

PID智能小车 直流电机调速



主讲：于涛

- PID是什么
- 绕不开的理论公式
- 实际如何下手
- 演示
- 更进一步的深入

生活中的例子

类比

例如一个底部有漏水的水池，如果我们要保持水位，在某一个刻度，需要通过水阀开度的大小来实现。

“高于刻度阀门就开小,低于刻度阀门就开大”

这本身就是PID里面的一部分：
P，即比例控制--只使用了PID的P部分。

题外话：使用开环也可以，pwm与转速对应表。

PID是什么?

P --比例
I --积分
D --微分

为什么使用PID

一般的开关控制只能简单的起停。
即控制电机的转动与停止。

使用PID的效果：实现更精细的控制。

- 1.保持转速恒定。
- 2.快速达到指定的转速。

常用单位：转/分钟 = RPM。

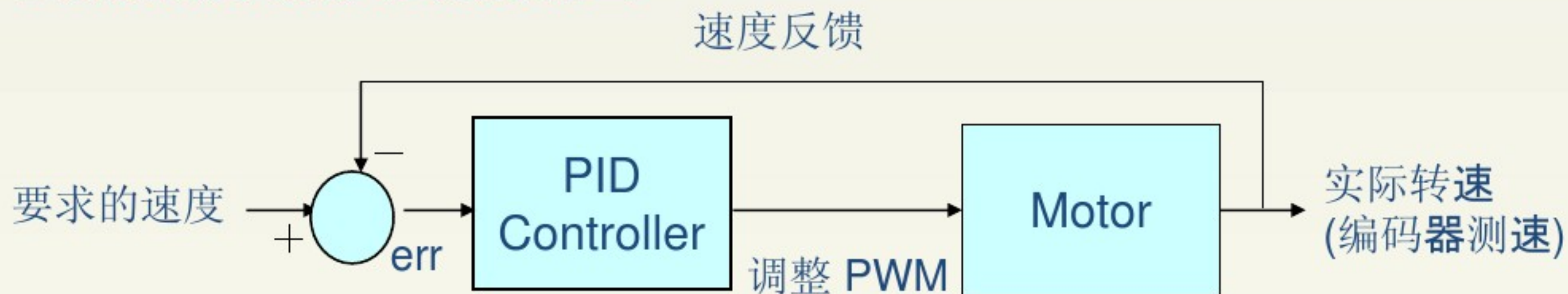
为什么我们不能告诉电机，按照我们要的速度转动？

电机的交互方式：PWM, 参数整定

PID是什么?

