

轮 趣 科 技

基于平衡车的电机 PID 控制 开发教程

推荐关注我们的公众号获取更新资料



版本说明:

版本	日期	内容说明
V2.0	2021/06/16	第一次发布

网址: www.wheeltec.net

目录

1. 基于平衡车的电机 PID 控制开发教程.....	3
1.1 PID 闭环控制.....	3
1.2 使用操作.....	6

1. 基于平衡车的电机 PID 控制开发教程

1.1 PID 闭环控制

PID 控制即比例-积分-微分控制，即通过对偏差进行比例-积分-微分控制，使得当前值趋于目标值的过程。一般来说，比例 P 控制是必须的，所以衍生出很多组合的 PID 控制器，如 PD、PI、PID 等。离散 PID 公式为：

$$u(k) = k_p e(k) + K_i \sum e(k) + K_d [e(k) - e(k-1)]$$

其中， $e(k)$ 为 k 时刻的偏差， $u(k)$ 为输出的控制量，对于电机来说即是 PWM。

① 位置闭环控制

根据位置闭环控制公式： $PWM = k_p e(k) + K_i \sum e(k) + K_d [e(k) - e(k-1)]$ ，位置信息由编码器读数得出，初始化编码器读数为 10000，偏差为目标位置和当前位置的差值，可以用 C 语言实现如下：

```
int Position_PID (int Position, int Target)
{
    static float Bias, Pwm, Integral_bias, Last_Bias;
    Bias=Target-Position;                //计算偏差
    Integral_bias+=Bias;                  //求出偏差的积分
    if(Integral_bias>100000) Integral_bias=100000;
    if(Integral_bias<-100000) Integral_bias=-100000;
    Pwm=Position_KP*Bias+Position_KI*Integral_bias/10+Position_KD*(Bias-Last_Bias);
    //位置式 PID 控制器
    Last_Bias=Bias;                       //保存上一次偏差
    return Pwm;                           //增量输出
}
```

入口参数为当前测量值与目标值，返回值为电机控制的 PWM。为防止超出执行器输出范围，我们在程序中作了积分限幅处理。

② 速度闭环控制

速度信息 M 法测速得出，即单位时间内读取编码器脉冲数得到速度。速度闭环控制就是根据单位时间获取的脉冲数与目标值进行比较，得到偏差，使偏差趋于 0 的过程。

PID 控制要求当误差为 0 时，控制器的输出也应该为 0，但是对于速度反馈，当误差为 0 时，即达到了我们目标速度，若输出 PWM 也为 0，那么必定不能维

持目标速度，故对于速度控制，我们使用增量式 PID，令所有 PWM 输出叠加作用于电机，使得速度保持。增量式 PID 控制公式为：

$$PWM += k_p[e(k) - e(k - 1)] + K_i e(k) + K_d[e(k) - 2e(k - 1) - e(k - 2)]$$

其中， $e(k)$ 是本次偏差， $e(k - 1)$ 是上次偏差， $e(k - 2)$ 为上上次偏差。增量式 PID 其实就是每次增加控制量的差值，和位置式 PID 相比不会积累误差，计算量较少。在我们速度闭环控制系统里只使用 PI 控制，因此 PID 控制器可以简化为：

$$PWM += k_p[e(k) - e(k - 1)] + K_i e(k)$$

程序实现代码如下：

```
int Incremental_PI (int Encoder, int Target)
{
    static float Bias, Pwm, Last_bias;
    Bias=Target-Encoder;           //计算偏差
    Pwm+=Velocity_KP*(Bias-Last_bias)+Velocity_KI*Bias; //增量式 PI 控制器
    Last_bias=Bias;                //保存上一次偏差
    return Pwm;                   //增量输出
}
```

③ 参数整定

以位置式 PID 为例子说明如何进行参数整定。首先我们需要明确我们的控制目标，也就是满足控制系统的 3 个要求：1.稳定性 2.快速性 3.准确性。具体的评估指标有最大超调量、上升时间、静差等。最大超调量是响应曲线的最大峰值与稳态值的差，是评估系统稳定性的一个重要指标；上升时间是响应曲线从原始工作状态出发，第一次到达稳态值所需的时间，是评估系统快速性的一个重要指标；静差是被控量的稳定值与给定值之差，一般用于衡量系统的准确性。

