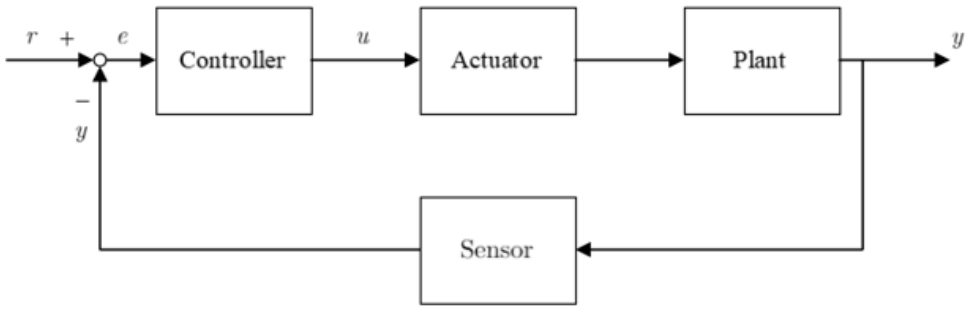
國立台灣科技大學 機器人控制實習 期末報告

課程代碼：ME3615301		任課老師：機械系 林紀穎 教授	
組別：1	學號 :B11003110	姓名 :林晃霆	繳交日期 :2024/06/07

[問題與討論]

1.參考以下閉迴路控制架構圖，回答以下三小題。



Closed loop

(1)定義二軸機械手臂之參考輸入(r)、控制輸入(u)、輸出(y)為何?(注意:參考輸入與輸出的單位須相同)

	角度控制	位置控制
參考輸入	目標(指定)角度(degree)	目標(指定)X、Y 座標(mm)
控制輸入	PWM	PWM
輸出	Encoder → degree	X、Y coordinate

過程會將輸入轉換成 encoder 脈衝數

(2)定義二軸機械手臂的 Controller、Actuator、Plant、Sensor，並嘗試描述此系統架構的運作流程。

Controller	Arduino Mega
Actuator	Stepper motor
Plant	Two-axis robotic arm
Sensor	Encoder(位置偵測)、infrared (極限保護)

角度模式：

透過序列埠輸入期望角度，乘上分辨率獲得移動到目標角度之脈衝數，透過建立 forward、backward 移動函式藉由當前編碼器脈衝數與目標角度之脈衝數差值為正、負之情形進行等速或 pid 控制 PWM 移動，直到完成

座標模式：

透過序列埠輸入期望座標，藉由反向運動學得出指定座標之對應軸 1、軸 2 角度，接著便是透過角度模式進行後續計算、控制以達到指定座標

(3)簡述二軸機械手臂的控制目標(control goal) (追跡、干擾抑制、強健性、穩定性?)

在開迴路控制下，軸二的穩定性、強健性、追跡效果較軸一來得差，如若使用 PID 進行閉迴路控制，則上述缺點都可以得到改善

在干擾抑制上，由於過程中不會有外界異物接觸、阻塞，因此無法確認

2.詳述二軸機械手臂之原點復歸程序(請列出詳細步驟 EX:Step1 關閉中斷……，可搭配 code 截圖說明)

Step1：

先建立 cmd1、cmd2 用作定義軸一、二是否要停止的控制參數

Step2：

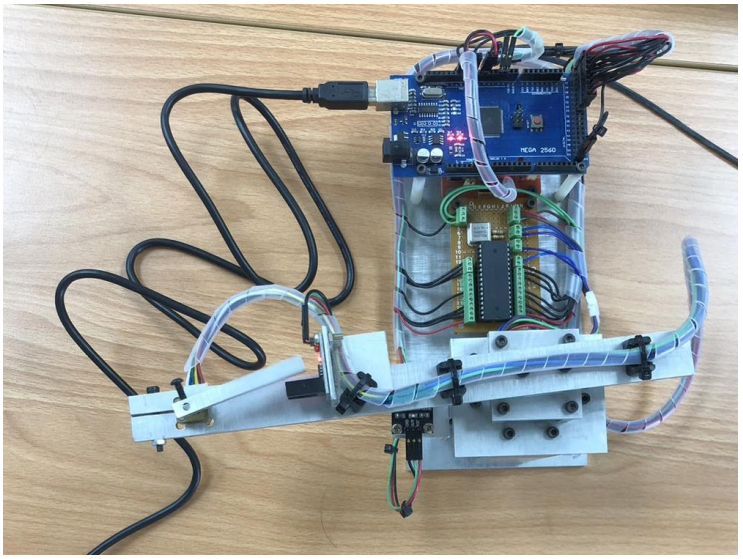
軸一、二開始反轉，反轉過程中若軸一紅外線讀值並非為 1，則即使軸二已經到復歸位置仍要繼續旋轉直到軸一到達，此時 cmd1 = cmd2 = 1，兩軸即可同時停止