PID的前提是反馈

必须有反馈才能知道如何调整



闭环：带有反馈的系统

开环：没有反馈的系统

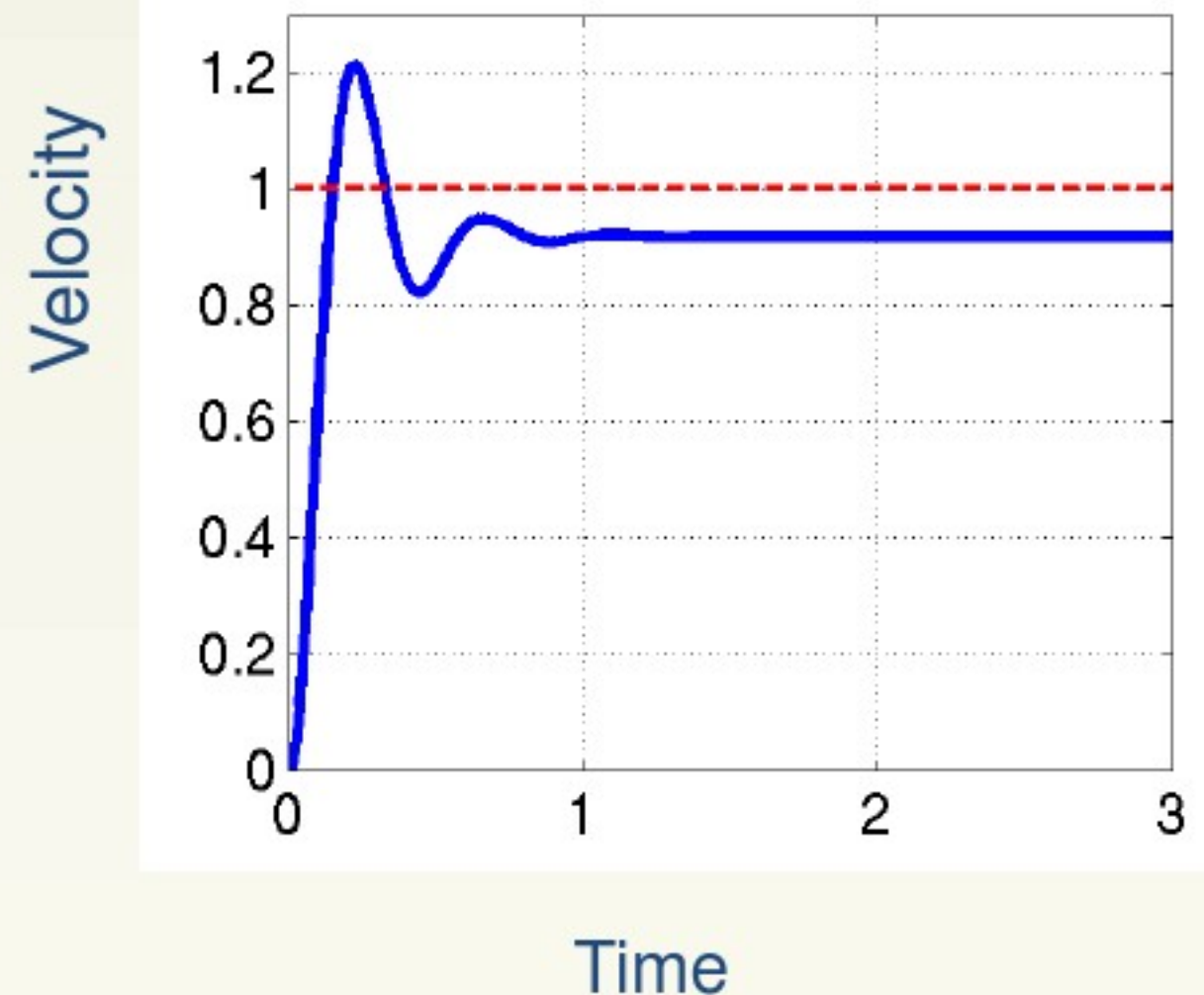


PID是什么?

P--比例控制部分

这个很好理解.比如说,速度传感器发现,当前速度是1200/分.而我们设定的速度值为1000 /分,那么差就是200 /分,这时,如果我设定P为0.1,而电机速度的控制量是“电压”,就应该是 $U_0 + 0.1 \times 200$ 。P的意思就是“倍数”,就是指你要把这个偏差放大多少倍.“放大”本身就是一个比例.单单一个P控制就足够可以控制电机的速度.

