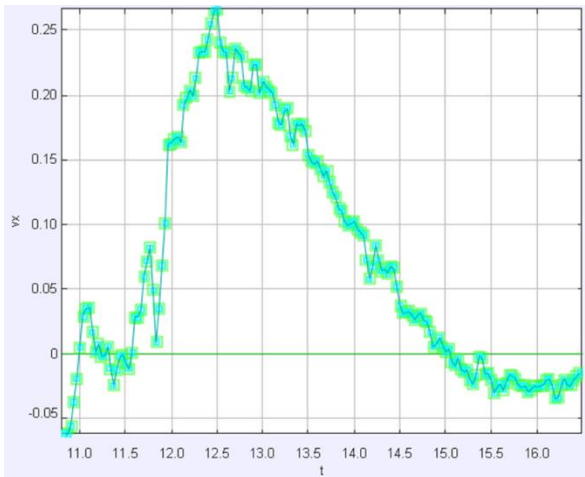


圖 16、右側車輛 x 方向 v-t 圖

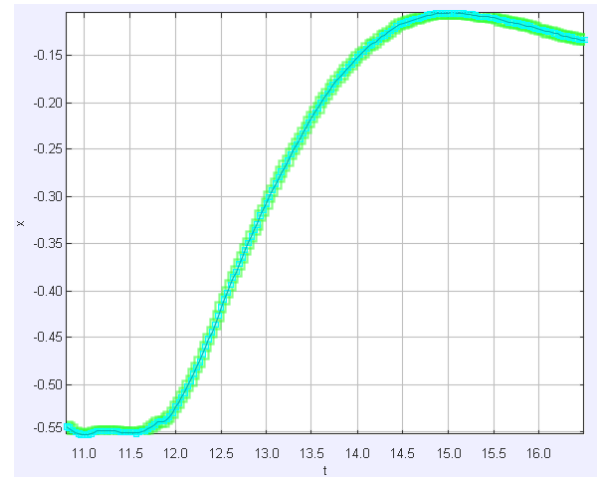


圖 17、右側車輛 x 方向 x-t 圖

右側車輛原先 x 方向速度為正值，經 PID 控制後轉向，x 方向漸小，有低於 0 的現象，再修正回 0，x-t 圖則為凹向下的曲線，因為修正而有一反曲點。可見當車輛過度轉向時，產生異號的左右距離差，車輛向反方向修正，最終垂直駛向牆面。

3. 後車轉向繞圓測試



圖 18、跟隨繞圓影片擷取

以傾斜紙板為跟隨目標，紙板以半圓、直線、半圓軌跡依序前進，並且紙板保持面對行進方向，後車根據左右感測器離紙板距離，能夠跟隨直線行進、轉向、繞圓，完成跟車測試。

三、軟體模擬的誤差修正

軟體模擬中，使用車身兩旁的距離(dL, dR)控制後車轉向，為更接近超音波真實的測距情況，計算並修正兩種方法的差異，得到準確值 dL', dR' ，PID 函式中包含左右距離差，距離差 R 修正為 R' 。

