

PID 控制器(比例-积分-微分控制器)，由比例单元(P)、积分单元(I)和微分单元(D)组成。透过 K_p ， K_i 和 K_d 三个参数的设定。PID 控制器主要适用于基本上线性，且动态特性不随时间变化的系统。

经典 PID

```
float Kp , Ki , Kd;

int PID_Control( int value , int target )
{
    int PWM, Bias,Bias_D;
    static int error, last_Bias;
    Bias = value-target;          //获取偏差
    error += Bias;                //偏差积分
    Bias_D = Bias-last_Bias;      //偏差微分
    PWM = Kp*Bias + Ki*error + Kd*Bias_D; //获取最终数值
    last_Bias =Bias;              //保存为上次偏差
    return PWM;
}
```

PID 位置控制

```
float Kp , Ki , Kd;

int Position_Control(int encoder,int target)
{
    static float error, Last_Bias;
    float PWM , Bias , Differential;
    Bias = encoder-target;        //获取偏差
    error += Bias;                //累加出位移
    Differential=Bias - Last_Bias; //求出偏差的微分
    PWM=Kp*Bias + Ki*error + Kd*Differential; //获取最终数值
    Last_Bias=Bias; //保存为上次偏差
    return PWM;
}
```

PI 速度控制

```
float Kp , Ki ;  
  
int Velocity_Control(int encoder,int target)  
{  
    static int PWM,Last_bias;  
    Int bias;  
    bias = target - encoder;    //获取偏差  
    PWM+=Kp*(bias-Last_bias)+Ki*bias; ////增量式累加  
    Last_bias = bias;//保存为上次偏差  
    return PWM;  
}
```

参数整定

以 PID 位置控制为例

P 是比例控制，在未超调的情况下，值越大响应越快，若过大会产生震荡。

I 是积分控制，一般用来消除余差，使静止时更接近目标值。

D 是微分控制，一般用来抑制震荡。某些获取参数滞后的场合，D 值会更大。

一般来说，调参时，先置 $K_i=0;K_d=0;$

然后逐步加大 K_p 的值，直到系统出现震荡，接着再逐步加大 K_d 的值，直到系统稳定，最后有需要再去调整 K_i 的值，使实际值更接近目标值。