



平衡球

指導老師：林家德老師

組員：26陳郁瑜 30黃允宸

31黃彥翔 32黃柏愷

目錄

- 前言
- 組員介紹
- 專題規劃
- 主要元件介紹
- 原理介紹
- 程式介紹
- 遭遇問題、Q&A

前言

- 在這個科技發展的時代，機器輔助人類生活的機會越變越多，而PID控制為業界最常見的控制演算法，我們將藉由平衡球實驗來清楚呈現PID的運算，並藉著調動 K_p 、 K_d 、 K_i 三個數值使球體可以快速且穩定的移動到平衡點。

組員介紹

組員介紹



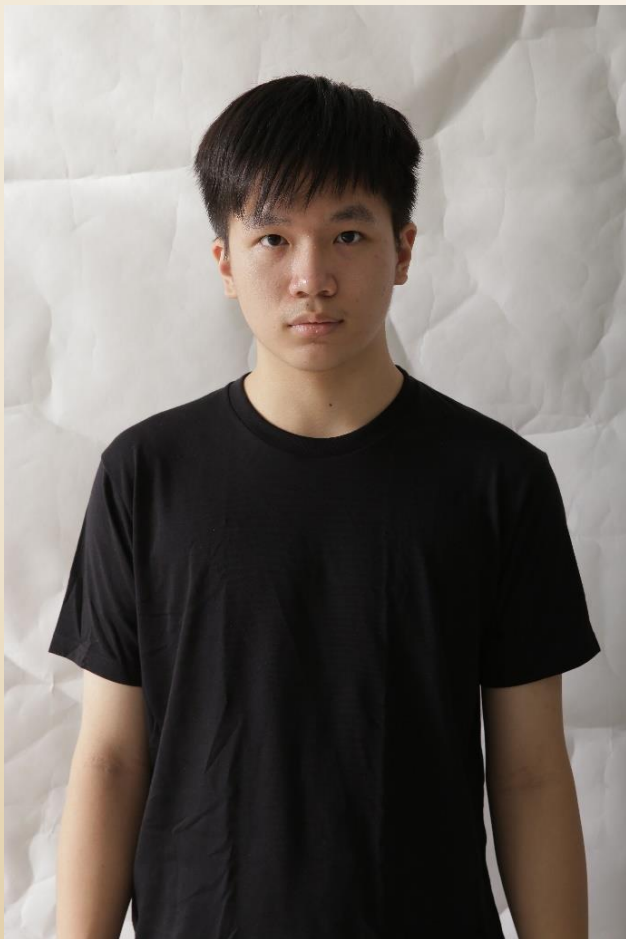
- 子三甲 32 黃柏愷
- 職位：組長
- 負責工作：程式撰寫
 電路設計

組員介紹



- 子三甲 26 陳郁瑜
- 職位：組員
- 負責工作：程式撰寫
硬體製作

組員介紹




- 子三甲 30 黃允宸
- 職位：組員
- 負責工作：材料採買
硬體製作

組員介紹



- 子三甲 31 黃彥翔
- 職位：組員
- 負責工作：文書處理
資料調查

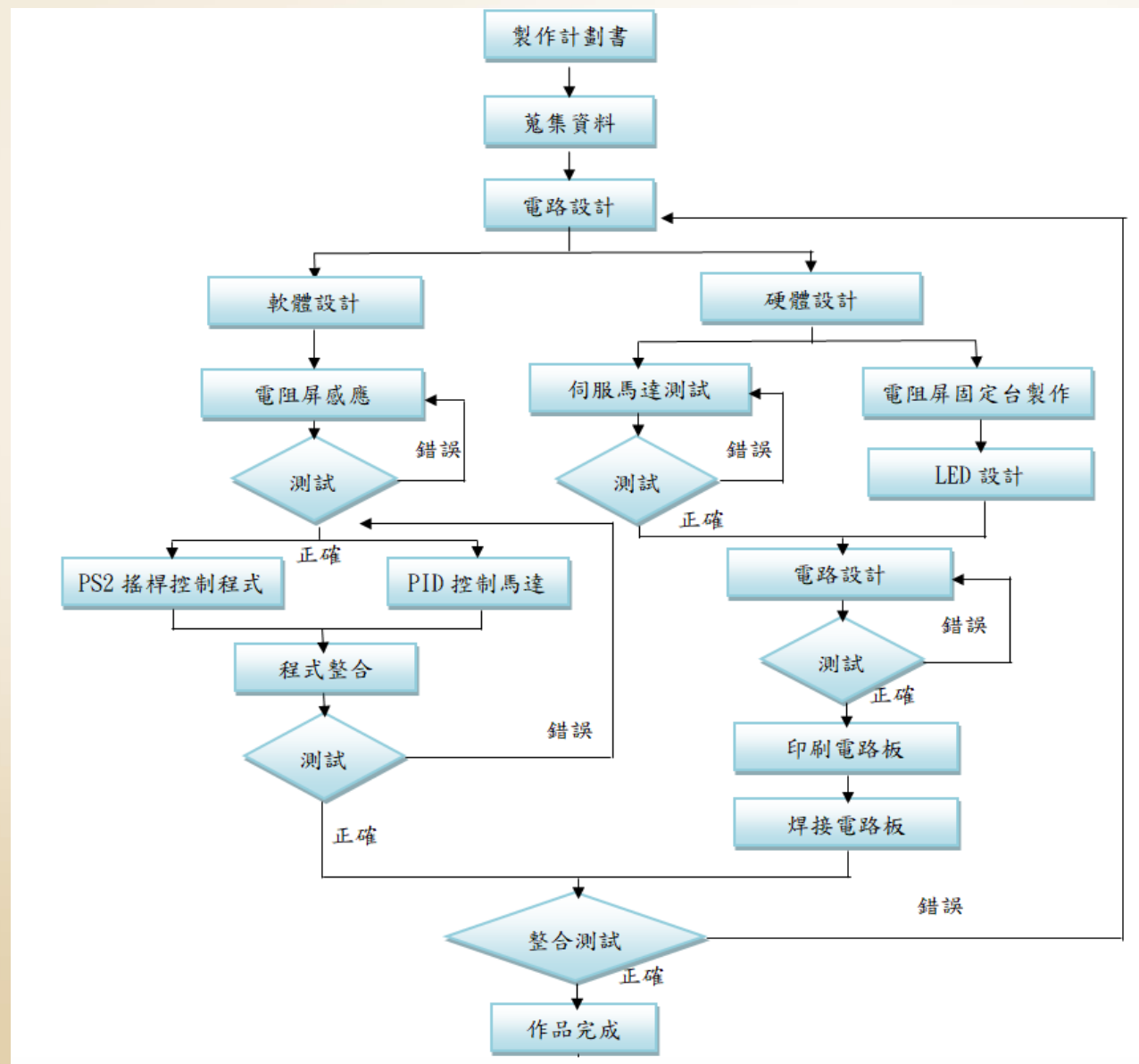


專題規劃

專題規劃 甘特圖

工作項目	週次	9	9	9	9	10	10	10	10	10	11	11	11	11	12	12	12	12	12	1	負責成員	
	(日期)	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/		
		3	10	17	24	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7		
		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
		9	9	9	9	10	10	10	10	11	11	11	11	11	12	12	12	12	1	1		
		/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/		
		7	14	21	28	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11		
寫計劃書		●	●	●																	允，柏	
資料蒐集			●	●	●																	全
電阻屏伺服機測試		●	●	●																		瑜，柏，允
Pid 控制				●	●	●	●	●	●	●												全
Arduino 程式						●	●	●	●	●	●	●										瑜，柏
LED 背板											●	●	●	●	●	●						彥
印刷電路板												●	●	●	●	●						彥，允
焊接															●	●						瑜，柏
軟硬體測試																●	●	●	●			允
口頭報告																			●	●		全
預定進度		3	6	10	15	20	25	40	45	50	54	56	60	65	70	75	85	90	100	100		累積百分比%

