雙輪車系統鑑別&建模

組員:楊翰祥0510379、宋其諭0510888

實驗目的:

簡化系統，使用轉移函數進行設計模擬傾角達到平衡，並再用藍芽的方式手機調整PID來達到平衡。

1. **雙輪車公式重新推導**

H(q)+ C(q.)+ G(q)=Qu

H(q)=ha+

C(q.)=

,

G(q)=

H(q)+ C(q.)+ G(q)=Qu

代入乘開可得

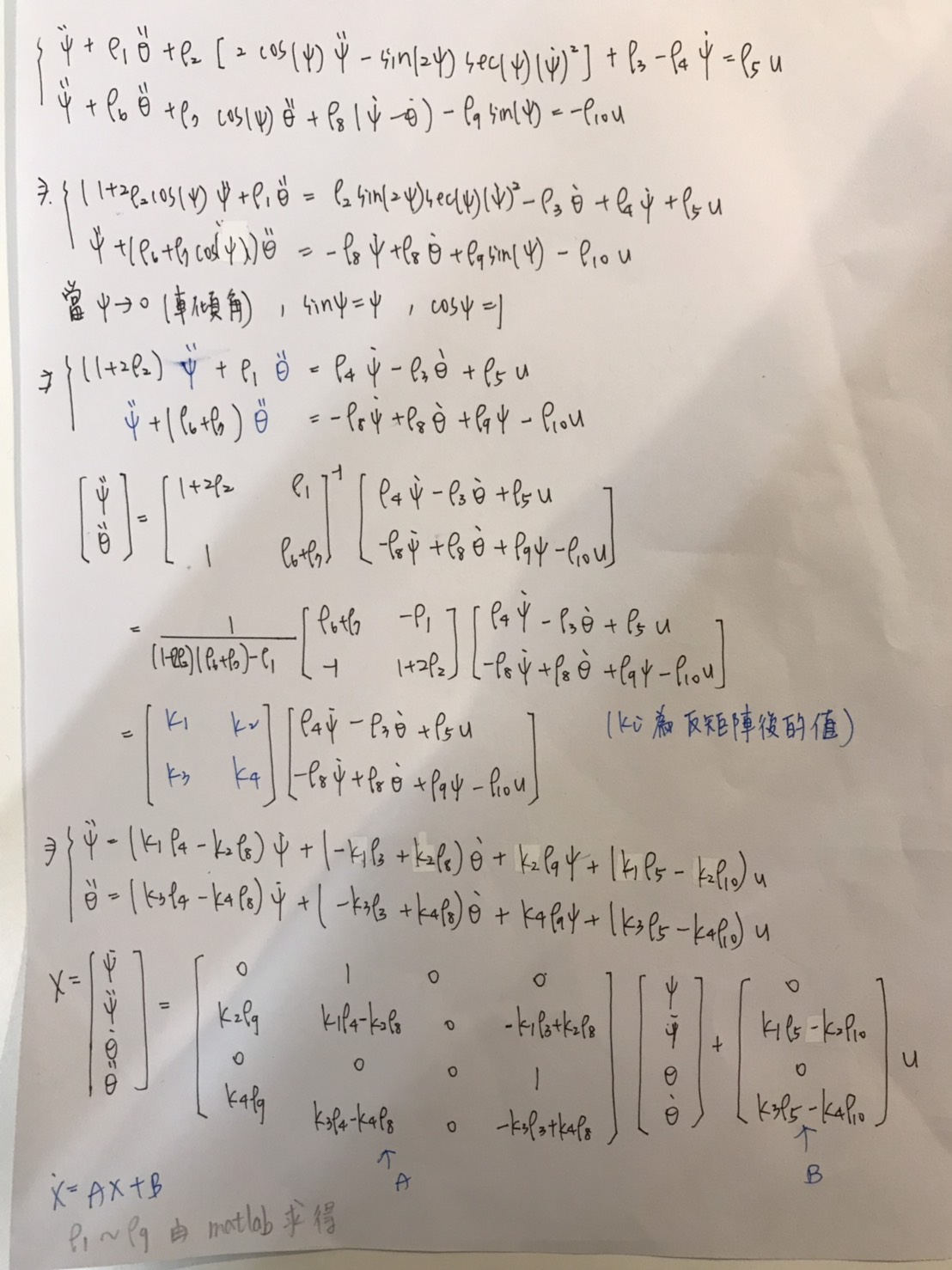
++ =

展開後比較第一列和第三列後可得以下兩條方程式 ()

關係式

接著因為考慮到車傾角角度在小角度時候滿足、cos(，

後利用最小平方法來推導。下圖為詳細的導證過程

****

1. **進行ID**

由於之前收集車子傾角、車輪角、和PWM資料時出現延遲的BUG，並將收集資料的function寫進“BalanceBotEncoder”、馬達驅動寫進”BalanceBotMotor”的library(原本全部放在Arduino)

檔案:

arduino: car.ino

matlab接收資料: receiveData.m

matlab分析: ID\_part1.m 、 ID\_part2.m

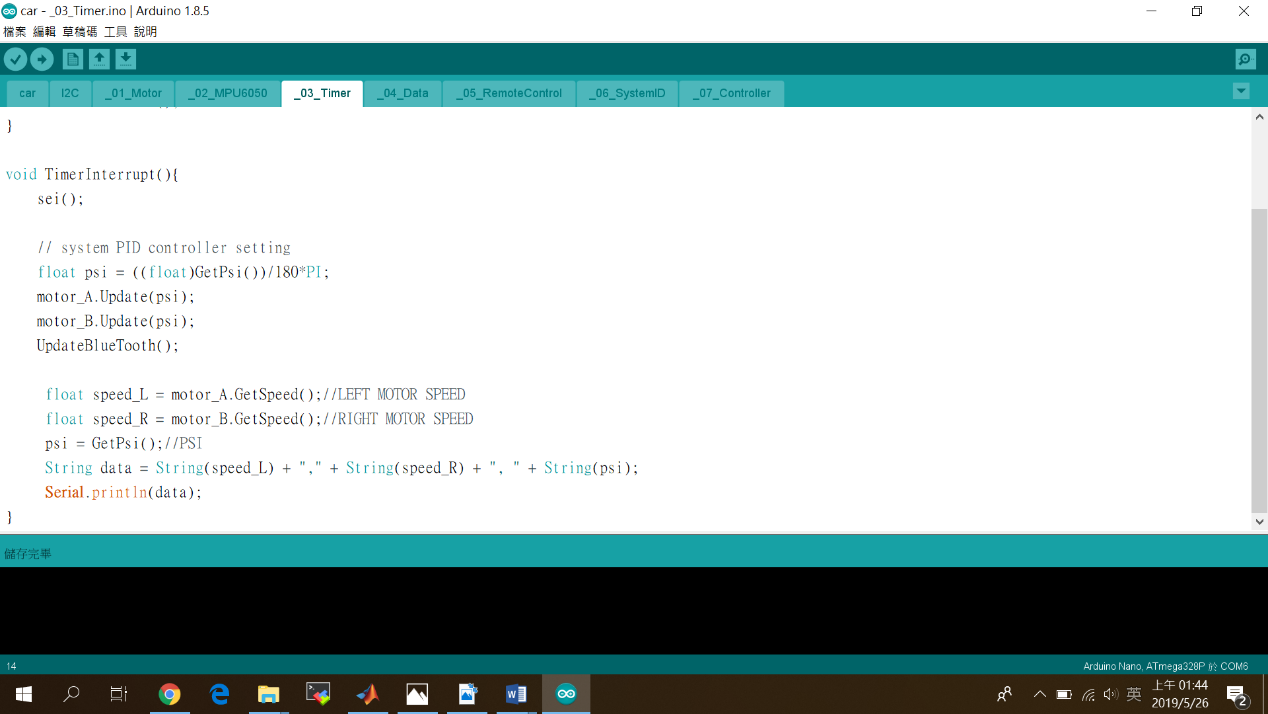
matlab 狀態矩陣: id\_controllable\_matrix

matlabc 畫跟軌跡、頻域分析: Root\_locus\_and\_blode.m

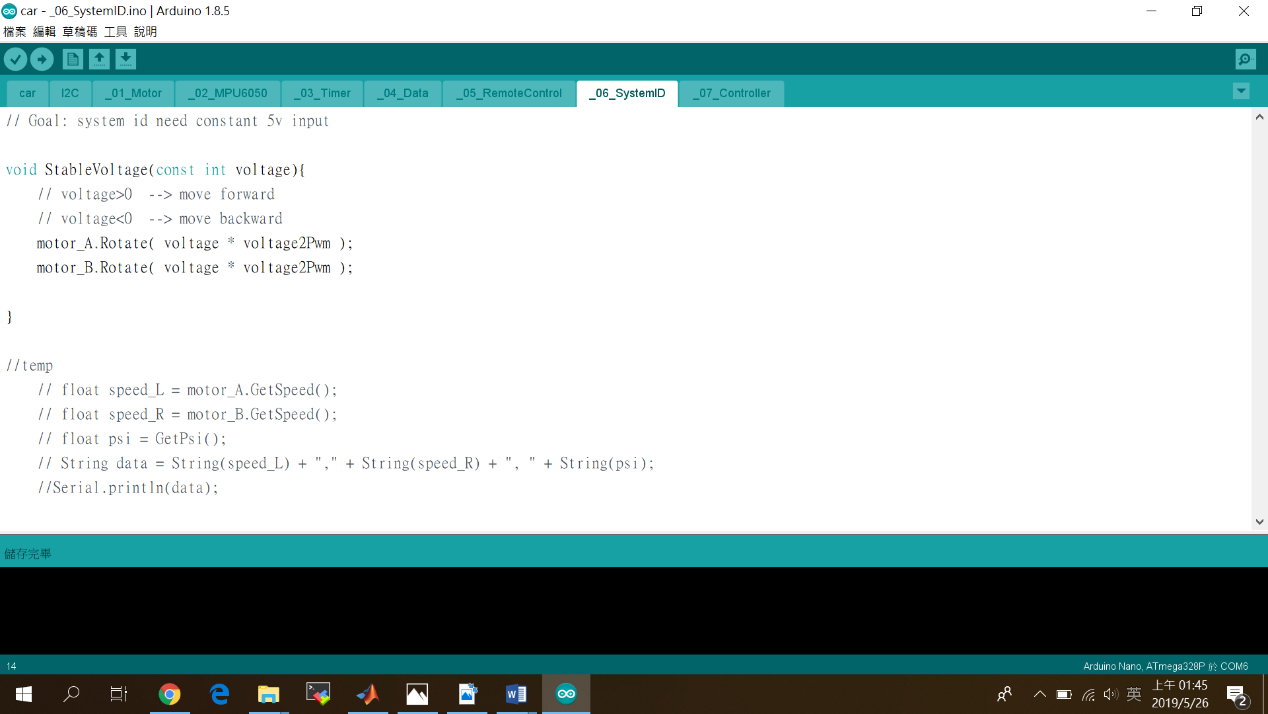
(i)收集系統參數

1.Arduino端:

