

座位号

实验日期 2021/5/11

实验名称	实验四 Matlab 混合编程和 PID 改进算法温度控制				
实验时间	年 月 日 周 - 节				
实验成绩	预习成绩	30%			
	操作成绩	50%			
	报告成绩	20%			
	总 评				
评改教师				评改日期	

一、预习说明：

请在实验开始前完成预习报告并提交，如未完成预习报告，不可参加实验课程。

1 实验目的

- 1) 熟悉温度 PID 控制原理；
- 2) 熟悉单水箱温度控制工作原理；
- 3) 了解 Matlab 的 M 文件编写方法和利用 Matlab 制作相应的 dll 文件，掌握 C#-Matlab 混合编程技术；
- 4) 学习 PID 改进算法的温度控制。

2、分析讨论题

- 1) 分析常规 PID 温度控制的特点。

常规 PID 温度控制具有非线性、高滞后性的特点。由于无有效的散热方式，一但出现超调量过大就会导致静态误差和稳态时间过大（散热速度远慢于加热速度）。所以常规的 PID 控制并不合适于温度控制。

- 2) 论述 PID 有哪些主要改进方法，各用于什么场合。

积分分离算法：积分分离算法可以避免计算得到的控制量超出执行机构的可能的最大可能动作。避免执行机构进入饱和区域。使系统进入长时间的振荡状态。

遇限削弱积分算法：在进入控制量的范围内后停止积分，可以避免控制量长时间的停在饱和区。

不完全微分 PID 算法：只在第一个周期内有微分控制作用，对于时间常数较大的系统，其调节作用很小，不能达到超前控制误差的目的。

微分先行 PID 控制算法：只对 $y(t)$ 进行微分，而不对偏差进行微分，适用于 $r(t)$ 频繁升降的场合。

二、实验部分说明：

请在开始实验前，阅读实验指导书，明确实验内容，记录实验过程和原始数据（包括实验截图）。

1、实验内容

- 1) 利用专用的“三容水箱参数标定实验软件”对液位进行校准；
- 2) 记录有关数据，用手工进行计算与实验结果进行对比；
- 3) 记录实验数据，分析实验结果，并完成实验报告。

2、实验步骤(根据实验要求简述步骤)

1) 设备的检查和连接

- ①连接 RS232 串口线，将串口选择开关拨向使用的 RS232 串口类型的方向。
- ②连接电源线，将电源插头接到 220V 电源上。
- ③打开电源，电源指示灯亮。
- ④使用万用表检查 5V，24V，220V 电源是否正常。

2) 基于 C#软件常规 PID 温度实验

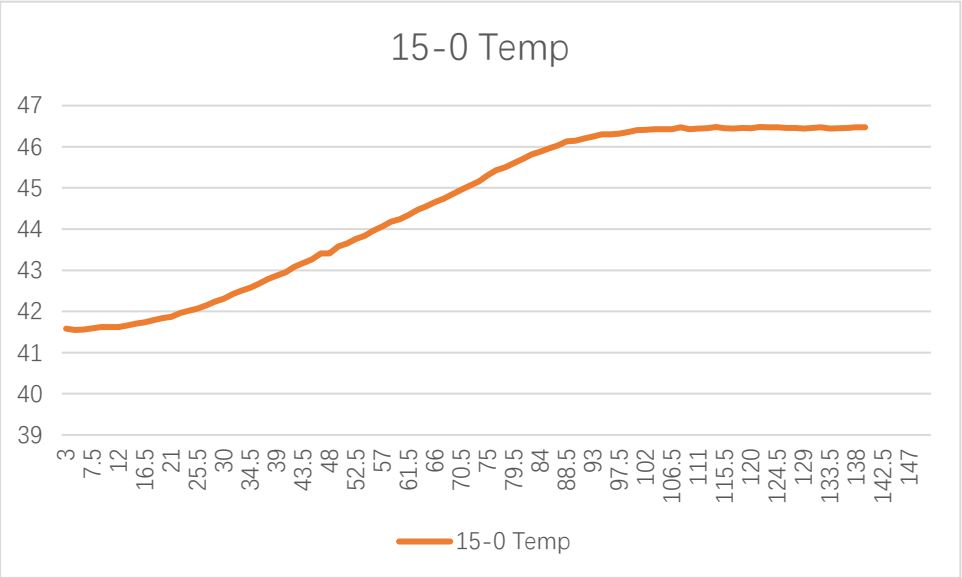
（请补充以下实验步骤）

5、数据处理与结果(记录实验过程，列出原始数据表格，注意实验过程和结果截图，并对实验数据进行必要的处理)