P 单独控制效果：
有误差，但是保持不变

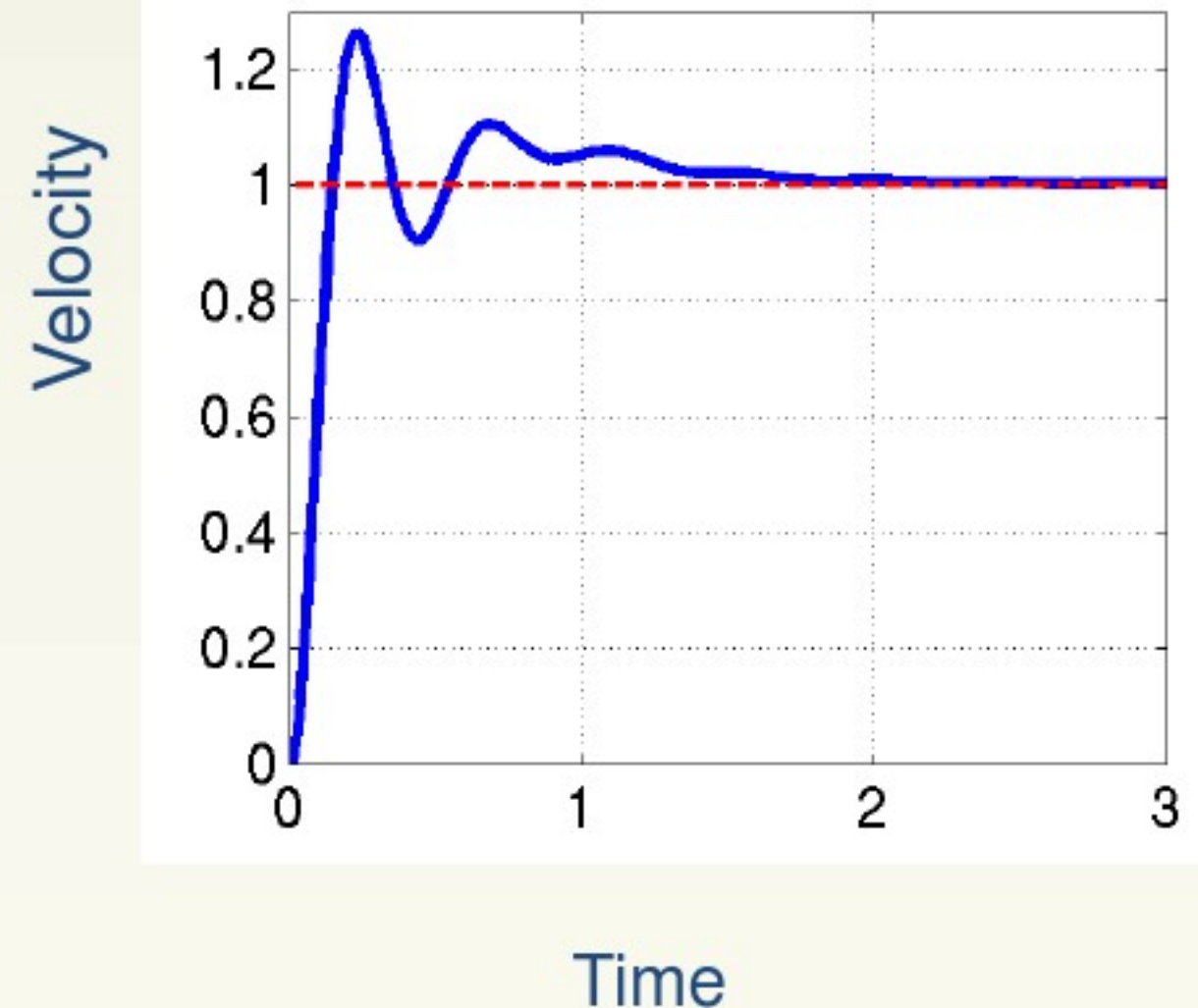


PID是什么?

I--积分控制部分

它是一个积分运算.有的时候偏差不是很大,可以继续运行下去,但会使系统始终存在一个偏差.如果你使用I运算将这个偏差累加起来,到了一定大小的时候就进行处理.这样就能防止系统的误差累计.其实,在程序中,这个过程就是对一个小偏差的连续累加.

PI 组合
消除了误差
带来了超调



PID是什么?

D--微分控制部分

所谓的微分就是对变量求导,意思就是一个量的变化率.其微分部分就是能够将变量变化率放入计算中.这个量在编程中其实就是求上次偏差和这次偏差的差.

PID 黄金组合

加快进入稳态
减少超调



绕不开的理论公式

误差：

$e(t)$ = 给定-测速结果

P控制器：

$$u(t) = K_p e(t)$$

PI控制器：

$$u(t) = K_p \left[e(t) + \frac{1}{T_I} \int_0^t e(t) dt \right]$$

离散化 $\cdot t \approx kT (k = 0, 1, 2, \dots)$

$$\int e(t) dt \approx \sum_{j=0}^k e(jT)T = T \sum_{j=0}^k e(jT)$$

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_I} \sum_{j=0}^k e(j) \right\}$$

绕不开的理论公式

PID控制器

本次的误差 $e(kT)$,上次的误差 $e[(k-1)T]$,2者的差值除以采样时间就是变化率

$$u(t) = K_p \left[e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right]$$