Step3：

Reset 軸一、二的 encoder 讀值為 0

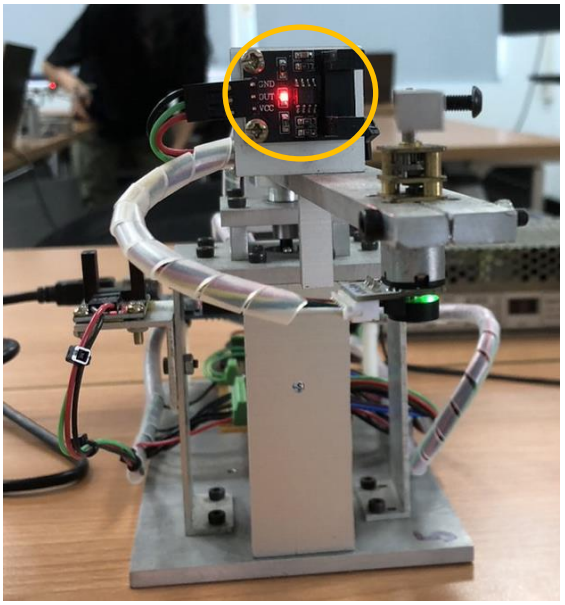
Step4：

軸一正轉 90 度、軸二正轉 180 度

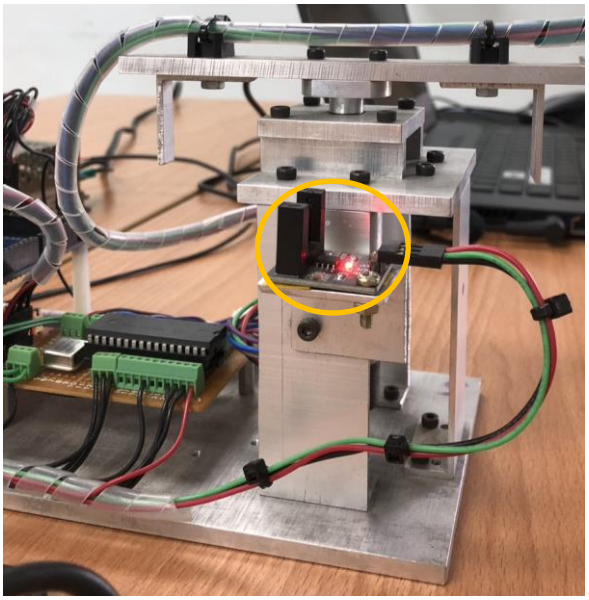


同時觸發極限開關->準備復歸

軸二極限開關



軸一極限開關

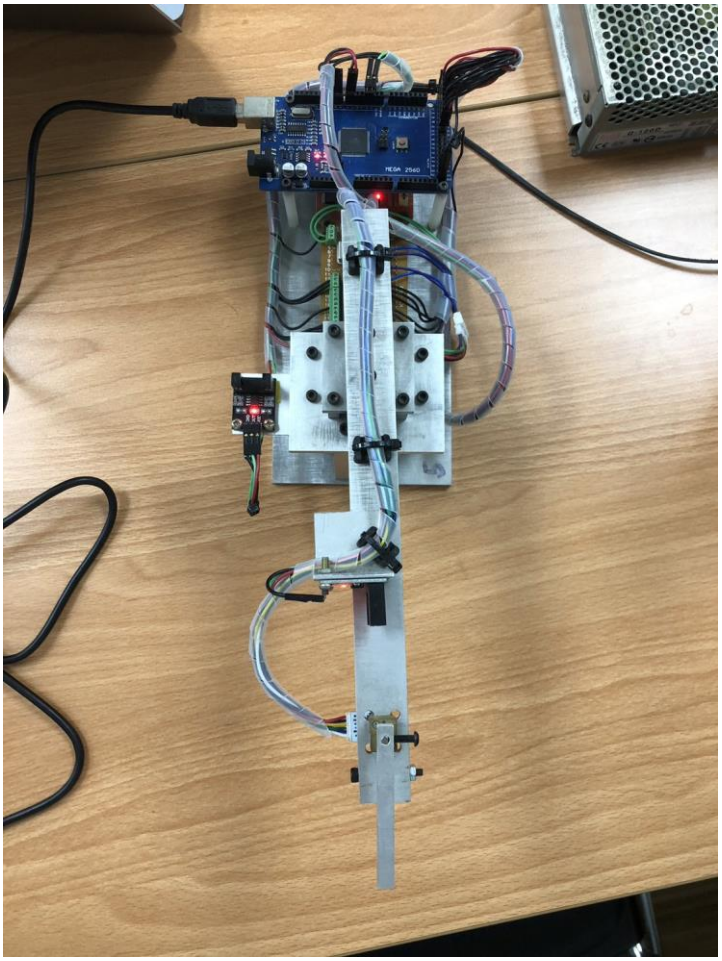


Step5：

Reset 軸一、二的 encoder 讀值為 0

Step6：

完成復歸



3.詳述二軸機械手臂之角度控制程序(列出詳細步驟，可搭配 code 截圖說明)

