实例分析：温度控制

1 问题描述

参考文献: 陈文科，刘强，王健雄，“基于专家 PID 的挤塑机温度控制系统设计，”合成树脂及塑料，2021，38 (4)：44-46，完成以下两个任务：

(1) 单位阶跃响应下，编程实现传统 PID 对上述系统的控制结果 (PID 参数固定)

(2) 编程实现专家 PID 对上述系统的校正结果 (PID 参数根据规则调整)。

2 系统描述

参考文献中，设计了一种基于 PLC+PROFINET 总线的挤塑机温度控制系统，其结构见图 1。该系统由控制核心 S7-1200 型 PLC、温度传感器、PROFINET 模拟量输入（AI）、PROFINET 数字量输出（DO）、人机界面 (HMI）等组成。

工作过程如下：使用 HMI 设定机筒和机头各温区温度后，PLC 借助 PROFINET DO 控制加热器工作，并借助各温区 PROFINET AI 获取温度传感器所检测到的温区温度信息，将 HMI 的温区温度设定值与各温区测量温度比对，PLC 采用专家 PID 调节加热器及冷却装置，以确保实际温度与设定温度一致。

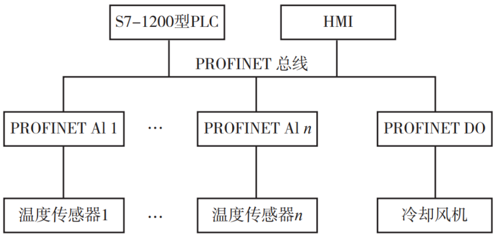

图 1: 控制系统结构示意图

挤塑机温度控制系统具有时变性、非线性和滞后性，其数学模型见式 (1)

$$G(s)=\frac{Ke^{-\tau s}}{Ts+1} \tag{1}$$

式中：K 表示稳态增益；τ 表示滞后时间；T 表示惯性常数；G（s），s 分别表示拉氏变换的通用输出和变量。

3 PID 控制原理

3.1 传统 PID 控制原理

PID 控制是一种结构简单、易于工程化的经典闭环控制算法，常用于控制温度、压力、流量等过程量信号。PID 控制时域数学模型见式 (2)，该式为典型的线性控制算法，其输出为对偏差进行比例、微分、积分计算后的加权求和值。

$$u(t)=K_{\mathrm{p}}e(t)+K_{\mathrm{i}}\int e(t)\mathrm{d}t+K_{\mathrm{d}}\frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{2}$$

式中：u(t) 表示输出；K_p，K_i，K_d 分别表示比例系数、微分系数、积分系数；e(t) 表示偏差。

PID 控制通过调节偏差来控制输出，使被控量朝着减小偏差的方向变化，其中，比例系数 K_p 用于控制偏差减小的速度，积分项用于减小稳态误差，微分项用于防止被控量的超调。

3.2 专家 PID 控制原理

针对非线性控制系统，PID 控制的不足之处在于比例、积分、微分参数由工程技术人员根据被控系统模型和经验整定而来，且为离线调整，不能根据系统的变化实时调整。因此，参考文献中提出了专家 PID 控制，从而根据现场温度偏差及温度偏差变化率实时调整优化 PID 控制参数。

专家 PID 控制结构图如下：

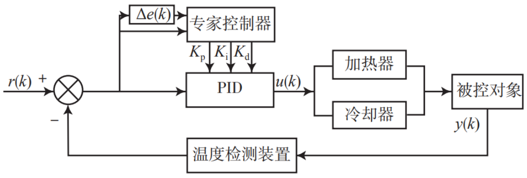

图 2: 专家 PID 控制

图中，Δe(k) 表示温度偏差变化率；r(k) 表示输入温度，℃；u(k) 表示输出控制信号；y(k) 表示反馈温度，℃。

专家 PID 控制的具体参数优化规则如下：

1. 若 $|e(k)|>$ 误差绝对值上限 (Max1)，此时 $|e(k)|$ 非常大，应该取较大的 K_p（根据所用的设备，此处为基础值的 5.0 倍），K_d 较小，K_i=0。
2. 若误差绝对值中间值 (Max2)<$|e(k)|$，e(k)×Δe(k)>0，此时增大 K_p，保持 K_i，K_d 不变。
3. 若 Max2>$|e(k)|$，e(k)×Δe(k)>0，此时保持 K_p，K_i，K_d 不变。
4. 若 e(k)×Δe(k)<0，Δe(k)×Δe(k-1)>0 且 Max2>$|e(k)|$，此时取较小 K_p（根据所用的设备，此处为基础值的 0.6 倍），使 K_i=0，K_d=0。
5. 若 e(k)×Δe(k)<0，Δe(k)×Δe(k-1)<0 且 Max2>$|e(k)|$，此时取较大 K_p（根据所用的设备，此处为基础值的 3 倍），使 K_i=0，K_d=0。
6. 若 $|e(k)|<\epsilon$，此时保持 K_p，K_i 不变，使 K_d=0。

其中，0<Max2<Max1，ϵ 为很小的正数。

4 实验步骤

4.1 构建系统的连续传递函数

上面已经建立了仿真模型，这里取稳态增益 K=0.3，滞后时间 τ=0.01，惯性常数 T=0.05，得到系统传递函数为：

$$G(s) = \frac{0.3e^{-0.01s}}{0.05s+1} \tag{3}$$

4.2 构建系统的离散传递函数

为了方便计算机处理，需要对该函数进行采样，设置采样时间为 0.001s，得到系统的离散传递函数为：

$$G(z) = \frac{m(1)\cdot z + m(2)}{n(1)\cdot z + n(2)} = \frac{0.0059}{z-0.9802} \tag{4}$$

