

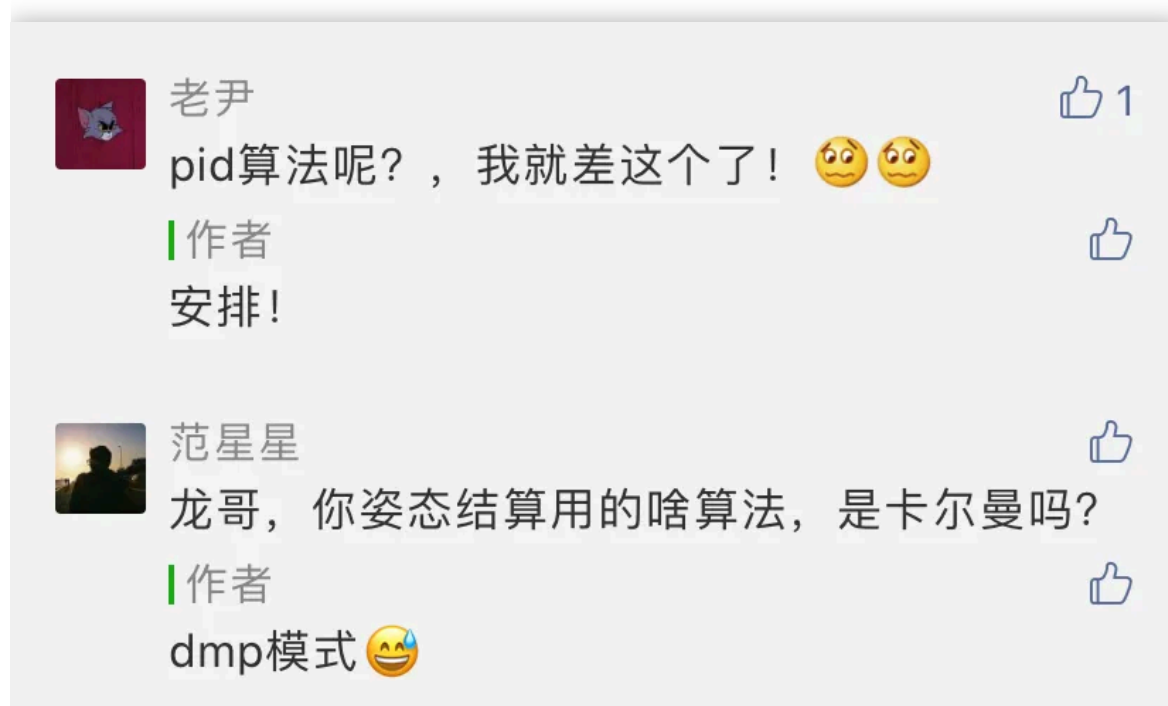
平衡小车PID，就该这么调！！

原创 张巧龙 大鱼机器人 2022-04-05 11:27

上一篇文章：[看完这篇文章，还不会做平衡小车，你来打我。](#)

描述了平衡小车的制作过程，也开源了一部分设计资料。

在上篇文章留言中，有朋友说：



安排，必须安排！

1、PID

关于PID的概念，网上相关的帖子太多，在此不再赘述。

之前也有过几篇关于PID的文章：

再论PID，PID其实很简单。。。

PID算法搞不懂？看这篇文章就够了。

这次我们聊下关于平衡小车中的PID，

在平衡小车的PID中，分为三种PID，分别是**直立环、速度环、转向环**。

任何一个PID最终计算出来的都是电机PWM，都是需要赋值给电机的。

环环嵌套，得出电机的最终控制PWM。

2、直立环PID

在直立环中，PID的入口参数为：平衡小车的姿态角和姿态角对应的角速度。

值得说明，MPU6050得出来的姿态角有三种：PITCH（俯仰角）、ROLL（翻滚角）、YAW（航向角）

一般来说，MPU6050都是平方且平行装在平衡小车上，总不会有人垂直装吧😂😂😂

如果平放且平行安装，那么直立环PID的入口参数为：Pitch或roll。

直立环中，有一个较为重要的概念，也就是**机械中值**。

通俗讲，小车在不接受任何外力或者电机作用，能够找到一个角度自我平衡。

如何理解这句话：

很简单，小车电机不转动，人的手扶着小车，小车总能找到一个角度，自我**短期**平衡。此时的角度就是**机械中值**。

直接看代码：

```
1  /*****
2  函数功能：直立PD控制
3  入口参数：角度、机械平衡角度（机械中值）、角速度
4  返回  值：直立控制PWM
5  作    者：公众号【大鱼机器人】
6  *****/
7  int balance_UP(float Angle,float Mechanical_balance,float Gyro)
8  {
9      float Bias;//角度误差
10     int balance;//直立环计算出来的电机控制pwm
11     Bias=Angle-Mechanical_balance;
12     //===求出平衡的角度中值和机械相关
13     balance=balance_UP_KP*Bias+balance_UP_KD*Gyro;
14     //===计算平衡控制的电机PWM  PD控制    kp是P系数 kd是D系数
15     return balance;
16 }
```

