

# 智能控制课程报告

王愉杰<sup>1,2</sup>

**摘要** 本文为智能控制课程报告，分为模糊控制、神经网络控制和专家系统控制三大部分。在模糊控制章节，设计电镀槽液温度模糊 PID 控制器，实现对 PID 三个参数进行在线调整。并与简单 PID 的效果进行对比分析，最后分析得出设计的模糊控制器性能良好。在神经网络控制章节，设计出一个神经网络模型，使其可以作为工业炉二阶系统动态特性的模型器，在工程开发和预测性分析中有广泛应用。在最后的阶跃信号跟踪测试中模型表现优秀。在专家系统控制章节，设计了炉温专家系统 PID 控制器，实时监测温度误差及其变化，并根据预设的控制规则调整加热元件的功率输出。在测试中发现设计的控制器性能良好。

**关键词** 智能控制, PID 控制, 模糊控制, 神经网络控制, 专家系统控制

**引用格式** 王愉杰. 智能控制课程报告. 自动化学报, 201X, XX(X): X-X

**DOI** 10.16383/j.aas.20xx.cxxxxxx

## Intelligent Control Course Report

WANG Yu-Jie<sup>1,2</sup>

**Abstract** This paper is an intelligent control course report, which is divided into three parts: fuzzy control, neural network control and expert system control. In the chapter of fuzzy control, the fuzzy PID controller of electroplating bath liquid temperature is designed, and the three PID parameters can be adjusted online. Compared with the effect of simple PID, the results show that the fuzzy controller has good performance. In the chapter of neural network control, a neural network model is designed, which can be used as a model of the dynamic characteristics of the second-order system of industrial furnace, and is widely used in engineering development and predictive analysis. In the last step signal tracking test, the model performs well. In the chapter of expert system control, the PID controller of furnace temperature expert system is designed to monitor the temperature error and its change in real time, and adjust the power output of heating element according to the preset control rules. The performance of the designed controller is found to be good in the test.

**Key words** Intelligent control, PID control, fuzzy control, neural network control, expert system control

**Citation** Wang Yu-Jie. Intelligent Control course report. Acta Automatica Sinica, 201X, China. Acta Automatica Sinica, 201X, XX(X): X-X

## 1 模糊控制

### 1.1 研究背景

随着现代控制系统理论的发展，逐步产生和实践出了模糊控制，总结工艺操作经验，建立模糊集合，利用语言规则进行模糊逻辑推理和决策，从而对复杂对象进行控制，逐渐成为智能控制的关键部分。

模糊控制论的诞生是以 1974 年美国 Mamdani 在实验室把模糊控制应用于锅炉和蒸汽机控制取得成功作为标志，这一创新离 1965 年 Zadeh 提出理论有 9 年时间，1975 年又在丹麦建造了模糊控制水泥窑。到 80 年代日本仙台地铁开通，成为当时最先进的地铁系统，自此很多家用电器、智能楼宇、电梯升降机等采用模糊控制算法开发，并接连研制成功投入市场。模糊集合理论，其本质是对复杂的工艺或者过程建立计算机语言的数学模型，让模糊语言

被 PC 或处理器接受，不仅成为解决客观模糊性问题的工具，也顺应了自适应控制的发展需求。

在传统的控制领域里，控制系统动态模式的精确与否是影响控制优劣的最主要关键。系统动态的信息越详细，则越能达到精确控制的目的。模糊逻辑控制 (Fuzzy Logic Control) 简称模糊控制 (Fuzzy Control)，是以模糊集合论、模糊语言变量和模糊逻辑推理为基础的一种计算机数字控制技术，是从行为上模仿人的模糊推理和决策过程的一种智能控制方法。

在之前，控制策略往往使用的是 PID，但是 PID 控制最大的问题在于三个参数的设置，常用的方法都是根据以往的曲线计算得出，但实际应用中，被控对象会受到很多环境因素的影响，导致 PID 自适应性差。模糊 PID 相比于传统 PID 的主要优点在于其能够实时优化参数以适应非线性时变系统的不确定性。这一点在面对被控对象结构和参数的不确定性时尤为明显，传统 PID 控制器的参数通常是

本文责任编辑 王愉杰

1. 中国地质大学 (武汉) 自动化学院 武汉 430074 2. 复杂系统先进控制与智能化湖北省重点实验室 湖北武汉 430074

特定环境调试后得到的,对于工作环境的变化适应性较差。而在模糊 PID 中,利用模糊逻辑和模糊规则可以对 PID 参数进行实时优化,从而克服了传统 PID 参数无法实时调整的缺点,提高了系统的适应性和鲁棒性。