(每0.05秒時間中斷來不斷更新街收到左右輪速度和車傾角)

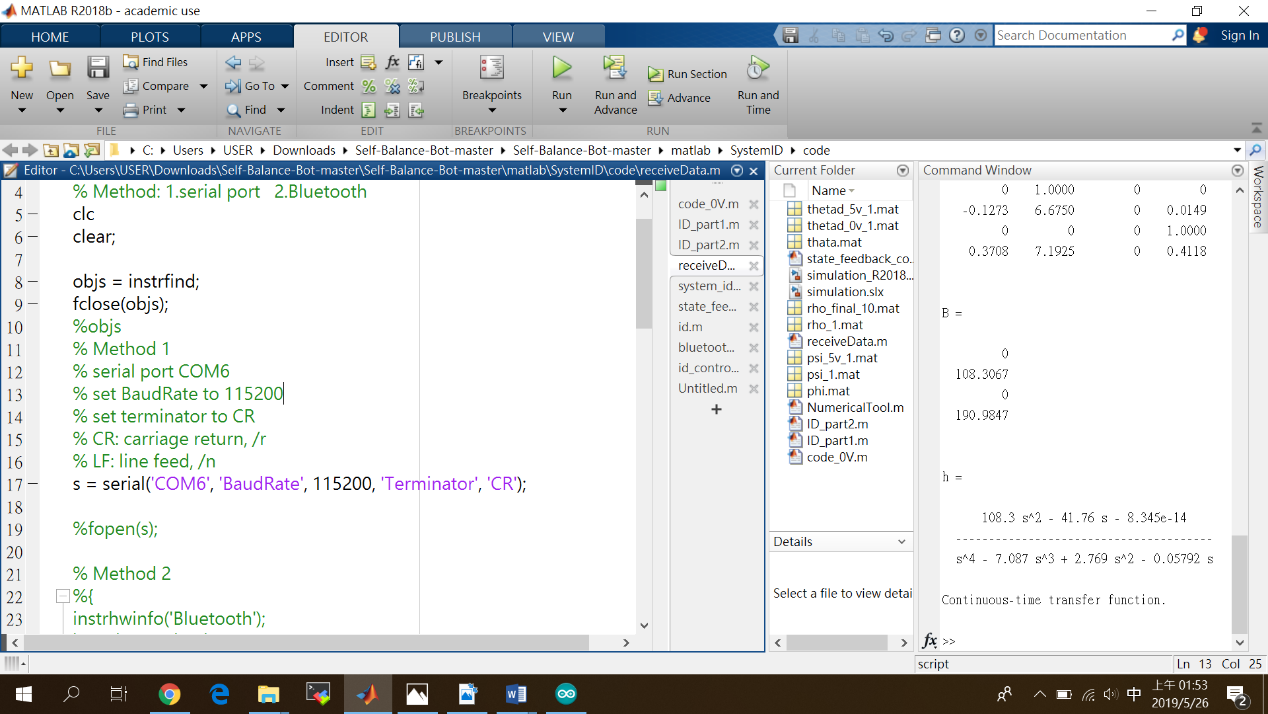
(將收到的資料包成一個string傳出去)

(當做5V分析時輪轉輸出)

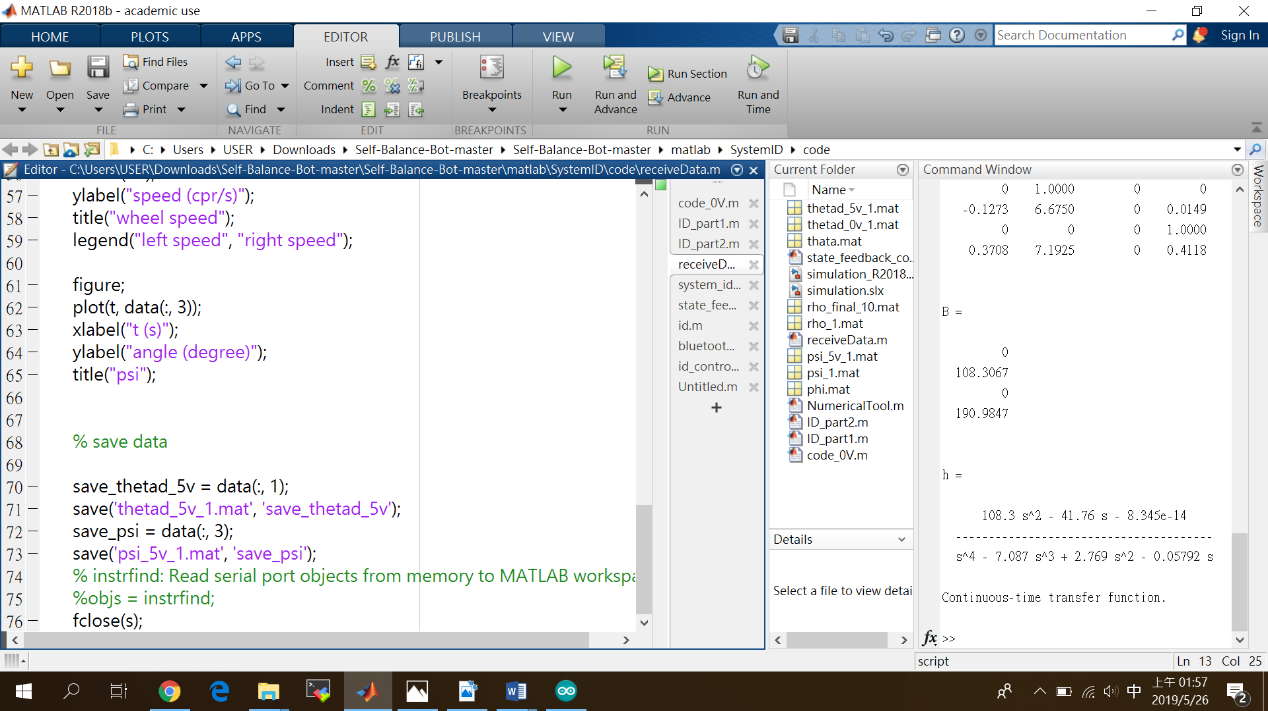
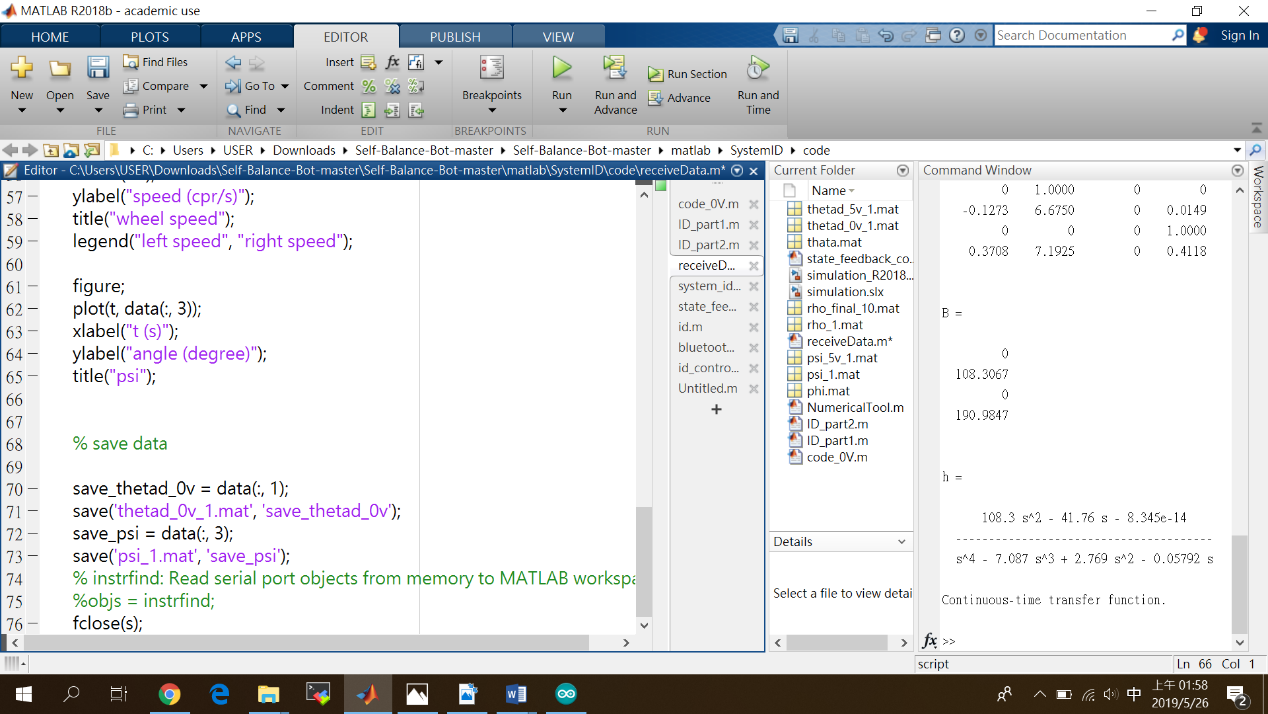


2.Matlab端(收資料->id):