專題規劃 專題流程圖



專題規劃 系統架構圖





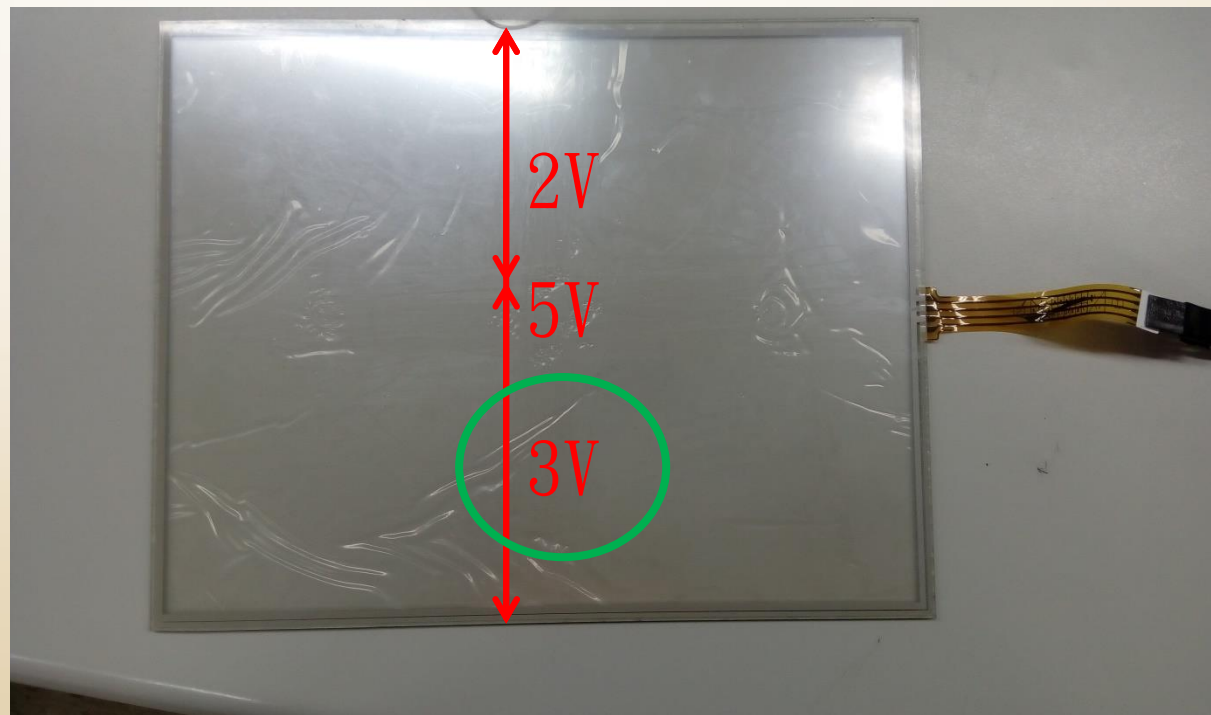
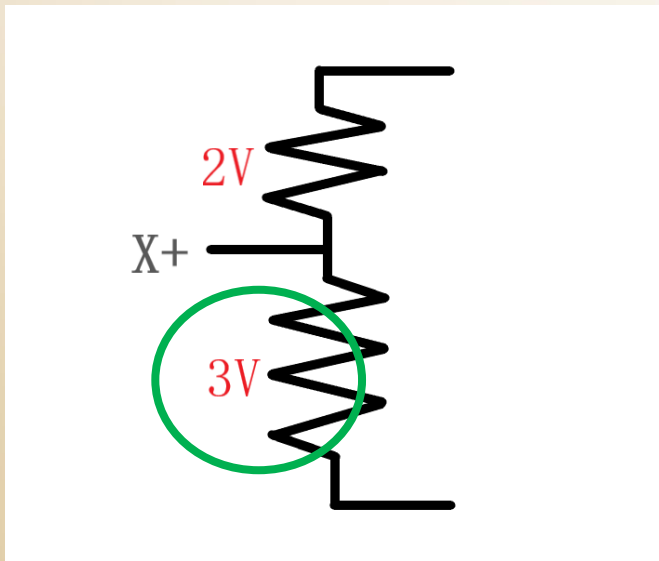
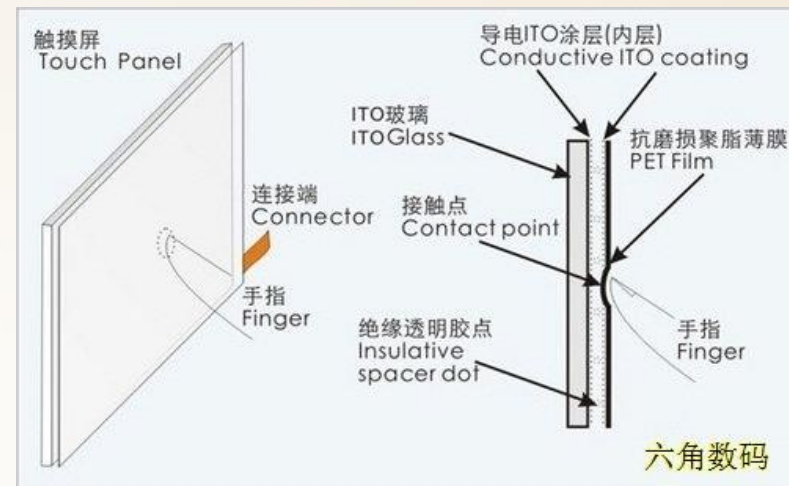
主要原件介紹

主要原件介紹 電阻屏

- 優點：
 - (1) 不受灰塵、水氣和油污的影響
 - (2) 能承受的溫度範圍也較廣
 - (3) 壓力感應因此只要施力就可被感測
- 缺點：
 - (1) 多層結構無法輕薄化
 - (2) 透光性不佳
 - (3) 不支援多點觸控

