# 闭环和PID的概念

<https://www.bilibili.com/video/BV17x411d7XR>

在各种电子器件的实际使用中经常会碰到器件的实际输出与输入的期望值不一致，甚至相差

很远的情况。以电机为例，假设我们输入电机的转速期望为 100rpm，当电机为空载时它的 输出转速也恰好是 100rpm，但是当我们给电机加上轮子或者其他负载之后，它的转速就无 法达到 100rpm 的期望值了，而是降低到 80rpm。假如电机转动了一段时间后电池电压开始下降了，电机的转速又会进一步的降低。

例子中的电机就是一个典型的无控制系统。在无控制系统中，输入期望和实际输出之间没有

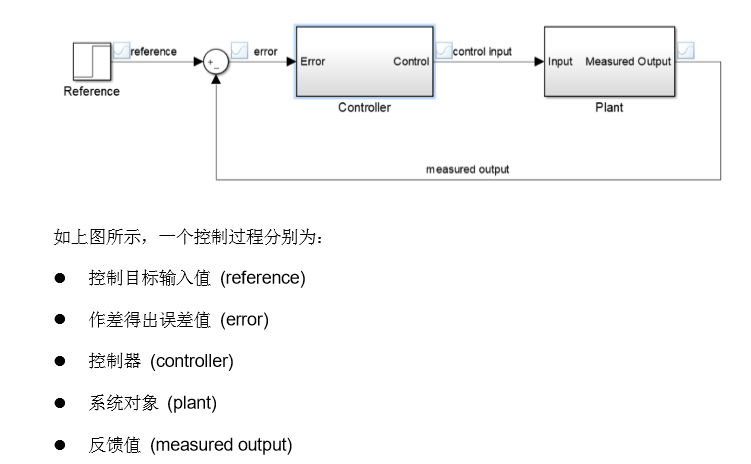
关联，两者之间可能会存在巨大的误差，从而导致系统可靠性很差。而控制系统则会根据实

际的输出对输入进行调整，从小缩小误差。

实现控制系统的一个常用方法是在无控制系统中加入 PID 控制器，PID 控制器能够根据实 际输出值和期望值之间的误差调整输入值，使最终的输出值靠近期望值。

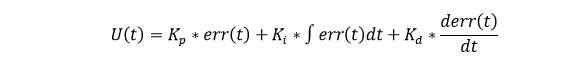
依然以电机为例，一个电机在挂了 10 斤负载的情况下，如果给电机的输入电压刚好是 24V， 则电机的实际输出只有 80rpm，输入和实际输出的误差为 20rpm，将这个差值 20rpm 输入PID 控制器， PID 控制器就会将输入量提高到 26V，这时电机的实际输出就能够达到 100rpm 了。如果差值变大到 40rpm，PID 控制器就将给电机的电压输入量继续提高。总之 PID 控制器会根据误差的大小自动完成对电机电压输入值的调节，从而保持输出值和期望值一致。

PID 是一种常用的控制算法，其基本思想是利用期望值和实际值的误差作为控制量决定最终 的输出。



而 PID 控制属于控制器的一种，由 P（比例），I（积分），D（微分）三项构成。

PID 输出值 U(t)的表达式如下，其中 err(t)即误差值，Kp，Ki，Kd 分别为比例，积分，微分 三项的系数。



比例项 Kp：控制器比例项输出值和误差值保持线性关系，误差值放大一倍则输出值也同样

放大一倍，误差值缩小一倍则输出值也同样缩小一倍。只依靠比例项进行控制的方法称为比

例控制，比例控制可以很简单的实现控制器的基本功能， 但往往存在静差以及过大引起系 统振荡的问题。

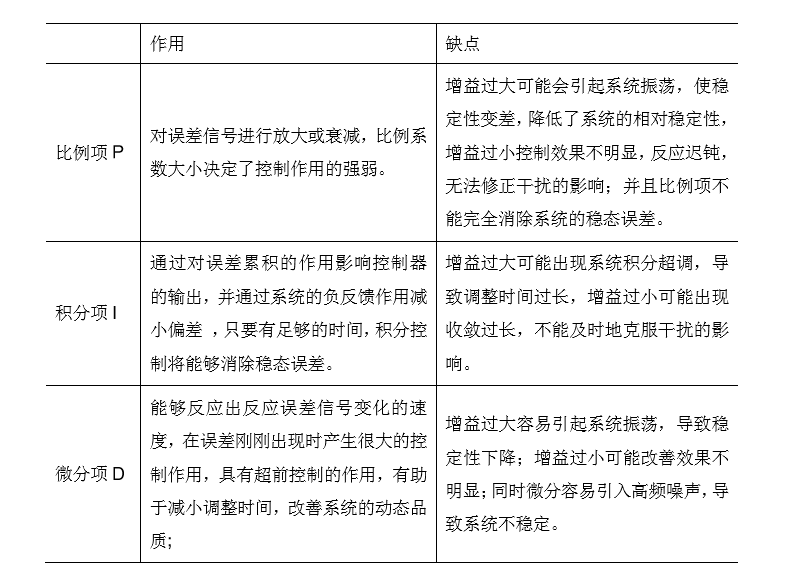
积分项 Ki：控制器积分项输出值与误差值的积分值成线性关系，即误差值的累计值乘以一