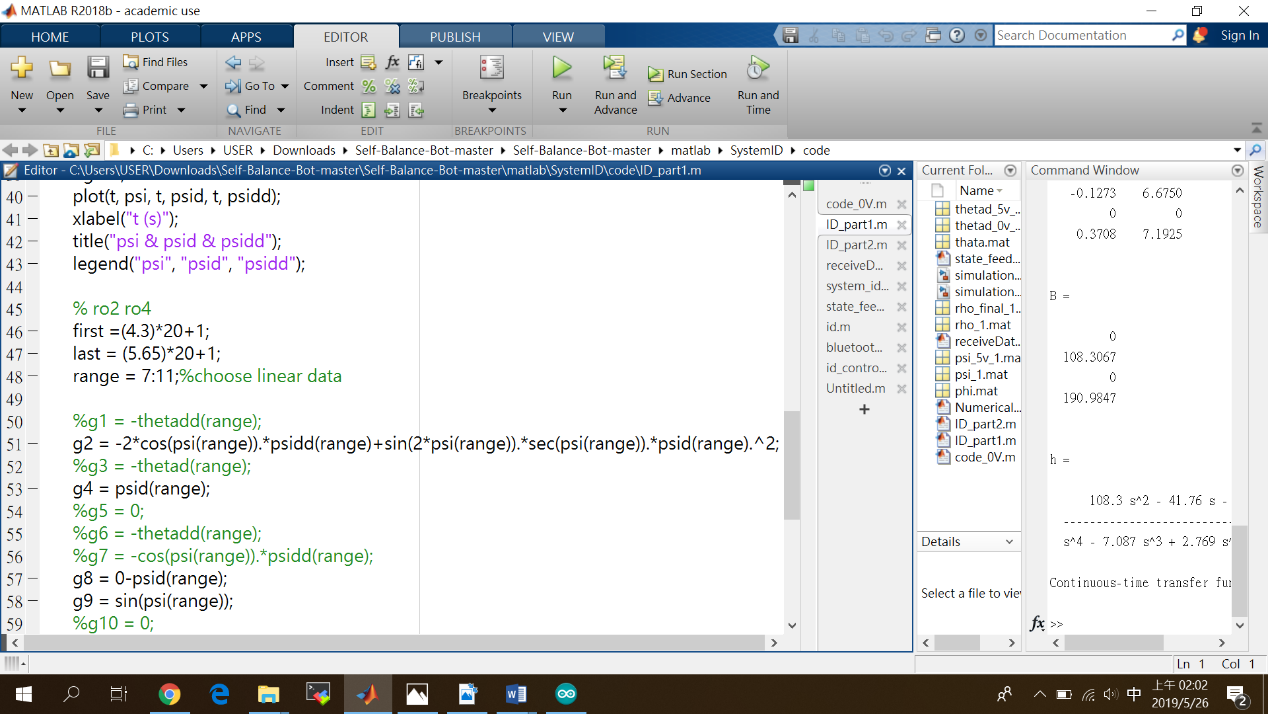
暫時使用serial port



將收進來兩組0V和5V的資料分別存在.mat檔，並嘗試將所鑑別出來的參數進行推導得出系統的轉移函數

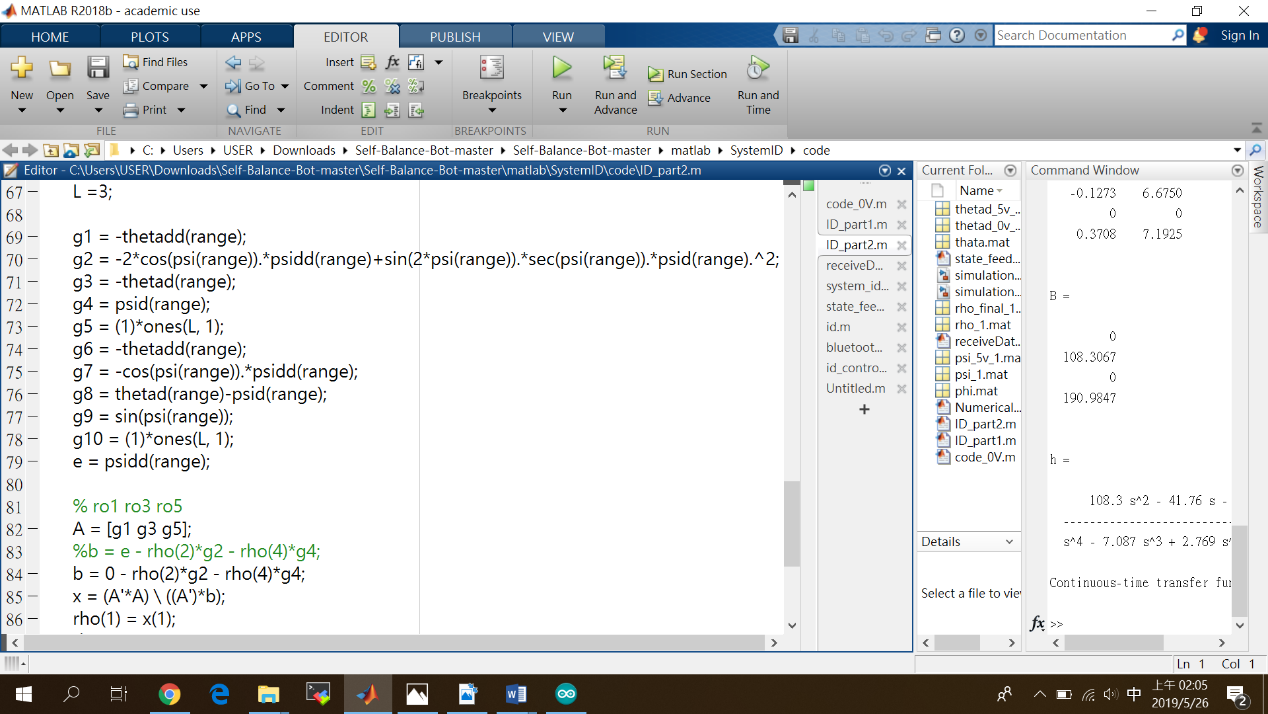


然後帶入推導的state equation

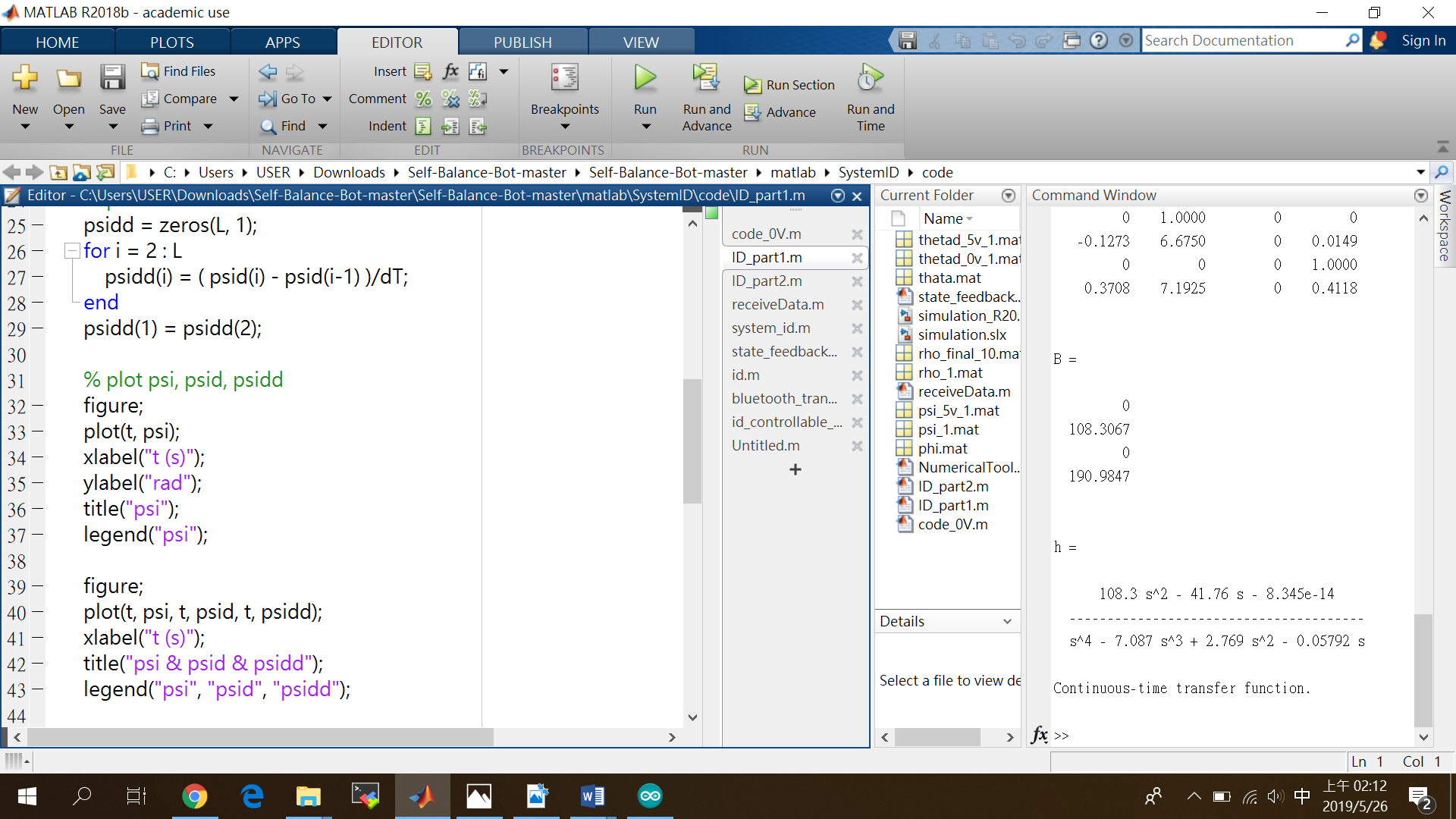
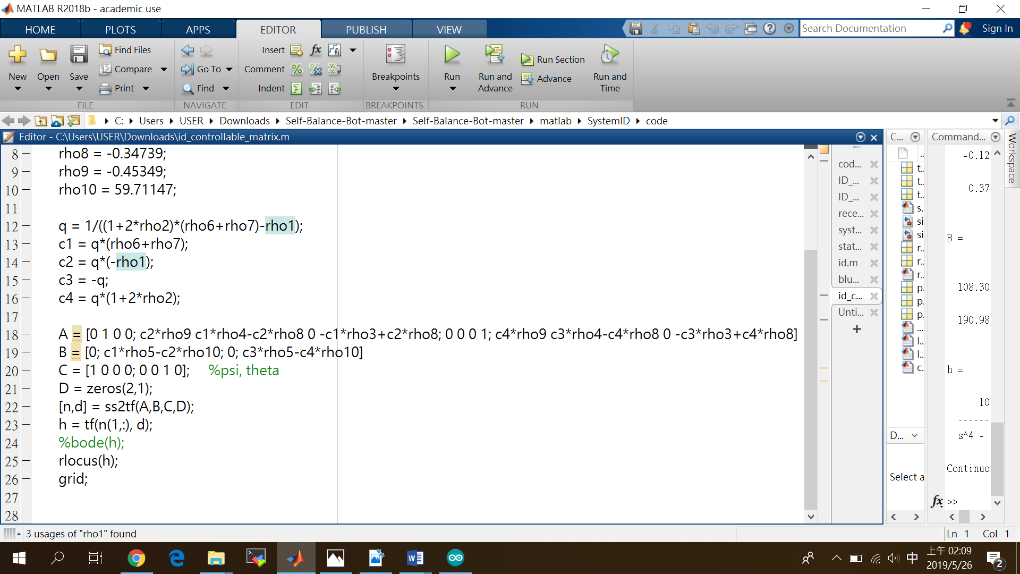
(0V僅和rho2、rho4、rho8、rho9有關)

