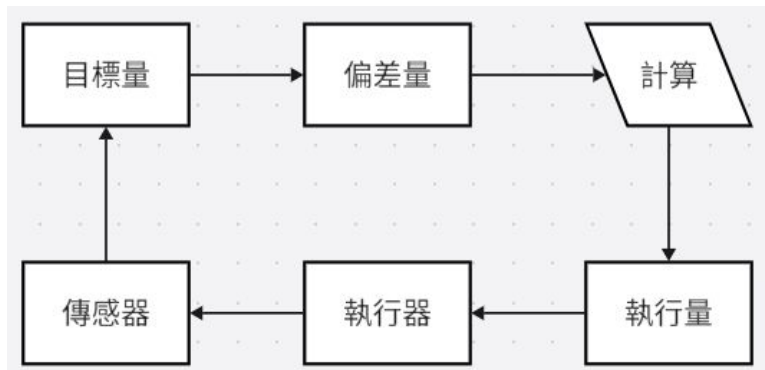


陀螺儀 PID 控制

PID 控制

簡單來說: 自動控制 -> 以達到精確的控制

閉環控制

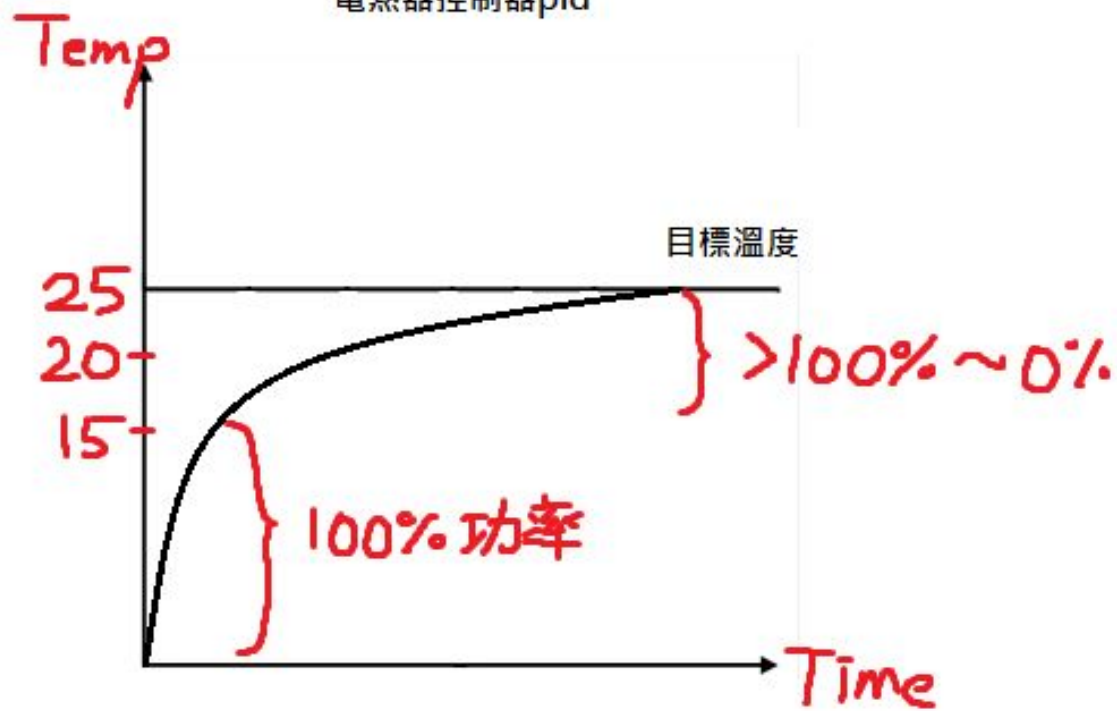


比例控制項

比例控制項

- 比例控制是一種最簡單的控制方式。
- 其控制器的輸出與輸入誤差訊號成比例關係。
- 比例控制考慮當前**誤差**，**誤差值**和一個正值的**常數 K_p (表示比例)**相乘。
- 比如說，一個電熱器控制器是在目標溫度和實際溫度差多於 10°C 時有100%的輸出，而其目標值是 25°C 。
- 那麼它在 15°C 以下的時候會輸出100%，在 20°C 的時候會輸出50%，在 24°C 的時候輸出10%，注意在誤差是0的時候，控制器的輸出也是0。