Step1：

個別輸入期望之軸一、二角度

Step2：

將角度轉換成脈衝數，並依照目標脈衝數與現在脈衝數差值開始正、反轉直到脈衝數在誤差範圍內

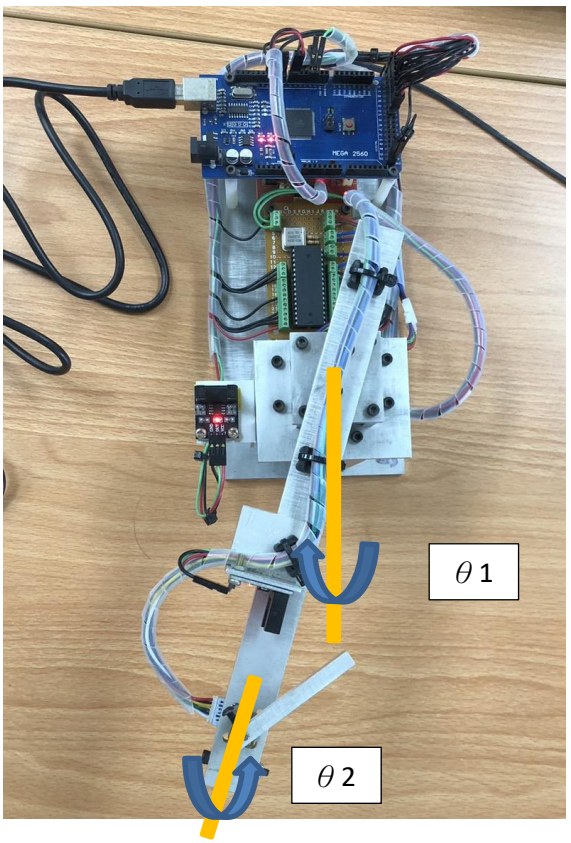
Step3：

到達指定位置後即可完成動作並停止

Ex ：

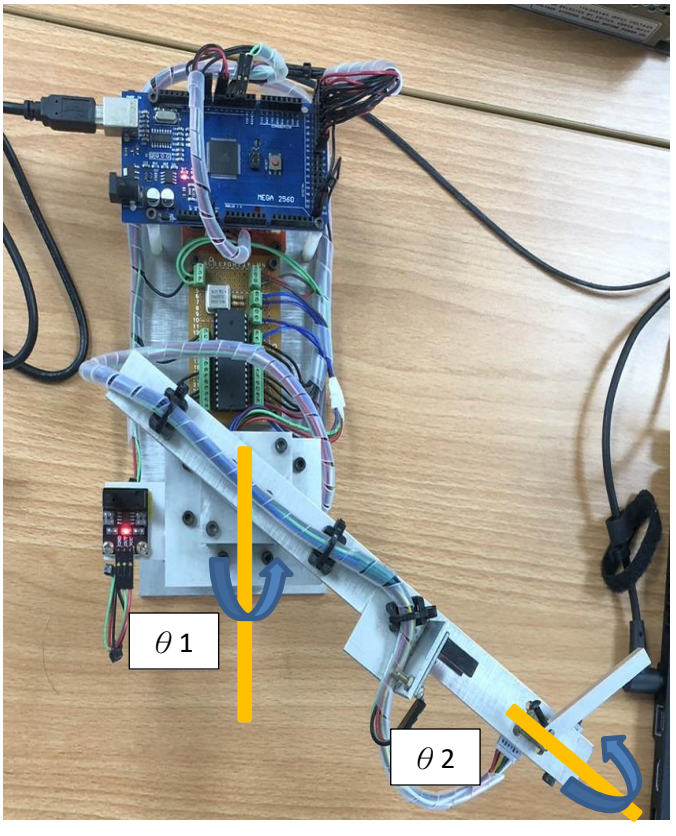
θ1：-25

θ2：150



θ1：50

θ2：75



4.詳述二軸機械手臂之座標控制程序(列出詳細步驟，可搭配 code 截圖說明)

Step1：

透過逆向運動學將輸入座標轉換成角度

```
Serial.println("start");
Serial.println("enter x cord");
while (Serial.available() == 0) {}
float X = Serial.parseFloat();
Serial.println("enter y cord");
while (Serial.available() == 0) {}
float Y = Serial.parseFloat();
float Theta2 = megaEncoderCounter.YAxisGetCount();
float Theta1 = megaEncoderCounter.XAxisGetCount();
```

獲取輸入 X、Y 座標

```
float Theta21 = acos((X*X+Y*Y-a1*a1-a2*a2)/(2*a1*a2));
float Theta11 = atan2(Y,X)-atan2(a2*sin(Theta21),(a1+a2*cos(Theta21)));
Theta21 = RAD_TO_DEG *Theta21;
Theta11 = RAD_TO_DEG *Theta11;


float Theta22 = -acos((X*X+Y*Y-a1*a1-a2*a2)/(2*a1*a2));
float Theta12 = atan2(Y,X)-atan2(a2*sin(Theta21),(a1+a2*cos(Theta21)));
Theta22 = RAD_TO_DEG *Theta22;
Theta12 = RAD_TO_DEG *Theta12;
```

透過逆向運動學計算出兩組解

若無解則不移動，若有解則依照兩組解的位移大小選擇較小的解

```
if(abs(Theta11)<=90 && abs(Theta21)<=180 && abs(Theta12)<=90 && abs(Theta22)<=180)
{
  Serial.println("both solution are in work region");