个常数。积分项科研加速系统趋近设定值的过程，但积分增益过大容易引起积分超调的现象。

微分项 Kd：微分项的大小和输出值的变化量成正相关，微分项计算误差的一阶导数，并和

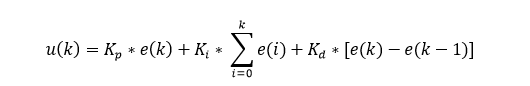
一个常数相乘，得到微分项的输出值。微分项可以对系统的改变做出反应，对系统的短期改

变很有帮助。



由于数字系统是离散的，在单片机中实现 PID 控制算法时，需要将 PID 控制器的输出表达

式改写成离散形式，其具体的做法就是将输出 u(t)和误差 e(t)由函数改成数组 u(k)和 e(k)， 积分换成求和，微分换成差分。离散化后的 PID 表达式如下：



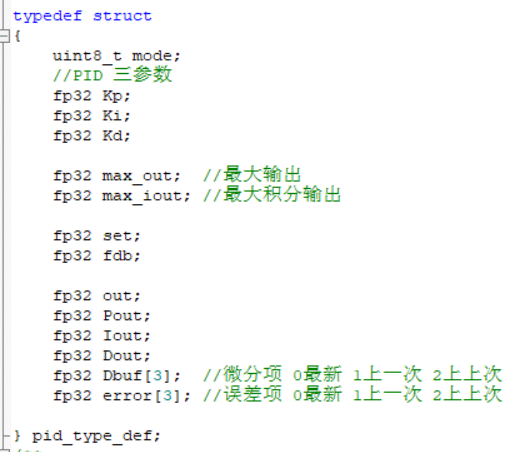
PID 控制器可以分为增量式 PID 控制器和位置式 PID 控制器，上文介绍的都是位置式 PID 控制器，即误差值直接决定最后的输出，而增量式 PID 控制器则用误差值来控制每次输出