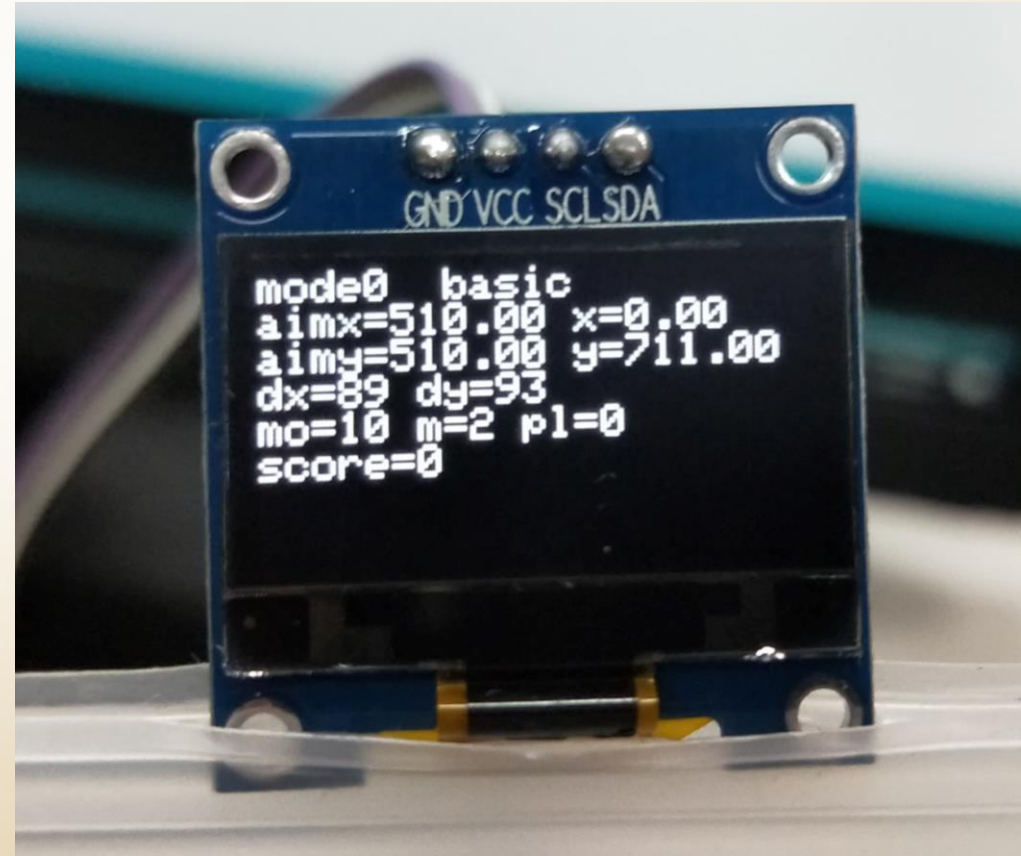
主要原件介紹 電阻屏

- 分為上(X)下(Y)層，各為完整的電阻
- 藉由外界施以壓力產生分壓點
- 由上(X)下(Y)兩層所得的電壓值來掌握球體位置(X, Y)



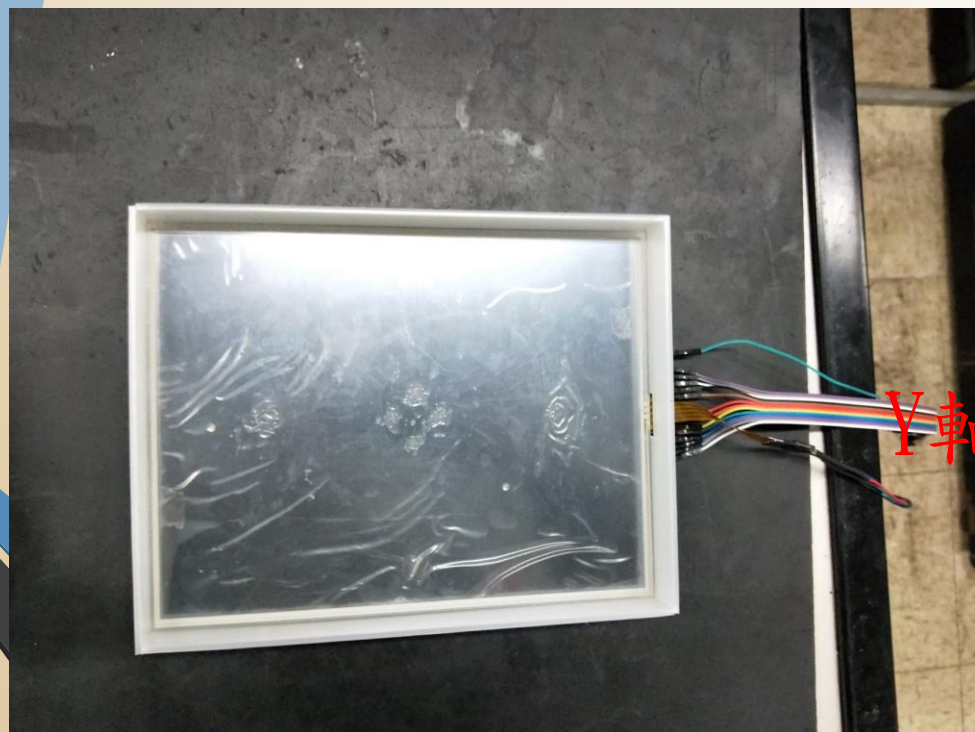
主要原件介紹 OLED

- 顯示平衡球資料
- 第一行: 模式
- 第二行: X目標位置及現在位置
- 第三行: Y目標位置及現在位置
- 第四行: 馬達起始角度
- 第五行: 遊戲時球體移動速度
- 第六行: 遊戲分數



主要原件介紹 伺服馬達平衡台

- 平衡球主要裝置
- 利用設置的2顆伺服馬達操縱平台XY角度
- 增設自製底座以13顆LED燈來應對球體位置



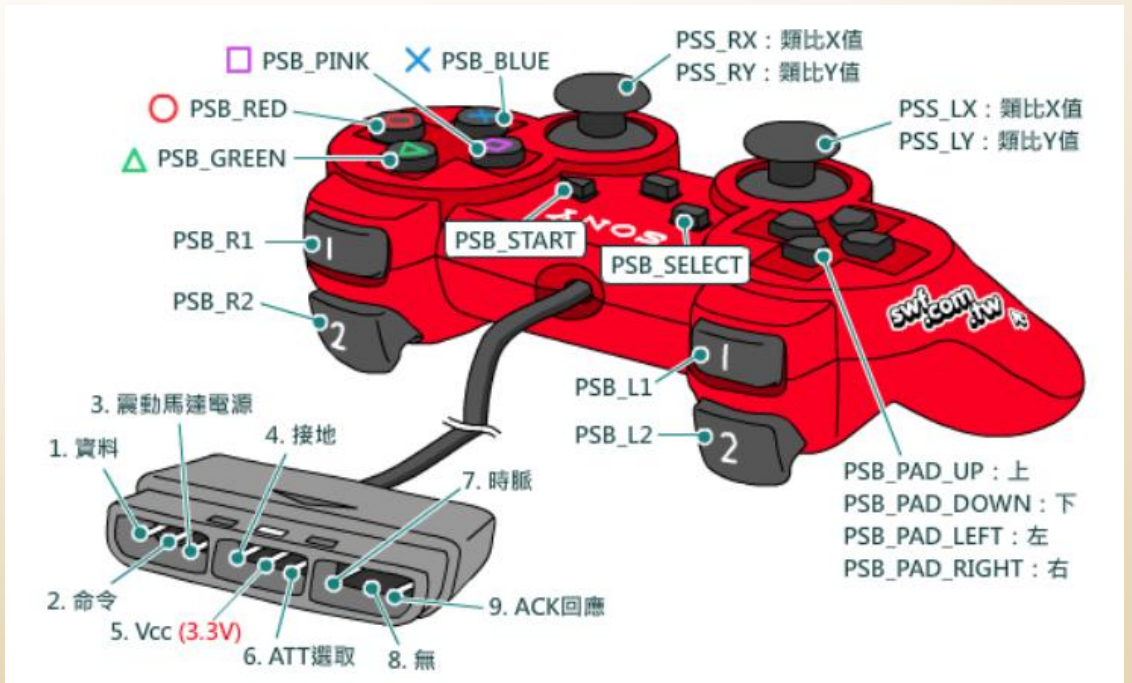
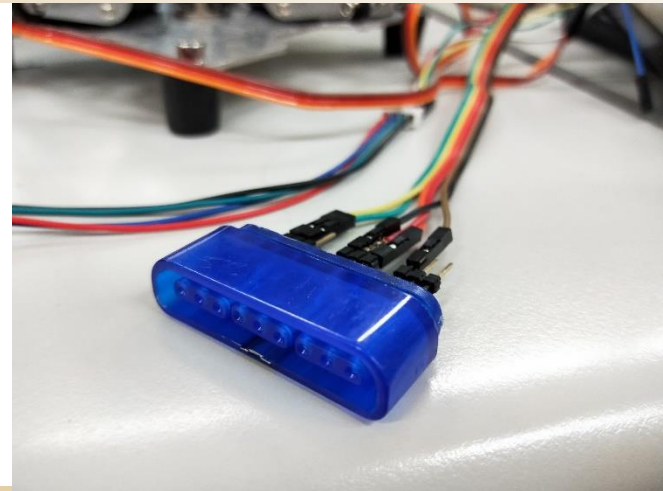
Y軸馬達

