

# 基于 MATLAB 的 PID 控制器设计

高林中<sup>1,2</sup>

(1.太原理工大学,山西太原 030024;2.山西煤炭职业技术学院,山西太原 030031)

**摘要:**介绍了 Ziegler-Nichols 规则的 PID 控制器设计,给出了基于 MATLAB 的实现方法和仿真。仿真结果表明,此算法设计的 PID 控制器有良好的性能指标。

**关键词:**PID;控制器;算法;MATLAB

**中图分类号:**TP273 **文献标识码:**A **文章编号:**1008-8881(2007)04-0150-02

## 一、前言

PID 控制是最早发展起来的经典控制策略,是用于过程控制最有效的策略之一。由于其原理简单、技术成熟,在实际应用中较易于整定,在工业控制中得到了广泛的应用。它最大的优点是不需了解被控对象精确的数学模型,只需在线根据系统误差及误差的变化率等简单参数,经过经验进行调节器参数在线整定,即可取得满意的结果,具有很大的适应性和灵活性。PID 控制中的积分作用可以减少稳态误差,但另一方面也容易导致积分饱和,使系统的超调量增大。微分作用可提高系统的响应速度,但其对高频干扰特别敏感,甚至会导致系统失稳。所以,正确计算 PID 控制器的参数,有效合理地实现 PID 控制器的设计,对于 PID 控制器在过程控制中的广泛应用具有重要的理论和现实意义。

在典型 PID 控制系统中,PID 控制器分别对误差信号  $e(t)$  进行比例、积分与微分运算,其结果的加权和构成系统的控制信号  $U(t)$ ,送给对象模型加以控制。PID 控制器的数学描述为:

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (1)$$

其传递函数可表示为:

$$G_{PID}(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (2)$$

从根本上讲,设计 PID 控制器也就是确定其比例系数  $K_p$ 、积分系数  $T_i$  和微分系数  $T_d$ ,这三个系数取值的不同,决定了比例、积分和微分作用的强弱。控制系统的整定就是在控制系统的结构已经确定、控制仪表和控制对象等处在正常状态的情况下,适当选择控制器的参数使控制仪表的特性和控制对象的特性相配合,从而使控制系统的运行达到最佳状态,取得最好的控制效果。本文介绍基于 MATLAB 的 Ziegler-Nichols 算法 PID 控制器设计。

## 二、PID 控制器的 Ziegler-Nichols 整定

在实际的过程控制系统中,有大量的对象模型可以近似地由一阶模型来表示。这个对象模型可以表示为:

$$G(s) = \frac{K}{1+sT} e^{-sL} \quad (3)$$

如果不能建立起系统的物理模型,可通过试验测取对

象模型的阶跃响应,从而得到模型参数。当然,我们也可在已知对象模型的情况下,由 MATLAB 通过 STEP() 函数得到对象模型的开环阶跃响应曲线。在被控对象的阶跃响应输出信号图(如图 1 所示)中,可获取  $K$ 、 $L$  和  $T$  参数,也可在 MATLAB 中由  $dcgain()$  函数求取  $K$  值。

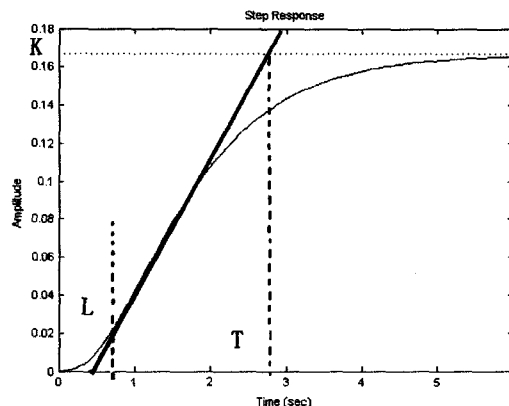


图 1 在开环阶跃响应曲线上确定 PID 参数

令  $\alpha = \frac{KL}{T}$ 。在求得  $L$  和  $\alpha$  参数的情况下,我们可通

过表 1 中给出的 Ziegler-Nichols 经验公式确定 P、PI 和 PID 控制器的参数。

表 1 Ziegler-Nichols 整定公式

控制器类型	由阶跃响应整定		
	$K_p$	$T_i$	$T_d$
P	$\frac{1}{\alpha}$		
PI	$\frac{0.9}{\alpha}$	$3.33L$	
PID	$\frac{1.2}{\alpha}$	$2L$	$\frac{L}{2}$

## 三、在 MATLAB 下实现 PID 控制器的设计与仿真

已知被控对象的  $K$ 、 $L$  和  $T$  值后,我们可以根据 Ziegler-Nichols 整定公式编写一个 MATLAB 函数  $ziegler\_std()$  用以设计 PID 控制器。该函数程序如下:

```
function [num,den,Kp,Ti,Td,H]=Ziegler_std(key,vars)
Ti=[];Td=[];H=[];
K=vars(1);L=vars(2);
T=vars(3);a=K*L/T;
if key==1,num=1/a; % 判断设计 P 控制器
elseif key==2,Kp=0.9/a;Ti=3.33*L; % 判断设计 PI 控制器
elseif key==3, % 判断设计 PID 控制器
Kp=1.2/a;Ti=2*L;Td=L/2;
end
switchkey
case 1,num=Kp;den=1; % P 控制器
```