的改变量Δu

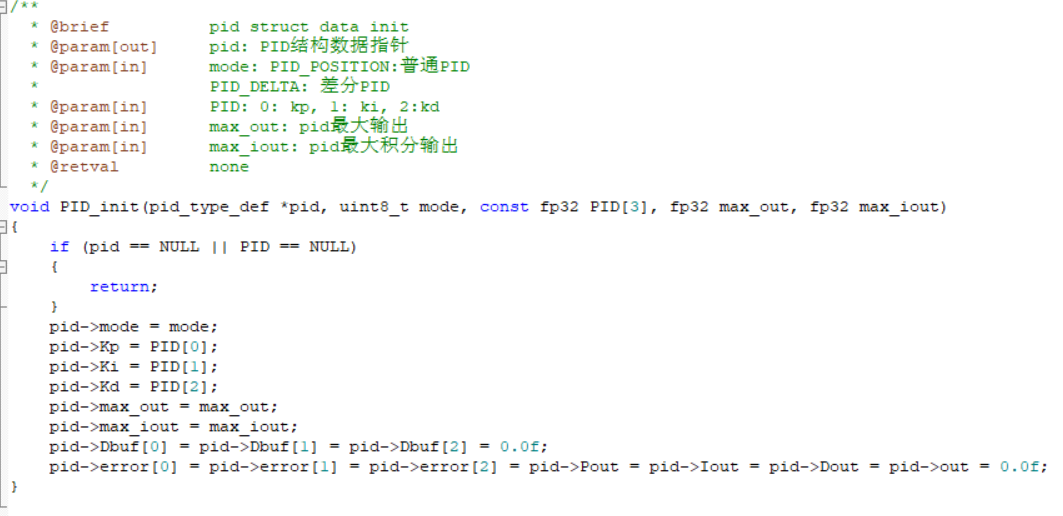


使用增量式 PID 时需要记忆上一次的输出值，将上一次的输出值和增量相加才能得到本次输出值。

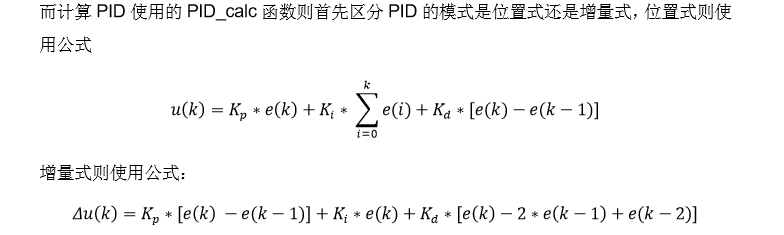
# PID 初始化以及计算函数



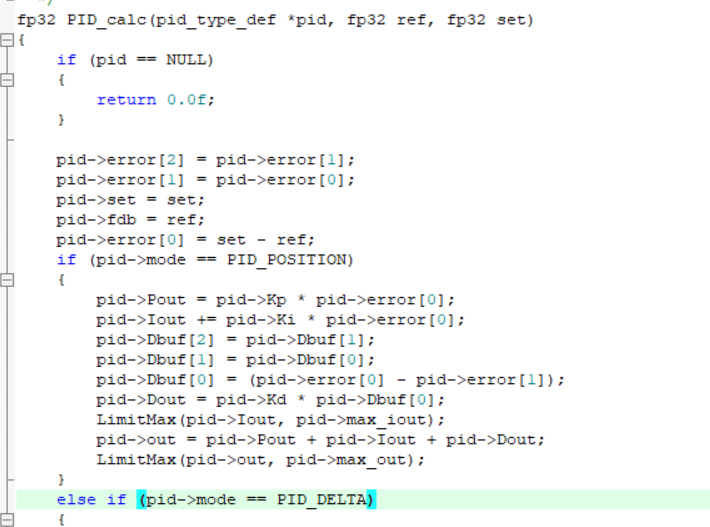
PID结构体定义



PID初始化函数，PID\_Init 函数的功能为设置 PID 控制器的模式，各项系数，最大输出值和积分上限，最后将 PID 控制器的输入和输出均初始化为 0。

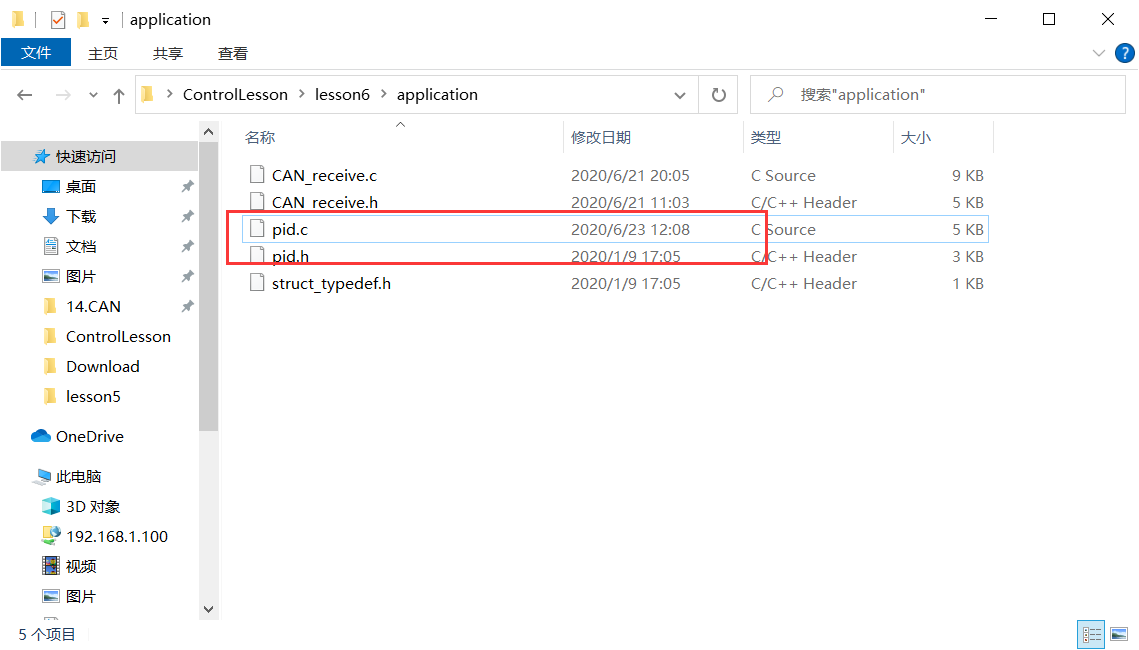


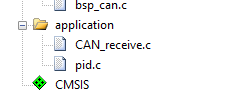
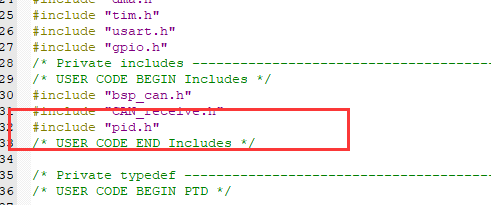
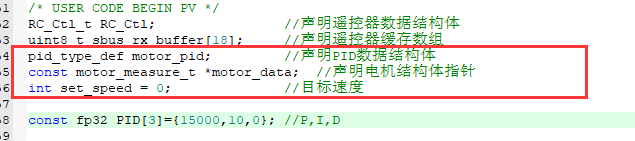
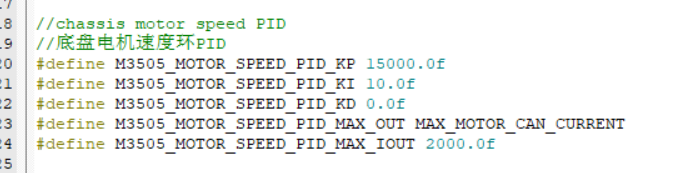
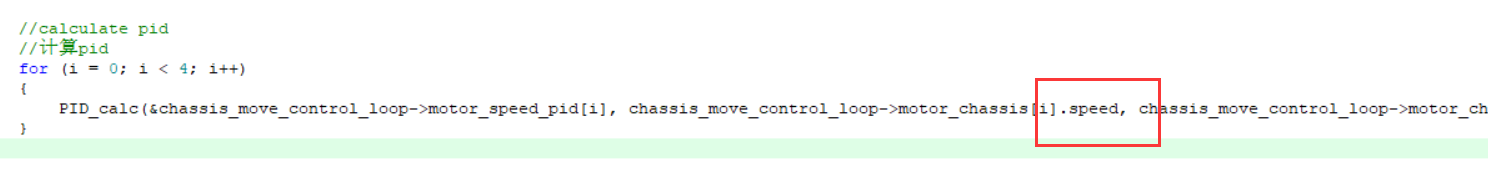
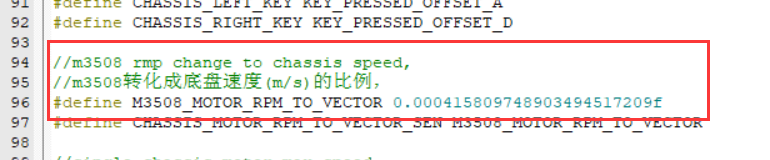
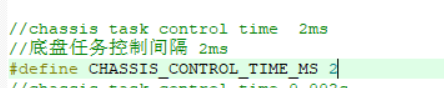
PID计算函数，可以更具位置式和增量式使用不同的计算公式