X軸馬達

中心支柱

主要原件介紹 PS2搖桿

- 將PS2的USB轉接頭拆開，連接到Arduino上
- 我們使用Sony PS2控制器的Arduino程式庫來做接腳的判定
- PS2搖桿來遙控球



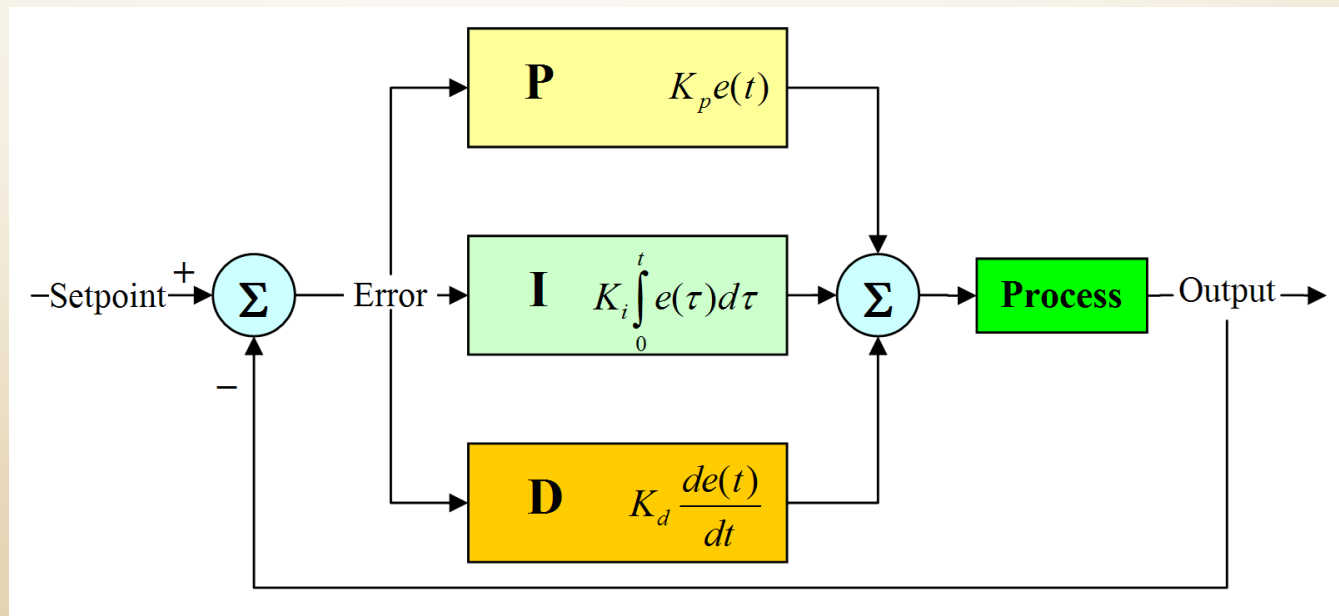
主要原件介紹 機架整體



原理介紹

原理介紹 PID控制

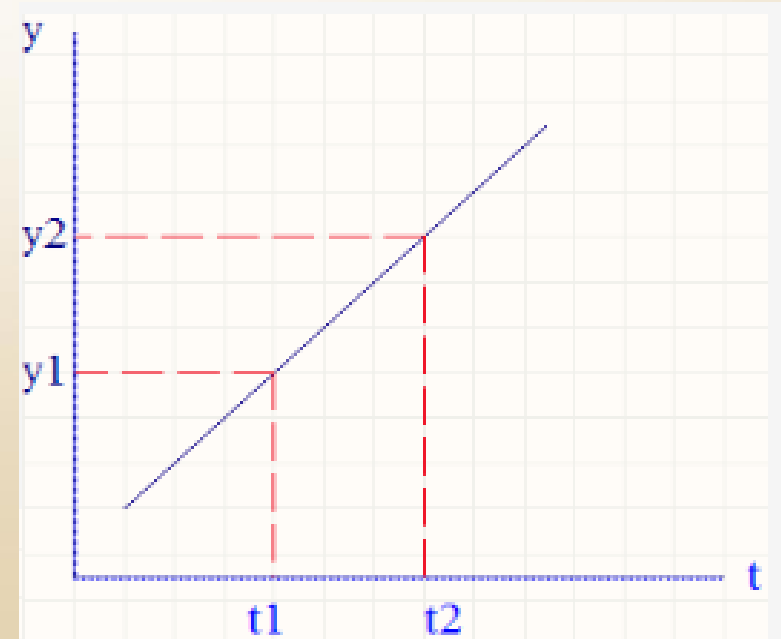
- PID控制器，由比例單元(P)、積分單元(I)和微分單元(D)組成
- 適用於需要透過適當的加減速以達到精準到位的系統
- 根據歷史數據和誤差的出現率來調整輸入值