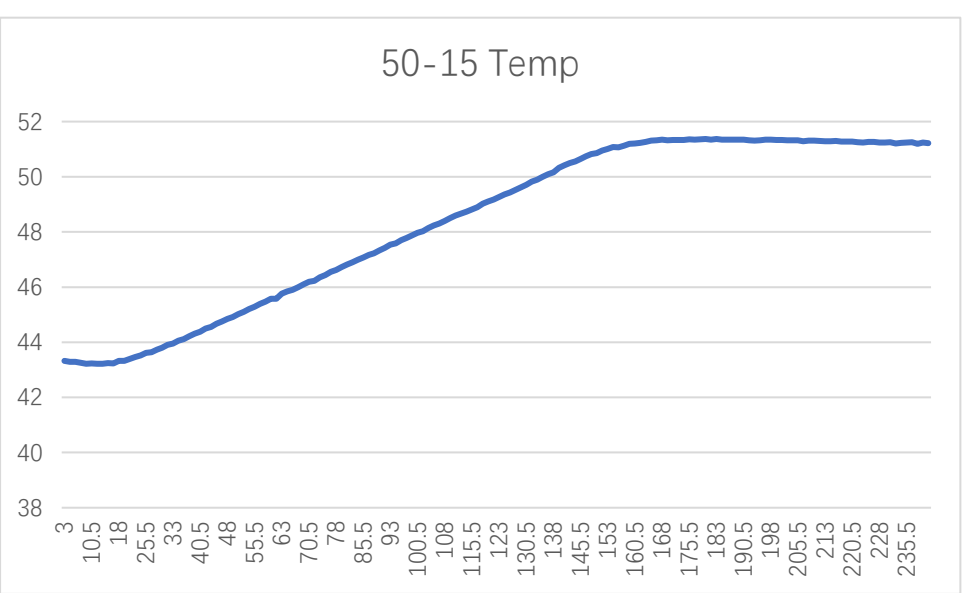
1) 基于 C#软件常规 PID 温度实验

通过上述实验，记录在保证预设液位一定的条件下，通过修改 PID 控制参数实现对温度控制的不同效果，观察实验现象，并记录数据，绘制温度控制曲线。

第一组 PID 参数下的温控曲线：

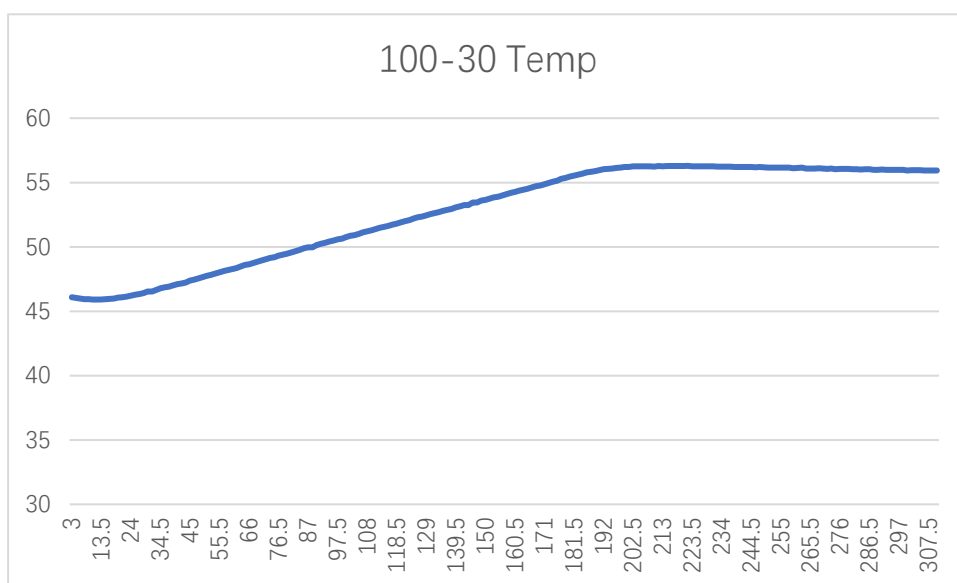


第二组 PID 参数下的温控曲线：



第三组

PID 参数下的温控曲线：



第四组 PID 参数下的温控曲线

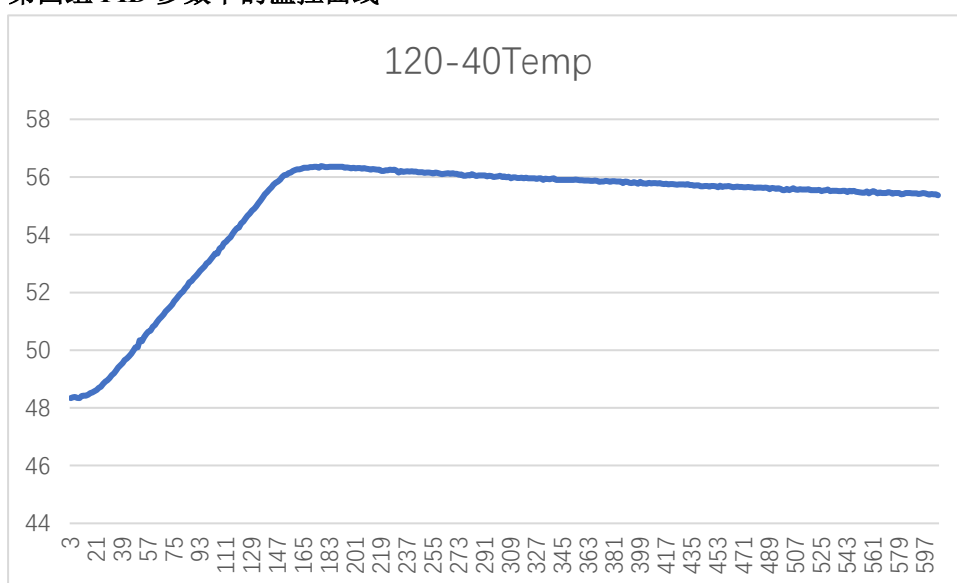


表 1 实验数据记录

参数	第一组	第二组	第三组	第四组
K_p	15	50	100	120
K_i	5	15	30	40
K_d	0	0	0	0
ess	1.46	1.240667	0.970714	0.415714
