**Control System Design Report**

Lab #3

Date: 2017.05.15

ECE106 0210749 賈恩宇

ECE106 0210825 蕭宇杰

1. **Objective**

本次實驗主要為設計控制器以達成對車體位置達成控制的目的。實驗中嘗試以兩種不同方式實作，其一為在已使用一PID控制器控制車子傾角的情況下，再使用另一PID控制器對其位置進行控制；另一種方式則為使用State-feedback控制，直接利用可量測到的系統State並對於State其中的位置進行控制。

1. **Principle & Derivation**

首先是嘗試使用PID控制位置的部分。在本次實驗中嘗試PID控制器時，有兩種配置方式，如圖(1)和圖(2)。

|  |  |
| --- | --- |
|  |  |
| 圖(1) | 圖(2) |

而其中平衡車系統使用的模型則是參考論文[5] [3][4][2]使用的動態方程式，其線性化後的方程式如式(1)與式(2)所示。

(1)

(2)

透過上述兩式，經過整理並帶入的實際數字後，可以得到輸入電壓與車傾角以及平衡車位置的兩個轉移函式，如式(3)與式(4)。

(3)

(4)

首先針對圖(1)的配置方式，可以將對於車傾角的PID控制器以及式(4)這兩部分整合為一inner loop，因此可以得到新的轉移函式作為新的plant，如式(5)。

(5)

接下來即可對於此新的plant進行控制。

而對於圖(2)的配置方式，則是直接以式(4)作為plant進行控制。

另外一種方式則是直接使用State-feedback的方式控制車子位置。同樣經過整理後，可以將式(1)與式(2)以式(6)與式(7)矩陣的形式表示。

(6)

(7)

其中,,,,,

由於上式中的state 皆可直接或間接透過平衡車程式的撰寫取得，再加上當控制於某一位置時，除了車子位置外，對於其餘state皆要控制至零，所以可以重新設計state ，其中的即是目標位置。藉由這樣的設計，可以在不改變原先state feedback設計的情況下，改變平衡的位置。接下來只要令輸入為，並選擇好Pole的位置，透過計算即可得設計好的矩陣。

1. **Data, Chart and Analysis**

首先是針對使用圖(1)的PID控制方式進行設計。圖(3)即是利用式(5)繪製的Root locus圖。

|  |
| --- |
|  |
| 圖(3) |

根據上圖，因為加入I與D控制器可能會更不穩定，因此可以設計出一組「P」控制器，可以勉強使其stable，但放到實際的平衡車上卻無法平衡，因此無法採用。

接下來則嘗試使用圖(2)的控制方式設計。對於圖(2)由於對於位置的轉移函數以及對於車傾角的轉移函數直接共用同一輸入，所以只要使用式(4)作為plant控制即可，所以可以利用以式(4)繪製出的Root locus來進行設計，如圖(4)。

根據圖(4)同樣可以設計出一PID控制器，然而同樣遇到將設計好的控制器放到實際的平衡車系統時會無法平衡，因此也無法採用。

|  |
| --- |
|  |
| 圖(4) |

所以最後只能嘗試使用State-feedback。利用試誤法，對於4個State設計新的極點的位置在，由此求出的增益依序為。值得注意的是，由於後兩項增益針對的是單位為rad的車傾角以及單位為rad/s的車傾角旋轉速度，而實際的平衡車中使用的單位則分別是degree以及degree/s，所以可以預先乘以，最後可得實際系統中使用的增益。

將上述參數帶入系統中，可得實際表現圖。圖(5)由左而右依序為車傾角、目標位置以及實際的位置。

|  |
| --- |
|  |
| 圖(5) |

由上圖可見，其車傾角在平衡車體時仍可保持平衡，且可以跑到預定的位置，然而仍會在大約正負10公分的範圍內震盪。我想可能和其到達該位置後仍需要做車傾角控制，而選擇的增益無法讓其在很小的範圍內即能做到平衡，才會有此現象。