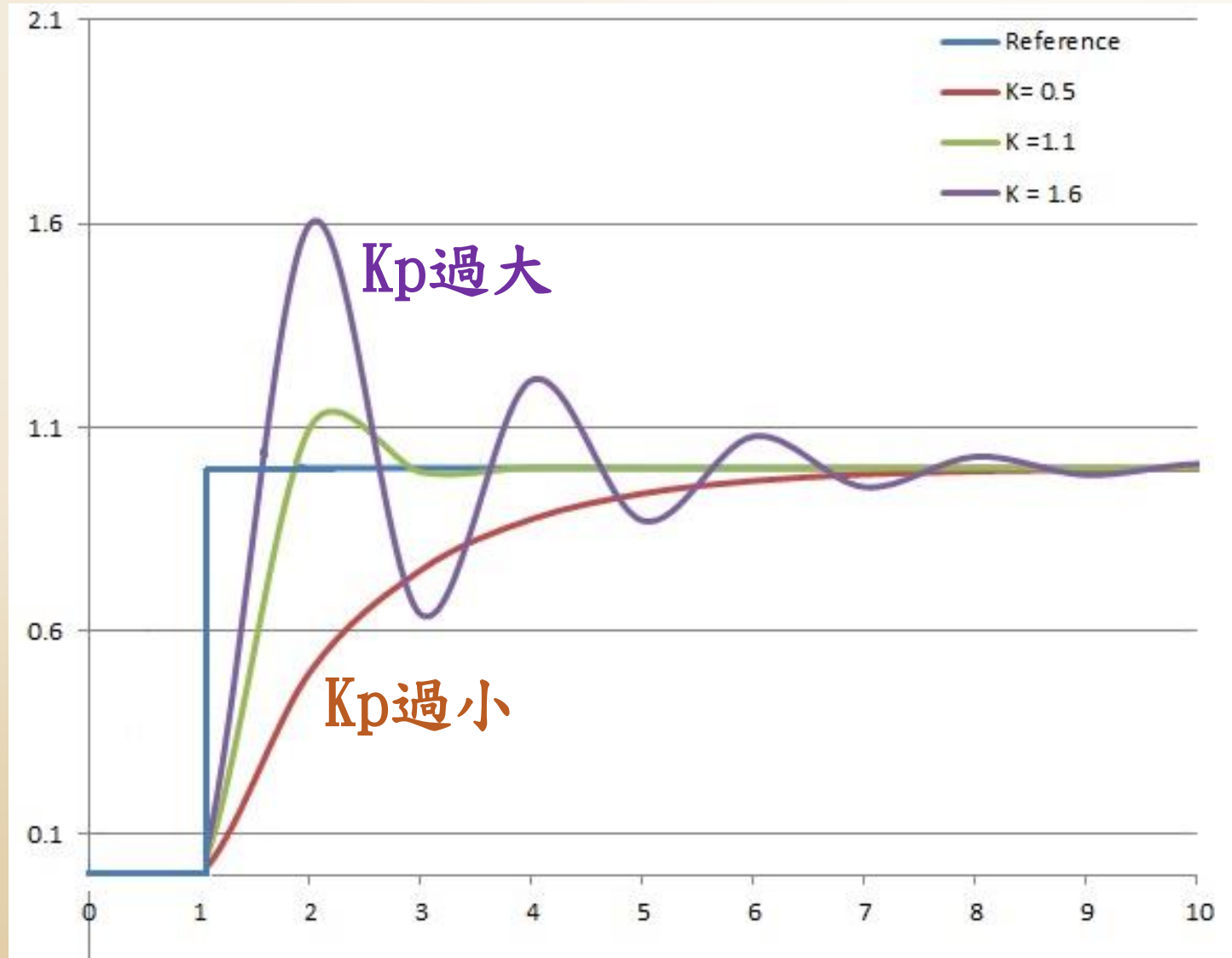
原理介紹 PID控制-比例單元(P)

- 比例控制考慮當前誤差，並依當前誤差來做出輸出的調變
- 比例控制為當前誤差(e)乘上一個常數(K_p)
- 比例增益太大會使系統不穩定，太小會使系統不敏感
- 誤差為0的時，輸出也是為0

$$P_{out} = K_p e(t)$$



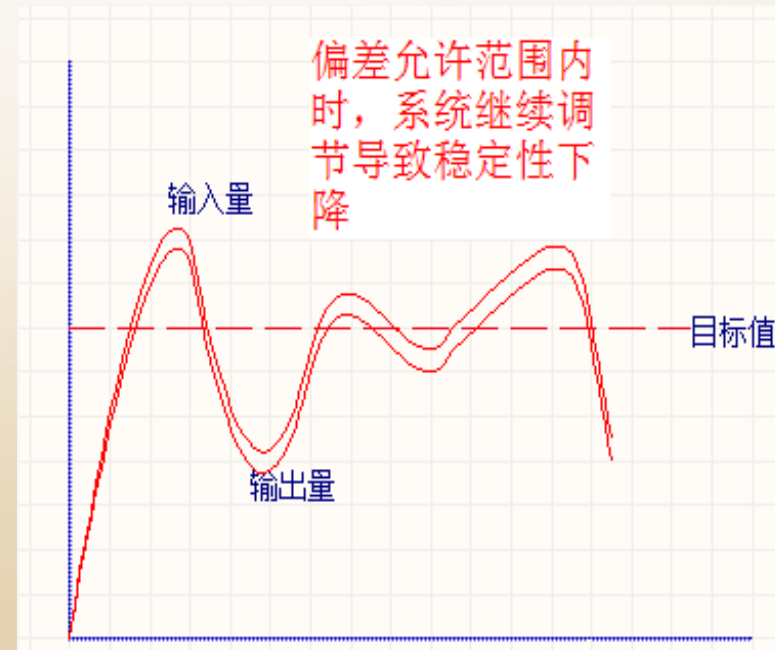
原理介紹 PID控制-比例單元(P)



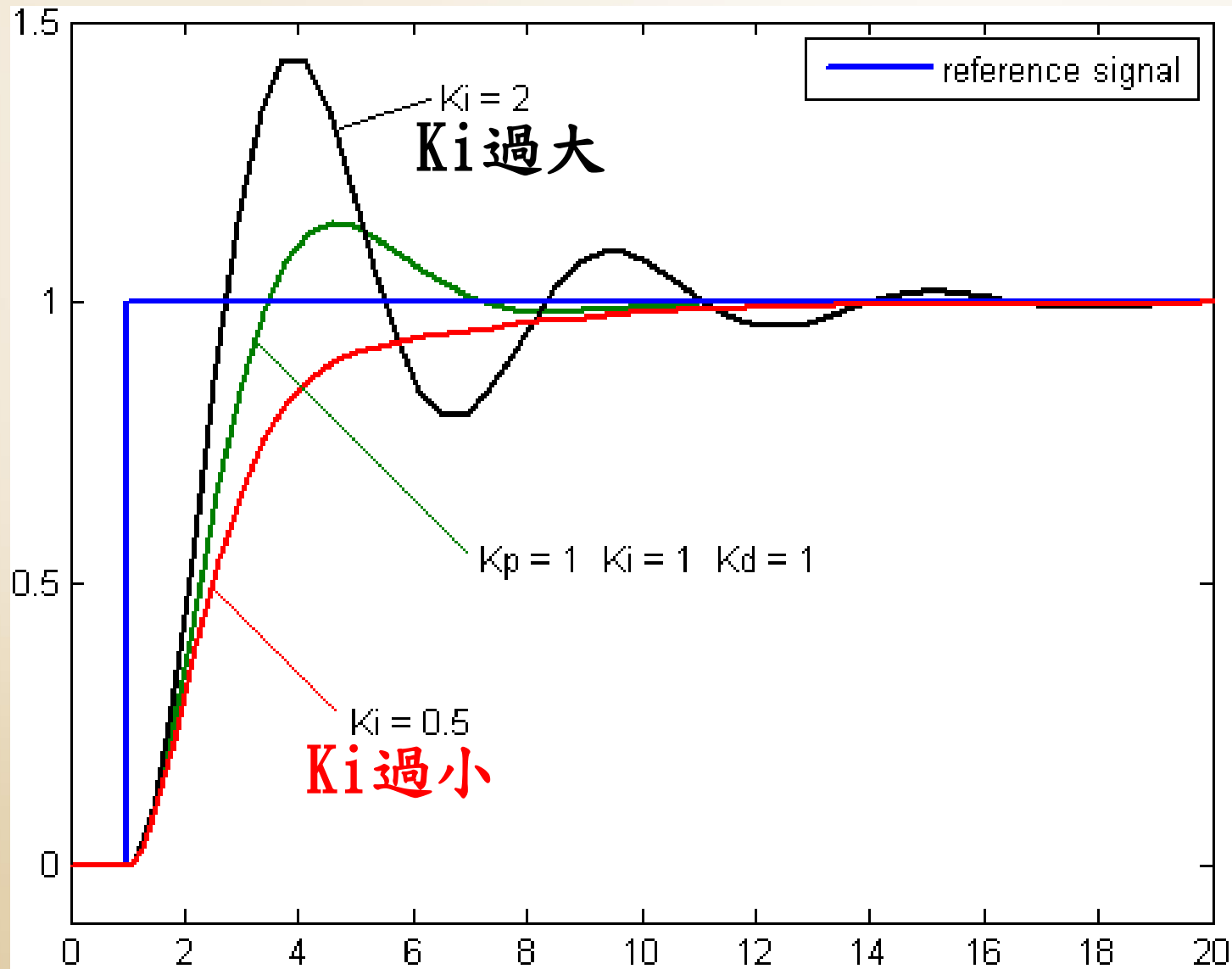
原理介紹 PID控制-積分單元(I)