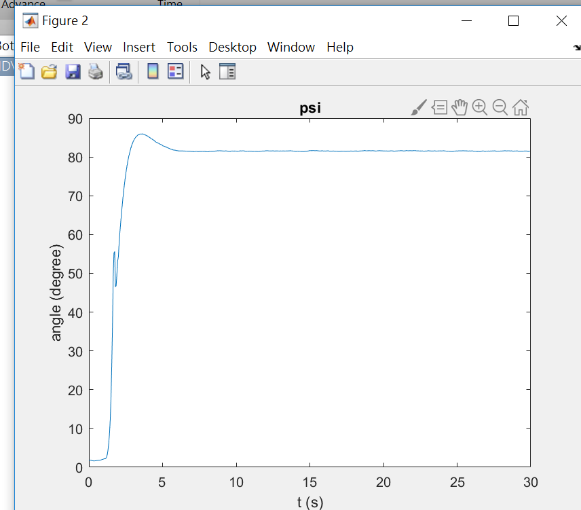
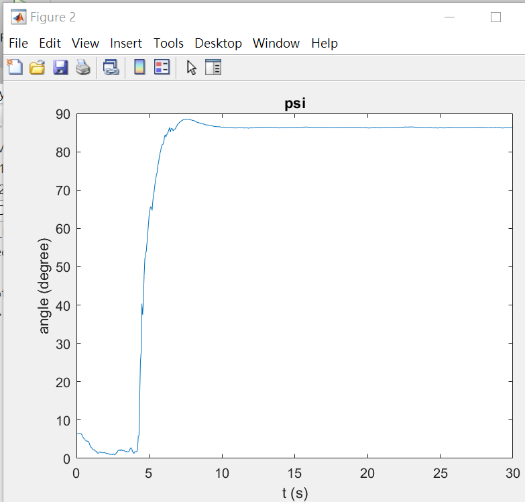
0V

再將已求得的rho2、rho4、rho8、rho9值傳到ID\_part2的5V分析中

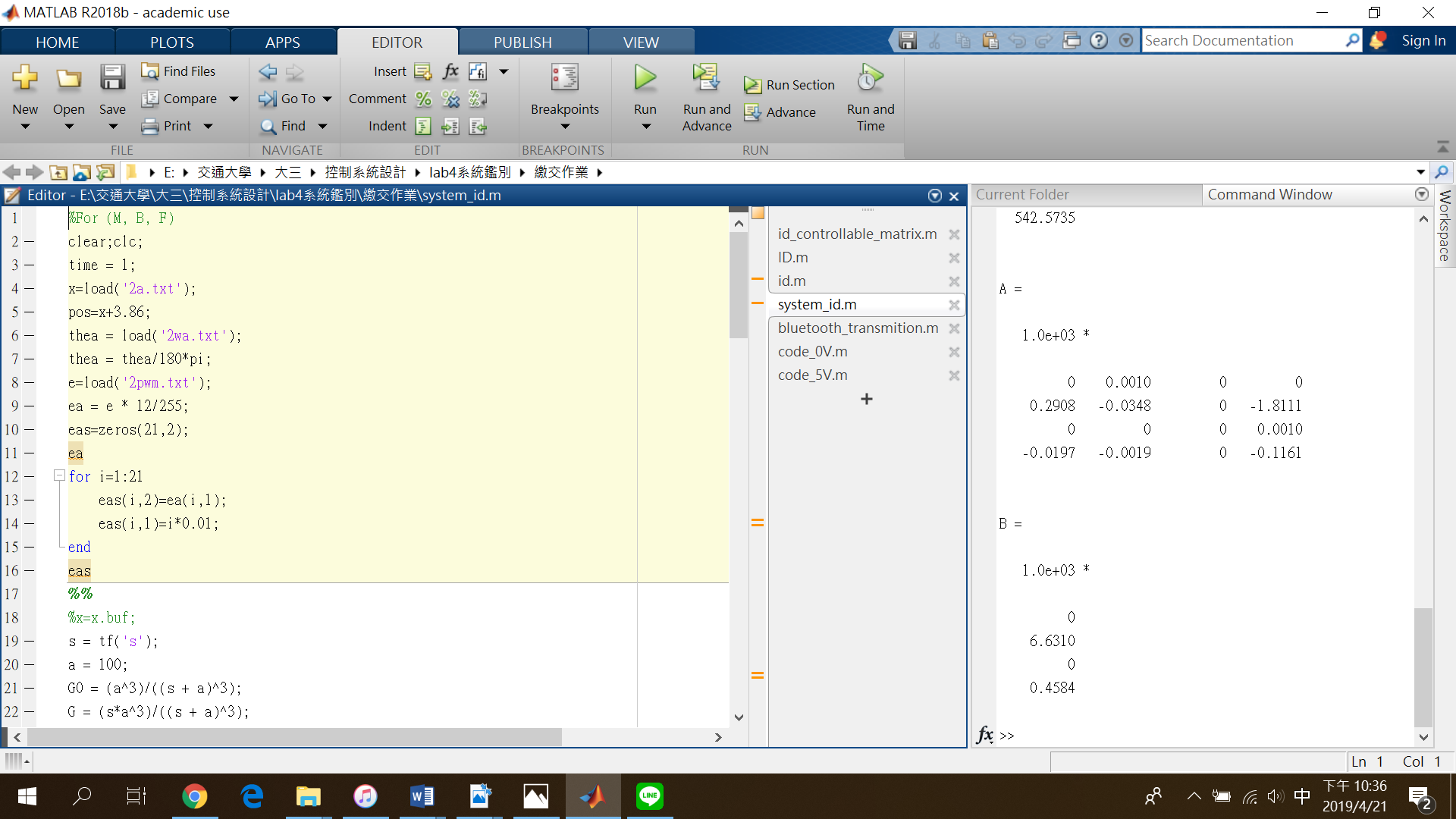


然後算出狀態方程式、根軌跡和車傾角的時間作圖

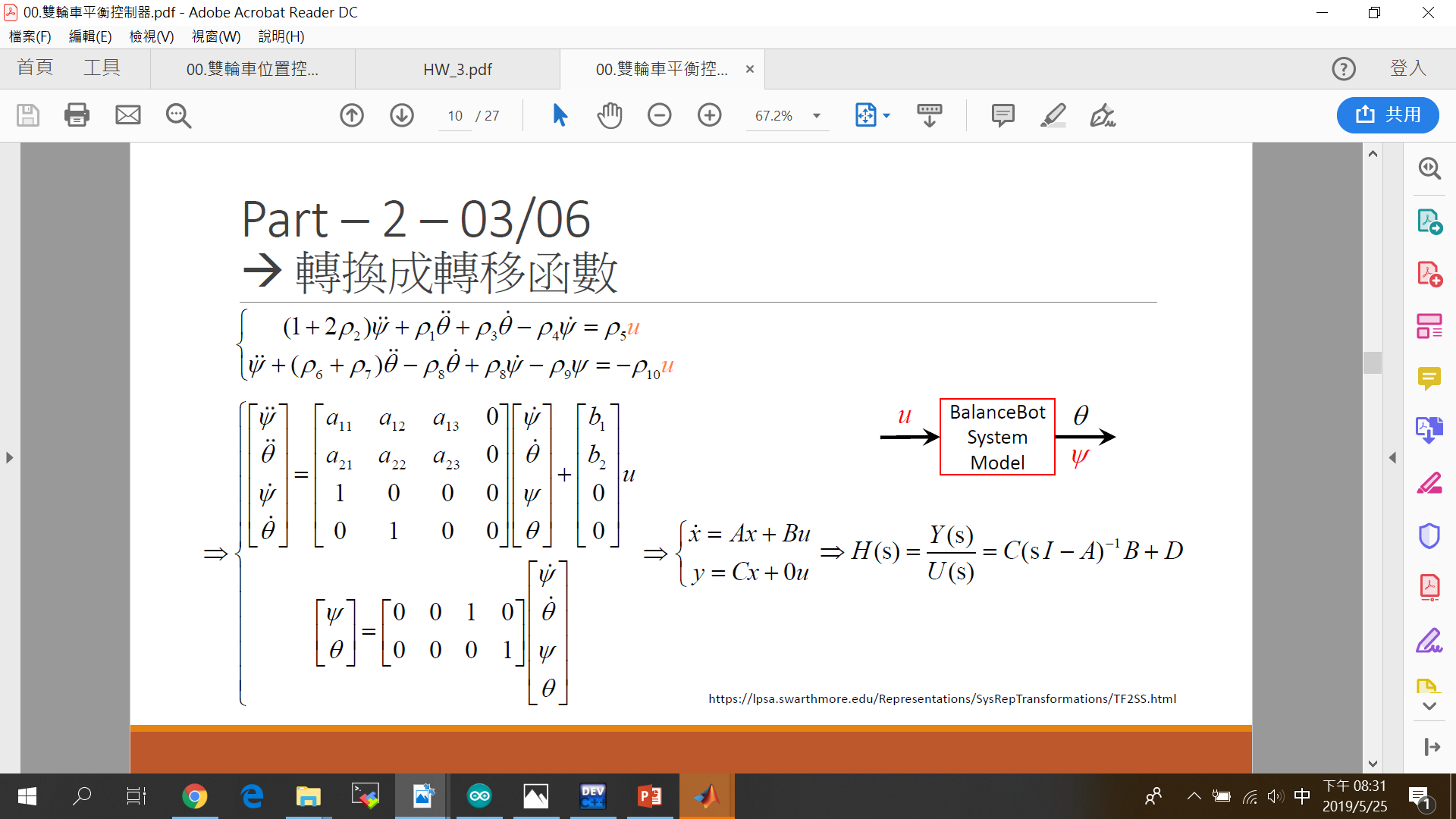
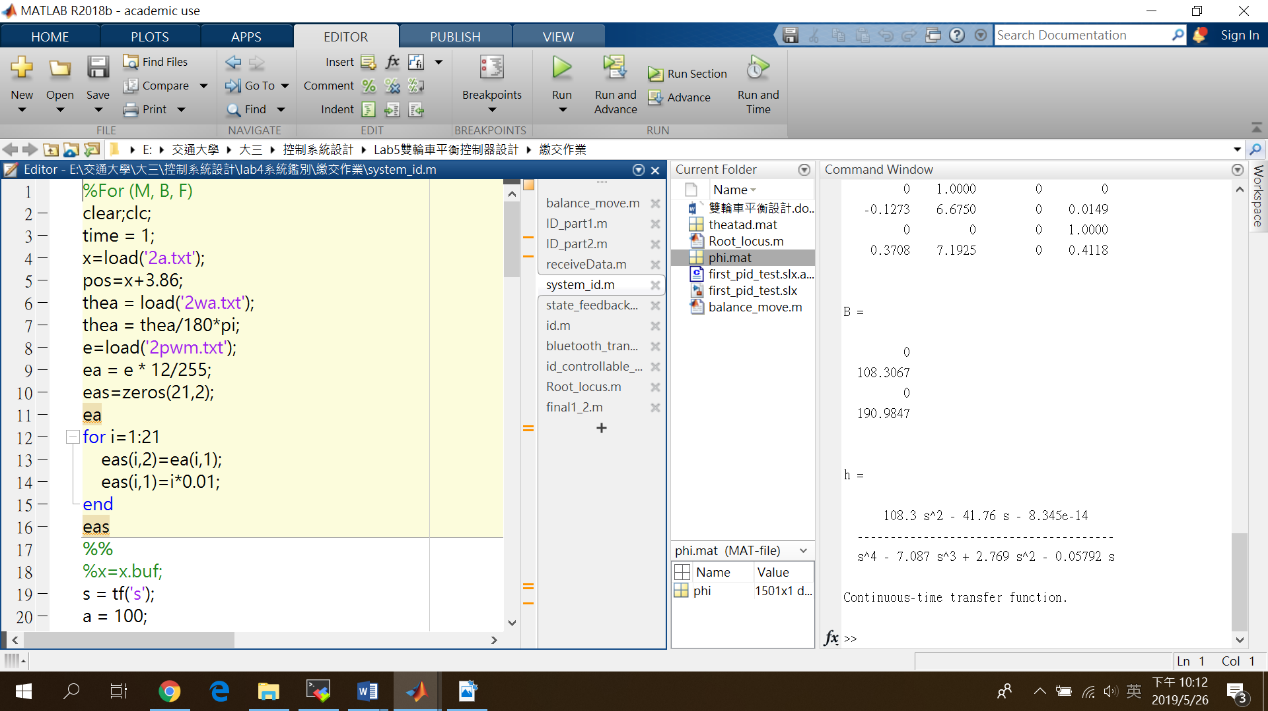