离散化 , $t \approx kT (k = 0, 1, 2, \dots)$ $e(t) \approx e(kT)$

$$\int e(t) dt \approx \sum_{j=0}^k e(jT) T = T \sum_{j=0}^k e(jT) \quad \frac{de(t)}{dt} \approx \frac{e(kT) - e[(k-1)T]}{T}$$

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_I} \sum_{j=0}^k e(j) + \frac{T_D}{T} [e(k) - e(k-1)] \right\}$$

$$= K_p e(k) + K_I \sum_{j=0}^k e(j) + K_D [e(k) - e(k-1)]$$

绕不开的理论公式

上面介绍的公式，叫做**位置PID**，也就是每次都使用整个系统的状态。
可以理解为：偏差使用的是所有的全量偏差。
还有一种算法是**增量式PID**。

摘录网上的话：“位置式PID算法每次输出与整个过去状态有关，计算式中要用到过去误差的累加值，因此，容易产生较大的累积计算误差。而增量式PID只需计算增量，计算误差或精度不足时对控制量的计算影响较小。”

上一次的输出为 $u(k-1)$ ，这次的输出为 $u(k)$ ，变化量为 $\Delta u(k)$

$$u(k-1) = K_p e(k-1) + K_I \sum_{j=0}^{k-1} e(j) + K_D [e(k-1) - e(k-2)]$$

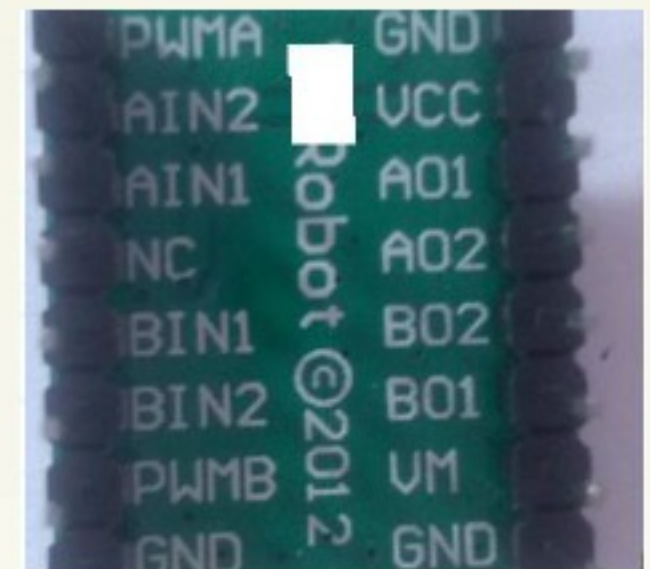
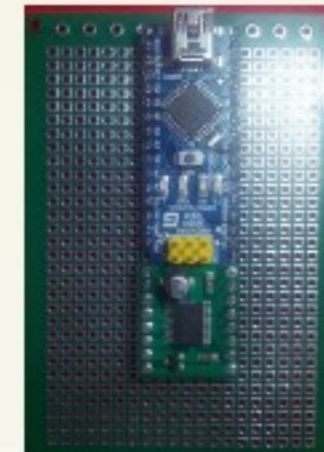
$$\Delta u(k) = u(k) - u(k-1)$$