  float displacement_1 = abs(Theta11 - Theta1) + abs(Theta21 - Theta2);
  float displacement_2 = abs(Theta12 - Theta1) + abs(Theta22 - Theta2);
  Serial.print("displacement_1 : ");
  Serial.println(displacement_1);
  Serial.print("displacement_2 : ");
  Serial.println(displacement_2);
  Serial.println("");

  if (displacement_1 < displacement_2){
    Serial.println("angle1 solution is ");
    Serial.println(Theta11);
    Serial.println("angle2 solution is ");
    Serial.println(Theta21);
  }

  else if (displacement_1 > displacement_2){
    Serial.println("angle1 solution is ");
    Serial.println(Theta12);
    Serial.println("angle2 solution is ");
    Serial.println(Theta22);
  }
}
```

挑選解 1

挑選解 2

Step2：

依照下述情形設定移動情形 (十→正轉, 一 →反轉)即可完成

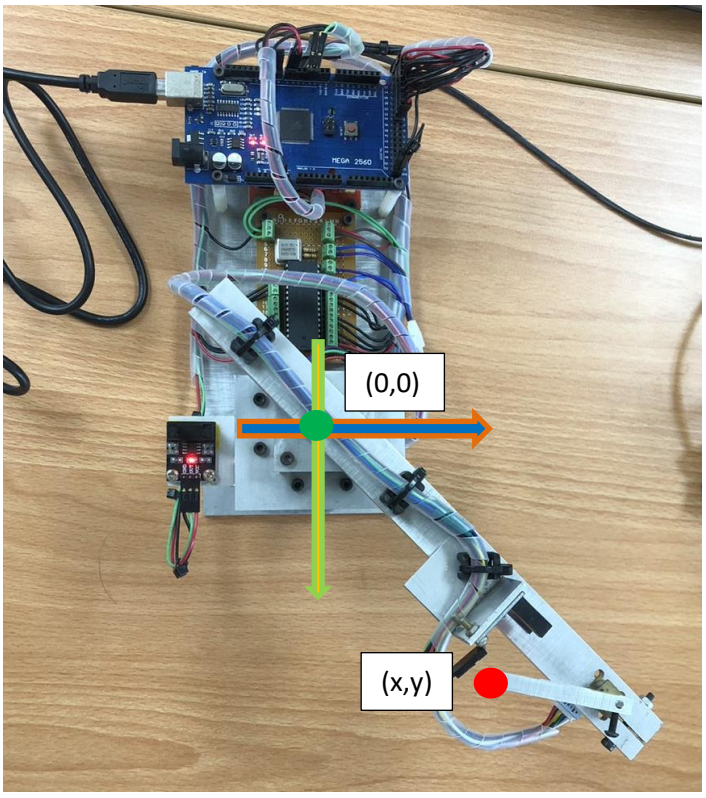
與目標差值	軸一	軸二	輸出
	十	十	(- , -)
	十	一	(- , +)
	一	十	(+ , -)
	一	一	(+ , +)

```
if (theta_1_count > encoder_1 && theta_2_count > encoder_2){
  encoder_1 = megaEncoderCounter.XAxisGetCount();
  encoder_2 = megaEncoderCounter.YAxisGetCount();
  M1Forward();
  M2Forward();
}

// both backward
else if  (theta_1_count < encoder_1 && theta_2_count < encoder_2){
  encoder_1 = megaEncoderCounter.XAxisGetCount();
  encoder_2 = megaEncoderCounter.YAxisGetCount();
  M1Backward();
  M2Backward();
}
```