0V5V

最後算出的狀態矩陣 、

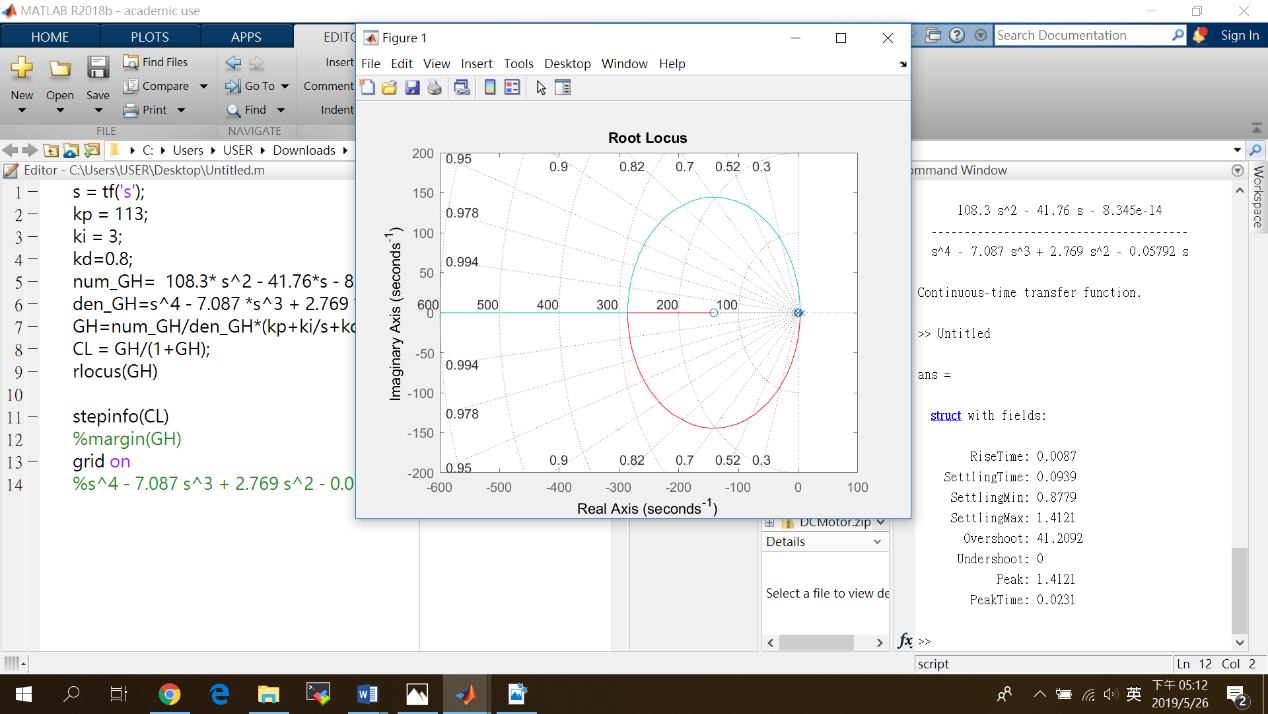
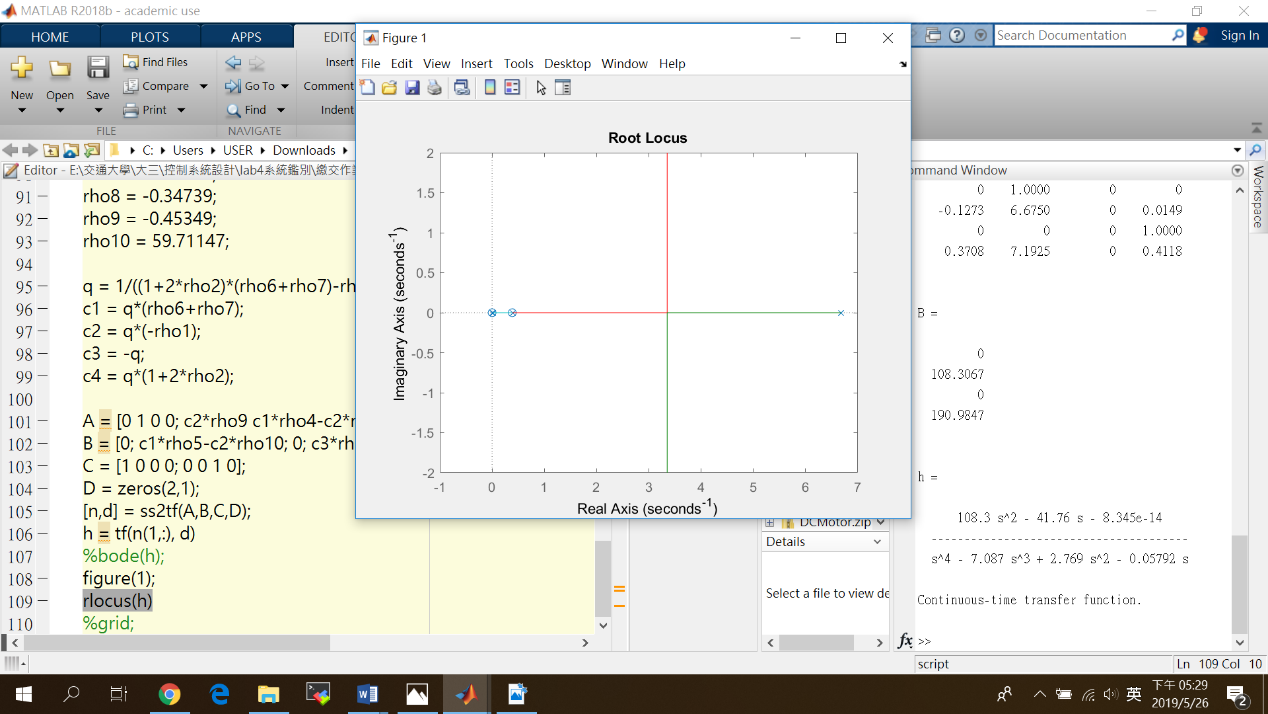
C = [1 0 0 0; 0 0 1 0];

註[B AB AB2 AB3]算出為full rank，可知系統為controllable system

又可求轉移函數: 