$$= K_p [e(k) - e(k-1)] + K_I e(k) + K_D [e(k) - 2e(k-1) + e(k-2)]$$

实际如何下手

硬件准备：

- 1.N20直流电机，配有1：145的金属变速箱。电机必须带测速。
- 2.Aduino nano控制板，主控芯片atmel328p。
- 3.实现电机控制的H桥电路板，带pwm输入。
- 4.供电电池。



软件准备：

- 1.Arduino IDE开发环境。
- 2.串口猎人，用于收集数据，实时绘制可视化图像。

实际如何下手

要控制就要涉及到三个量：
 设定值（控制目的）、被调量
 对应到直流电机上就是：
 设定值：希望（给定）电机
 被调量：PWM占空比
 输出量：实际电机输出的转速

这个就是我们的
控制目标，一般
通过上位机命令
给出

编码器体现速度的数字是
 单位时间内的电平改变次数（ticks）。
 我设置的是引脚变化触发计数，有改变就计数。因此，速度 = ticks。
 对应的量就变为：

设定值：希望的单位时间内ticks
 被调量：PWM占空比
 输出量：实际单位时间内ticks

设定速度

PWM占空比

实际速度

实际如何下手 代码实现

速度测定：

程序可以通过定时器来计数，利用单片机上的引脚变化中断实现。

问题是这个采样时间怎么取？

这里针对我买的电机，测速是放在转轴上未减速前的，采样周期可以设短为30ms。电机每圈有12个ticks，就设为15ms测速一次，每测2次求平均值，作为当前速度值。有了采样周期，就可以得到：输出量。

这样就完全具备了条件：

设定值：给定转速ticks,命令给出，看作常数。

被调量：PWM占空比，通过带入公式计算得到。(PWM周期)

输出量：通过周期内的编码器输出得到。

同时以30ms为PID计算周期，计算结果输出给电机。

位置式PID代码实现：

```
double error = mySetpoint - inputMeter;
errSUM += error;
outputPWM = kp*error + ki*errSUM + kd*(error - errorLast);
//arduino 的调节范围就是0 - 255
if(outputPWM > 255) outputPWM = 255;
if(outputPWM < 0) outputPWM = 0;
```

这个就是全部代码仅仅5行就实现了！

实际如何下手 代码实现

增量式PID：

```
#define PVAL (error - last1Error)
#define IVAL ((error+last1Error)/2)
#define DVAL (error - 2*last1Error + last2Error)
/*Compute all the working error variables*/
double input = *myInput;
double error = *mySetpoint - input;
/*Compute PID Output*/
double output = *myOutput + kp * PVAL + ki * IVAL + kd * DVAL ;
if(output > outMax) output = outMax;
else if(output < outMin) output = outMin;
*myOutput = output;
/*Remember some variables for next time*/
last2Error = last1Error;
last1Error = error;
```


参数整定

参数整定就是通过实验得到合适的 3 个数值: K_p, K_i, K_d 。

调整PID三个参数是不断试验的烦杂过程，需要有图形化界面直观的了解，我采用的是试凑法：

(1) 确定比例系数 K_p

确定比例系数 K_p 时，首先去掉PID的积分项和微分项，可以令 $T_i=0$ 、 $T_d=0$ ，使之成为纯比例调节。输入设定为系统允许输出最大值的60%~70%，比例系数 K_p 由0开始逐渐增大，直至系统出现振荡；再反过来，从此时的比例系数 K_p 逐渐减小，直至系统振荡消失。记录此时的比例系数 K_p ，设定PID的比例系数 K_p 为当前值的60%~70%。

(2) 确定积分时间常数 T_i

比例系数 K_p 确定之后，设定一个较大的积分时间常数 T_i ，然后逐渐减小 T_i ，直至系统出现振荡，然后再反过来，逐渐增大 T_i ，直至系统振荡消失。记录此时的 T_i ，设定PID的积分时间常数 T_i 为当前值的150%~180%。