# 实验：使用遥控器闭环控制电机转速

1. 复制上一节课控制电机的工程到新文件夹中
2. 复制PID的库文件到新工程中



1. 加入源文件
2. 包含头文件
3. 声明PID数据结构体，电机数据指针（因为电机数据的变量在can接收的代码里已经声明）和目标速度变量
4. 
5. 查看大疆例程给电机的PID参数
   1. 
   2. 注意，大疆这里的PID计算时是带入的是整车的速度，不是轮子的转速，所以不能直接将15000等带入！
   3. 
   4. 
6. 声明包含PID三个参数的数组
   1. 
7. 初始化PID
   1. PID\_init(&motor\_pid,PID\_POSITION,PID,16000,2000); //PID结构体，PID计算模式，PID参数，最大值，最大I值
8. 获得对应电机的数据指针
   1. motor\_data = get\_chassis\_motor\_measure\_point(2); //获取ID为3号的电机数据指针
9. 在while循环中将遥控器的值映射到目标的速度rpm
   1. set\_speed=map(RC\_Ctl.rc.ch0,364,1684,-500,500);
10. 使用PID公式，将PID结构体，时间电机转速和目标转速带入
    1. PID\_calc(&motor\_pid,motor\_data->speed\_rpm,set\_speed); //PID结构体，实际速度，设定速度
    2. 这个函数最后会将计算出的电流赋值到motor\_pid.out上
11. 最后将计算出的电流发送至CAN总线
12. 注意最后的延时时间要和大疆的例程相同，因为不同的循环时间，对应的最优PID参数也是不同的

# PID的调参

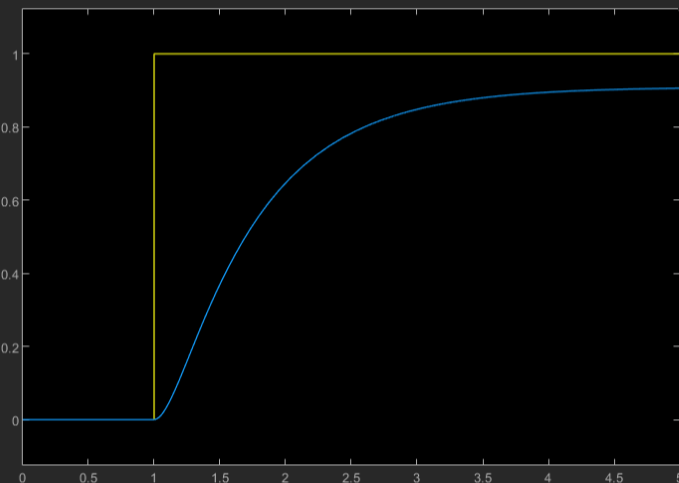
比例项 Kp 过小时，PID 控制器的反应速度较慢且存在静差。静差是指控制器的最终输出保

持为一个和期望值存在一定误差的值，引发静差的原因时由于比例控制的输出和误差成线性

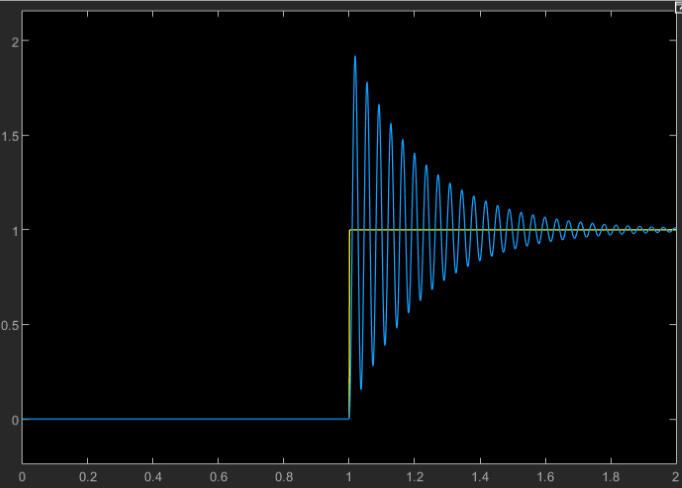
关系，如果当误差值减小时，比例控制器的输出值同样会减少，导致比例控制器不可能达到

和期望值完全相同，即误差值为零的情况，因为如果误差值为零则比例控制器的输出也会为

零。 下图为 PID 控制器比例项过小的仿真，蓝色曲线为系统的输出，黄色曲线为设定的输入，可 以看到最终输出值和期望值之间保持一个恒定的误差值，这个误差值就是静差。