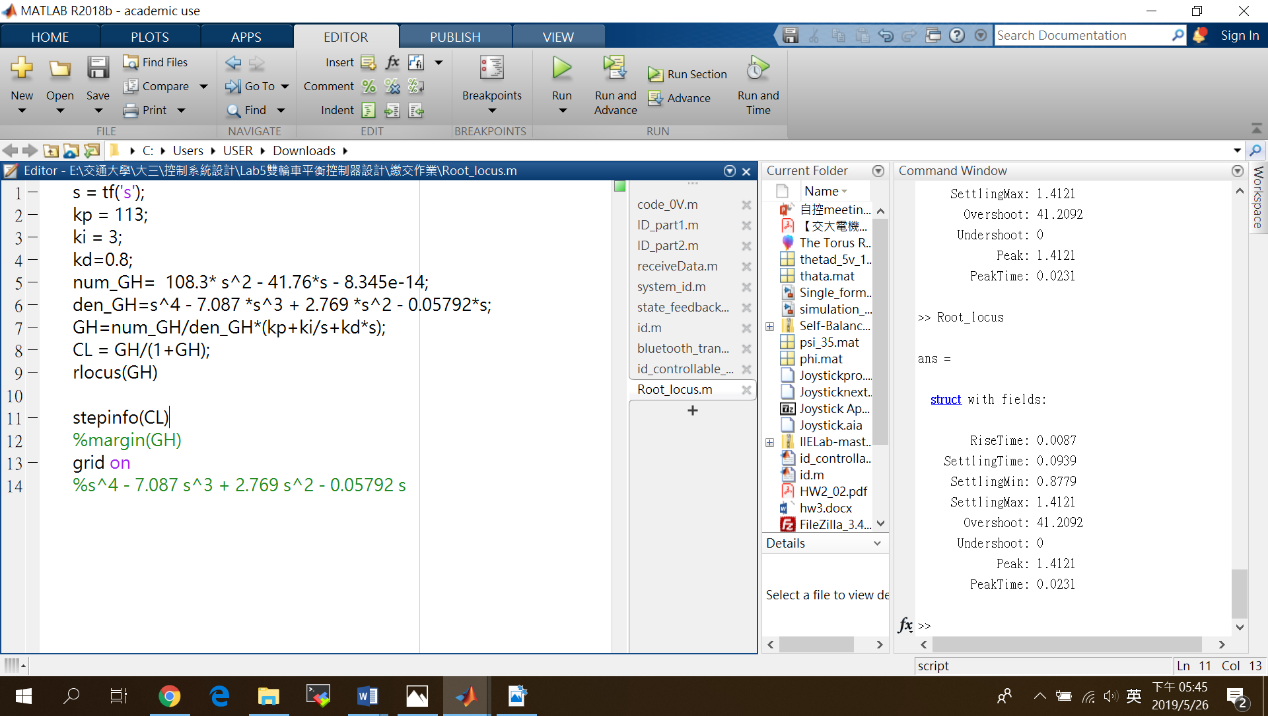
(ii)導證

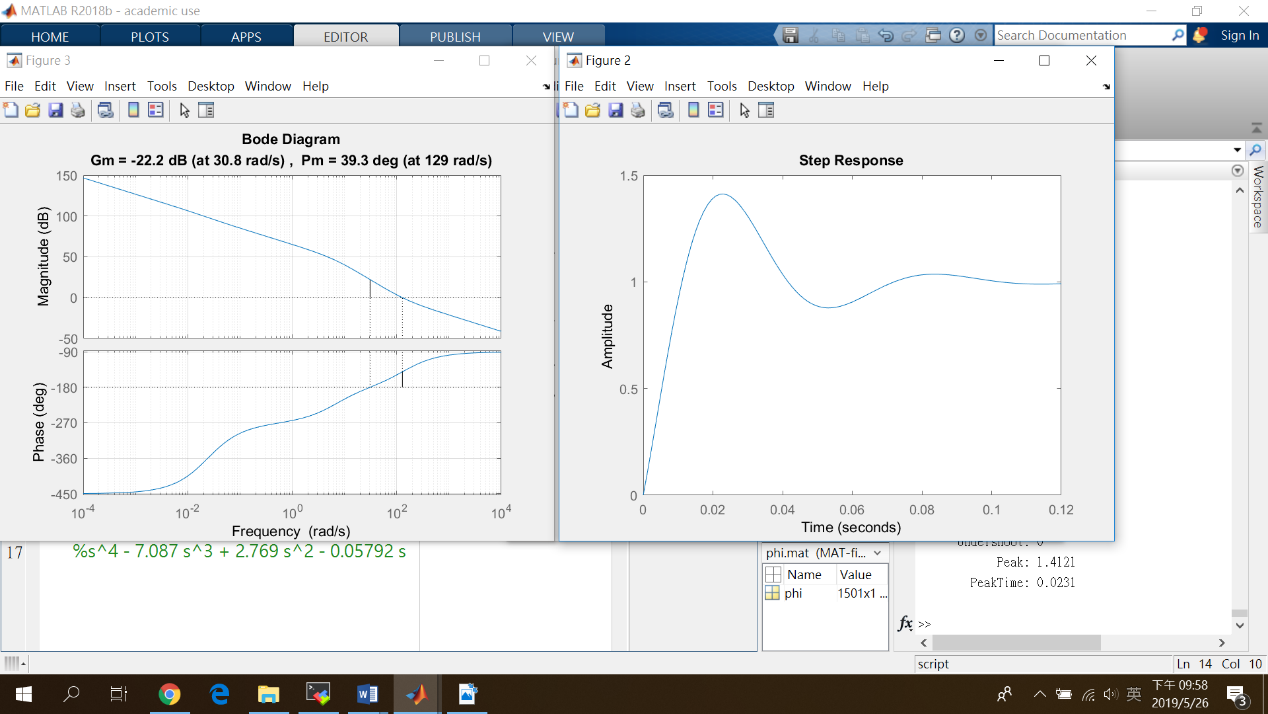
由原系統Root locus 可看出圖形皆在右半平面，會造成系統不穩定的狀況發生(無法平衡)，故我們嘗試將放入不同的pid把圖形拉到左半平面，有相對應的zero與其相消，達到pole-zero cancellation使得系統仍為穩定系統。(目前的經驗是加入遠離原點的pole較能讓系統達到往左移動的效果