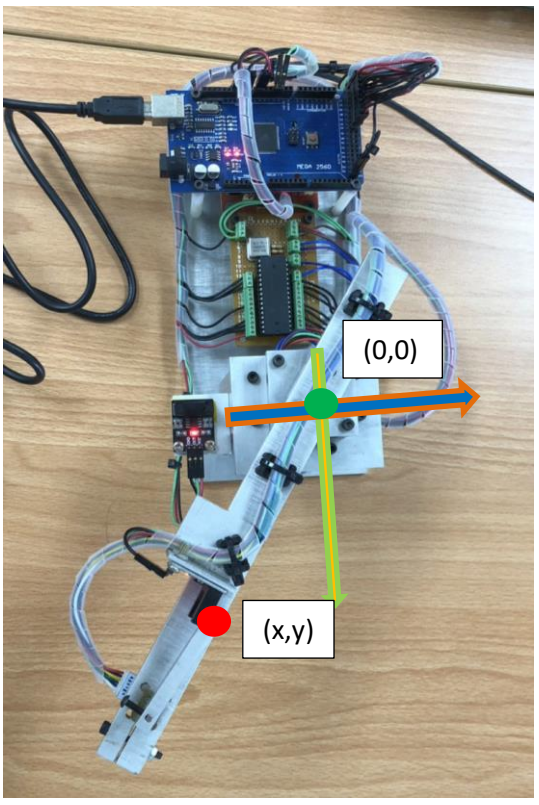
```
// 1 forward 2 backward
else if (theta_1_count > encoder_1 && theta_2_count < encoder_2){
  encoder_1 = megaEncoderCounter.XAxisGetCount();
  encoder_2 = megaEncoderCounter.YAxisGetCount();
  M1Forward();
  M2Backward();
}

// 1 backward  2 forward
else if (theta_1_count < encoder_1 && theta_2_count > encoder_2){
  encoder_1 = megaEncoderCounter.XAxisGetCount();
  encoder_2 = megaEncoderCounter.YAxisGetCount();
  M1Backward();
  M2Forward();
}
```

● 端點位置　● 原點位置　X 軸　Y 軸

輸入(80, 60)



輸入(80, -60)



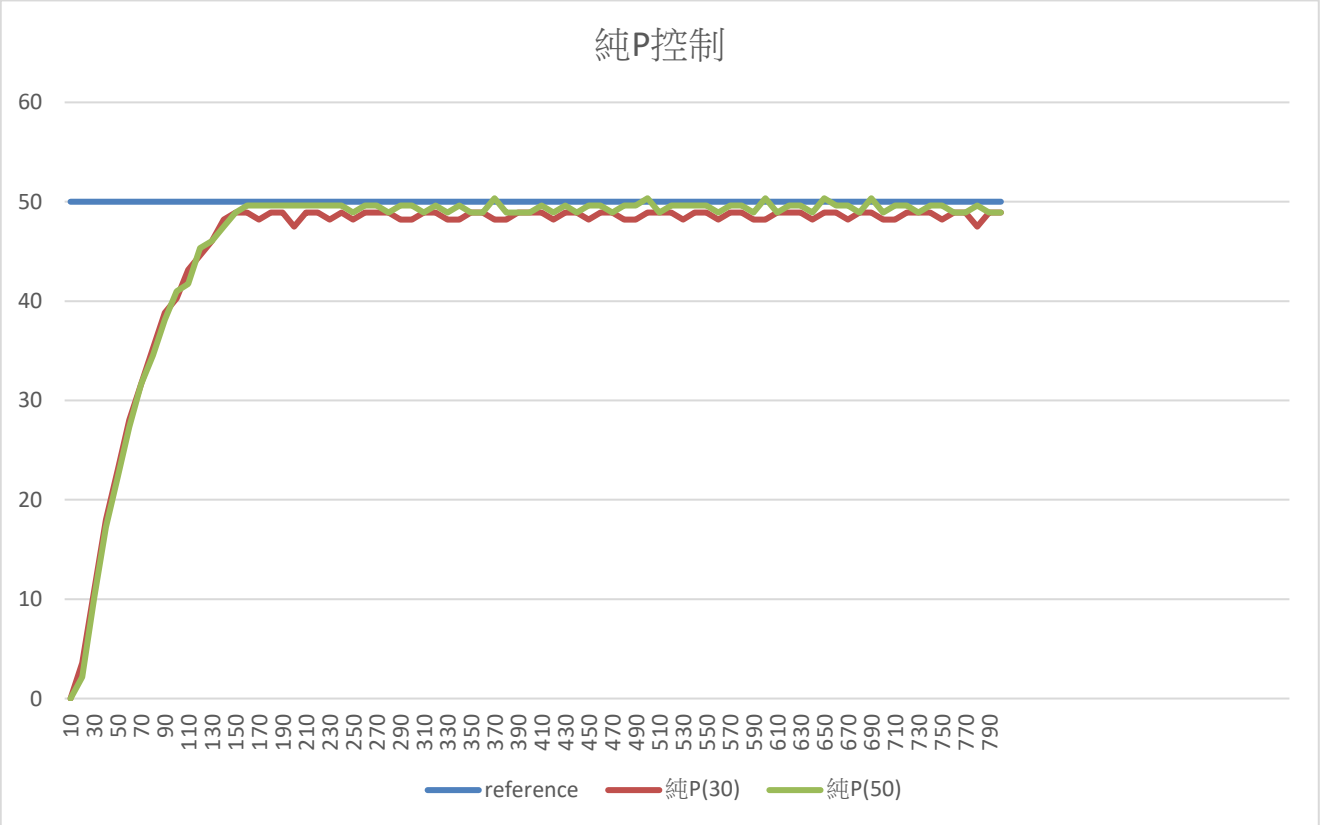
5.簡述你對 PID 控制器的理解&用一句話(10 個字以內)定義何謂控制器?