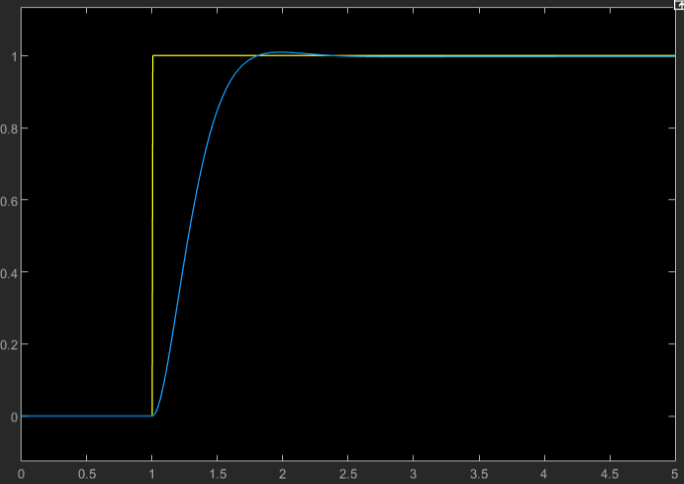


通过增大比例系数 Kp 可以减小静差，提高反应速度。但如果 Kp 过大则会导致输出严重震 荡，如下图所示。震荡会使系统的稳定性大大下降。



为了克服静差，可以引入积分项，由于积分的累积性，积分项的输出值由之前所有时刻的误

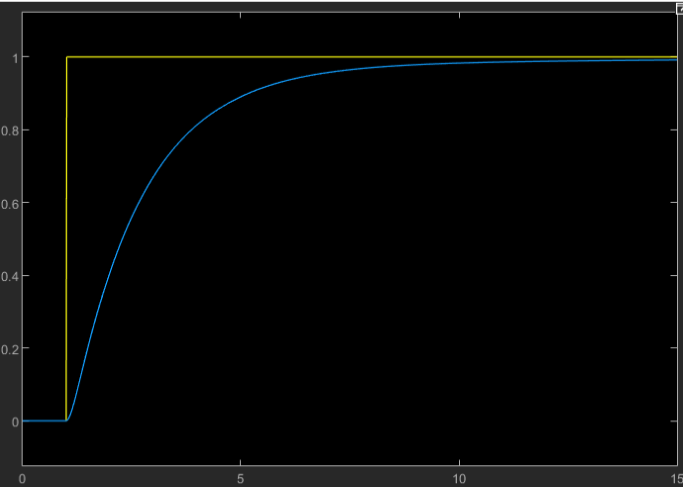
差值决定。因此即使在误差值为零的情况下，积分项也会保持作用，使控制器的输出量不为