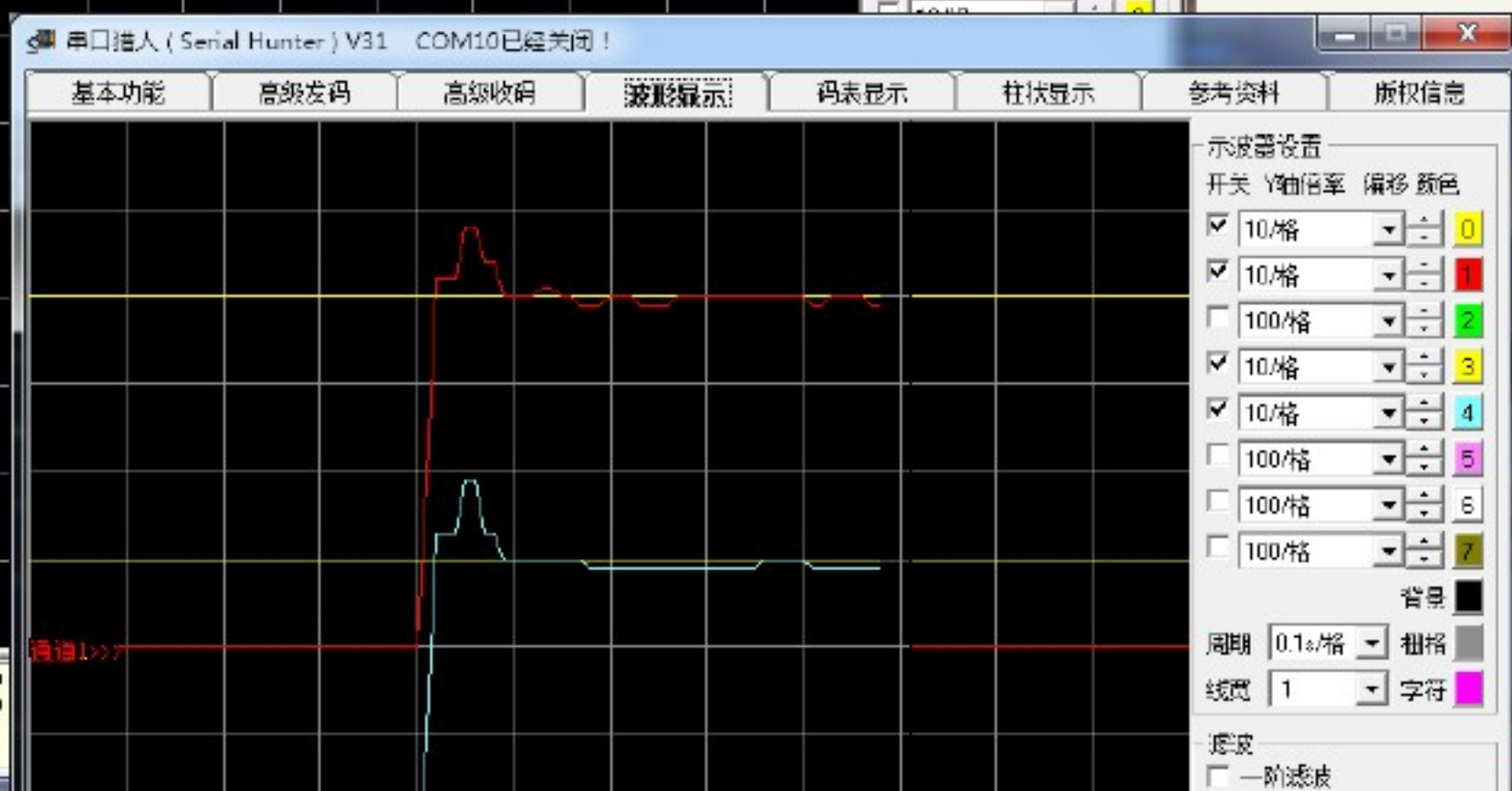
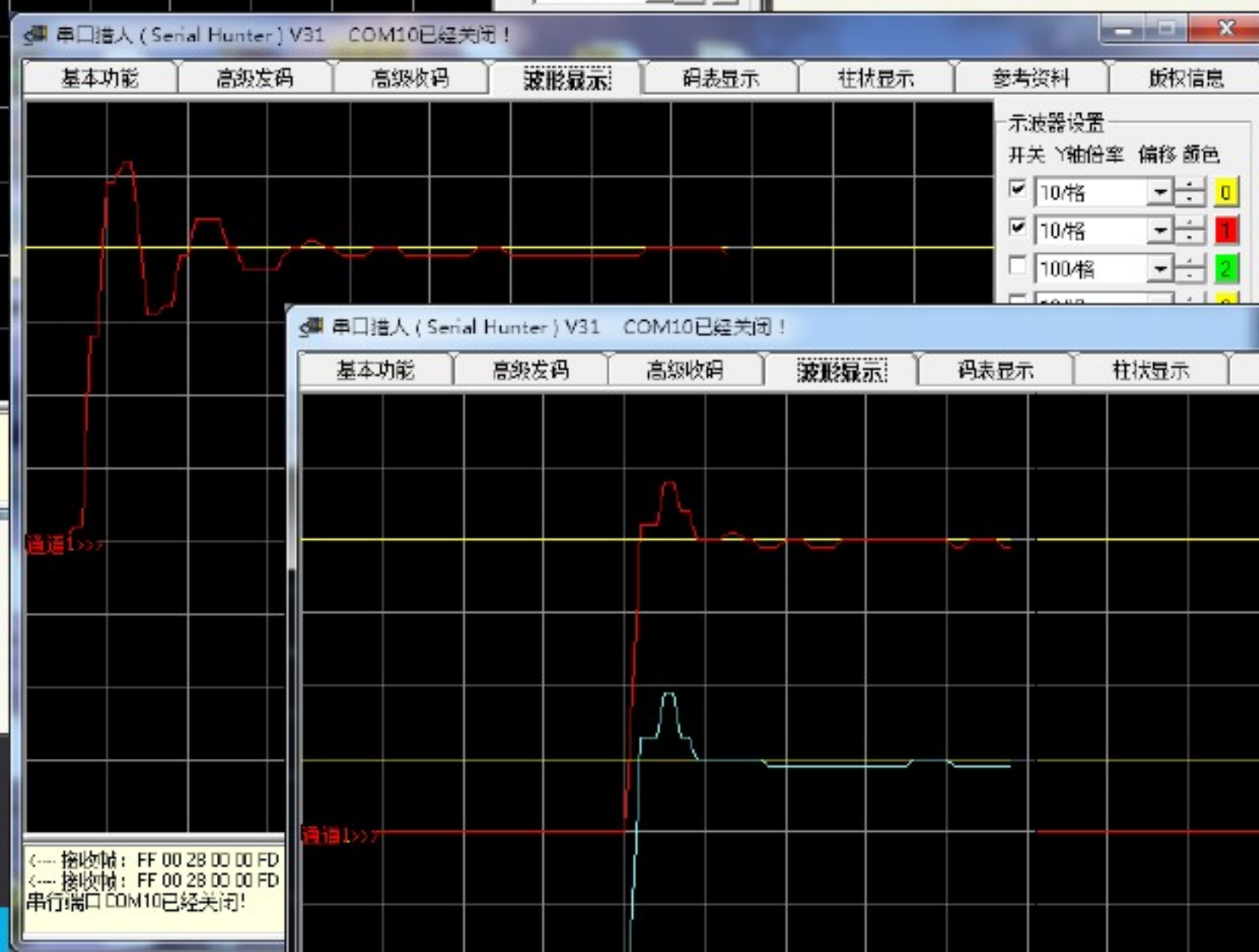