PID 控制器是基於誤差的大小給予輸出的一種閉迴路控制方法，可以有較好的控制效果，包含上升時間、暫態響應、穩態誤差等，但是如果需要有精確的位置、路徑控制，則不建議使用 PID

將輸入與輸出有效逼近

6.說明轉速追跡 PID 控制器參數之調整步驟，並附上課堂中調整之 EXEL 追跡響應圖，至少三張，並請比較說明這些追跡響應圖的成因，(例如:A 圖 Kp 比 B 圖大，因此 A 圖 Overshoot > B 圖)

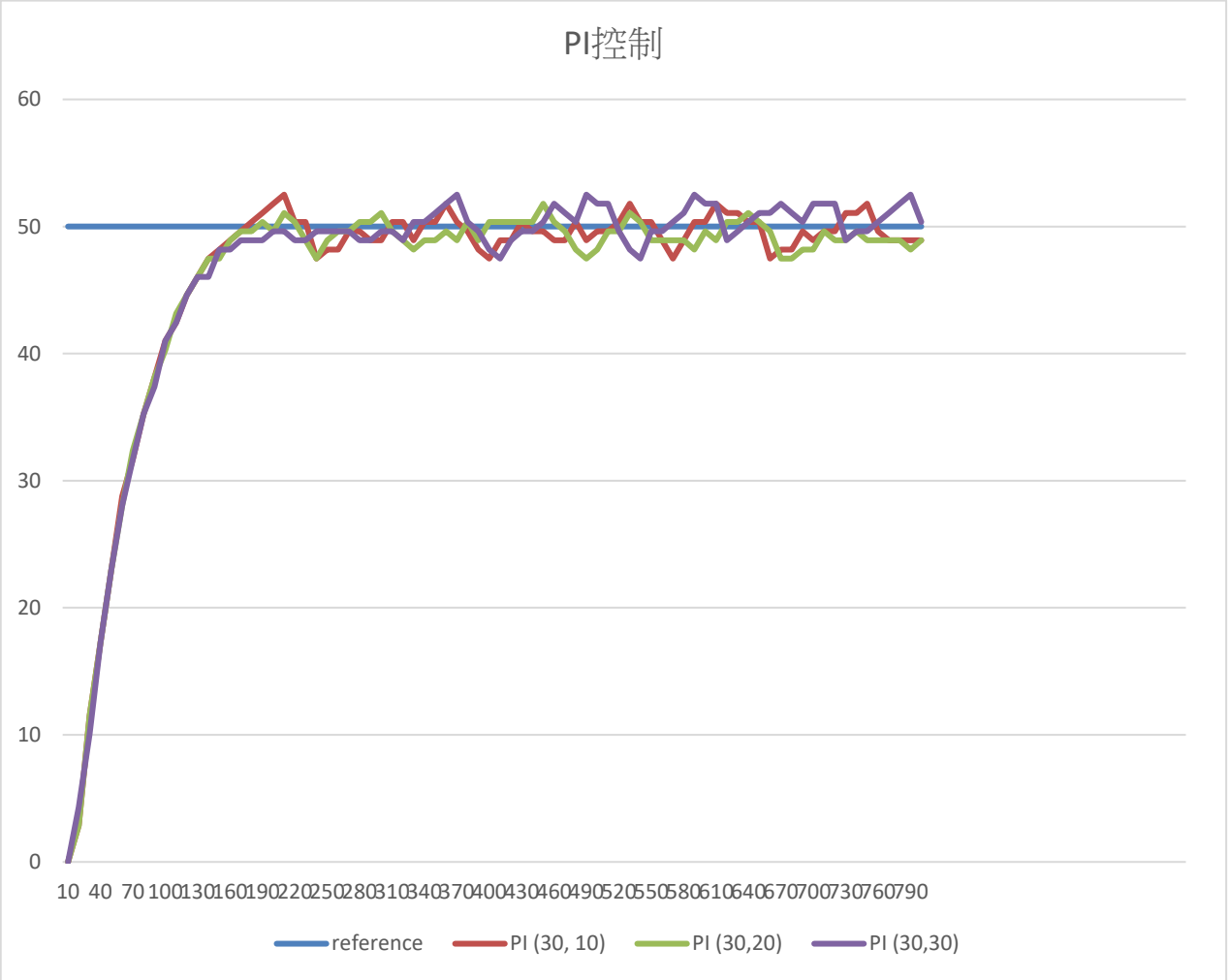
純 P 控制



從此圖來看可以發現 Kp=50 之穩態誤差較小，但是暫態有較多抖動

經過測試，發現抖動乃是因為取樣時間間隔與 encoder 讀值時，兩者頻率並不一樣，因此才會有此抖動情形

PI 控制



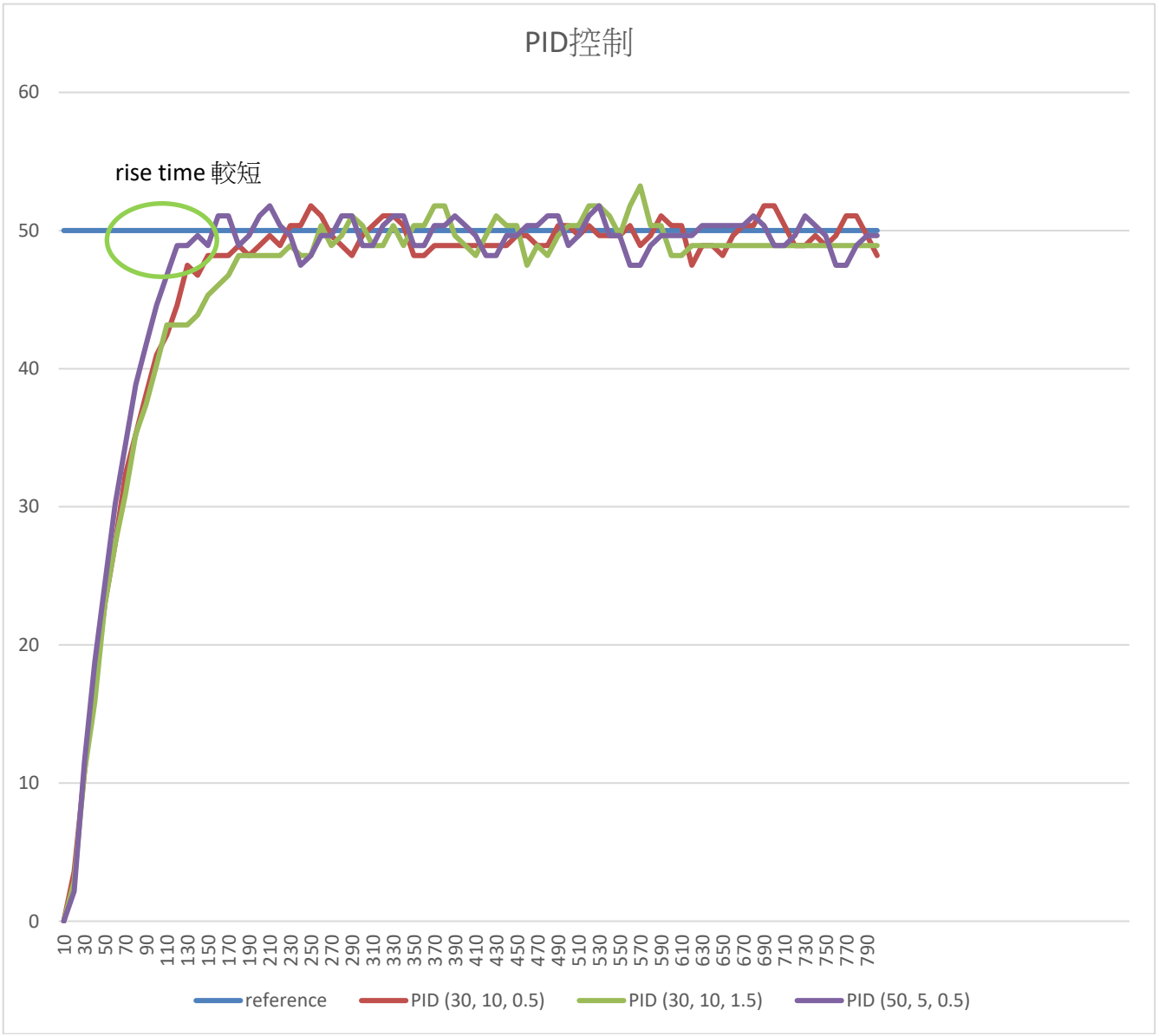
從上 PI 控制圖可以看出當 Ki = 10、20、30 時對穩態追跡的效果大致相同

