PID算法技术报告

PID算法是一种经典的控制策略，P表示Proportion（比例），I表示Integral（积分），D表示Defferential（微分）。PID控制对于一个我们不太了解的系统和控制对象很适用。它是根据系统的偏差值，利用比例，积分和微分运算得到控制量来控制系统。

**一．概述：**

**PID控制器的全公式**为：

我基于公式来阐述一下**PID的原理**：

我们先预设一个线性系统的被控制量的目标值为L，然后我们输入的控制量是u，被控制量目标值与实际值的差值记为e.

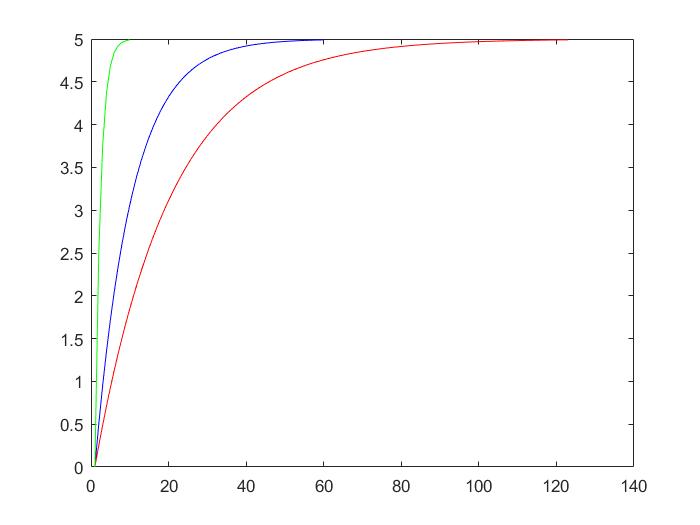
PID控制的目标就是使被控质量达到目标值L,这意味着我们要使e的值为0.

PID算法通过三个环节来使e趋近于0。先假设e为正值，也就是说目标值比当前值要大。

我们先看**比例环节**，该环节将上一次的误差乘以一个比例系数Kp作为系统的输入量，因为该系统是线性的，误差e会慢慢减小，直到等于0，并且如果Kp越大，则越快达到目标值，

增大比例系数主要是加快系统的反应速度。

用matlab仿真的结果如下：



绿色的曲线是Kp = 0.5，蓝色的线Kp = 0.05，红色的线Kp = 0.01.

但是如果系统中存在一个阻尼量，光是比例环节无法使输出达到目标值，会有一个误差，称为稳态误差，想要**消除系统的稳态误差**，需要**积分环节**。该环节对系统历史的偏差进行积分，如果误差为负值，说明目标值小于当前值，积分的增量减小，可以慢慢减小稳态误差，直到误差恒为0，则比例项和积分项都为0。总结一下，积分环节的主要作用是减小稳态误差，提高精度。