- 積分控制考慮過去誤差，會將過去累積下的誤差做一次性的補足
- 積分控制為誤差和乘上一個常數(K_i)
- 積分增益越大，趨近設定值的速度越快，但是數值過大可能造成過衝

$$I_{out} = K_i \int_0^t e(\tau) d\tau$$



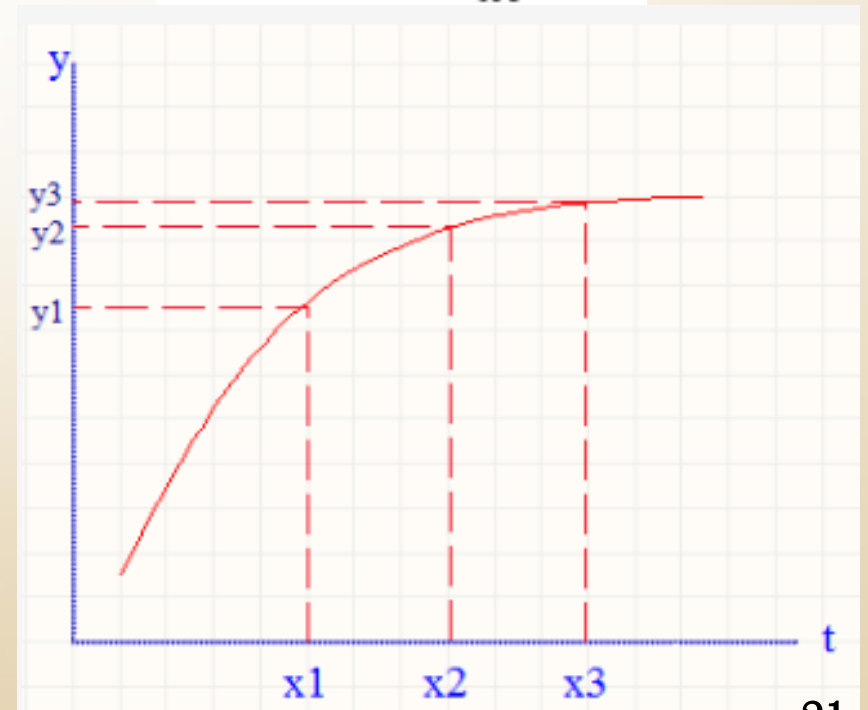
原理介紹 PID控制-積分單元(I)



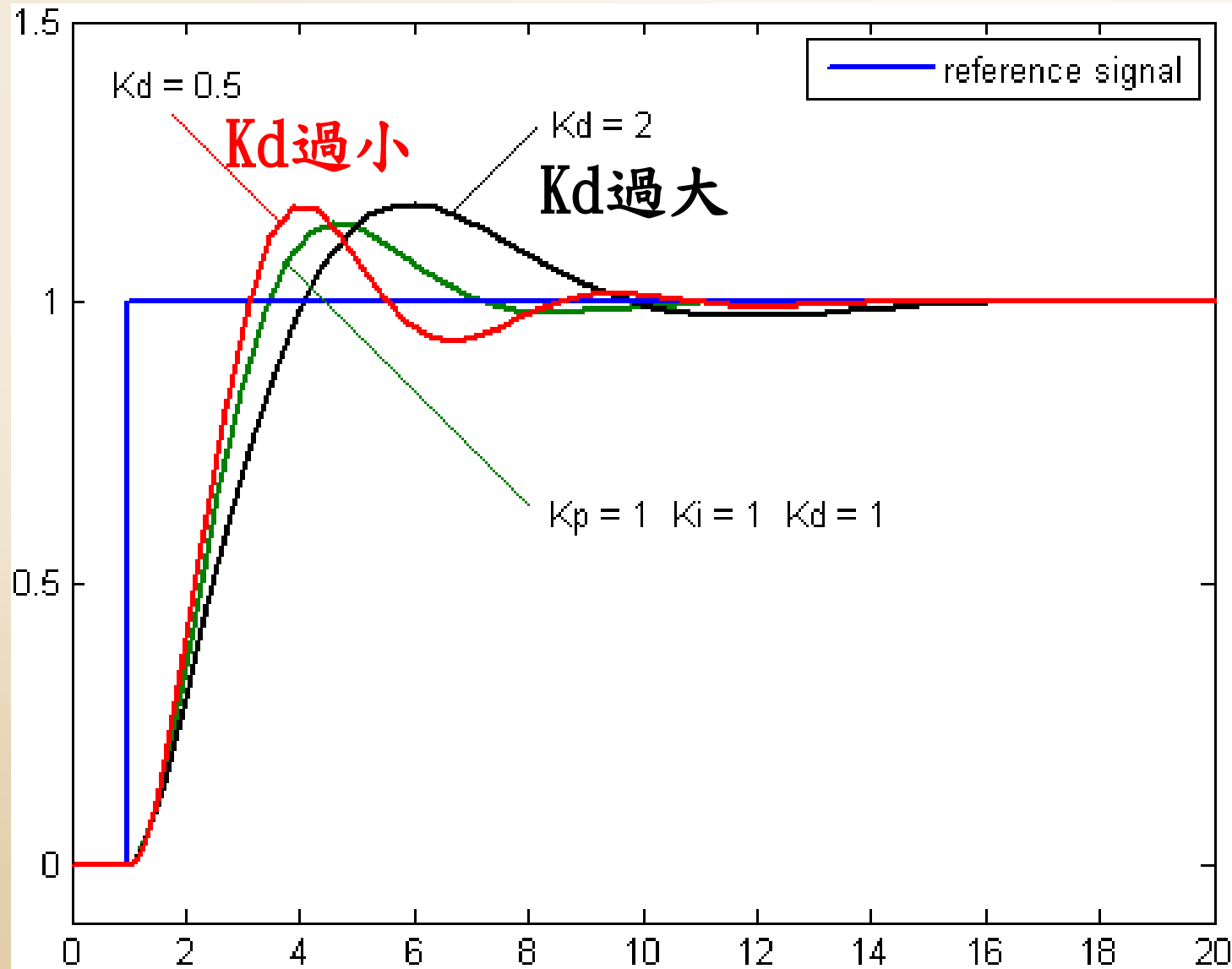
原理介紹 PID控制-微分單元(D)