這是由於我們使用程式控制馬達轉速時乃是控制其 PWM 值，因此到穩態的 PWM 一定會趨近我們所輸入的值

所以我們主要在乎的應該是馬達追跡的 rise time 及 settling time，可以使得馬達快即迅速安定至目標值

而從圖中當 Ki = 20 時有相對較短的安定時間，而上升時間由於三者的 Kp 皆相同，因此差距不大

PID 控制



從圖可以看到許多對照組

(1) (30, 10, 0.5)與(30, 10, 1.5)當 kd 調大後，暫態時的起伏次數有明顯改善

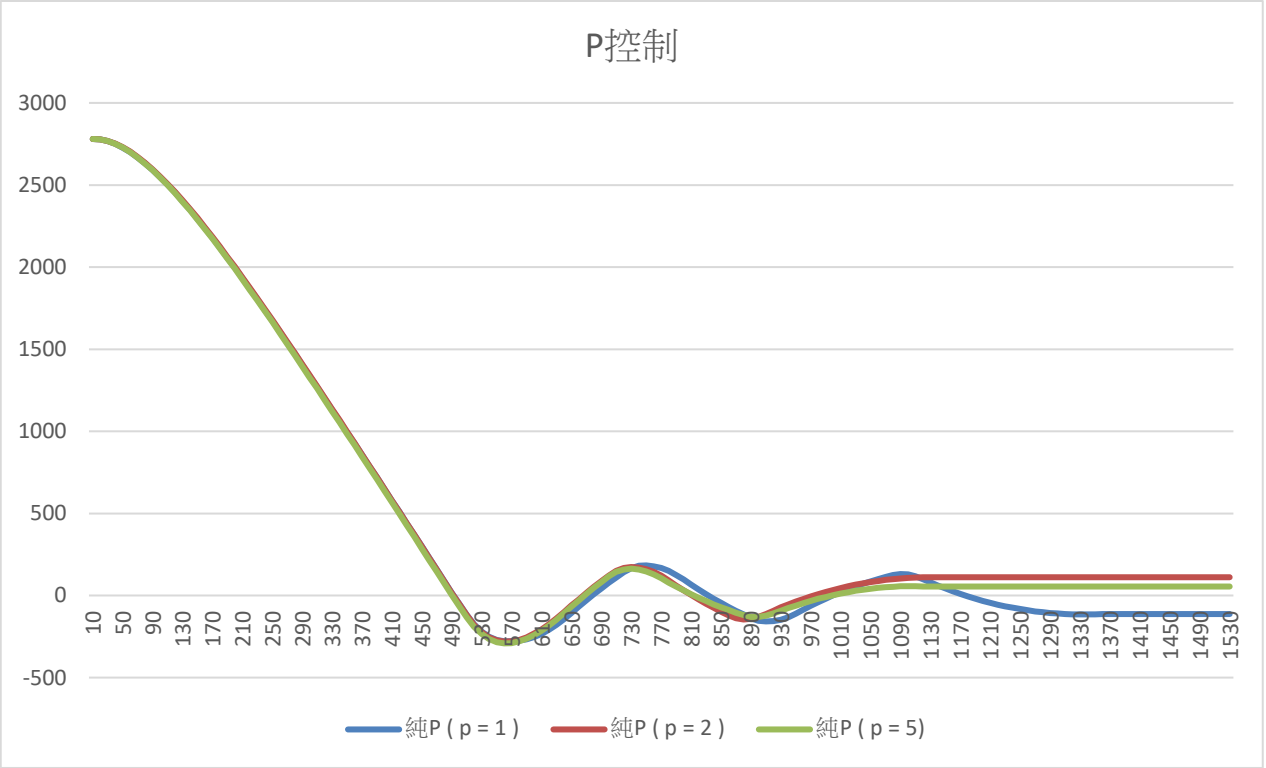
(2) (30, 10, 0.5)與(50, 5, 0.5)當 kp=30 時，rise time 大於 kp=50，響應速度快

可以從上圖看出當 PID 參數為(50, 5, 0.5)時，穩態誤差較小，響應速度較快，因此可以視作較優之參數

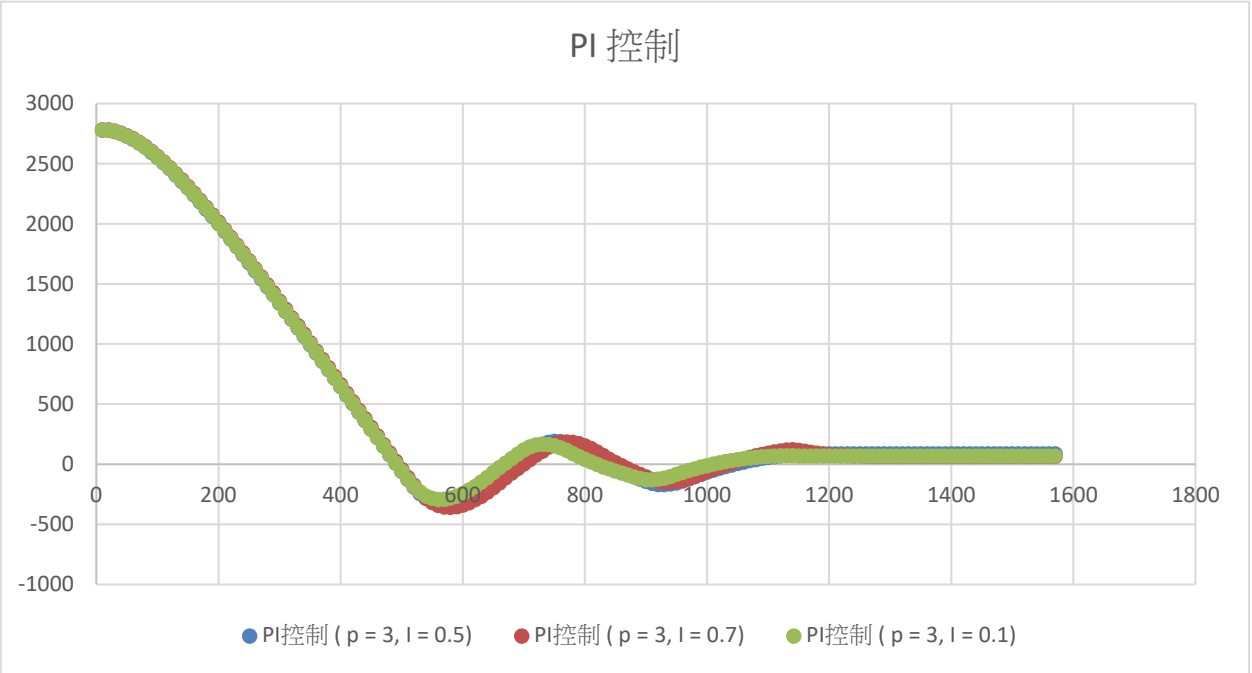
註：由於馬達減速齒輪限制，當輸入轉速大於 65 以上時，馬達轉速無法達成，故使用 50rpm 最為追跡參考

7.說明位置追跡 PID 控制器參數之調整步驟，並附上課堂中調整之 EXEL 追跡響應圖，至少三張，並請比較說明這些追跡響應圖的成因，若結果不收斂，請嘗試解釋原因。