電熱器控制器pid



比例值 Proportional

- 當錯誤值越大, 修改值越大

算式:

偏差值 = 目標量 - 當前量

比例P = 偏差值 * 比例P系數(調整數值範圍)

不過:

由於比例P系數不能動態地調整, 當誤差很小時修改值亦會很小, 從而無法消除穩態誤差

積分I (參考)

解決比例P過小的問題

算法：

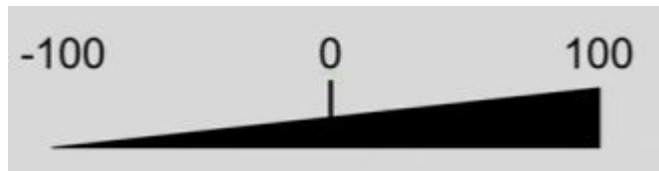
積分I = 積分I + 偏差量 * 積分I 系數

執行量 = 積分I + 比例P

例子：

如果偏差量維持偏向正數，積分I 就會不斷增大，以彌補比例P不足的問題

如果偏差量在正負數不斷浮動，積分則會互相抵消，從而不產生效果



微分D (參考)

防止P過大產生超出目標的問題

算法：

微分D = (偏差量 - 上一次偏差量) * 微分D系數

執行量 = 積分I + 比例P + 微分D

上一次偏差量 = 偏差量

例子：

當偏差值大幅移動時(偏差量 >>> 上一次偏差量)，大幅增加執行量

利用PID control 使小車轉至指定角度

先做好前設任務

```
1 #include <JY901.h>
2 // Define the pins for your motor control
3 #define RF_BI1 34
4 #define RF_BI2 35
5 #define RF_PWM 12
6
7 #define RR_BI1 37
8 #define RR_BI2 36
9 #define RR_PWM 8
10
11 #define LF_BI1 43
12 #define LF_BI2 42
13 #define LF_PWM 9
14
15 #define LR_BI1 29
16 #define LR_BI2 39
17 #define LR_PWM 5
```

```
25 void setup() {
26     // Initialize the motor control pins
27     pinMode(RF_BI1, OUTPUT);
28     pinMode(RF_BI2, OUTPUT);
29     pinMode(RF_PWM, OUTPUT);
30
31     pinMode(RR_BI1, OUTPUT);
32     pinMode(RR_BI2, OUTPUT);
33     pinMode(RR_PWM, OUTPUT);
34
35     pinMode(LF_BI1, OUTPUT);
36     pinMode(LF_BI2, OUTPUT);
37     pinMode(LF_PWM, OUTPUT);
38
39     pinMode(LR_BI1, OUTPUT);
40     pinMode(LR_BI2, OUTPUT);
41     pinMode(LR_PWM, OUTPUT);
42     Serial.begin(9600);
43     Serial3.begin(9600);
44     // Initialize the JY901 sensor
45     JY901.attach(Serial3);
46 }
```