4.3 将传递函数转换为差分方程

在仿真过程中我们需要求解出时域表达式，因此需要借助差分方程解决，对于以下的 Z 变换：

$$Y(z) = G(z) \cdot U(z) = \frac{m(2)}{n(1) \cdot z + n(2)} \cdot U(z) \tag{5}$$

对上式进行反 Z 变换，可以得到:

$$y(k) = -n(2)y(k-1) + m(1)u(k) + m(2)u(k-1) \tag{6}$$

4.4 构建传统数字 PID 控制

在原理中给出的 PID 公式含有积分和微分，在计算时，需要进行转换。这里用差分代替微分环节，用累加和代替积分环节，比例环节则保持不变，可以得到 PID 的控制器输出：

$$u(k) = K_p \cdot e(k) + K_d \cdot (e(k) - e_1) + K_i \cdot E_e \tag{7}$$

4.5 传统 PID 和专家 PID 仿真建立

有了上面的准备后，需要确定各参数数值。由于原论文中并未给出相关参数的具体数值，这里进行反复尝试，最终设定 K_p=0.6，K_i=0.2，K_d=0.05，max1=0.8，max2=0.5。

将各公式代入循环，即可得到传统 PID 的控制效果仿真，在传统 PID 的基础上，加入各参数的更新规则，即可得到专家 PID 的控制效果仿真。

5 结果分析

仿真结果如下图所示：

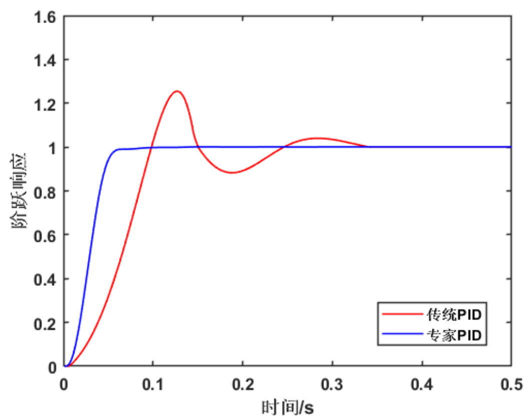

图 3: 仿真结果

从图中可以看出，传统 PID 控制和专家 PID 控制的上升时间（指响应曲线从零时刻到首次达到稳态值的时间）分别为 0.100，0.075s；超调量分别为 25%，0；稳定时间分别为 0.10，0.33s。显然，专家 PID 控制在响应速度、超调量和温度稳定速度等方面优于传统 PID 控制。

6 程序代码

matlab 代码：

```matlab
clear;
clc;
%% 参数定义
Ts = 1e-3;% 采样时间
e_sum = 0;% 多次误差和
e_sum2 = 0;
% PID参数
kp = 0.6;% 比例
ki = 0.2;% 积分
kd = 0.05;% 微分
kp2 = 0.6;% 比例
ki2 = 0.2;% 积分
kd2 = 0.05;% 微分
%% 建立被控系统
s_sys = tf(0.3,[0.05 1],'inputdelay',0.01); % 根据传递函数建立被控系统的模型,延迟设为
z_sys = c2d(s_sys,Ts,'z');      % 拉氏变换——>z变换
[m,n] = tfdata(z_sys,'v');
%% 开始PID控制
T = 500;% 设置仿真运行时间
r = 1.0;% 期望输出值
max1 = 0.8;
max2 = 0.5;
% 预先分配内存
u = zeros(1,T);% PID输出初始值
y = zeros(1,T);% 被控系统响应输出
e = zeros(1,T);% 误差信号
u2 = zeros(1,T);
y2 = zeros(1,T);
e2 = zeros(1,T);
for k=2:1:T
y(k) = -n(2)*y(k-1) + m(1)*u(k) + m(2)*u(k-1);% 计算被控系统输出
e(k) = r - y(k);   % 计算误差
u(k) = kp*e(k) + ki*e_sum + kd*(e(k)-e(k-1)); %根据误差调整PID控制量输出
e_sum = e_sum+e(k);% 计算多次误差和

% 专家控制PID
y2(k) = -n(2)*y2(k-1) + m(1)*u2(k) + m(2)*u2(k-1);% 计算被控系统输出
e2(k) = r - y2(k);   % 计算误差
u2(k) = kp2*e2(k) + ki2*e_sum2 + kd2*(e2(k)-e2(k-1));
e_sum2 = e_sum2+e2(k);% 计算多次误差和
if abs(e2(k)) >= max1
kp2 = 5*kp2;
kd2 = 0.5;
```

```
ki2 = 0;
end
if abs(e2(k))>= max2 && e2(k)*(e2(k)-e2(k-1))>0
kp2 = kp2 + 0.1;
end
if abs(e2(k))< max2
if e2(k)*(e2(k)-e2(k-1))<0 && e2(k)*e2(k-1)>0
kp2 = 0.6*kp2;
ki2 = 0;
kd2 = 0;
elseif  e2(k)*(e2(k)-e2(k-1))<0 && (e2(k)-e2(k-1))*(e2(k-1)-e2(k-2))<0
kp2 = 3*kp2;
ki2 = 0;
kd2 = 0;
end
end
if abs(e2(k))<=0.001
kd2 = 0;
end

end
% 绘制过渡过程的曲线
t = 1:1:T;
figure(1);
plot(t/1000,y,'r--',t/1000,y2,'b--','LineWidth',1.2); %t/1000 时间单位转换成秒
% ,time,y,'b',
xlabel('时间/s');
ylabel('阶跃响应');
set(gca,'FontSize',12,'LineWidth',1.2,'Fontname', 'Microsft YaHei UI','FontWeight','Bold','YLim',[0 1.6])
```