- 微分控制考慮將來誤差，能預知未來誤差，使系統作出更快速的反應
- 計算誤差的導數並乘上一個常數(K_d)
- PID被稱為可預測的控制器的原因
- 實際上較少用到微分控制

$$D_{out} = K_d \frac{d}{dt} e(t)$$



原理介紹 PID控制-微分單元(D)



原理介紹 PID控制-參數調試(人工調整)

- 先將 K_i 及 K_d 設為零，增加 K_p 一直到迴路輸出震盪為止
- 再增加 K_i 減少穩態時的誤差
- 最後再慢慢調高 K_d 加速其穩定速度

調整方式	(on) 上升時間	超調量	安定時間	穩態誤差	穩定性 ^[10]
$\uparrow K_p$	減少 \downarrow	增加 \uparrow	小幅增加 \nearrow	減少 \downarrow	變差 \downarrow
$\uparrow K_i$	小幅減少 \searrow	增加 \uparrow	增加 \uparrow	大幅減少 $\downarrow\downarrow$	變差 \downarrow
$\uparrow K_d$	小幅減少 \searrow	減少 \downarrow	減少 \downarrow	變動不大 \rightarrow	變好 \uparrow

程式介紹

程式介紹 中心點平衡程式

- 以TouchScreen.h函式庫測量電阻屏得知目前球的位置
- 以目前位置和中心點之間的位置誤差做PID運算得知伺服馬達所需移動的角度

```
X=478,90.00,Y=534,90.00  
X=479,90.00,Y=533,90.00  
X=480,90.00,Y=533,89.00  
X=480,90.00,Y=532,90.00  
X=480,90.00,Y=532,90.00  
X=480,90.00,Y=532,90.00  
X=480,90.00,Y=532,90.00  
X=479,90.00,Y=533,90.00
```

```
TSPoint p = ts.getPoint();  
Inputx=p.x;  
Inputy=p.y;  
myPIDx.Compute();  
myPIDy.Compute();  
Outputx = map(Outputx, -255, 255, -20, 20);  
Outputy = map(Outputy, -255, 255, -20, 20);  
vocX = 90 + Outputx;  
vocY = 90 + Outputy;  
myservoY.write(vocY);  
myservoX.write(vocX);
```

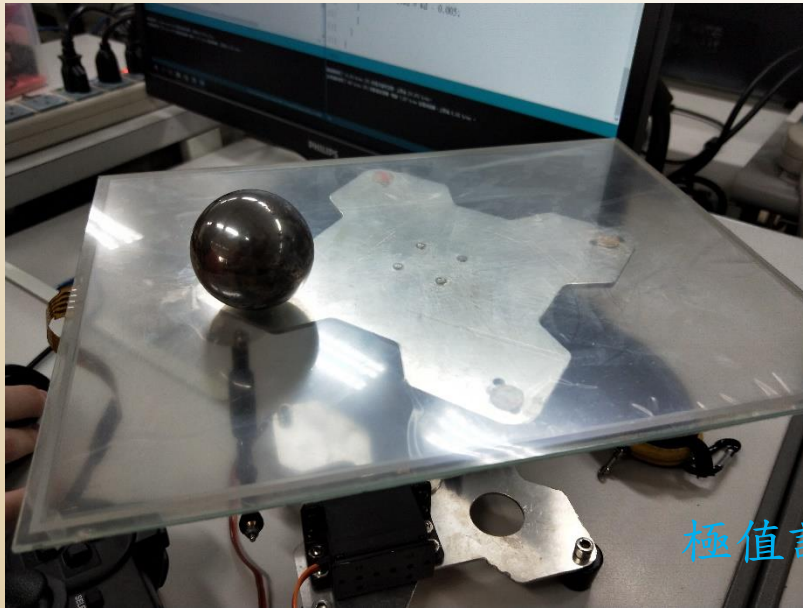
測量電阻屏讀值

PID運算

馬達輸出

程式介紹 搖桿控制球的平衡點

- 用PS2把手上的PSS_LX和PSS_LY(搖桿)來改變目標位置
- 做出移動位置的邊緣限制



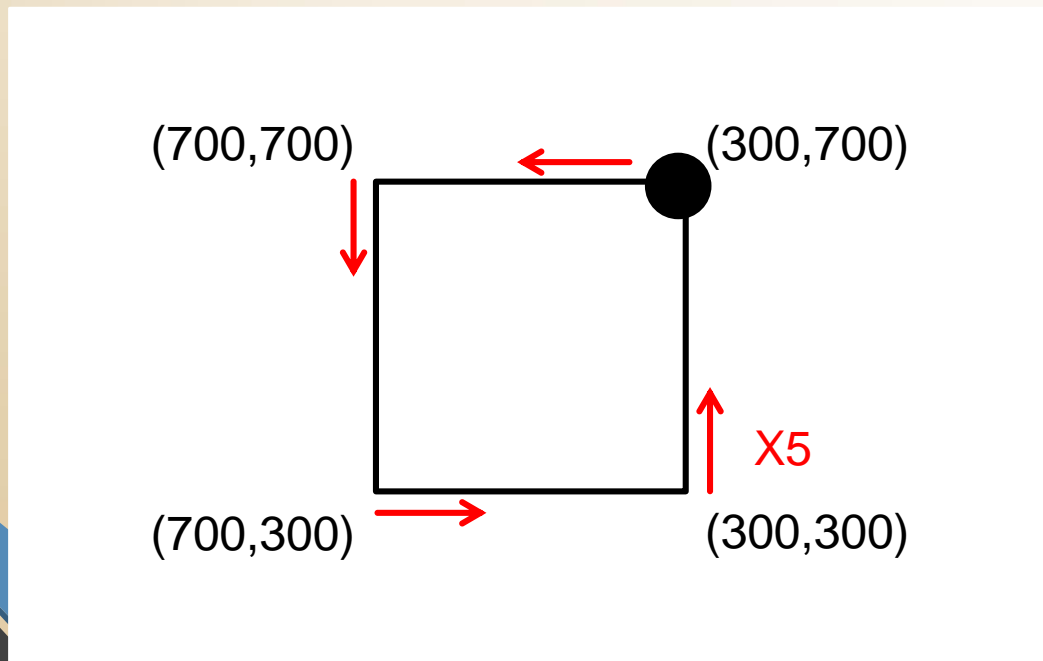
```
ps2x.read_gamepad(false, vibrate);  
apx = ps2x.Analog(PSS_LX);  
apy = ps2x.Analog(PSS_LY);  
apx = map(apx, 0, 255, 5, -5);  
apy = map(apy, 0, 255, -5, 5);  
aimx = aimx + apx,  
aimy = aimy + apy;  
if (aimx < 200)aimx = 200;  
if (aimx > 840)aimx = 840;  
if (aimy < 220)aimy = 220;  
if (aimy > 800)aimy = 800;
```