零，引入积分项之后控制器的输出值才有可以真正等于期望值。

但是引入积分控制器后会延长系统的调节时间，比例-积分控制器的输出值需要相较比例控制器更长的时间趋于稳定，导致系统的响应速度变慢。

如果积分项过小，或者积分上限限幅过小，则输出收敛的时间会非常长，这就导致系统的响

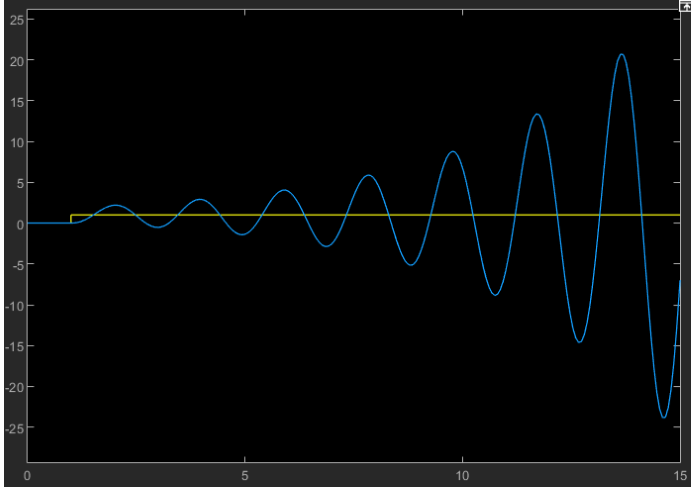
应速度非常慢，如下图所示。



而如果积分项过大，或者积分上限过大，则会带来严重的超调。超调即系统输出的峰值远远

高于期望值，超调会导致系统需要很长时间才能够趋于稳定，甚至可能无法趋于稳定，会严

重影响系统的稳定性，降低系统响应速度，如下图所示。

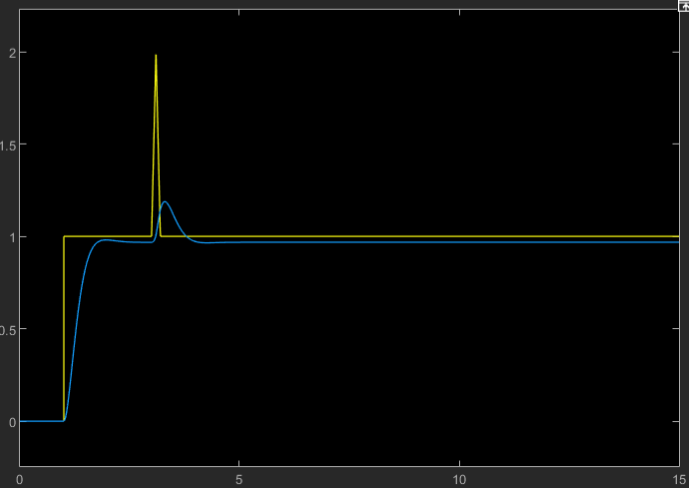


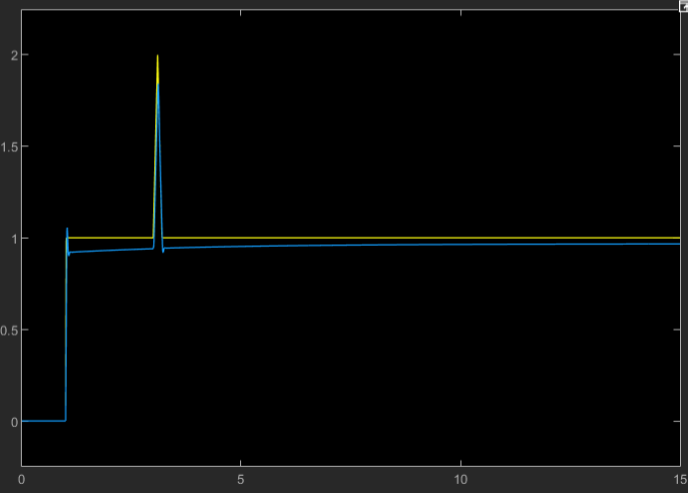
为了提高系统的响应速度和减少超调量，我们可以引入微分项，微分项的大小和输出值的变

化量成正相关，因此每当输入产生急剧变化时，就会产生一个很大的微分项来迅速跟上这一

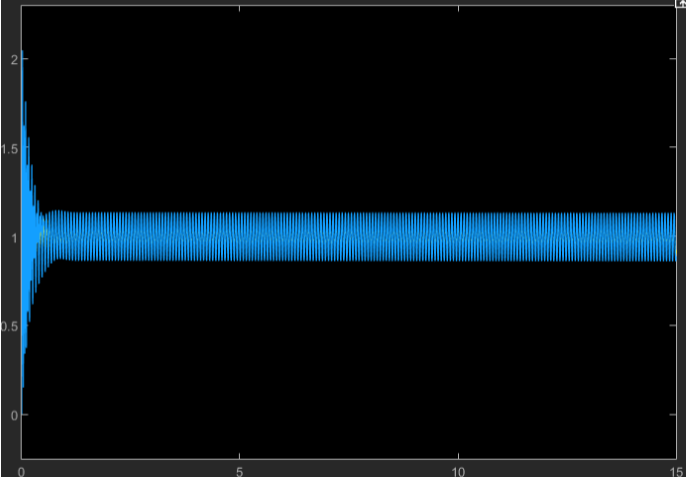
变化，这就可以使得超调量减小。另外由于微分项抵消变化的这一性质，输出波动就可以更

