**Report - Lab1**

0310705\_王郁馨 0310772\_康力仁

1. **實驗目的：**

這次的實驗主要是學會使用藍芽模組和電腦連線進行讀值。一組接在車子上面進行傳值，將車子資訊（車子角度、左輪速度、右輪速度）傳出來，再利用一組接在UNO版上的藍芽模組連接到電腦進行讀值。

1. **實驗步驟與結果分析：**
2. **設定藍芽模組：**

|  |  |
| --- | --- |
|  | KEY先不接  VCC接到5V  GND接地  TXD接到pin 10  RXD接到pin 11  STATE不用接 |

一開始需要進入AT COMMAND設定藍芽模組的資訊，藍芽模組要進到AT COMMAND，KEY要接到高電位。

將藍芽模組接到UNO版上，在將設定AT COMMAND的程式(在程式檔資料夾裡的sketch\_apr12a)燒錄到板子上。進到AT COMMAND的藍芽模組閃爍頻率會變得很慢，大概兩秒一次。燒錄完程式後須將藍牙模組拔掉再重新插入，才會進到AT COMMAND。

1. **先設定接收值的藍芽模組：**

輸入AT+ROLE:0，將此藍芽模組設成Slave，再輸入AT+ADDR，並記錄其位址。

1. **再設定傳輸值的藍芽模組：**

輸入AT+ROLE:，將此藍芽模組設成Master，再輸入AT+BIND=上面藍芽模組所得到的位址

(這裡要注意指令等號那裡要加空格，不然會產生ERROR)

兩者輸入完後，接上電源，藍牙模組如從一開始閃爍頻率極高，變成較緩慢，且兩者幾乎同步，代表連線成功。

確認成功後，再將助教給的兩份code分別燒入車子上的NANO板與電腦端的UNO板，就可以開啟read\_id.m開始讀取車子上的資訊了。

1. **透過藍芽傳接收值：**

藍芽模組連線之後，開啟read\_id.m讀取車子傳送過來的資料

在車子自己平衡的狀態，重複做了兩次，結果紀錄如下表：

|  |  |  |
| --- | --- | --- |
| 1st結果 | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 2st結果 | | |
|  |  |  |

由兩次結果發現，車子的傾角會一直再0度上下震動，因車子為了保持平衡，需要讓車子保持在0度左右，但因整個控制系統還沒設定到非常穩定，所以震動幅度也較大。

而從上面結果也可發現，左輪速度和右輪速度幾乎是一樣的，因為車子再做平衡的時候，左右輪幾乎是同步往前往後的。

1. **透過實驗計算車子的參數：**
2. **將PWM設為0，計算ρ3和ρ4**

將車子傾角從0度自然掉落，重複做五次，所得結果如下：

(result/1/1~5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1st | 2nd | 3rd | 4th | 5th |
|  |  |  |  |  |

再下降部分取一小段(我們取五個數)，丟入anglesmooth.m檔中，再將運算出的結果丟入leastsquare\_p3p4.m運算ρ3和ρ4，結果如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1st 取18~22 | 2nd 取17~21 | 3rd取17~21 | 4th取11~15 | 5th取22~26 |
| ρ3 | 2.49938 | 2.73913 | 3.3 | 2.69908 | 3.05916 |
| ρ4 | -5.08923 | -6.28121 | -9.192522 | -6.552467 | -8.76021 |

因為這是逼近算法，所以將五份資料的五個值取平均，得到一組最準確的ρ3和ρ4，得：ρ3 = 3.0437，ρ4 = -7.6662。

其算法如下(Least Square)：

將ψ"-ρ3ψ‘-ρ4sin(ψ) = 0改寫成Ax = b的形式 🡪 [ψ‘ sin(ψ)] [ρ3 ρ4] = ψ"

帶入𝑥=(𝐴^𝑇 𝐴)^(−1) 𝐴^𝑇 𝑏，求出ρ3 ρ4

此算法是在藉由已知多個ψ值，透過最小誤差的平方找出最適合的解。

1. **將PWM設為87，計算ρ1、ρ2和σ1、σ2、σ3、σ4、σ5**

將車子從傾角0度，放在地面上，車子因馬達施力而倒下，重複做五次，再將得到的值，丟入程式資料夾裡的o1o2.m結果如下：

(result/2/1~5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1st | 2nd | 3rd | 4th | 5th |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1st 取28~32 | 2nd 取16~20 | 3rd取26~30 | 4th取15~19 | 5th取27~31 |
| ρ1 | -35.7645 | 38.0879 | -24.8153 | -64.5516 | -15.6376 |
| ρ2 | 46.9746 | -150.1848 | -30.4831 | 36.4200 | 74.0450 |

由上表可觀察，ρ1ρ2相差滿大的，我們認為應該是因為在取馬達轉動角時，因為左右輪速度已經為0了，這樣取道的轉動角也為0，可能會造成誤差。但還是將五組資料的五個點取平均，得到一組最接近的值：ρ1 = -20.5387、ρ2 = -4.6832。

|  |  |  |  |
| --- | --- | --- | --- |
| ρ1 | ρ2 | ρ3 | ρ4 |
| -20.5387 | 4.6832 | 3.0437 | -7.6662 |