除此之外，車子移動到預定位置也需要較長的時間，如上述移動50公分需時大約10秒，若調整的速度過快，則因為馬達的輸出有限，無法同時保持平衡車的車傾角維持平衡。

1. **Simulation**

接下來來使用模擬驗證。下圖(6)與圖(7)為對應使用圖(1)與圖(2)兩種控制方式其模擬的結果。其中圖(1)使用的是P控制器，，而圖(2)使用的則是PID，。

|  |  |
| --- | --- |
|  |  |
| 圖(6) | 圖(7) |

由上圖可見，無論採用何種控制方式，皆會震盪，甚至看起來會不穩定，而實際將參數帶入到平衡車系統時，情況更糟，因此我們才會改採用State-feedback來控制。

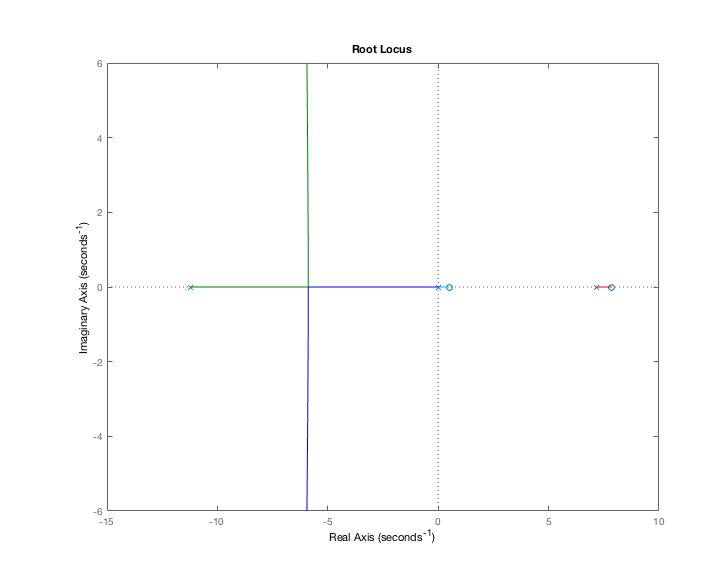
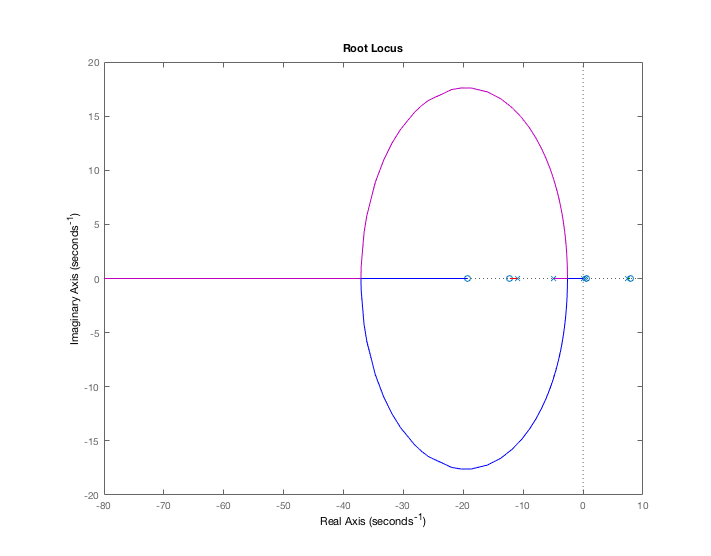
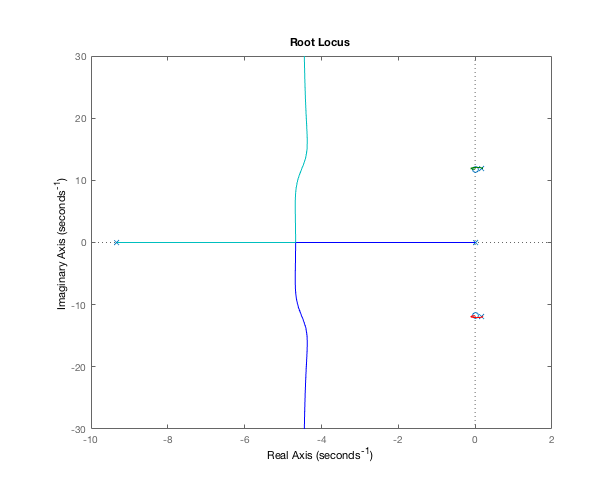
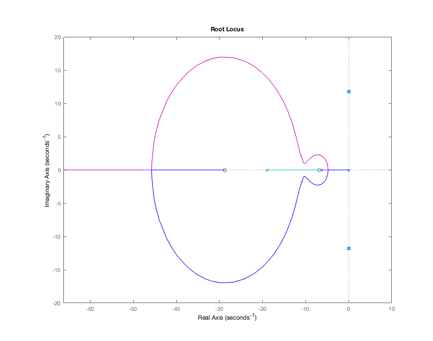
下圖(8)則為模擬State-feedback的方塊圖。而圖(9)則是將模擬車子位置的圖。