如果阻尼量不是一个常量，而是一个变化量，就需要用到**微分环节**了，微分主要针对误差的变化，它根据**误差的变化趋势**来改变输入。

**二．常见PID算法：**

常用的基本PID算法有位置式PID，增量式PID。

**1.位置式PID：**

也称为全量算法。它的输出是控制量的绝对值，输出能够直接控制系统的阀门位置，适用于不带积分元件的执行器。它的公式如下：

**2. 增量式PID：**

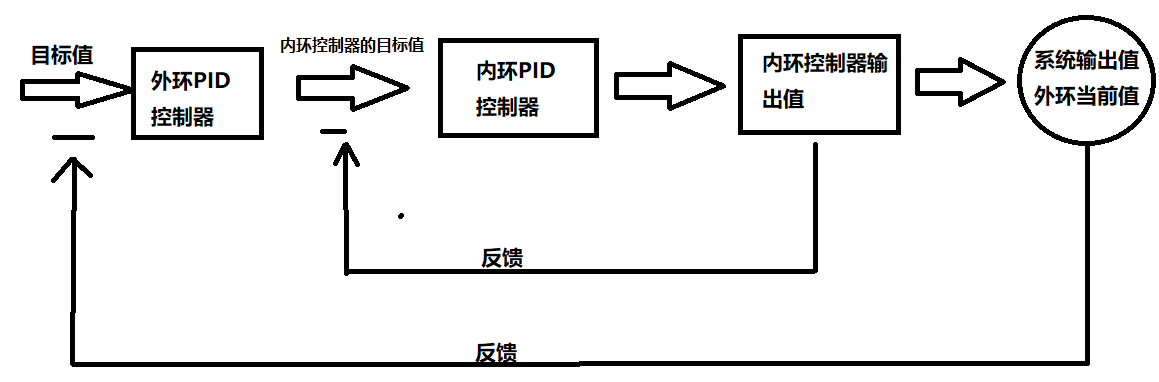
增量式PID算法输出的是控制量的增量，这种PID算法只需要最近三个误差值e(k),e(k-1),e(k-2)就可以了。公式如下：

该算法比较适用于自身带有积分记忆元件的器件。另外由于位置式算法必须对误差不断积分，会使得计算量不断加大，而且一旦累加过程中有错误，会对控制系统造成很大的影响，而且容易出现积分饱和现象，改进方法是进行积分限幅。

而增量式算法不需要累加，只用到最近三次的误差值，不会出现积分饱和现象。

**3.串级PID**

串级PID是将两个PID控制器串联工作，这样能够使系统控制更快，提高了稳定性。两个PID控制器分为内环和外环，一般将外环PID控制器的输出作为内环PID的目标值。工作的框图如下:



**4.改进基本的PID算法：**

**1）积分分离：**引入积分环节能够消除静差，但如果系统有大幅度的变化，会使得输出的控制量超过最大值，引起较大的超调，为了避免这点，使用积分分离，即在被控制量与设定值相差过大时，不进行积分，而在被控量趋近于稳定状态时，引入积分，消除静差。

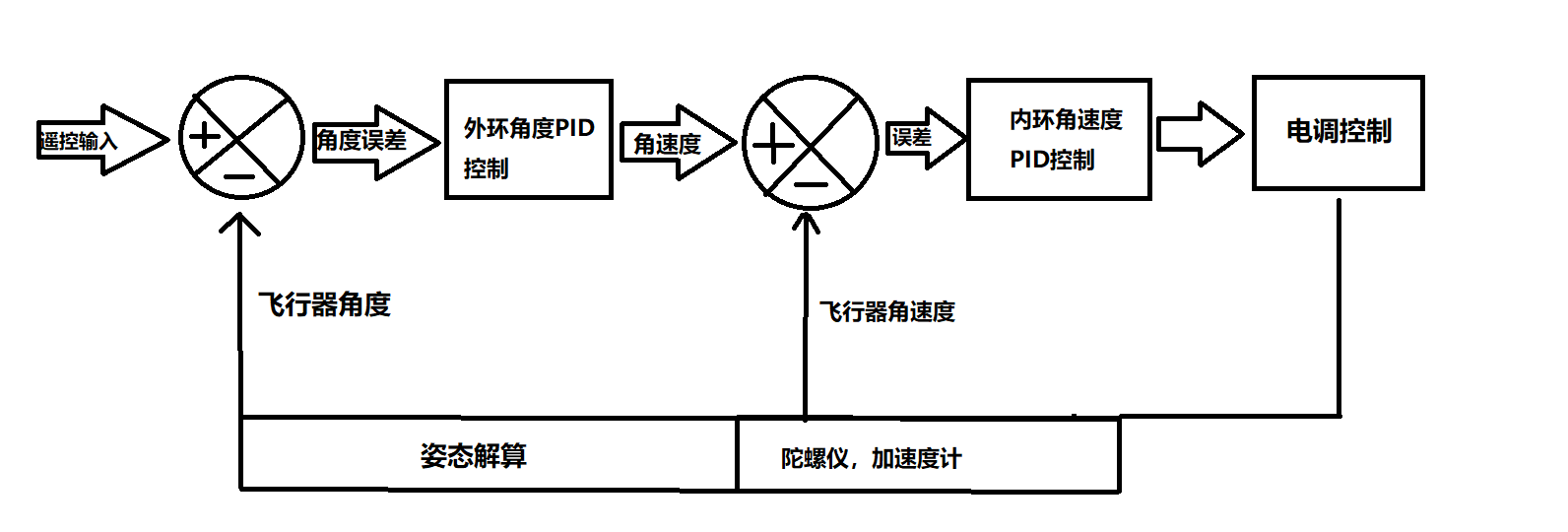
实现步骤为：设定一个阈值k，当误差e>k时，不累加，采用PD控制，e<k时，采用PID控制。