从程序上看：

balance_UP_KP 为直立环的P，

balance_UP_KD 为 直立环的D。

如何确定P和D的大小和极性？

2.1 直立环 P 范围确定：

需要先确定PWM的范围，例如，定时器最大的PWM为7200，此时占空比为100%，电机应该是全速运行。

如果小车需要直立，摆幅，差不多就要 $\leq 10^\circ$ 。如果超过此范围，小车抖动较为厉害

根据直立环的程序：

```
1 balance=balance_UP_KP*Bias+balance_UP_KD*Gyro;
```

由于PWM最大是**7200**，角度在 **10°** ，反推可以得到：

直立环的P可选范围应该在**0~700**。

当然这只是个大致的范围，具体多少还需要进一步调试。

2.2 直立环 P 极性确定：

极性也就是符号，P到底是给正的，也是负的。

直接给kp正负值，然后观察现象：

正常出现的现象是负反馈，小车往那边倒，电机转动使得小车往要倒的方向去追。使得小车能够往反方向站起来！

如果出现正反馈，车往哪边倒，电机转动使得小车快速倒下。这种现象就是不对的。

2.3 直立环 P 大小确定：

慢慢试错，从小到大，响应慢慢加快也就是小车倒下后恢复直立的时间越来越短，直到小车出现大幅度的**低频抖动！**

此时的P可以确定。

2.4 直立环 D 极性确定：

D的极性较为好确定，设P为0，D给正负值，分别去试，看效果。

当拿起小车进行旋转时，小车的轮子应该是小车旋转方向相同，**此时说明极性是对的。**

如果小车的轮子转动和小车的转动方向不相同，**说明此时极性是反的！**

2.5 直立环 D 大小确定：

D的大小，需要联合P去调试，在P调好的基础上，加入D，从小到大慢慢去试，从程序PD可以看到，D对应的是角速度，由于角速度都是四位数以上的数值，所以可以从0.1开始试。

一直到小车出现**高频的剧烈抖动**。

需要说明的是，如果小车各方面机械机构都分布较为均匀，重量分布较好，重心较低，小车靠单纯的直立环能够暂稳。

但一般来说，没有谁的小车机械结构做的很好。

所以说，单纯靠直立环是无法将小车站稳的。**需要再加入速度环**。

单纯的直立环能使小车站稳 **5s** 就说明调的很好了！

3、速度环

速度环中，采用PI控制，积分控制和比例控制有一定的比例关系。

这里可以确定为200，别问为什么，没有为什么。问就是200！

速度环的入口参数，为小车的2个电机编码器数值，也就是测速！

没有小车速度的实时反馈，谈何速度闭环。

看代码：

```
1  /*****
2  入口参数：电机编码器的值
3  返回值：速度控制PWM
4  作者：公众号【大鱼机器人】
5  *****/
6  int velocity(int encoder_left,int encoder_right)
7  {
8      static float Velocity,Encoder_Least,Encoder,Movement;
9      static float Encoder_Integral;
10     //=====速度PI控制器=====//
11     Encoder_Least =(Encoder_Left+Encoder_Right)-0;
12     //===获取最新速度偏差==测量速度（左右编码器之和）-目标速度（此处为零）
13     Encoder *= 0.7;          //===一阶低通滤波器
14     Encoder += Encoder_Least*0.3;    //===一阶低通滤波器
15     Encoder_Integral +=Encoder; //===积分出位移 积分时间：10ms
16     if(Encoder_Integral>10000)    Encoder_Integral=10000;
17     //===积分限幅
18     if(Encoder_Integral<-10000)    Encoder_Integral=-10000;
```

```

19    //===积分限幅
20    Velocity=Encoder*velocity_KP+Encoder_Integral*velocity_KI;
21    //===速度控制
22    if(pitch<-40||pitch>40)    Encoder_Integral=0;
23    //===电机关闭后清除积分
24    return Velocity;
25 }

```

3.1 速度环 P 范围确定：

同样的，和直立环P的大小范围确定一样，我们需要得到电机编码器的最大值和PWM的最大值的关系！

从程序中可以看到，我们应该比较的是，2个电机的速度偏差和pwm的关系。

比如：用STM32定时器的正交解码模式对电机进行测速，10ms一次。

小车电机满速旋转时，左右两个电机，编码器相加可达160。

假设速度偏差（实际测量值与理想值）达到50%时满转。

那么有， $160/2=80$ ， $7200/80=90$ ，也就说kp最大为90。

(注意，这里只是在假设50%的前提下)。

90只是一个参考值，具体多少，还是需要根据，实际测试的效果。

3.2 速度环 P 极性确定：

确定P的极性，需要关闭前文的直立环，也就是说整个系统的控制参数只能有速度环的P。

单单靠直立环控制小车，小车能短暂直立，但会出现往前走或往后走，然后倒下，那么速度环就是用来抑制此现象的出现。

从上文程序中可以看到：

```

1    Encoder_Least =(Encoder_Left+Encoder_Right)-0;
2    //===获取最新速度偏差==测量速度（左右编码器之和）-目标速度（此处为零）