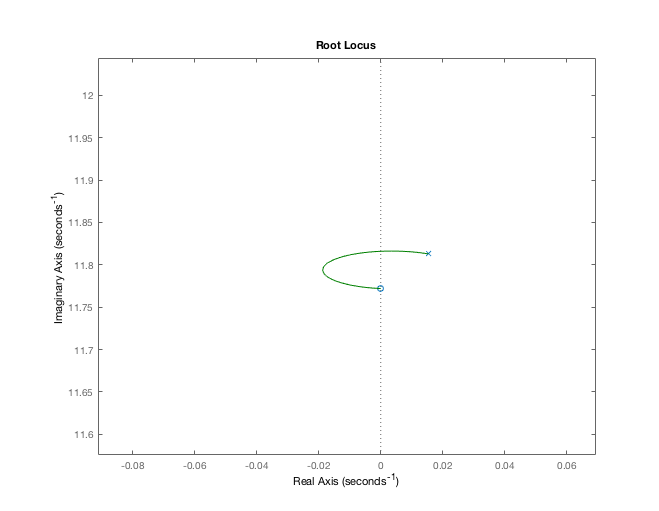
|  |  |
| --- | --- |
|  |  |
| 圖(8) | 圖(9) |

圖(10)則是比較模擬以及實際車傾角的部分，較為特別的是，其實際的車傾角並非如模擬的趨勢變化，且實際的抖動極為明顯，但相對其變化範圍較小。

|  |  |
| --- | --- |
|  |  |
| 圖(10) | 圖(11) |

圖(11)則是比較模擬以及實際位置的部分，由圖可見兩者的趨勢相當接近，然而會受制於實際車傾角抖動的情況，而無法一致的往目標位置移動。

1. **Questions and Discussions**
   1. **為何第二層ＰＩＤ設計失敗？**
      1. **講義動態方程式：  
         **以上兩張分別是在B部分的圖(1),(2)的Root locus圖，一張圖有包含對的PID設計一張沒有，如圖可以看到兩種方式都會產生右半平面的zero,pole，pole直接往zero跑不到左半平面，導致不論哪種對的PID控制器配置方式都還是會在unstable的狀態，因此做到這裡我們改用paper 的動態方程式
      2. **paper動態方程式：  
         **一樣，以上兩張分別是對paper裡(位置)的Root locus圖，一張圖有包含對的PID設計一張沒有，當中二張在虛軸表現類似，有一個虛軸上的zero跟很靠近的右半平面pole，放大該區域圖如下

****雖說pole可以被拉到左半平面，但是由於虛軸部份過大且實軸過度靠近虛軸，導致震盪幅度太大，overshoot約略100％，最終第二層PID控制器設計也宣告失敗

* 1. **為何state feedback可以控制，PID卻不行?**state feedback是直接可以assign pole的位置，直接設計pole皆在左半平面，且同時去控制，然而PID設計難以同時對兩者一起設計會互相影響，並且PID設計參數設計限制較多，須滿足一定的規格條件，多一個zero可能方程式就無解
  2. **為何位置控制不能一次控制在遠處**車子馬達速度有限，當前進的時候馬pwm以255全力衝刺，當抵達目的地的時候，需要更快的速度來平衡，但極限就是pwm=255不可能再大，因此會失去控制

1. **Conclusion**

本次實驗嘗試過兩種paper的pid設計，但由於經歷了重重失敗，經過分析覺得有點難達成目標，因此先往state feedback去做設計，並且增加可以指定他要在離出發距離多遠處平衡的功能，以及指定速度移動。

1. **Improvement of experiment process**

將原本的state space

將Ａ矩陣變成3\*3，去掉位置控制，也就是少去原矩陣第一列與第一行，為了要控制速度所以先不管位置。

跟前面一樣利用一樣的方法去對速度控制，設計state ，其中的即是目標速度。藉由這樣的設計，可以在不改變原先state feedback設計的情況下，改變平衡的位置。接下來只要令輸入為，並選擇好Pole的位置，透過計算即可得設計好的矩陣。

下圖(12)即為實際測試的圖，由左而右分別是車傾角、目標速度以及實際速度。在n=800時設定速度為-5 cm/s。

|  |
| --- |
|  |
| 圖(12) |

由圖可見，實際速度並非平衡再-5 cm/s附近，此原因為我們僅設定目標的速度，卻沒有考慮平衡時的車傾角，因此在計算時同時想保持車傾角為零，所以整體的速度反而要較大才能使最終結果是可以平衡的。

1. **Reference**
2. Modeling and model verification of an intelligent self-balancing two-wheeled vehicle for an autonomous urban transportation system
3. JOE: A Mobile, Inverted Pendulum
4. Modeling, Control of a Two-Wheeled Self-Balancing Robot
5. Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller Board
6. Balancing a Two-Wheeled Autonomous Robot
7. Adaptive Neural Network Control of a Self-Balancing Two-Wheeled Scooter