**作者简介:**高林中(1968-),太原理工大学 2005 级电路与系统专业在读高校教师硕士,山西煤炭职业技术学院讲师。

收稿日期:2007-08-22

```

case2,num=Kp*[Ti,1];den=[Ti,0];    % PI 控制器
case3                                % PID 控制器
p0=[Ti*Td,0,0];
p1=[0,Ti,1];p2=[0,0,1];
p3=p0+p1+p2;
p4=Kp*p3;
num=p4/Ti;
den=[1,0];

```

endl

假设对象模型为一个三阶的传递函数  $G(s)=20/(s+1)(s+2)(s+3)$ ,则在 MATLAB 中输入下列语句:

```

>>num=20;den=conv([1,1],conv([1,2],[1,3]));
step(num,den);K=dcgain(num,den)

```

Step(num,den)绘制的阶跃响应曲线如图 2 所示。从该图中,我们可以近似地提取出带有延迟的一阶环节模型  $L=0.67, T=2.70$ ,而  $K$  由  $dcgain()$ 函数得到: $K=3.3333$ 。

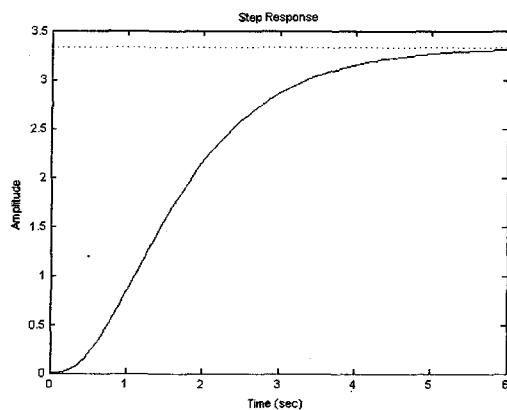


图 2  $G(s)$ 的阶跃响应曲线

故 PID 控制器可以由函数  $Ziegler\_std()$ 得到

```
>>K=3.3333;L=0.67;T=2.70;
```

```
[num1,den1,Kp,Ti,Td]=Ziegler_std(3,[K,L,T])
```

运行结果如下:

```

num1=
    0.4860    1.4508    2.1653
den1=
     1     0
Kp=
    1.4508
Ti=
    1.3400
Td=
    0.3350

```

至此,我们可在动态仿真集成环境 Simulink 下构造系统模型(如图 3 所示):

在 Simulink 窗口下点击开始仿真按钮,双击 SCOPE 图标,即得到阶跃响应曲线(如图 4 所示)。其中,上面曲线为未接入 PID 的阶跃响应曲线,下面曲线为接入 PID 的闭环响应曲线。

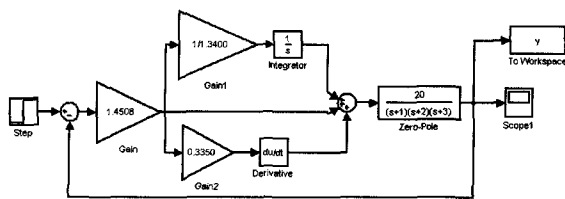


图 3 控制系统模型图

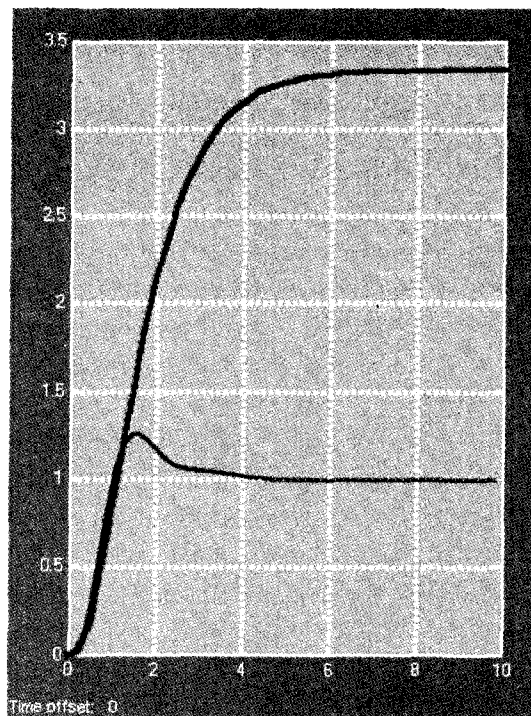


图 4 阶跃响应曲线

#### 四、结论

从系统接入 PID 控制器前后的阶跃响应曲线中,我们可以明显地看到系统性能的提高。

利用 MATLAB 可以实现 PID 控制器的离线设计和整定,并可实现实验室仿真。但是这种常规的 PID 控制不具有自适应性,在长期工作时对象参数会产生偏移,系统具有时变不确定性,也存在非线性,工况点附近小范围的线性化假设在整个工作范围中不能成立时,就难以达到理想的控制效果。为此,我们常常考虑自适应的 PID 控制算法。

#### 参考文献:

- [1]李国勇,谢克明.控制系统数字仿真与 CAD[M].北京:电子工业出版社,2003.
- [2]金以慧.过程控制[M].北京:清华大学出版社,1993.
- [3]邹治军,张晓江.基于 MATLAB 的 PID 控制器的设计算法及其实现[J].仪器仪表用户,2005,12(5).