**2）抗积分饱和：**如果系统存在一个方向的偏差，PID算法的输出会随着时间而不断增加，最终超过输出的上限，进入饱和区，在饱和区中系统的控制不正常。为了防止积分饱和，如果输出量u已经超过了极大值，就只累加负的偏差，如果u小于极小值，就只累加正的偏差，避免进入饱和区。

**3）不完全微分：**微分环节可以改变系统的动态特性，但对于高频的干扰比较敏感，所以可以在PID控制器前面或者后面加入一个低通滤波器，减低高频干扰。

**三．PID怎么运用到四轴飞行器上?**

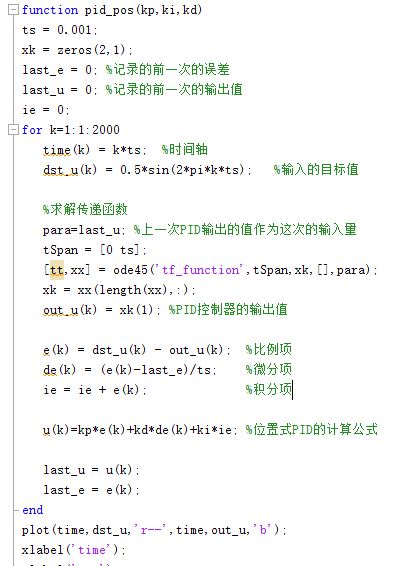
飞行器一般采用两级PID闭环串联控制的方式，一般用角速度控制环作为内环。角度控制环作为外环。为什么要用角速度作为内环呢？原因是角速度由陀螺仪采集，抗干扰能力强，而且角速度变化灵敏，受到干扰后恢复得很快。一般是用位置式数字PID控制。外环角度控制的目标值由遥控器输入，外环输出为角速度，作为内环的目标值，然后内环输出给电调用来控制飞行器。框图如下：



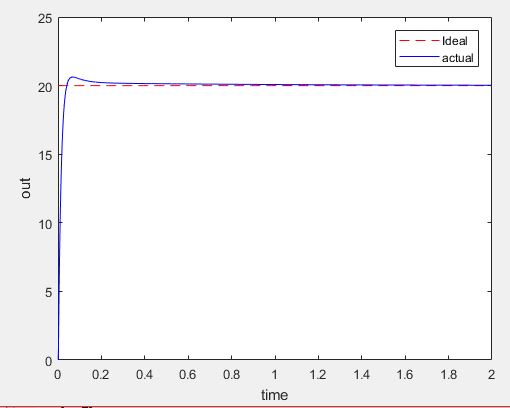
**四．仿真PID改进算法**

**1.位置式PID控制算法仿真：**

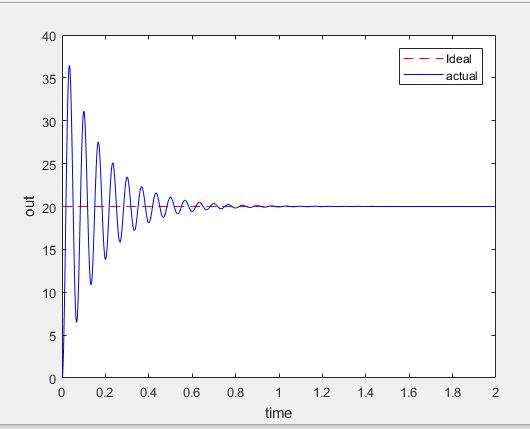
假设被控对象为电动机，传递函数为G(s)=，传递函数反映了系统激励与输出的关系。