先做好前設任務 cont.

```
48 //LF LR
49 void controlRF(int BI1, int BI2, int speed){
50     if (speed > 0){
51         digitalWrite(RF_BI1, BI1);
52         digitalWrite(RF_BI2, BI1);
53         analogWrite(RF_PWM, constrain(speed, -255, 255));
54     }
55     else if(speed < 0){
56         digitalWrite(RF_BI1, BI2);
57         digitalWrite(RF_BI2, BI1);
58         analogWrite(RF_PWM, constrain(-speed, -255, 255));
59     }
60 }
61
62 void controlRR(int BI1, int BI2, int speed){
63     if (speed > 0){
64         digitalWrite(RR_BI1, BI1);
65         digitalWrite(RR_BI2, BI2);
66         analogWrite(RR_PWM, constrain(speed, -255, 255));
67     }
68     else if(speed < 0){
69         digitalWrite(RR_BI1, BI2);
70         digitalWrite(RR_BI2, BI1);
71         analogWrite(RR_PWM, constrain(-speed, -255, 255));
72     }
73 }
```

```
75 void controlLF(int BI1, int BI2, int speed){
76     if (speed > 0){
77         digitalWrite(LF_BI2, BI2);
78         digitalWrite(LF_BI1, BI1);
79         analogWrite(LF_PWM, constrain(speed, -255, 255));
80     }
81     else if(speed < 0){
82         digitalWrite(LF_BI2, BI1);
83         digitalWrite(LF_BI1, BI2);
84         analogWrite(LF_PWM, constrain(-speed, -255, 255));
85     }
86 }
87
88 void controlLR(int BI1, int BI2, int speed){
89     if (speed > 0){
90         digitalWrite(LR_BI2, BI2);
91         digitalWrite(LR_BI1, BI1);
92         analogWrite(LR_PWM, constrain(speed, -255, 255));
93     }
94     else if(speed < 0){
95         digitalWrite(LR_BI2, BI1);
96         digitalWrite(LR_BI1, BI2);
97         analogWrite(LR_PWM, constrain(-speed, -255, 255));
98     }
99 }
```

背後原理

1. 先設定目標角度
2. 再計算角度誤差值
3. 利用($K_p * \text{誤差值}$)作為麥克姆輪的加減速
4. 若角度誤差值為0則停止, 否則:
5. 利用if else 判斷偏離方向, 在其偏離方向的麥克姆輪向前, 另一方向的麥克姆輪減向後, 使車子返回正軌

void loop()

```
101 void loop() {  
102     // Read the sensor data from the JY901  
103     JY901.receiveSerialData();  
104     // Read the current angle  
105     float currentAngle = JY901.getYaw(); // Adjust the axis as needed  
106     Serial.println(currentAngle);  
107     // Calculate the error  
108     float error = targetAngle - currentAngle;
```

void loop() cont.

```
118 // Update the motor control based on the PID output
119 if(error < 1 && error > -1){
120     controlRF(1, 0, 0);
121     controlRR(1, 0, 0);
122     controlLF(0, 1, 0);
123     controlLR(0, 1, 0);
124 }
125 else if (error > 1) {
126     controlLF(0, 1, -output);
127     controlRR(1, 0, output);
128     controlRF(1, 0, output);
129     controlLR(0, 1, -output); //turn left
130 }
131 else if(error < -1) {
132     controlRF(1, 0, output);
133     controlLR(0, 1, -output);
134     controlLF(0, 1, -output);
135     controlRR(1, 0, output); //turn right
136 }
137 // Add a delay to control the loop rate
138 //delay(10);
139 }
```

如果將目標角度設成-179度會發生什麼？

Program

```
109  if (error > 180) {  
110      error -= 360;  
111  } else if (error < -180) {  
112      error += 360;  
113  }
```

利用PID control 使小車走直線

先做好前設任務

```
1 #include <JY901.h>
2 // Define the pins for your motor control
3 #define RF_BI1 34
4 #define RF_BI2 35
5 #define RF_PWM 12
6
7 #define RR_BI1 37
8 #define RR_BI2 36
9 #define RR_PWM 8
10
11 #define LF_BI1 43
12 #define LF_BI2 42
13 #define LF_PWM 9
14
15 #define LR_BI1 29
16 #define LR_BI2 39
17 #define LR_PWM 5
```

```
25 void setup() {
26     // Initialize the motor control pins
27     pinMode(RF_BI1, OUTPUT);
28     pinMode(RF_BI2, OUTPUT);
29     pinMode(RF_PWM, OUTPUT);
30
31     pinMode(RR_BI1, OUTPUT);
32     pinMode(RR_BI2, OUTPUT);
33     pinMode(RR_PWM, OUTPUT);
34
35     pinMode(LF_BI1, OUTPUT);
36     pinMode(LF_BI2, OUTPUT);
37     pinMode(LF_PWM, OUTPUT);
38
39     pinMode(LR_BI1, OUTPUT);
40     pinMode(LR_BI2, OUTPUT);
41     pinMode(LR_PWM, OUTPUT);
42     Serial.begin(9600);
43     Serial3.begin(9600);
44     // Initialize the JY901 sensor
45     JY901.attach(Serial3);
46 }
```