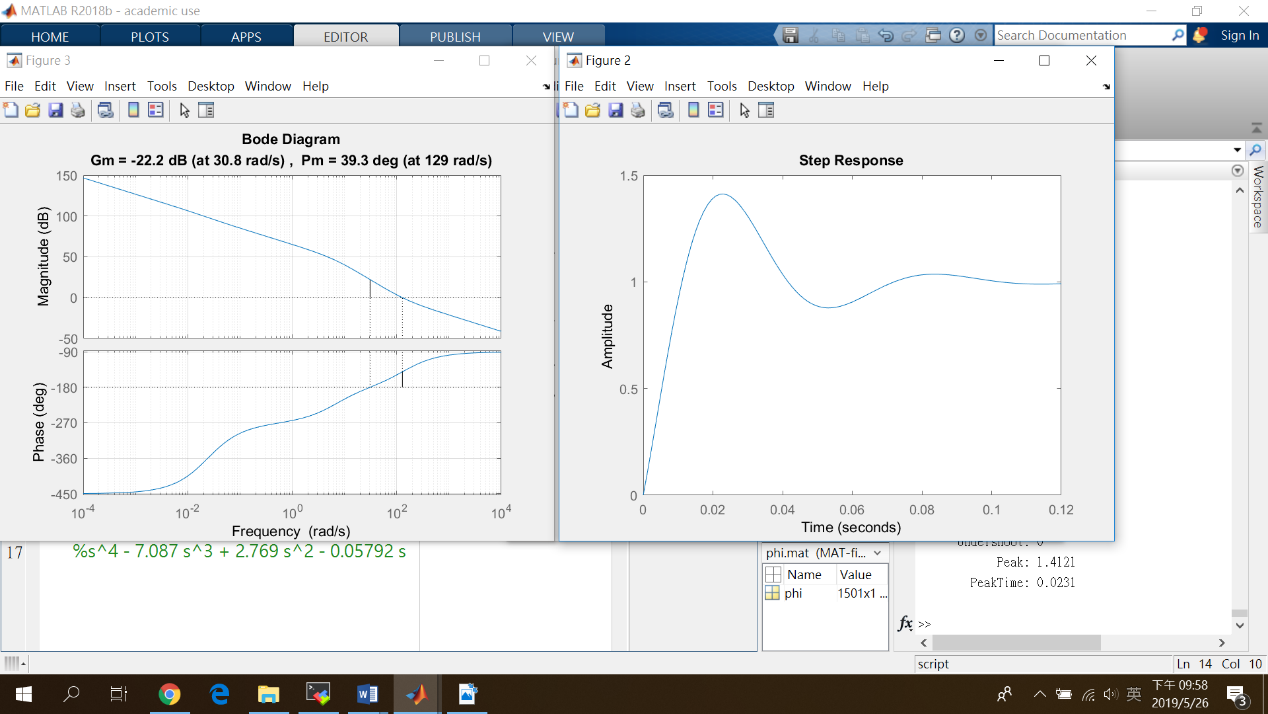


加入PID後

未加入PID後

其系統的上升時間(0.0087)、穩定時間(0.0939)皆在合理範圍內

且其頻域分析blode diagram為

輸出時域圖為

故將此系統放simulink做模擬。

1. **雙輪車系統模型(Simulink)**

**檔案Simulink:** **first\_pid\_test.slx**

**Ψ=0**

**PID controller**

**System**

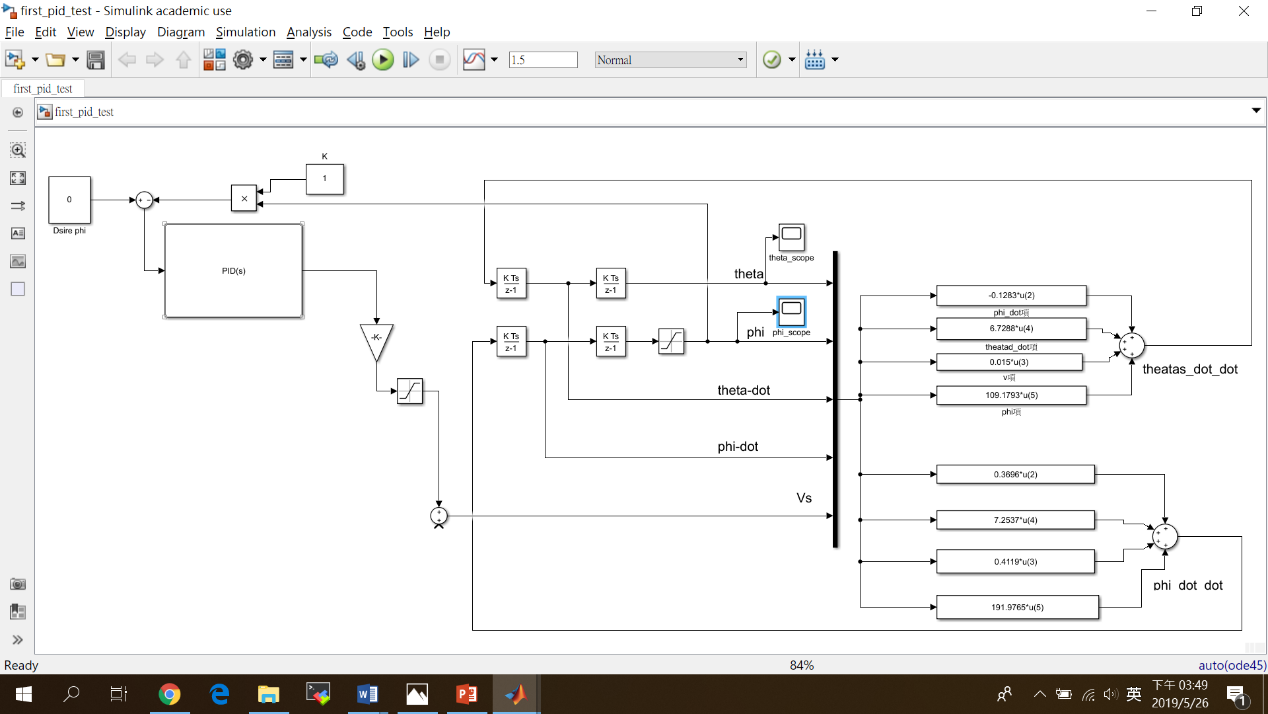
車**傾角Ψ**

**+**

**-**

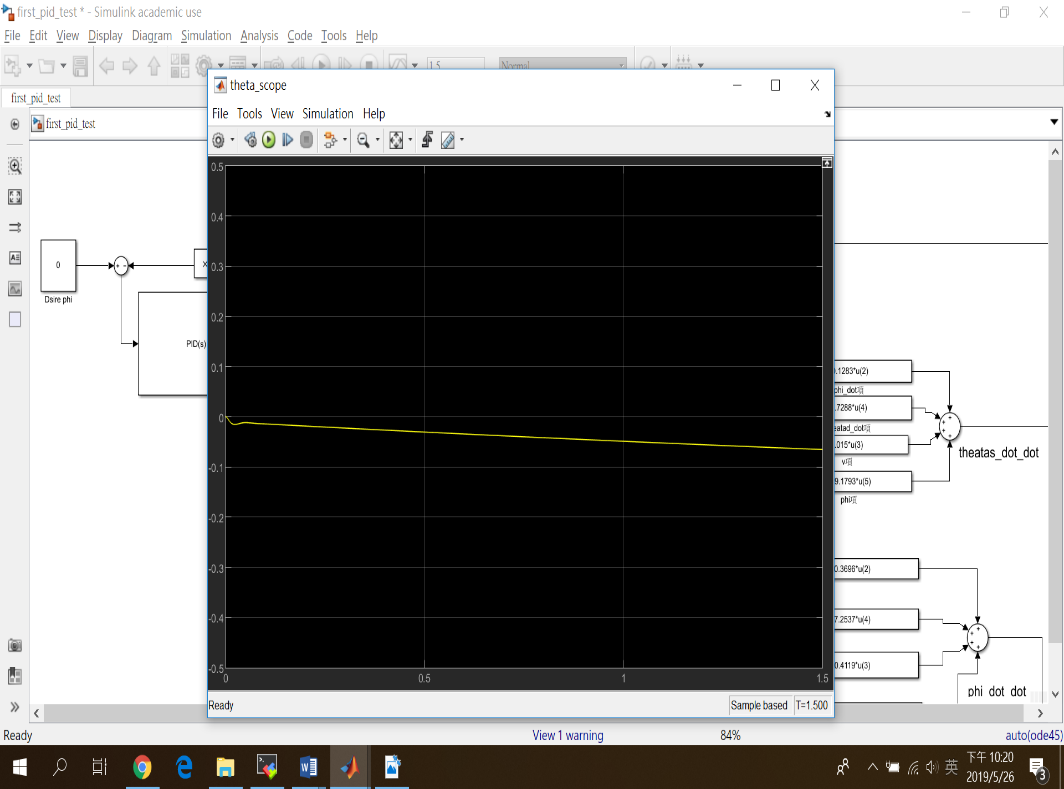
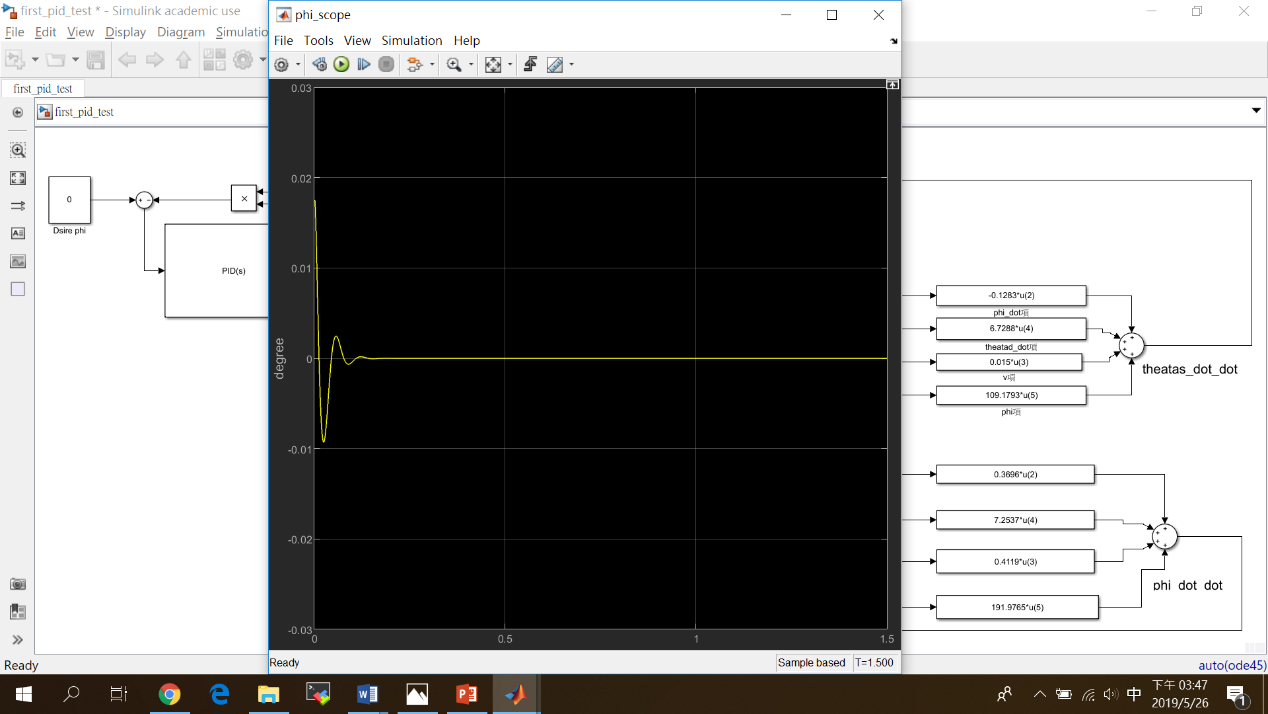
**Vs**

**e**



**我們將車傾角直接回授到pid**

時間:sec 角度:degree



使用PID控制器(P=100、I=3,D=1)

從模擬可發現此組數據是可以達到平衡效果。

1. **實現**

先將寫好的Ardunio燒入硬體，並用藍芽傳輸的方式，遠端於手機連線，直接在硬體上將PID值做改變。

檔案:

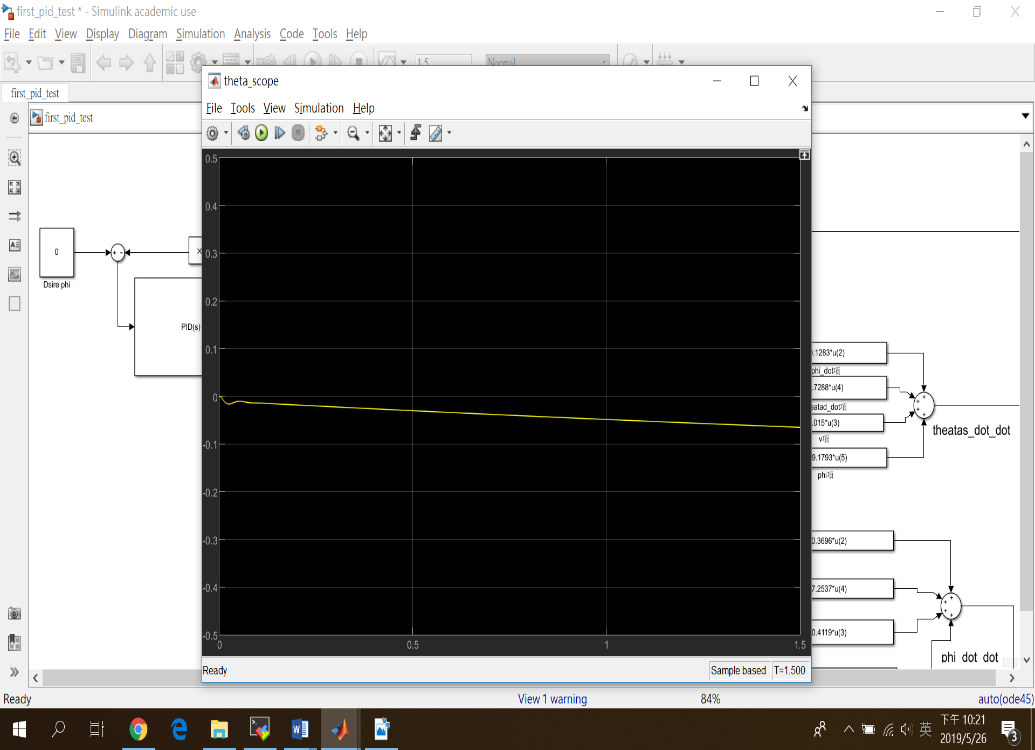
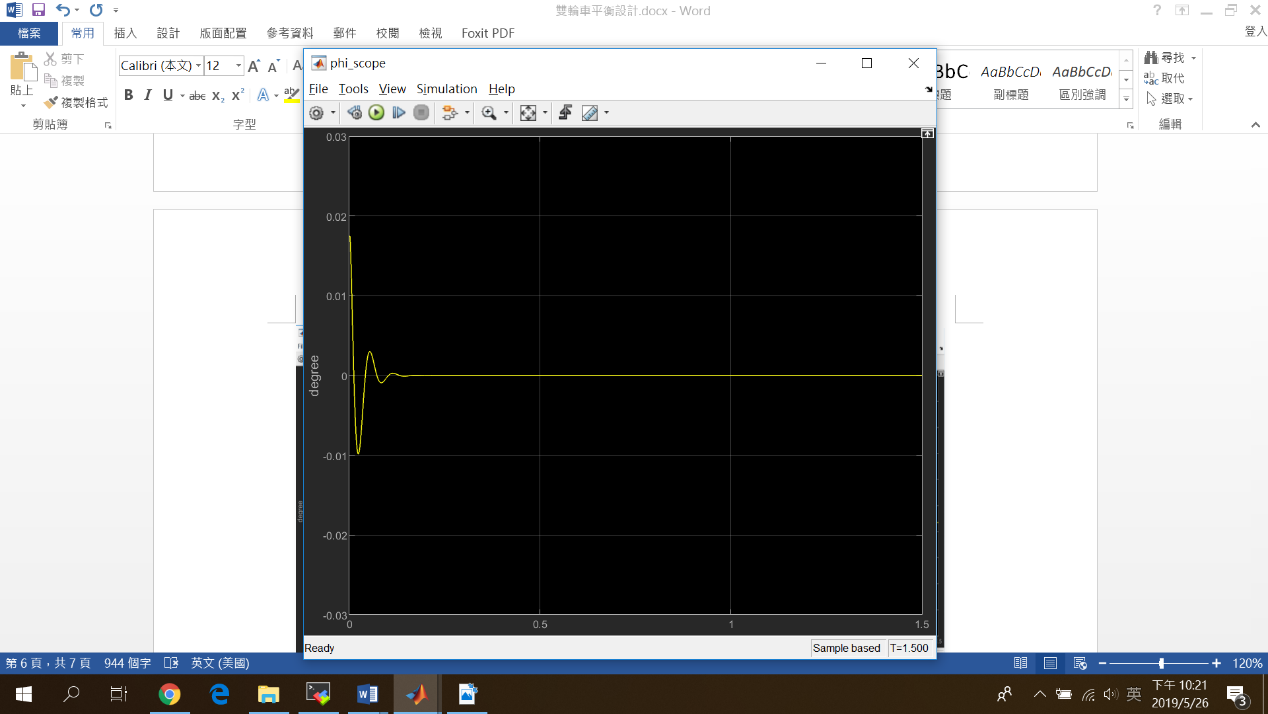
Ardunio: car.ino