純 P 控制

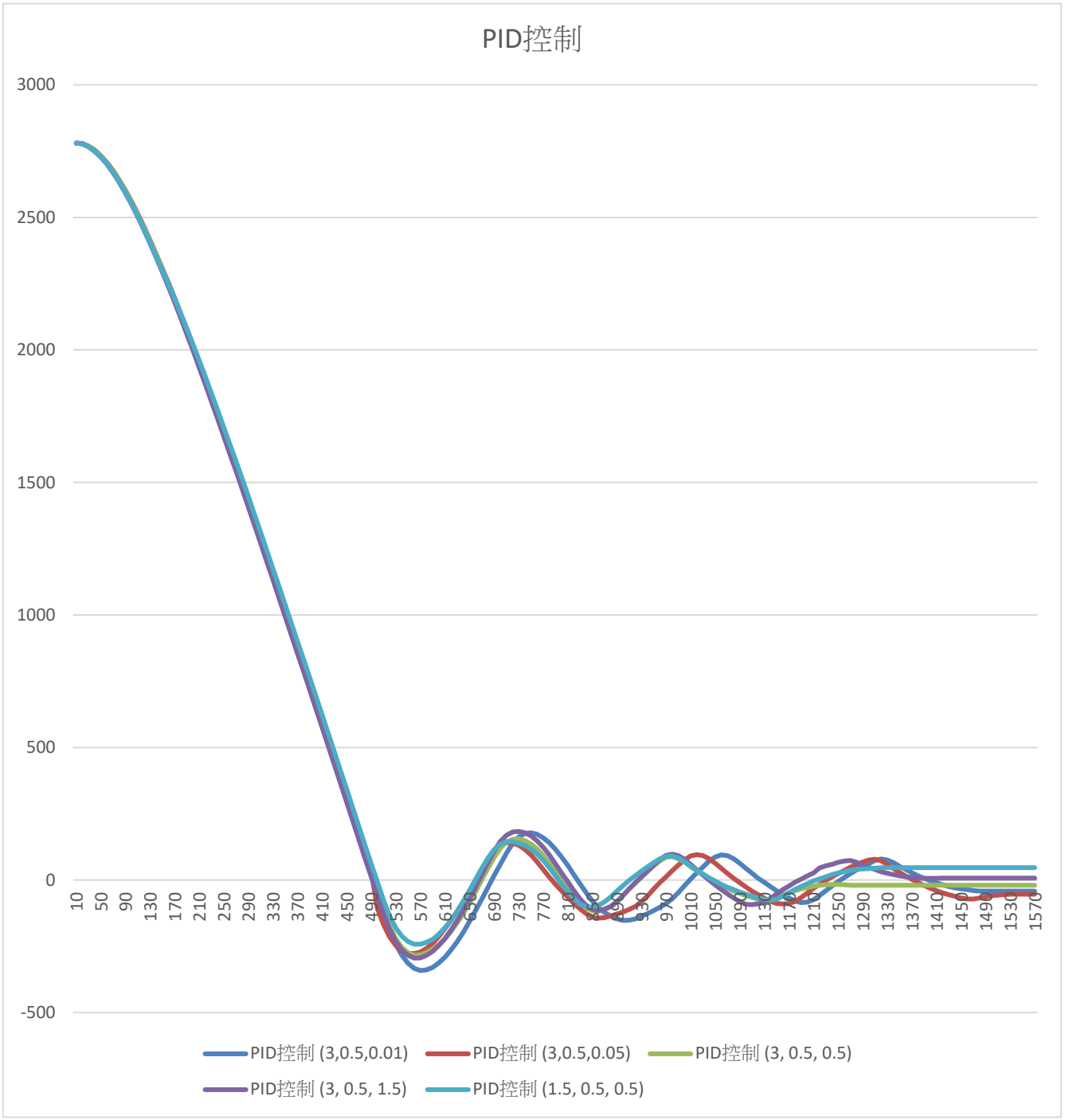


從圖可以看出 kp = 5 時有較好穩態追跡，使用 0.6kp 並加入 ki



其中 ki = 0.5、0.7、0.1 的穩態誤差皆沒有明顯差異，因此可以加入 kd 優化暫態響應

PID 控制



從五組參數中可以看出當 kp、ki、kd 分別為 3、0.5、0.5 時不僅追跡準確、安定時間短、overshoot 也較小

8. 請簡述「速度軌跡規劃」的意義為何(加分題)

單純使用 PID 控制時，可能會由於初始誤差較大造成輸出大而有 overshoot 情形，速度軌跡規劃則可以預先規劃出在該時間點所需具備的速度，再與目前速度坐回授控制，可以避免不同時間段的誤差劇烈變化而有 overshoot

9. T-curve 是一種速度規劃的方式，除了 T-curve 以外，還能使用哪些方式來進行速度規劃?分別簡述之(加分題)

S-curve，為較平滑的速度曲線規劃，可以避免大急跳度

10.比較 T-curve 和 S-curve 兩種速度規劃的差異，試著說明為何 S-curve 在加減速時會比 T-curve 還要平穩(提示：位置對時間的三次微分) (加分題)

由於 T-curve 並不是一條連續曲線，因此在兩曲線交會處的微分會是趨近無限大，進而有很大的 jerk

S-curve 是一條連續的曲線，因此再經過微分時只會有遞減的多項式->常數->0，並不會有過大的 jerk，所以在加減速時才能較為平緩

11.寫出對於機器人控制實習的上課心得，歡迎給予評論指教。

當這個學期結束後，我才發現其實我已經完成了手臂控制中的大部分功能，其中程式碼也是相當龐大，但是助教們在一開始就將內容切分成許多獨立部分，這讓我們在學習的過程中部會一下子有太大的負擔，學習起來也更容易上手，其中也很感謝助教們在我有問題的時候都能夠即予協助，也能提供我額外的時間到實驗室將缺失的進度給補齊，這堂課讓我學習到了許多控制的實作、也加深了我對 PID 控制的理解。

P.S.(報告中有附結果圖的地方，請詳細解釋，不可只放圖片)

助教： 徐瑋駿、林冠瑋、林冠宏、王品仁　　實驗室: T3-603-1