算法一樣用Least Square，公式如下：

Ψ”+ρ1θw”+ρ2cos(ψ)θw”-ρ3(θw'+ψ')-ρ4sin(ψ)=0，ρ3 = 3.0437,ρ4 = -7.6662

🡪Ψ”= [θw” - cos(ψ)θw” θw'+ψ' sin(ψ)][ ρ1 ρ2 ρ3 ρ4]

利用𝑥=(𝐴^𝑇 𝐴)^(−1) 𝐴^𝑇 𝑏，求出ρ1 ρ2。

將read\_id.m裡讀到的(SPEED\_R+SPEED\_L)/3.45資料丟進程式資料夾裡的vs.m檔裡進行運算，結果如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1st 取28~32 | 2nd 取16~20 | 3rd取26~30 | 4th取15~19 | 5th取27~31 |
| σ1 | -11.9043 | NaN | -9.5303 | NaN | -17.1599 |
| σ2 | 0.3715 | NaN | 4.7183 | NaN | 2.7861 |
| σ3 | -22.7497 | NaN | 68.8477 | NaN | 3.1386 |
| σ4 | 1.5507 | NaN | 0.276 | NaN | 7.1681 |
| σ5 | 0.3363 | NaN | -0.2753 | NaN | -0.1605 |

和ρ1、ρ2一樣，每組誤差大，甚至有兩組算不出值，但還是取五組資料的平均，得到一組最接近的值：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| σ1 | σ2 | σ3 | σ4 | σ5 |
| 2.1831 | -5.2418 | -151.275 | 5.2752 | -2.1064 |

算法一樣用Least Square，公式如下：

θw”+σ1ψ”+σ2(2cos(ψ)ψ”-sin(2ψ)sec(ψ)ψ')-σ3ψ'+σ4θw'=σ5vs，其中vs = 87

🡪θw” = [-ψ” -2cos(ψ)ψ”-sin(2ψ)sec(ψ)ψ' ψ' -θw' vs][ σ1 σ2 σ3 σ4 σ5]

利用𝑥=(𝐴^𝑇 𝐴)^(−1) 𝐴^𝑇 𝑏，求出σ1 σ2 σ3 σ4 σ5。

1. **模擬結果：**

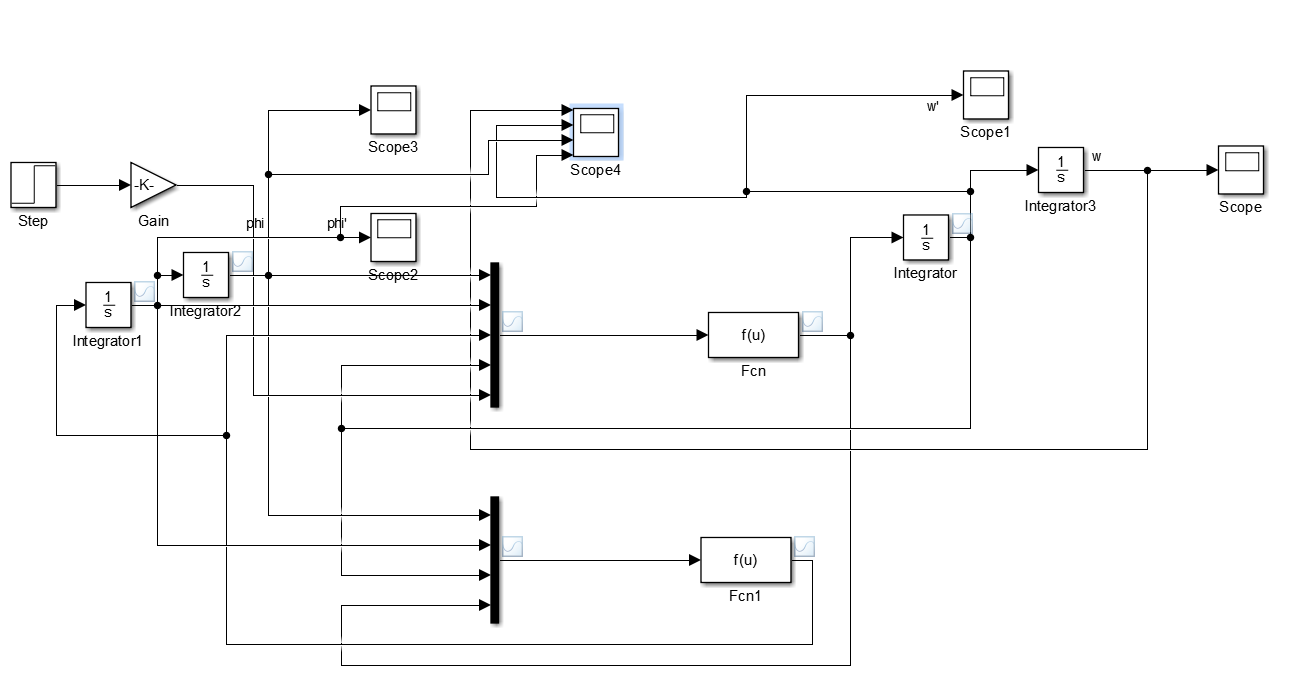
利用上面實驗所求得的車子參數：ρ1、ρ2、ρ3、ρ4、σ1、σ2、σ3、σ4、σ5，還有兩項方程式：