App inventor: simple.aia

影片:於交通大學自動控制系統教學社團中

結果:在系統鑑別後，將所得的PID值做些微的調整後，能在無外力情況下平衡約兩分鐘的時間。實際Kp:113 ,Ki:3 ,Kd:0.8。

起初我們藍芽baudrate設為115200封包傳輸過快，所以將baudrate調降成38400，傳輸過程就不會發生遺失的現象



根據之前學的控制器設計理論，Kp愈大則系統的暫態響應會變快、Ki愈大則會加速穩態響應而Kd愈大則可以加速暫態響應的同時減少系統overshoot。若比較兩張圖中的兩個系統輸出訊號可以看出系統二的暫態響應與穩態響應速度皆較系統一迅速而且系統二的overshoot也較系統一小，這與理論推論相符合。

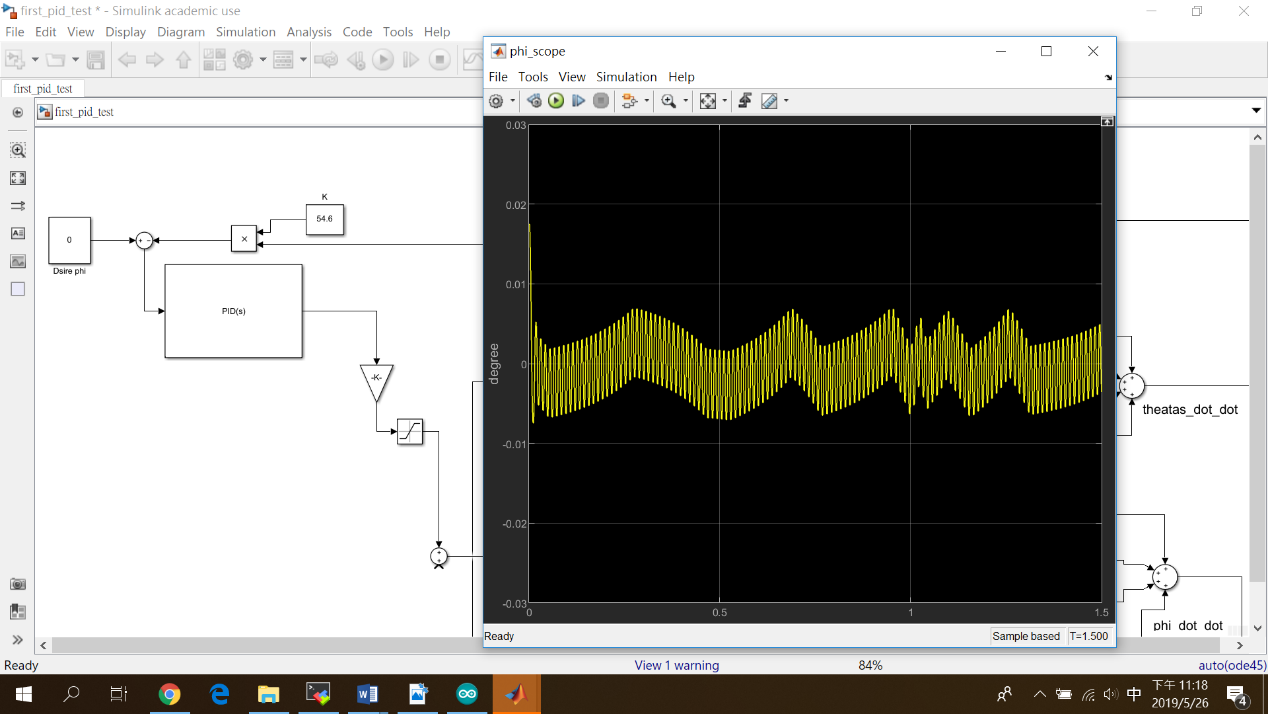
1. **結論**

這次實驗一開始因為ID結果做得不好，所以很難將模擬出來的PID值放入實際硬體當中，然後我們又重新系統鑑別模擬後，又發現在軟體所偵測的0度並非真正的0度，所以又增加了一個改變初始值的參數調整reference，然後因為我們需要判斷的程式碼過多，所以有別於之前把code全部放在arduino ，這次我們啟用助教的library，將不同功能的函式分門別類，雖然最還是花了”一整個後末!!!”做平衡，但能讓車子能夠平衡兩分鐘實在真的真的好開心!

1. **其他**

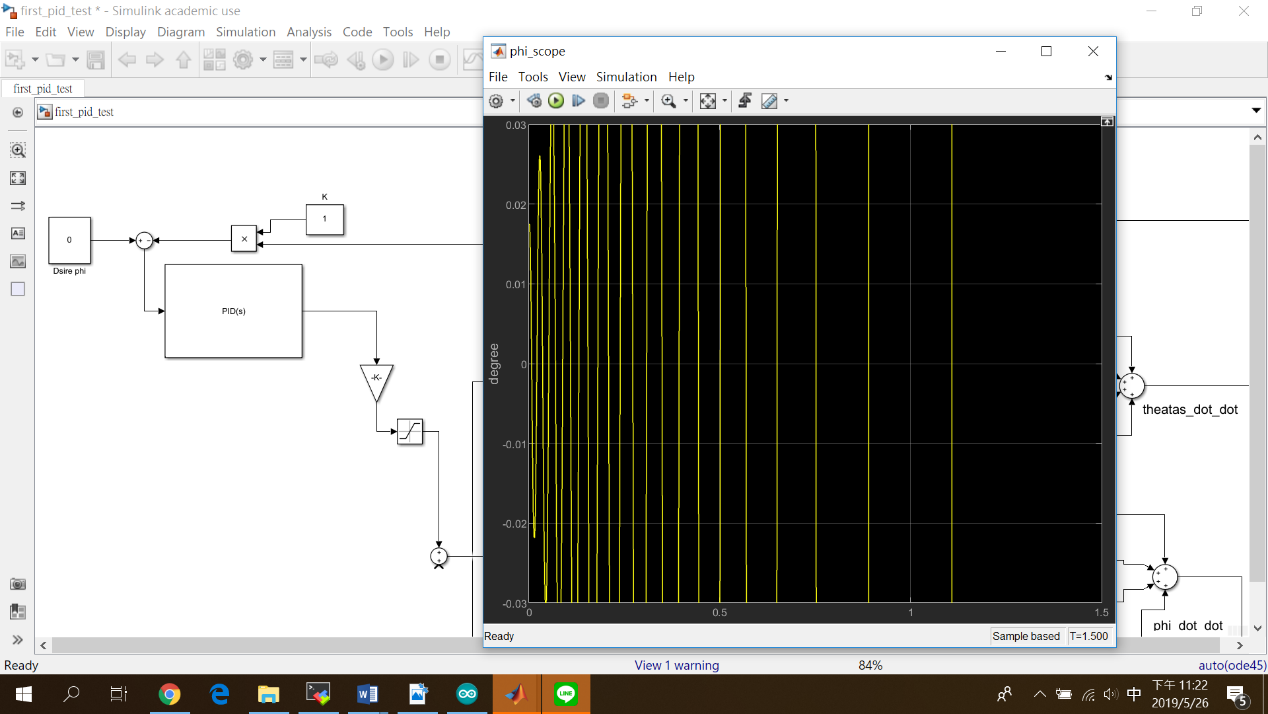
以下為我們取樣幾次調整PID過程的時域分析圖

狀況一



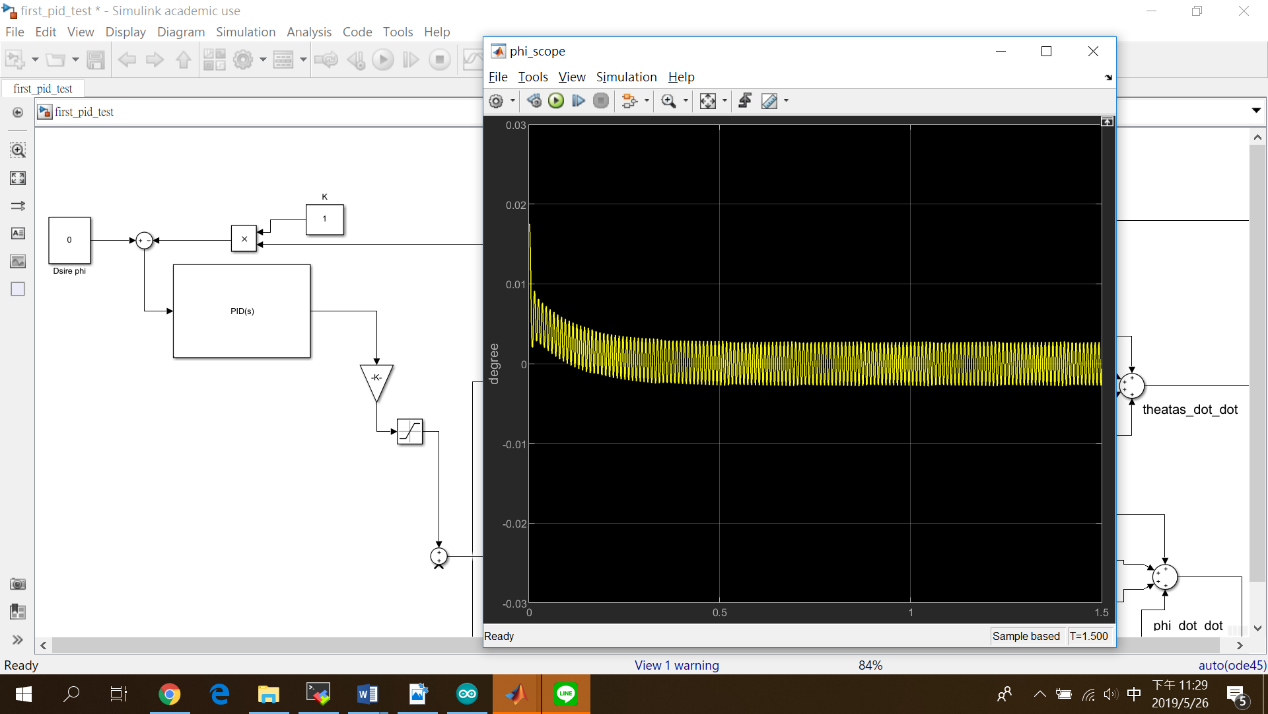
當I太大，會造硬體下半部(輪子)左右擺動劇烈，I 的功用是補足P的不足，補太多會來回震動

狀況二



當P太大，車傾角校正量太多，車子位置來回移動，最後演變成無法平衡的狀況

狀況三



I太大，造成沒辦法平衡，依據實驗經驗發現，I盡量要小，可讓車子穩定