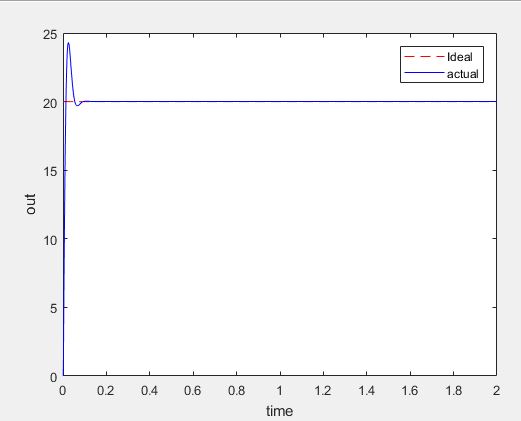
当kp=10,ki=0.01,kd=0.1时，输入阶跃信号20，pid能够使输出与目标基本相符合。



**2.增量式PID仿真：**

仿真对象仍然是之前的电动机模型，传递函数也相同，如果先输入kp=60,ki=0.01,kd=10,输出的图像如下：  


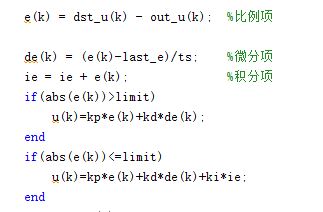
抖动很严重，考虑调大微分项的系数kd，可以消除，调整后图像如下：



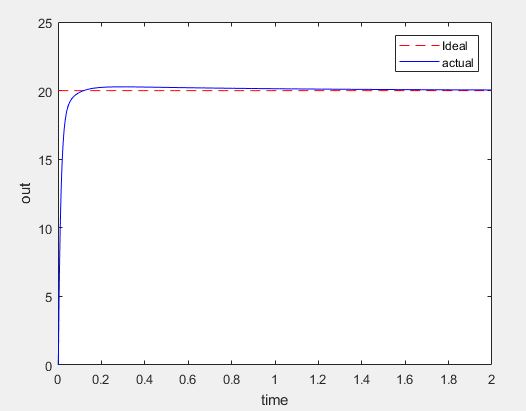
效果好很多了。

**3.积分分离PID控制算法仿真：**

相对于位置式PID算法，关键在这一部分



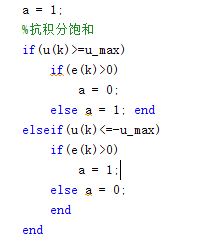
图像如下：



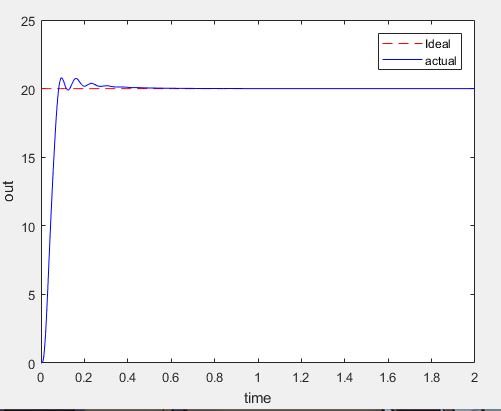
**4.抗积分饱和PID控制算法仿真：**

这次的控制对象的传递函数为：

抗积分饱和主要是对PID的积分部分进行判断，如果积分的值大于阈值，那么就只累积负的误差，若小于阈值，就只累积正的误差。



kp=0.87,ki=5,kd=0,仿真出来的图像如下：



如果不采用抗积分饱和，相同的参数，输出的结果如下：  