δp	0.0148	0.0137	0.013	0.0138
t	42	103	141	99

2) C#-Matlab 混合编程实验

①通过混合编程参数整定，得到较满意的温度控制曲线，确定 PID 控制参数

最优 PID 参数下的温控曲线：

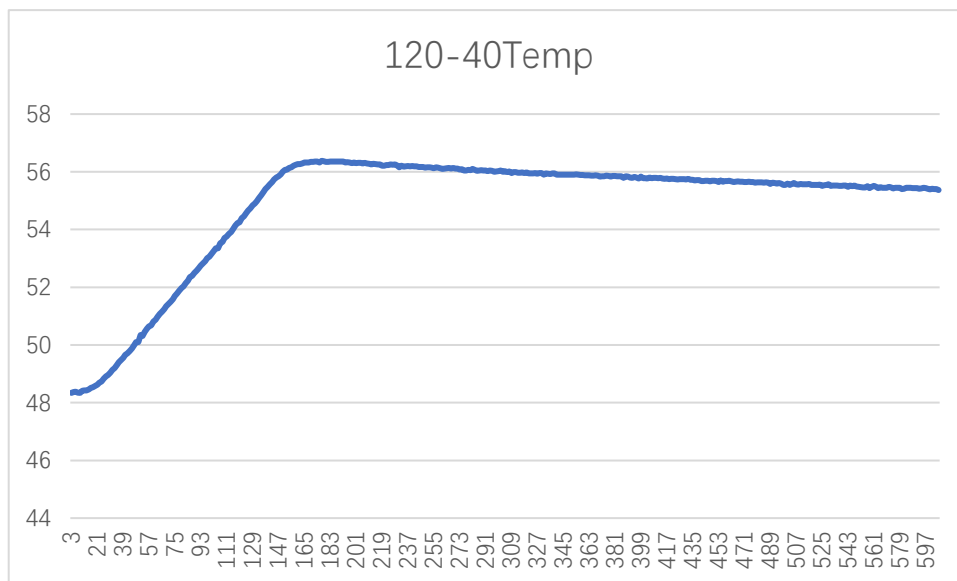


表 2 最佳 PID 控制参数记录

K_p	120	K_i	40	K_d	0
ess	0.415714	δp	0.0138	t	55.41

②维持 PID 参数不变，利用预习作业中的设计积分分离 PID 控制算法（当 $|e(k)| > 1$ ，也即偏差值较大时，采用 PD 控制，可避免大的超调，又使系统有较快的响应；当 $|e(k)| \leq 1$ ，也即偏差值较小时，采用 PI 控制，可保证系统的控制精度），实现温度控制。