PID 参数的大致作用如下：

KP：调节到达目标的时间。增大 KP 会提高响应速度，但可能会造成超调，有静态误差。

KI：调节 KI 参数能够减少静差，但容易造成振荡。

KD：相当与阻尼力的作用，增大 KD 参数能够抑制振荡，尽快稳定，但有可能造成调节周期过长。

我们需要调节目标值为 10390，当前值为 10000，观察响应曲线的变化。一般我们进行 PID 参数整定的时候，首先设 KI 和 KD 的值为零，然后把 KP 参数

从 0 逐渐增大，直至出现振荡。

当 $K_P=62$, $K_I=0$, $K_D=0$ 时响应曲线如下：

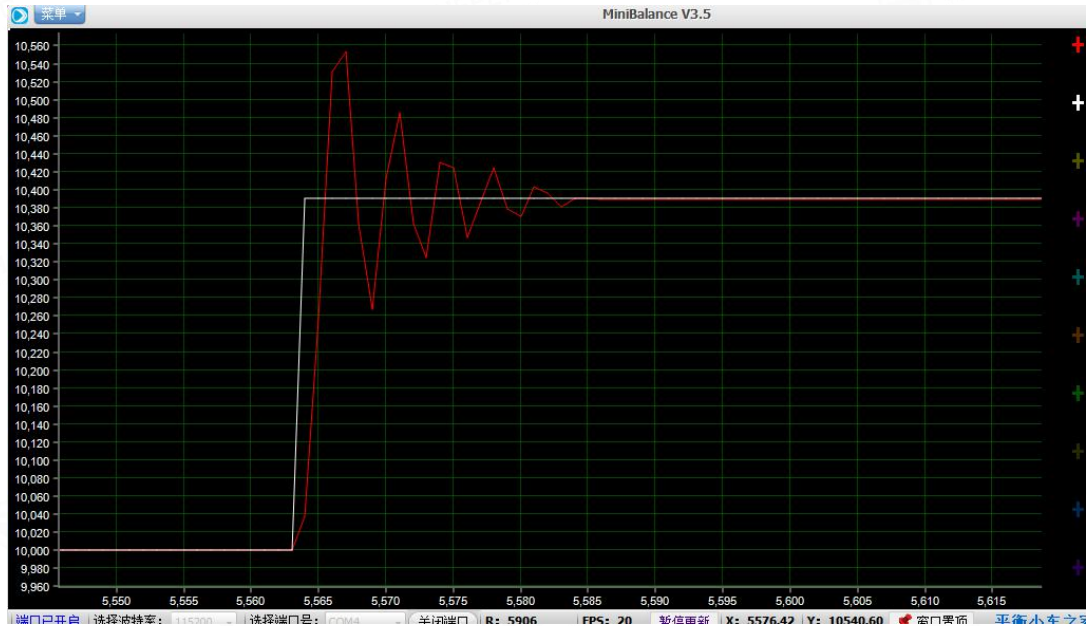


图 1-1

图中，白线为目标值，红线为当前值。可以看到，这个时候 K_P 值较大，出现了振荡现象。为什么 K_I 值为零，还是没有出现静差呢？因为这个时候 K_P 值已经很大了，静差一般是在 K_P 值较小而 K_I 值为 0 的时候出现的。