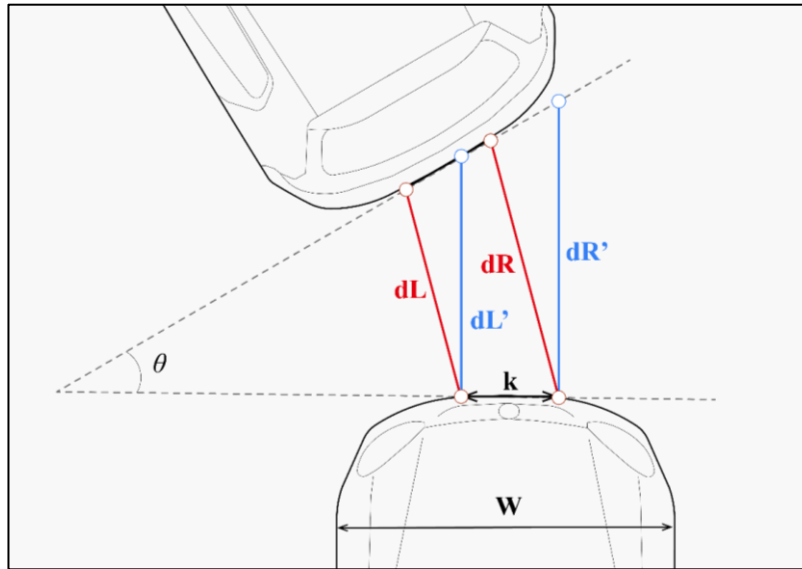


圖 19

$$\begin{cases} dL = 2 \sin \frac{\theta}{2} \cdot \left(d + \frac{W-k}{2} \right) \\ dR = 2 \sin \frac{\theta}{2} \cdot \left(d + \frac{W+k}{2} \right) \end{cases} \quad \begin{cases} dL' = 2 \tan \theta \cdot \left(d + \frac{W-k}{2} \right) \\ dR' = 2 \tan \theta \cdot \left(d + \frac{W+k}{2} \right) \end{cases}$$

$$R = dR - dL = 2 \sin \theta \cdot k$$

$$R' = dR' - dL' = \tan \theta \cdot k$$

$$2 \sin \frac{\theta}{2} \cdot k = R$$

$$\sin \frac{\theta}{2} = \frac{R}{2k}$$

$$\theta = 2 \arcsin \left(\frac{R}{2k} \right)$$

$$\begin{aligned}
R' &= R \cdot \frac{\tan(\theta) \cdot k}{2 \sin\left(\frac{\theta}{2}\right) \cdot k} \\
&= \tan\left(2 \arcsin\left(\frac{R}{2k}\right)\right) \cdot k \\
&= \frac{\sin\left(2 \arcsin\left(\frac{R}{2k}\right)\right)}{\cos\left(2 \arcsin\left(\frac{R}{2k}\right)\right)} \cdot k \\
&= \frac{2 \sin\left(\arcsin\left(\frac{R}{2k}\right)\right) \cos\left(\arcsin\left(\frac{R}{2k}\right)\right)}{1 - 2 \sin^2\left(\arcsin\left(\frac{R}{2k}\right)\right)} \cdot k \\
&= \frac{2\left(\frac{R}{2k}\right) \cdot \cos\left(\arcsin\left(\frac{R}{2k}\right)\right)}{1 - 2\left(\frac{R}{2k}\right)^2} \cdot k \\
&= \frac{\frac{R}{k} \cdot \sqrt{1 - \sin^2\left(\arcsin\left(\frac{R}{2k}\right)\right)}}{\frac{2k^2 - R^2}{2k^2}} \cdot k \\
&= \frac{\frac{R}{k} \cdot \sqrt{1 - \frac{R^2}{4k^2}}}{\frac{2k^2 - R^2}{2k^2}} \cdot k \\
&= \frac{R}{k} \cdot \frac{\sqrt{4k^2 - R^2}}{2k} \cdot \frac{2k^2}{2k^2 - R^2} \cdot k \\
&= R \cdot \frac{\sqrt{4k^2 - R^2} \cdot k}{2k^2 - R^2}
\end{aligned}$$

肆、結論與應用

一、前後車有速度差時，以距離為基礎使用 PID 控制跟車，後車速度有震盪的現象，改變

K_p 、 K_i 值可以有效改善。

二、P 控制的比例控制項、PI 控制改變積分控制項，決定系統在應對煞車、起步的靈敏度。

- 三、經過變換車道、圓型軌跡、轉向等環境測試，分析出 P 值在各個情境的作用，及其大小的優劣或限制。
- 四、使用 Arduino 車輛，經由 PID 控制、PWM 輸出，設計出以超音波測距跟隨前車方向及速度的系統，並實際檢驗其跟車性能。
- 五、分別在軟體模擬的理想環境，及真實環境的車輛輸出，總結如何以 PID 方式實現跟車系統。

伍、參考資料及其他

- [1] Tilbury, D. Introduction: PID Controller Design. Retrieved from Control Tutorial for Matlab and Simulink:
<https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlPID#7>
- [2] 許祥傑. (2016 年 7 月). 基於最佳預估控制之節能自適應巡航控. 台灣: 逢甲大學自動控制工程學系.
- [3] 陳偉杰、陳柏全、古昆隴、張志豪. (2018). 具縱向速度調變之自動換車道系統. 中華民國第二十三屆車輛工程學術研討會. 臺灣·苗栗: 國立聯合大學 能源工程學系.

【評語】 100020

本研究利用超音波傳感器感測與前車距離、相對速度等數據，以 PID 的原則設計跟車模組。這個作品構成一個不錯的 PID 教學模組。若能利用督卜勒效應感測速度，據以精進控制參數與流程，當可使作品的價值更上層樓。