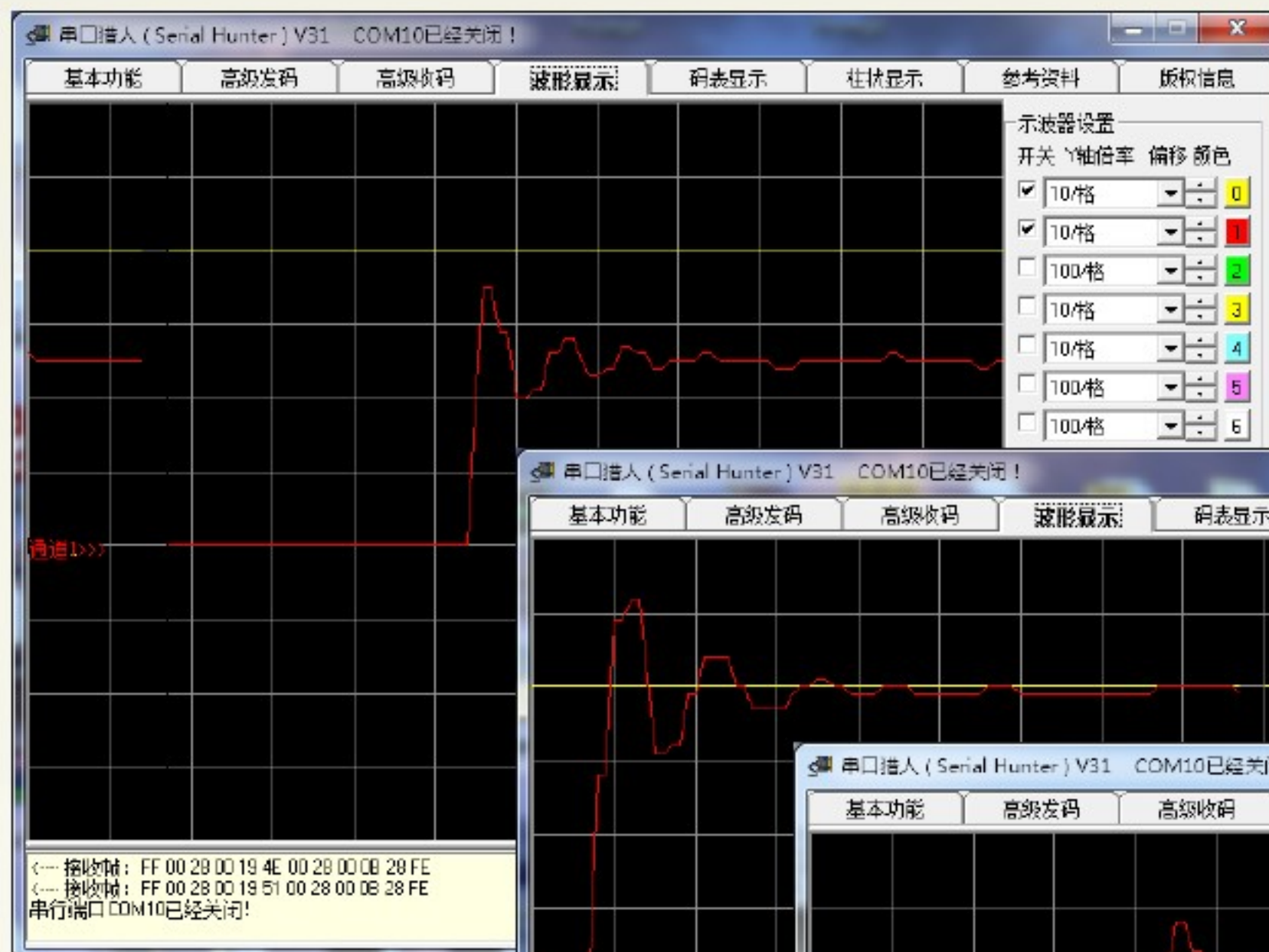
(3) 确定微分时间常数 T_d