Ψ”+ρ1θw”+ρ2cos(ψ)θw”-ρ3(θw'+ψ')-ρ4sin(ψ)=0 -- (1)

θw”+σ1ψ”+σ2(2cos(ψ)ψ”-sin(2ψ)sec(ψ)ψ')-σ3ψ'+σ4θw'=σ5vs --(2)

使用simulink來模擬整個系統。

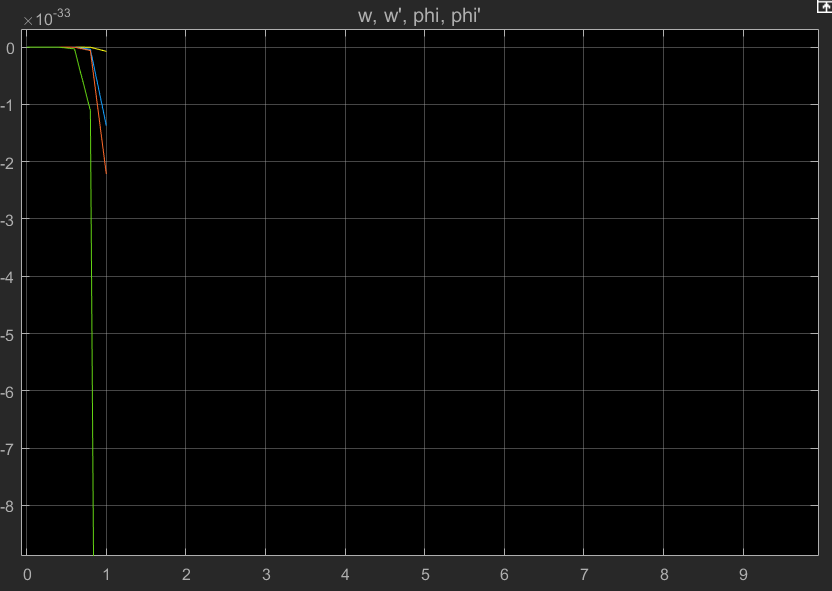
模擬電路圖如下(程式資料夾裡的simu.slx)：



Fcn的f(u) = -2.1064\*u(5)-2.1831\*u(3)+5.2418\*2\*cos(u(1))\*u(3)-5.2418\*sin(2\*u(1))\*(1/cos(u(1)))\*u(2)-151.275\*u(2)-5.2752\*u(4)，其輸出為θw”

Fcn1的f(u) = 20.5387\*u(4)+4.6832\*cos(u(1))\*u(4)+3.0437\*(u(3)+u(2))-7.6662\*sin(u(1))，其輸出為Ψ”

下圖為θw、θw’、ψ、ψ’的圖：



分開顯示：

|  |  |
| --- | --- |
| θw | Ψ |
|  |  |
| θw' | Ψ’ |
|  |  |

1. **問題與討論：**
2. 一開始藍牙設定時，一直發生ERROR，後來發現是AT的指令要加上空格。
3. 後來連線完成後，跑matlab讀值，一直讀到很奇怪的數字，甚至程式執行起來一直有錯誤訊息，後來發現是在車子上的藍牙模組裝錯地方了。車子上有BLU1和BLU2，一開始很直覺地接到BLU1，但比較電路圖和arduino code後發現是要接到BLU2。接對之後問題就解決了。
4. 在測試讓車子倒下的實驗時，我們發現，如果要讓車子往+90倒下，電腦端會在車子倒下時突然讀不到值，因為藍芽模組會有點受到撞擊而掉出，導致藍牙斷線。後來往-90度倒進行實驗，就沒有這個問題了。
5. 在網路上給的兩份matlab檔(anglesmooth和leastsquare\_p3p4)一開始跑出的值都特別大，後來發現要將t由0.01改為0.1，算出來的值才是正確的。
6. 當車體快速倒下時，三軸陀螺儀有時反應不及，只會有60度的傾角(實際觀測為90度)，且一次實驗重複測量時，角度會有累積誤差。所以之後可能會考慮換個較好的三軸陀螺儀。
7. **心得感想：**

這次在做實驗的時候一開始有一點無助，因為不太知道實驗要求為何，也因為之前從來沒有接觸過，所以有點迷惘。不太知道甚麼程式該用，或該怎麼用，所以花了很長一段時間研究。時間有點趕，因為有很多莫名的錯誤，光是解決就花了很長的時間了，還有測出來的東西也不太確定其正確性。之後還需要再做更深入的了解。