温控曲线：

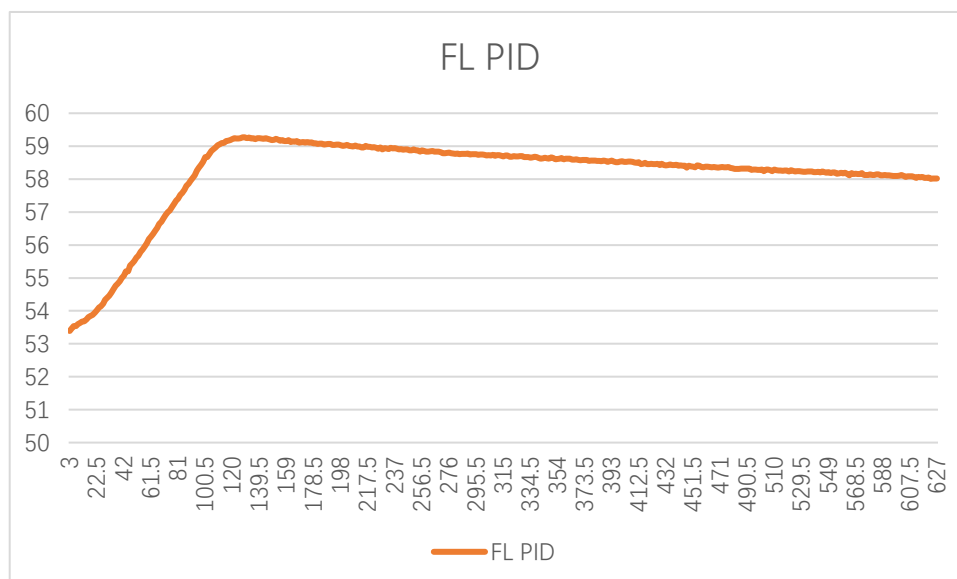


表 3 温度响应曲线参数

e_{ss}	0.0127	δp	0.03	t	58.03
----------	--------	------------	------	-----	-------

此时所使用的.m 文件程序代码:

```
function result =EasyTempPID(Set_Temp,Actual_Temp,i)
Kp=50;% 推荐在 50 左右
Ki=4; %推荐在 10
Kd=0;% 推荐 20 以下
global e
global u
global fl
if(i==0); %初始化
    e=[0 0 0];
    u=[0,0];
end
if(Set_Temp- Actual_Temp>1) % 执行 PI
    e(3) = e(2);
    e(2) = e(1);
    e(1) = Set_Temp-Actual_Temp;
    u(2)=u(1)+Kp*(e(1)-e(2))+Ki*e(1);
    u(1)=u(2);
    fl = u(2);
end
if(0<(Set_Temp- Actual_Temp)<1) % 执行 PD
    e(3) = e(2);
    e(2) = e(1);
    e(1) = Set_Temp-Actual_Temp;
    u(2)=u(1)+Kp*(e(1)-e(2))+Kd*(e(1)-2*e(2)+e(3));
    u(1)=u(2);
    fl = u(2);
end
```

```

if((Set_Temp- Actual_Temp)<=0)
    f1 = 0;
end
f = f1;%控制量 f1
end

```

三、完成实验后：

1、实验结果讨论与误差分析(试分析实验中常规 PID 与积分分离 PID 控制效果)

	常规 PID	积分分离
Kp	120	
Ki	40	
设定值	55	58
ess	0.39	0.03
超调量	0.0138	0.0127

实验结果如上图所示：在同样的 600s 内，积分分离算法比普通 PID 算法有着更小的静态误差（积分分离算法的静态误差几乎为 0）、有着更小的超调量。控制效果上明显由于普通 PID 算法。

2、试比较温度 PID 控制中 Kp 的值与实验三中液位 PID 控制时 Kp 值的大小，并请根据实验台温度和液位控制系统结构原理分析原因。

试验台温度控制系统使用加热管使温度上升，但散热却只能依靠空气散热，实验环境下一段时间内的散热效率可以视为一致的。而液位控制系统由水泵使得液面上升，当液位超过要求时可以通过水阀进行放水，可以按照需求进行控制。这也就导致了实验台温度控制静态误差的控制难度会远高于液位控制，所以温度控制的 Kp 会高于液位控制的 Kp，以确保尽可能小的静态误差。

3、实验心得

本次实验是试验台温度 PID 控制和 C#MATLAB 混合编程的实验。本次实验让我了解到了温度控制在 PID 中的控制实际（比液位等更难控制，对算法要求更高）。也实际了解了各

种的 **PID** 算法在实际系统应用中如何发挥作用,对积分分离算法有着一次实际操作的体验。理解到了改进型 **PID** 控制算法的强大效率。