当 $K_P=2$, $K_I=0$, $K_D=0$ 时，响应曲线如下：

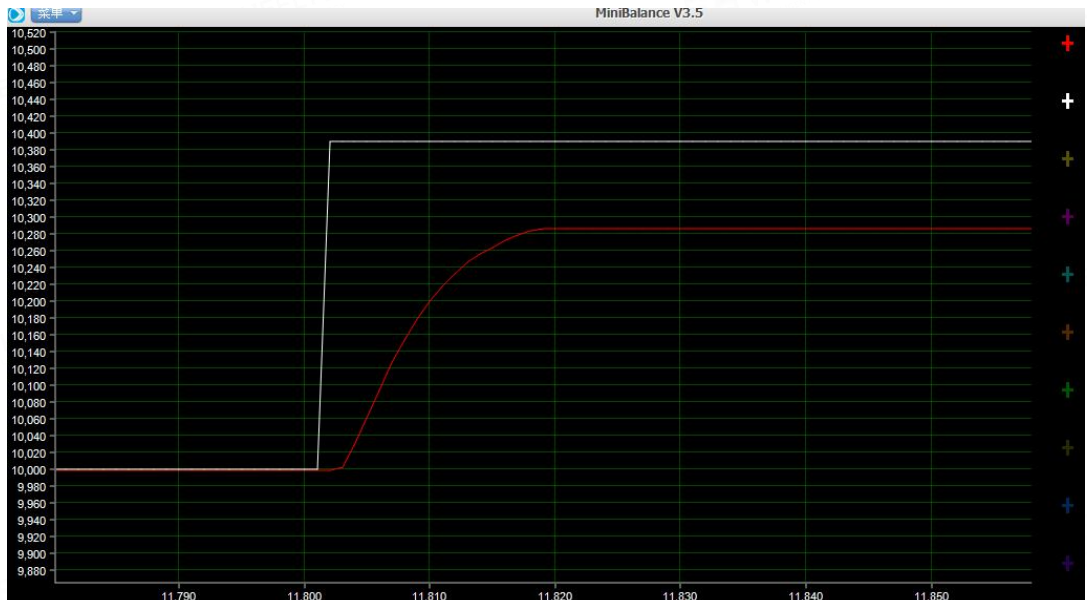


图 1-2

可以看到这个时候 K_P 较小，出现了静差，响应速度也变慢了，验证了我们

的想法。提高 K_P 能提高响应速度，但会导致振荡，可以加入微分控制抑制振荡。

当 $K_P=62$, $K_I=0$, $K_D=188$ 时，响应曲线如下：



图 1-3

当加入微分控制的时候，可以看到，系统的振荡得到了抑制，但是系统的响应变慢了。因为微分控制相当于一个阻尼力，引入微分控制相当与增大系统的阻尼。这个时候需要结合 K_P 和 K_I 进行进一步的优化。在实际生产中，需要对不同项目进行评估，比如一个系统对快速性要求不高，但对稳定性和准确性的要求很高，那么就需要严格控制超调量和静差。

位置 PID 控制的调节经验可以总结为：先只使用 P 控制，增大 P 系数知道系统振荡，然后加入微分控制增大阻尼，消除振荡之后再根据系统对响应和静差等具体要求，调节 P 和 I 参数。

1.2 使用操作

我们提供了基于平衡小车的 PID 例程源码，能够在小车上学习 PID 的知识。下载好 PID 学习套件的程序，打开电源开关，可以观察到 OLED 显示屏被点亮，我们只需按照提示进入相应的模式即可。程序有两种模式，分别为位置模式和速度模式。显示屏内容如下：

Trun Right Wheel	第一行：提示“转动右轮”
Encoder: 10000	第二行：显示当前右轮编码器的读数
Forward To	
Position	第四行：提示向前转动进入位置模式

Backward To
Velocity

第六行：提示向后转动进入速度模式

若向前转动右轮则进入位置模式，向后转动右轮则进入速度模式。可用数据线连接上位机，选择波特率 115200 查看波形。

① 位置模式

向前转动右轮进入位置模式后，可以看到显示屏显示如下：

POSITION	MODE	第一行：显示当前模式（位置/速度）
VOLTAGE:	11.21V	第二行：显示当前电池电压
KP: 40.0	Y A: 2.0	第三行：显示 KP 参数值和是否可修改，每次修改的幅度
KI: 40.0	N A: 0.1	第四行：显示 KI 参数值和是否可修改，每次修改的幅度
KD: 40.0	N A: 3.0	第五行：显示 KD 参数值和是否可修改，每次修改的幅度
T: 10000	P: 10000	第六行：显示目标值与当前值