目標改變

極值設定

程式介紹 規律繞方

- 用PS2把手上的PSB_PINK(粉色)
- 以定時改變座標來畫出方形
- 五個循環後回到中心



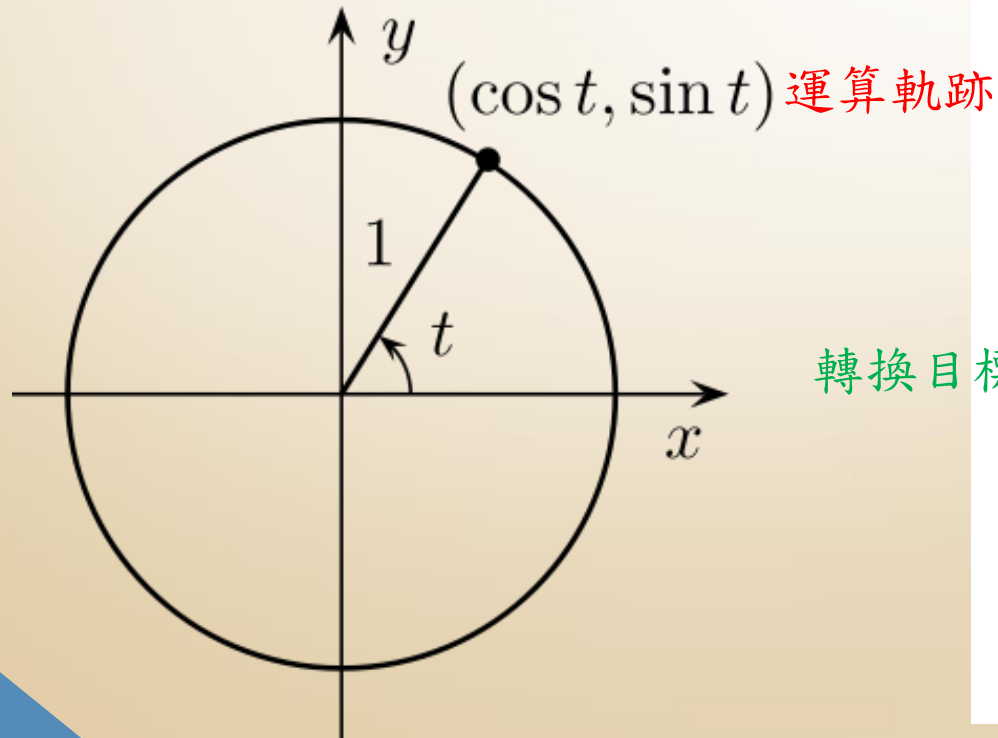
```
for (d = 0; d < c / 2; d++) {  
    for (f = 330; f <= 720; f++) {  
        aimy = f;  
        delay(sp / 4);  
    }  
    delay(150);  
    for (g = 250; g <= 780; g++) {  
        aimx = g;  
        delay(sp / 4);  
    }  
    delay(150);  
    for (h = 720; h >= 330; h--) {  
        aimy = h;  
        delay(sp / 4);  
    }  
    delay(150);  
    for (i = 780; i >= 230; i--) {  
        aimx = i;  
        delay(sp / 4);  
    }  
}
```

→ 改變座標

燈亮

程式介紹 規律繞圓

- 用PS2把手上的PSB_RED(紅色圓圈)來啟動程式
- 以Arduino運算出圓形軌跡
- 將圓形軌跡轉換為目標



```
if (ps2x.ButtonPressed(PSB_RED)) {  
  for (b = 0; b <= 360 * c ; b++) {  
    deg = b;  
    deg = deg / 180 * 3.1415926;  
    cx = cos(deg);  
    cy = sin(deg);  
    aimx = cx * 200 + 500;  
    aimy = cy * 200 + 500;  
    delay(sp);  
  }  
}
```

轉換目標

程式介紹 追點遊戲

- 用PS2把手上搖桿遙控球體
追逐LED亮點
- 以OLED顯示分數
- 分數越高球體移動越快

```
if (ps2x.ButtonPressed(P SB_GREEN)) {  
    score = 0;  
    pl = 0;  
    m = 2;  
    mo = 10;  
    w = random(1, 14);  
    aimx = 510;  
    aimy = 510;  
}
```

→ 初始設定

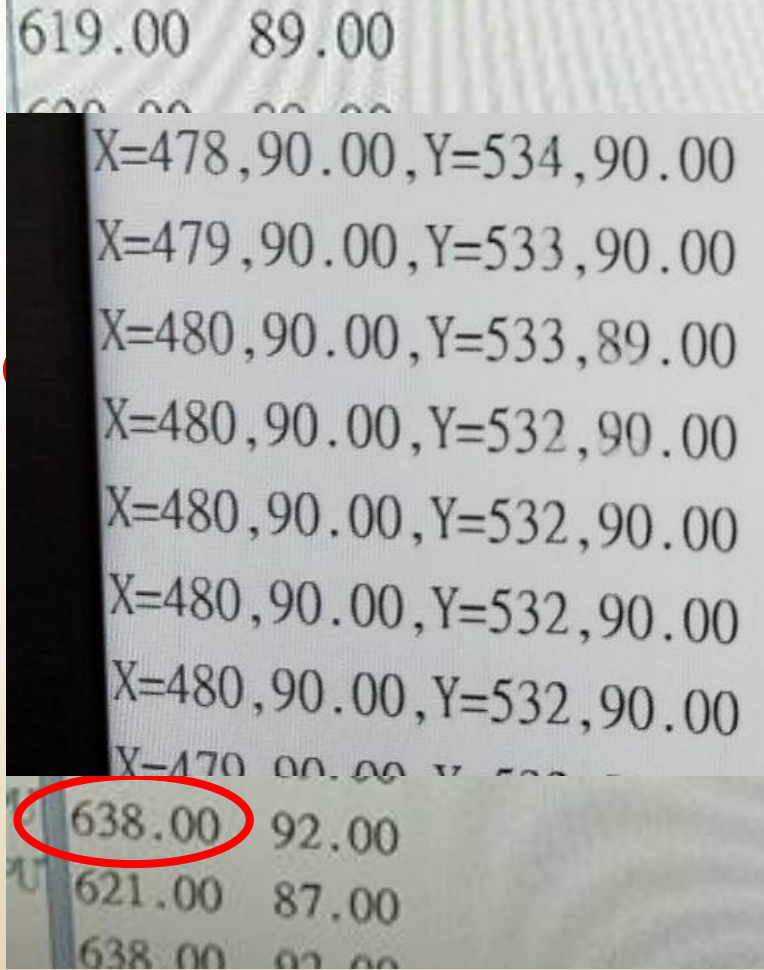
```
if (w == 13) {  
    digitalWrite(led13, HIGH); s13 = 1;  
    if (inx > 860 & inx < 890 & iny > 815 & iny < 845) {  
        digitalWrite(led13, LOW); s13 = 0;  
        w = random(1, 14);  
        m = m + 1;           亂數抽取LED燈  
        mo = mo + m;  
        pl = pl + 5;  
        score = score + 10 + pl;  
    }  
}
```

```
if (aimx > 950)aimx = 950;  
if (aimy < 70)aimy = 70;  
if (aimy > 950)aimy = 950;
```

→ 極值設定

遭遇問題

- Q: 球到達指定點後的穩定性不佳
- A: 以 TouchScreen.h 函式庫做出穩定測量
- Q: 專題較無具體實用性
- A: 向遊戲育樂方面發展



619.00 89.00
600.00 80.00
X=478,90.00,Y=534,90.00
X=479,90.00,Y=533,90.00
X=480,90.00,Y=533,89.00
X=480,90.00,Y=532,90.00
X=480,90.00,Y=532,90.00
X=480,90.00,Y=532,90.00
X=480,90.00,Y=532,90.00
X=470,00.00,Y=500,00.00
638.00 92.00
621.00 87.00
638.00 92.00



Q&A



報告結束