快的衰减，减少系统输出稳定需要的时间，从而加快响应速度。

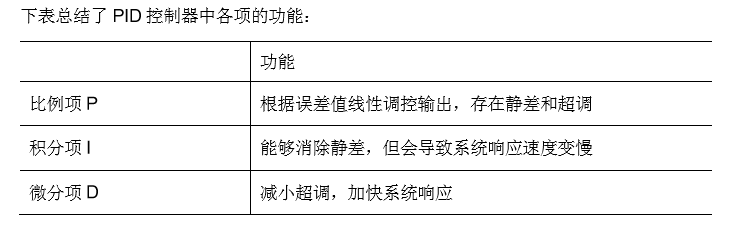




但是如果 D 项过大则输出的一点小变化都会产生很大的输出变化，导致系统发生震荡，如 下图所示：

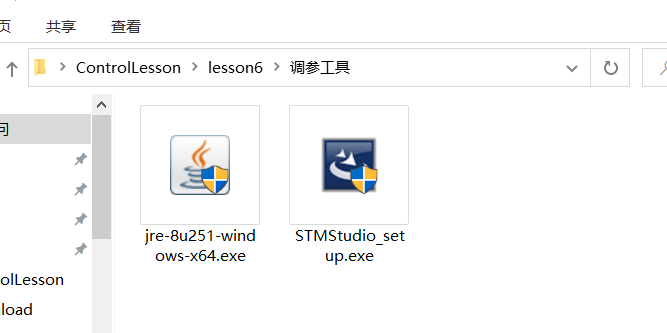


一般我们设计PID控制器时，需要根据PID的实际输出情况对PID 三项的系数的进行调节， 从而让系统的输出快速且稳定。

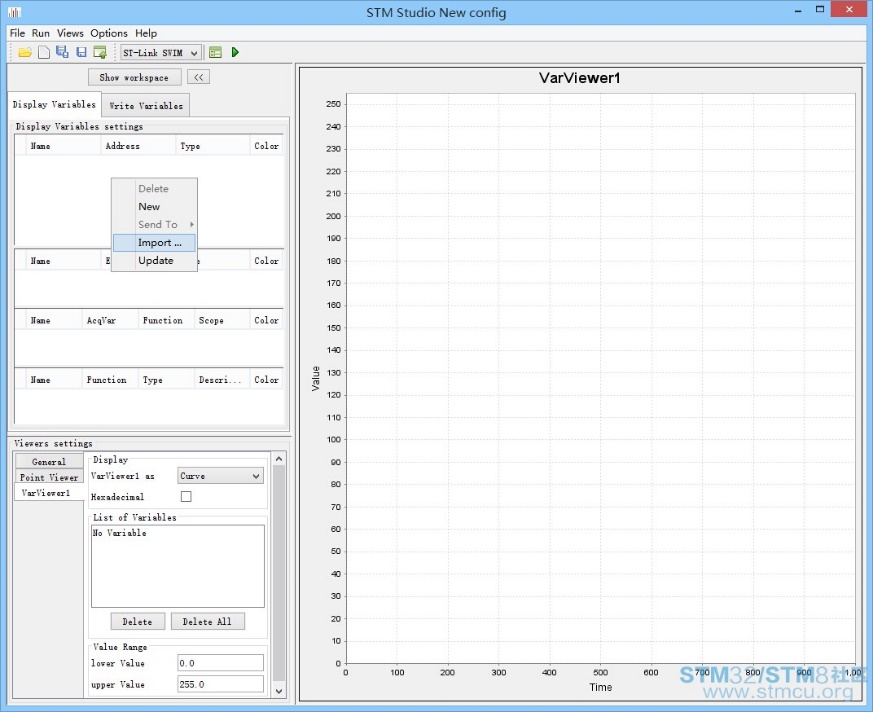


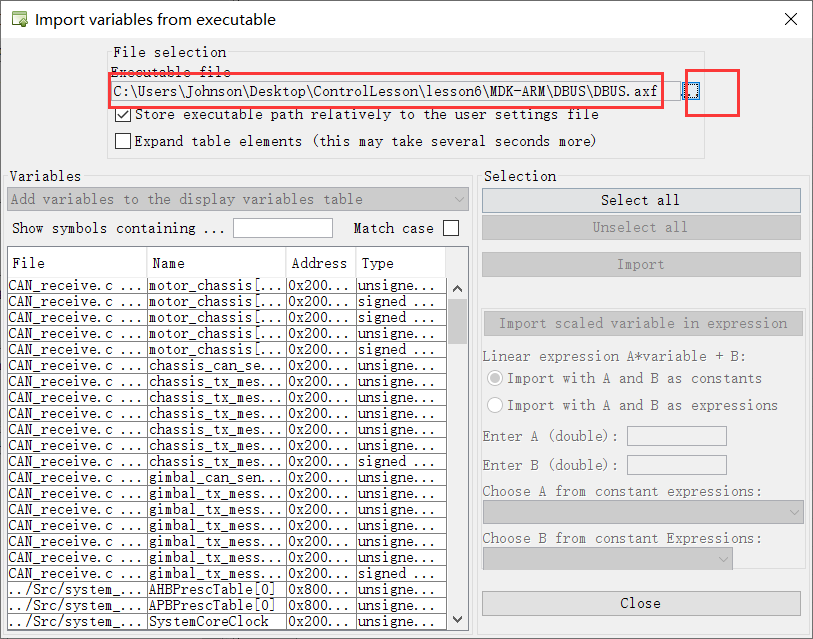
# 示波器调参软件使用方法

下载安装STMStudio，（st link专用），需要按照提示安装java支持（使用管理员权限）

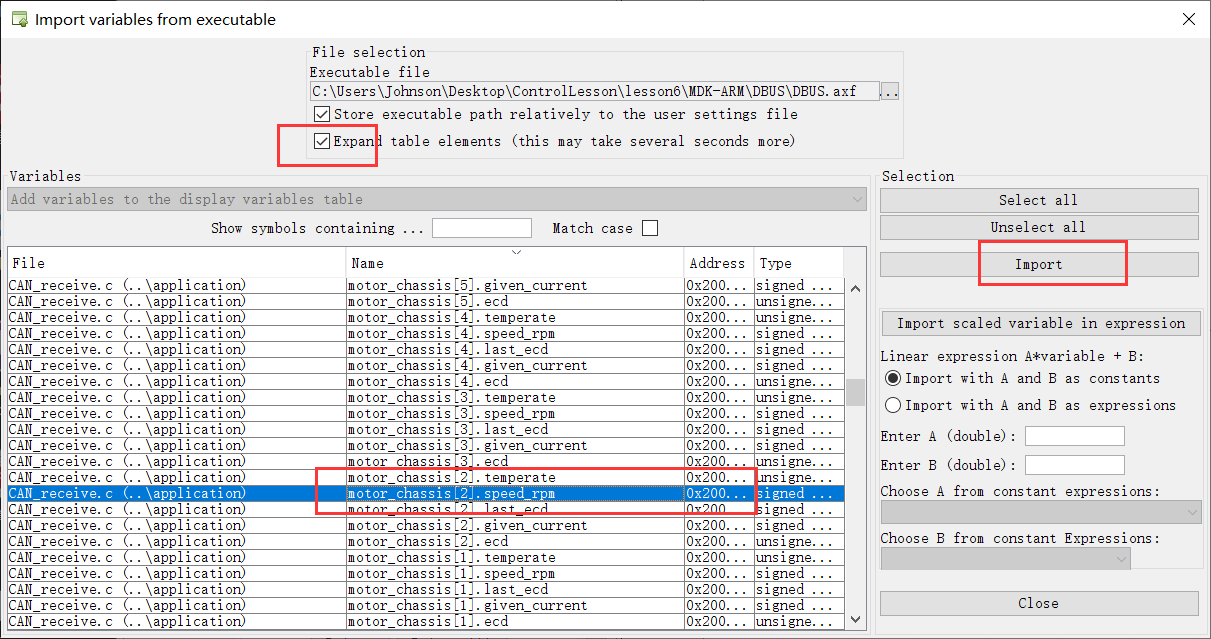