在每个参数后面都有标识查看当前参数是否可选修改，转动右轮即可修改数值。如 KP 参数后面有 Y(YES)，表示可修改，转动右轮即可改变大小；KI 和 KD 参数后面为 N(NO)表示不可修改。长按用户按键 2s 可以选择修改的参数，单击用户按键会加大的编码器目标值，双击用户按键会减少编码器目标值，此时 PID 控制器会开始调节，使当前值趋于目标值。

按键动作	单击	双击	长按
功能	增大目标值	减少目标值	选择修改参数

表 1-1 按键动作说明

电池电压若低于 11.1V，此时电机将会停止，请及时充电。

② 速度模式

往后转动右轮进入速度模式，可以查看显示屏显示内容如下：

VELOCITY	MODE	第一行：显示当前模式（位置/速度）
VOLTAGE:	11.21V	第二行：显示当前电池电压
KP: 5.0	Y A: 1.0	第三行：显示 KP 参数值和是否可修改，每次修改的幅度
KI: 5.0	N A: 1.0	第四行：显示 KI 参数值和是否可修改，每次修改的幅度
T: 40	P: 39	第六行：显示目标值与当前值

进入速度模式后，左轮会自动维持在目标速度，当单击/双击用户按键时，会增加/减少目标速度，这时程序会控制左轮趋于目标速度。

③ 上位机波形查看

我们可以使用 MiniBalance 上位机来查看波形，评估 PID 调节的效果。进入位置/速度模式后，程序会以 115200 波特率发送数据给上位机，上位机界面如图

1-4 所示。



图 1-4

我们选择相应的波特率和端口，点击打开端口，可以查看到波形如下，以位置模式为例。

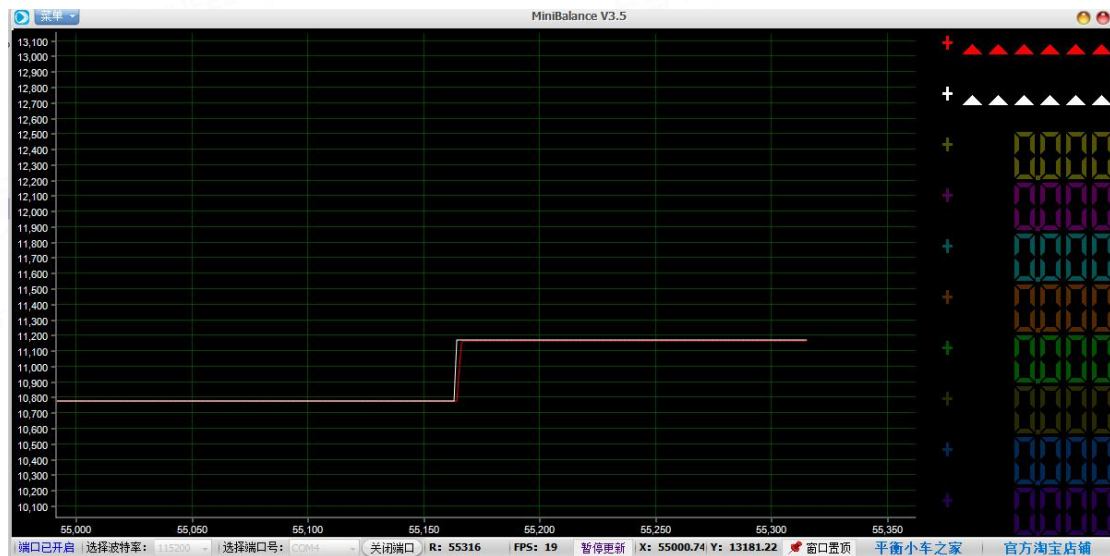


图 1-5

白线是目标值，红线是当前的值。可以用于评估 PID 参数是否合理，选择增大或减少 PID 各个参数的大小。