微分时间常数 T_d 一般不用设定，为0即可，此时PID调节转换为PI调节。如果需要设定，则与确定 K_p 的方法相同，取不振荡时其值的30%。

(4) 系统空载、带载联调对PID参数进行微调，直到满足性能要求。

演示

www.do-idea.org



更进一步深入

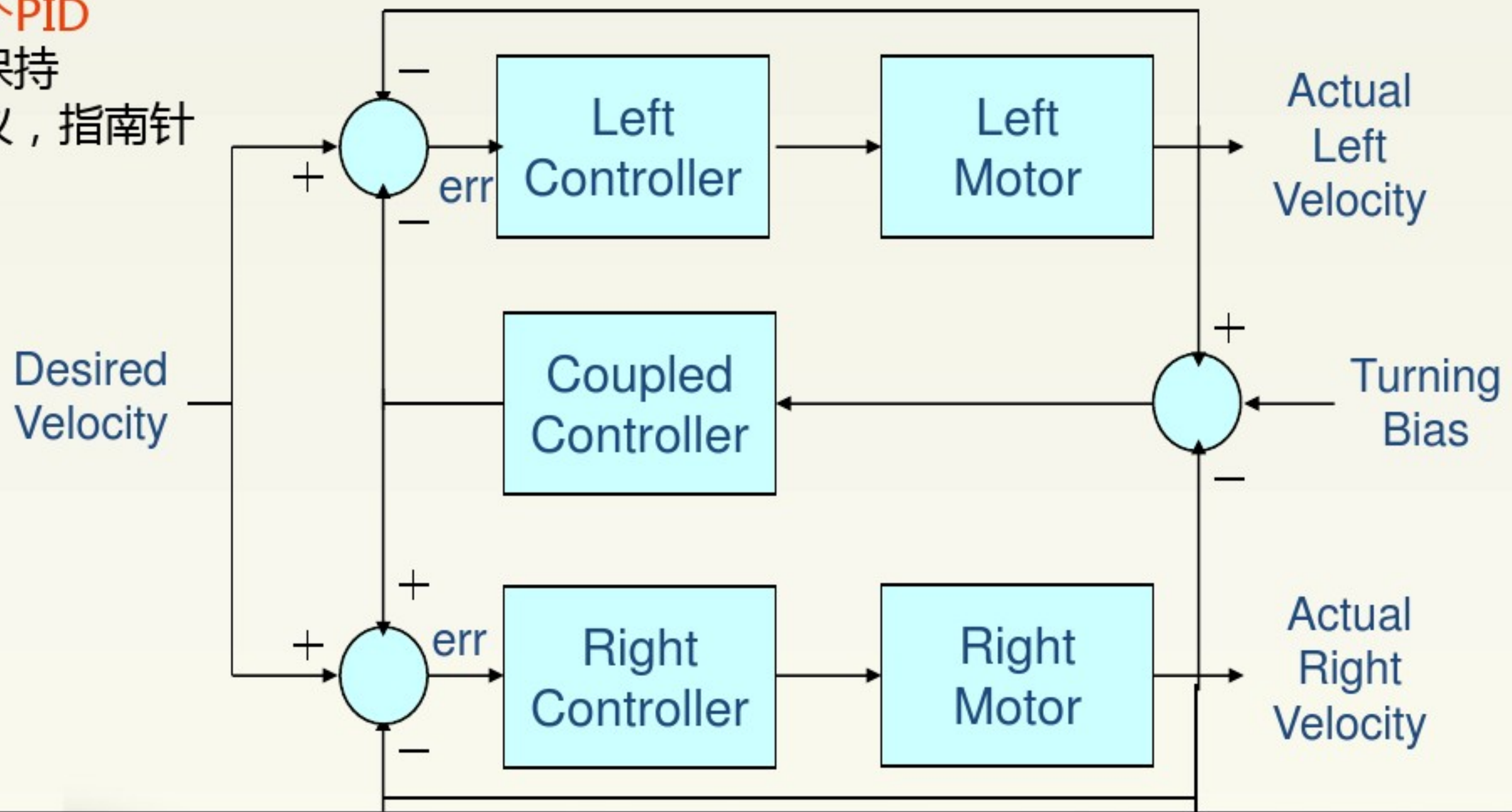
小车走直线

烦恼：2个电机速度一样也无法直线

引入第三个PID

角度姿态保持

需要陀螺仪，指南针



更进一步深入

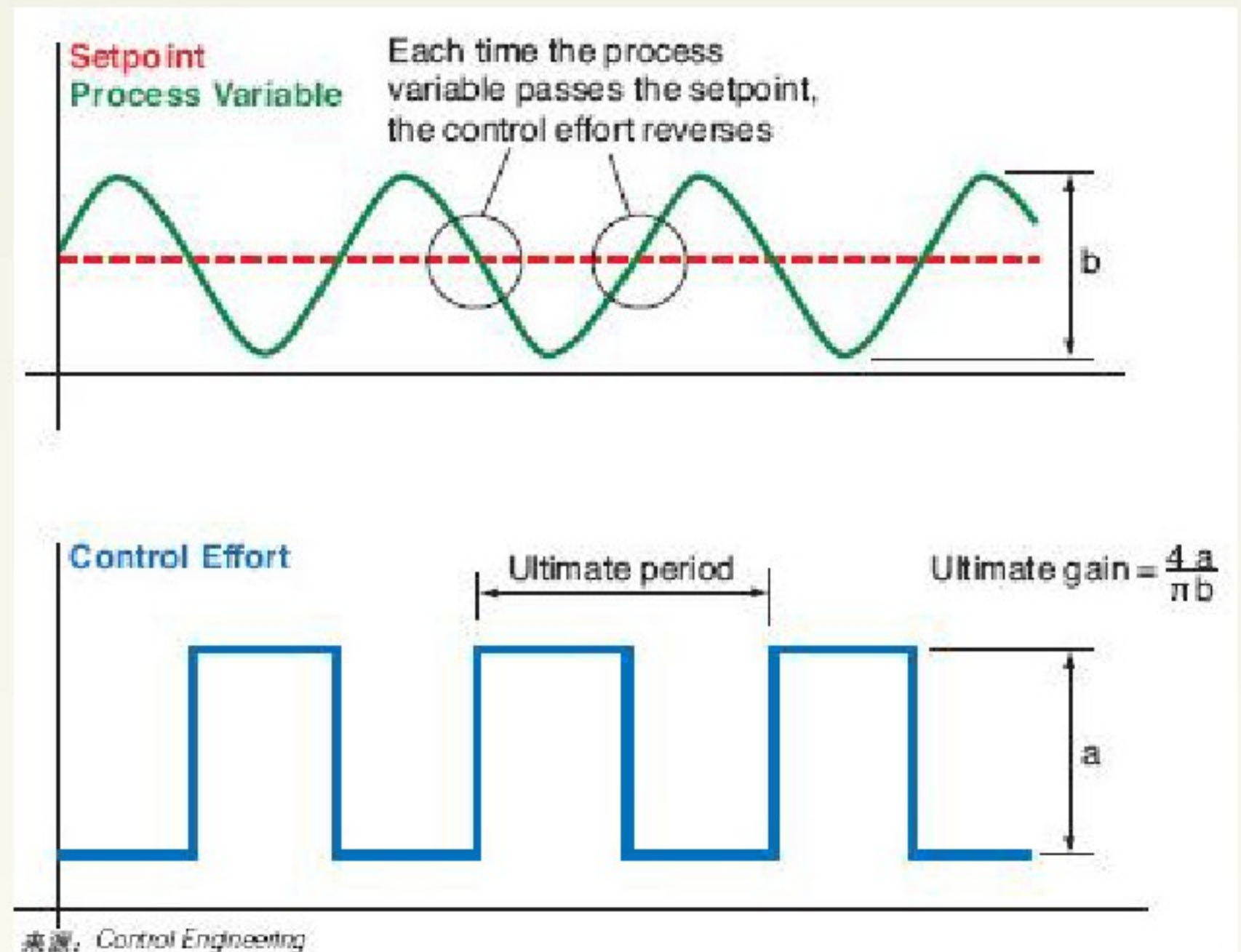
- 不完全微分
- 积分分离
- 微分先行

- 高级主题

自整定

临界比例法

临界周期 T_u 和临界增益 P_u



THANK YOU

[Contact Us](#)