先做好前設任務 cont.

```
48 //LF LR
49 void controlRF(int BI1, int BI2, int speed){
50     if (speed > 0){
51         digitalWrite(RF_BI1, BI1);
52         digitalWrite(RF_BI2, BI2);
53         analogWrite(RF_PWM, constrain(speed, -255, 255));
54     }
55     else if(speed < 0){
56         digitalWrite(RF_BI1, BI2);
57         digitalWrite(RF_BI2, BI1);
58         analogWrite(RF_PWM, constrain(-speed, -255, 255));
59     }
60 }
61
62 void controlRR(int BI1, int BI2, int speed){
63     if (speed > 0){
64         digitalWrite(RR_BI1, BI1);
65         digitalWrite(RR_BI2, BI2);
66         analogWrite(RR_PWM, constrain(speed, -255, 255));
67     }
68     else if(speed < 0){
69         digitalWrite(RR_BI1, BI2);
70         digitalWrite(RR_BI2, BI1);
71         analogWrite(RR_PWM, constrain(-speed, -255, 255));
72     }
73 }
```

```
75 void controlLF(int BI1, int BI2, int speed){
76     if (speed > 0){
77         digitalWrite(LF_BI2, BI2);
78         digitalWrite(LF_BI1, BI1);
79         analogWrite(LF_PWM, constrain(speed, -255, 255));
80     }
81     else if(speed < 0){
82         digitalWrite(LF_BI2, BI1);
83         digitalWrite(LF_BI1, BI2);
84         analogWrite(LF_PWM, constrain(-speed, -255, 255));
85     }
86 }
87
88 void controlLR(int BI1, int BI2, int speed){
89     if (speed > 0){
90         digitalWrite(LR_BI2, BI2);
91         digitalWrite(LR_BI1, BI1);
92         analogWrite(LR_PWM, constrain(speed, -255, 255));
93     }
94     else if(speed < 0){
95         digitalWrite(LR_BI2, BI1);
96         digitalWrite(LR_BI1, BI2);
97         analogWrite(LR_PWM, constrain(-speed, -255, 255));
98     }
99 }
```

背後原理

1. 先設定目標角度為0
2. 再計算角度誤差值
3. 利用 K_p 倍大誤差值作為麥克姆輪的額外加減速
4. 若角度誤差值為0則繼續直走, 否則:
5. 利用if else 判斷偏離方向, 在其偏離方向的麥克姆輪加速, 另一方向的麥克姆輪減速, 使車子返回正軌

void loop()

```
101 void loop() {  
102     // Read the sensor data from the JY901  
103     JY901.receiveSerialData();  
104     // Read the current angle  
105     float currentAngle = JY901.getYaw(); // Adjust the axis as needed  
106     Serial.println(currentAngle);  
107     // Calculate the error  
108     float error = targetAngle - currentAngle;
```

void loop() cont.

```
115 // Calculate the PID output
116 float output = Kp * error;
117 //Serial.println(output);
118
119 // Update the motor control based on the PID output
120 if(error < 1 && error > -1){
121     controlRF(1, 0, 100);
122     controlRR(1, 0, 100);
123     controlLF(0, 1, 100);
124     controlLR(0, 1, 100);
125 }
126 else if (error > 1) {
127     controlLF(0, 1, 100-output);
128     controlRR(1, 0, 100+output);
129     controlRF(1, 0, 100+output);
130     controlLR(0, 1, 100-output); //turn left
131 }
132 else if(error < -1) {
133     controlRF(1, 0, 100-(output*-1));
134     controlLR(0, 1, 100+(output*-1));
135     controlLF(0, 1, 100+(output*-1));
136     controlRR(1, 0, 100-(output*-1)); //turn right
137 }
138 // Add a delay to control the loop rate
139 //delay(10);
140 }
```