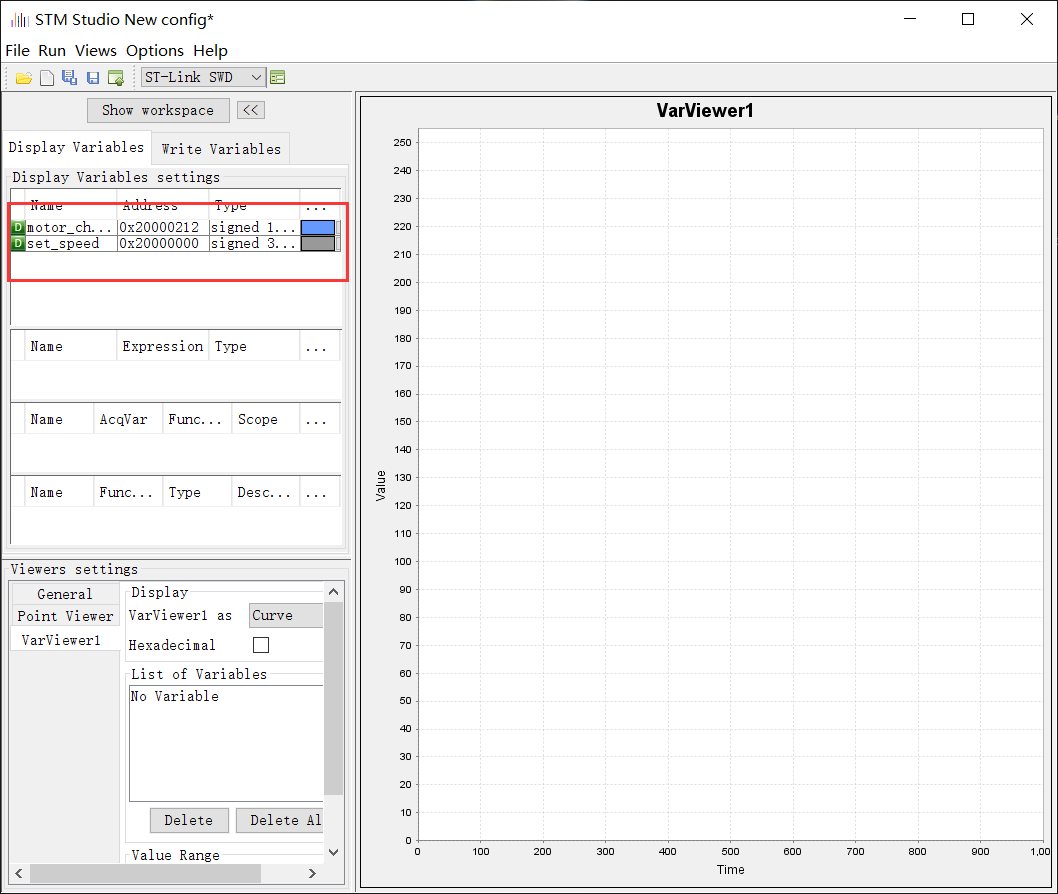
打开软件后载入KEIL编译生成的二进制文件



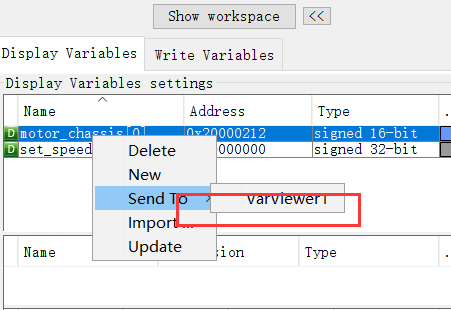


勾选将所有数组都 显示，在列表中找到实际速度和目标速度，点击Import

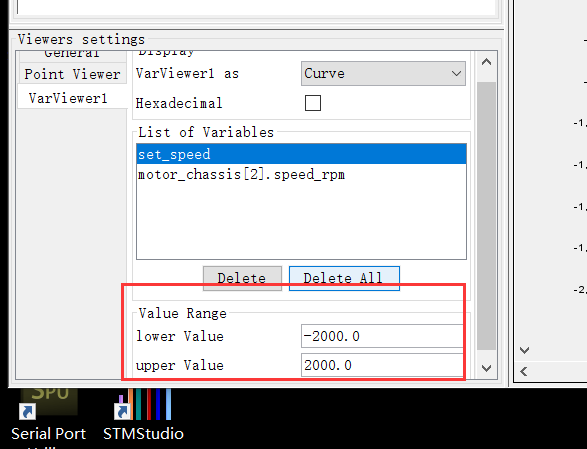


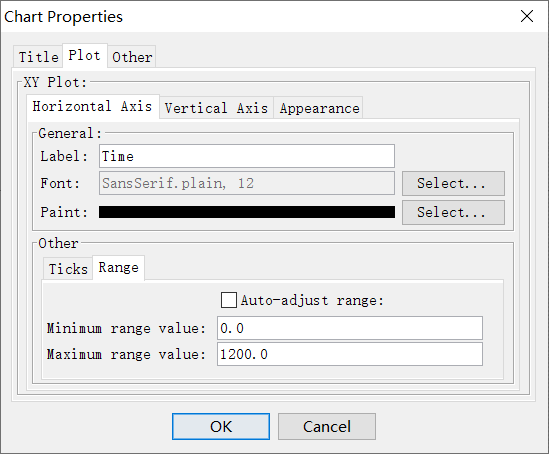


将这两个变量送入观察框中



设置一下图表y轴速度的显示范围





右键图表可以设置x轴时间的范围

**按照PID调参的方法，可以通过STMstudio实时写入目标速度的功能将目标转速设置成越阶的跳变，（需要在keil里面将遥控器的映射函数注释掉）然后只设置P，从最小值开始，观察示波器的波形，调整至理想后，加入I变量，一般不需要D，即可比较完美的实现电机的闭环控制**