```

这句程序的意思就是，获取最新速度偏差，控制小车目标速度为0。

直立环中控制小车不倒下是用来控制小车的角度，所以直立环的机械中值是：**角度**

速度环控制小车不倒下是用来控制小车的速度，所以速度环的“中值”就是：**速度为0**

应该不难理解！

那么如何抑制小车速度为0呢？

既然我们可以知道小车的当前速度，**只要速度环的P为正反馈即可，意思就是假如向前倒，那么小车就要以更快的速度向前冲，保持直立。**

同样的，屏蔽前文的直立环，分别给速度环P正负值，看现象。

正反馈的现象为：

当旋转其中一个轮子，两个轮子往相同方向旋转，到速度最大值。此时应该为正反馈。此时的现象说明，**速度环的P极性是对的！**

如果出现旋转其中一个轮子，另外一个轮子往反方向转动，让偏差趋向于零。这就是负反馈，**此时说明P极性错误！**

3.3 速度环 P 大小确定：

确定P极性和大小之后，由于P和I有比例关系且P为I的200倍！P和I的大小可以一同调试，可以将P和I慢慢从小到大的参数去试，观看小车效果。

如果出现以下效果：

1、小车放在地上，慢慢的，随着时间越来越长，小车会来回晃动，此时可以认为P和I的参数过小。

2、小车放在地上，用手去推，如果小车无法回到初始位置，一直来回晃动，来回晃动的时候，车身出现较为大的倾斜，此时可以认为P和I的参数过大。如果车身没有出现较大的倾斜，只是小车来回晃动，此时可以认为P和I的参数过小。

关于速度环的初步调试，大致就讲到这里，这种试错的方法是较为愚钝的，但却是较为方便且简单的一种方法。

可以帮助大家很快速的调试站立好小车。

4、转向环

关于转向环，其实没什么好讲的，很多种转弯方式，P控制，固定电机pwm差速控制，唯一牵扯到的就是转向环可以矫正小车走直线的问题，这个可以单独拉出来进行聊聊。在这里不再聊了。

最后，我给大家准备了一个简易版的小车直立程序，只有直立环和速度环。

关注公众号【**大鱼机器人**】后台回复关键词【**平衡小车**】即可获取！

资料包中也有参考电路，就是封面图的掌上mini小车！比可乐还要小~

dyuRobot_Bliz平衡车程序\dyuRobot_Bliz平衡车代码（简易版）				
名称	修改日期	类型	大小	
CORE	2017/10/2 18:37	文件夹		
HARDWARE	2017/10/2 18:37	文件夹		
OBJ	2020/3/27 15:44	文件夹		
STM32F10x_FWLib	2017/10/2 18:37	文件夹		
SYSTEM	2017/10/2 18:37	文件夹		
USER	2020/3/27 15:44	文件夹		
清除临时文件	2011/5/9 19:17	Windows 批处理...	1 KB	

1, Bliz平衡小车原理图	2017/9/7 11:34	文件夹	
2, dyuRobot_Bliz平衡车程序	2017/10/2 18:53	文件夹	
3, OLED显示屏资料	2017/9/30 17:59	文件夹	
4, MPU6050资料	2017/9/30 18:21	文件夹	
5, 电机相关资料	2017/9/30 18:10	文件夹	
6, 蓝牙资料	2017/9/30 18:10	文件夹	
7, MDK5 (开发环境)	2017/10/10 13:16	文件夹	
8, PID参考资料	2017/9/7 13:29	文件夹	
9, STM32参考资料	2017/9/12 13:29	文件夹	
10, 软件与驱动	2017/9/13 0:29	文件夹	
11, 电脑上位机以及手机APP	2017/9/7 13:39	文件夹	
12, 快速DIY平衡车教程 (待更, 请关注...	2017/10/10 1:00	文件夹	
PDF Bliz平衡车上电测试教程	2017/9/27 23:25	看图王 PDF 文件	2,475 KB
V2.0版本资料更新说明	2017/10/10 13:15	文本文档	1 KB

-END-

往期推荐：点击图片即可跳转阅读



这篇文章后，不要再问怎么做一台智能车了



看完这篇文章，还不知道怎么学单片机，来打我！



看完这篇文章，还不会做平衡小车，你来打我。

最后

若觉得文章不错，**转发分享**，也是我们继续更新的动力。

5T资源大放送！包括但不限于：C/C++，Linux，Python，Java，PHP，人工智能，PCB、FPGA、DSP、labview、单片机、等等！

在公众号内回复「**更多资源**」，即可免费获取，期待你的关注~

一个试着将硬件与软件相结合的公众号

给作品以身形

再用代码赋其灵魂

软硬结合

无所不能



长按识别图中

二维码关注

喜欢此内容的人还喜欢

嵌入式硬件工程师的崩溃日常~~~

大鱼机器人



看完震撼了！一位复旦毕业电子工程师的8年履历！

大鱼机器人



商汤牛逼！彻底爆了。。。

大鱼机器人