### 1.2 研究对象

电镀是以电化学原理为基础的一种表面处理技术。电镀过程中,槽液温度对电镀膜质量有很大影响。通过控制槽液温度,可以优化结晶形貌、均匀性、致密性等电镀膜的物理属性,从而提高电镀质量。不同的电镀反应在不同温度下具有最佳的反应速率,通过控制槽液温度,可以使电镀反应在适宜的温度范围内进行,提高反应速率,缩短电镀时间,提高电镀效率。为获得较好的电镀效果,开展电镀槽液的温度控制研究具有重要意义。

电镀槽液温度的调控依赖于阀门的调节以及占空比(占空比为阀门开启时间与周期  $T$  的比值)。在实际电镀过程中,由于加热能量的传输存在延迟,因此电镀槽液温度的控制会出现相应的延迟。为此,引入一个稳态增益  $K$ ,并获取电镀槽液温度与蒸汽阀开通时间之间的传递函数  $G_s$ :

$$G(s) = \frac{K}{as^2 + bs + c} \quad (1)$$

由于不同的设备中传递函数的参数不同,实验中假设  $a=2$ 、 $b=5$ 、 $c=4$ 、 $K=10$ 。则传递函数可写为:

$$G(s) = \frac{10}{2s^2 + 5s + 4} \quad (2)$$

很多时候企业都是直接利用 PID 对电镀槽液的温度进行控制,有学者提出一种基于改进粒子群优化的比例-积分-微分控制算法,该方法在建立基于 PID 的电镀槽液温度控制器的过程中,采用改进的粒子群算法实现控制器三个参数的优化与自动调整,以保证电镀槽液温度的控制效果。但这样的方法未对由于环境等因素引起的数据波动进行修正, PID 控制器的参数是根据特定环境调试后得到的,对于工作环境的变化适应性较差,导致温度控制效果不佳、鲁棒性不强。

本文将模糊控制理论引入到 PID 控制中,增强了系统温度控制的自适应能力,实现 PID 参数的模糊调节,并提高控制精度。

### 1.3 研究原理

本章的研究分为两部分。第一个是设计电镀槽液温度 PID 控制器:以固定幅值的信号作为激励信号,表示电镀槽液期望的温度,通过 PID 控制器实现对信号的跟踪控制,使得电镀槽液达到期望值。

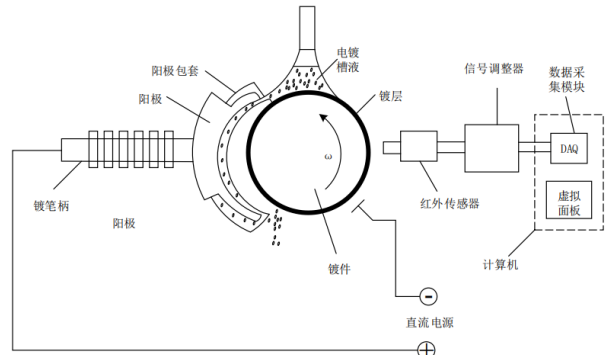


图 1 研究对象示意图

第二个是设计电镀槽液温度模糊 PID 控制器:模糊 PID 控制是结合 PID 控制和模糊控制得出的一种新型控制方式,结构如图 2 所示。PID 模糊控制重要的任务是找出 PID 的三个参数  $K_p$ 、 $K_i$ 、 $K_d$  与误差  $E$  和误差变化率  $E_c$  之间的模糊关系,在运行中不断检测  $E$  和  $E_c$ ,根据确定的模糊控制规则来对三个参数进行在线调整,满足不同  $E$  和  $E_c$  下三个参数的不同要求。模糊控制器有两个输入,三个输出。输入分别为跟踪误差  $E$  与误差的导数  $EC$ ,输出为  $K_p$ 、 $K_i$ 、 $K_d$  各自的变化量。通过 MATLAB 自带的 fuzzy 工具箱,设计不同语言变量的隶属度函数和对应的规则,最后在 Simulink 中运行结果,将两种控制策略的效果进行对比分析。

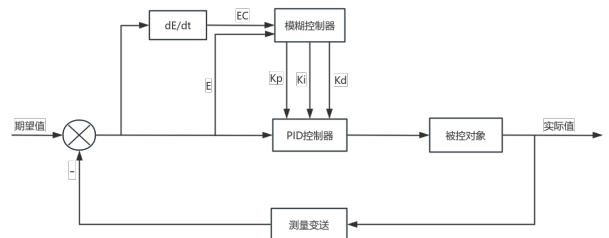


图 2 模糊 PID 结构图

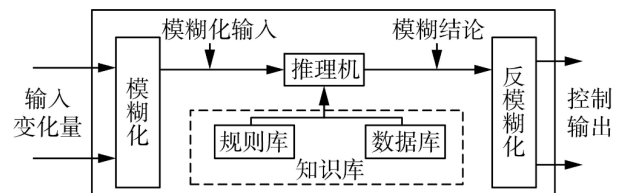


图 3 模糊控制器内部图

### 1.4 电解槽液 PID 控制

在 1.2 节中已知控制对象的传递函数为  $G(s) = \frac{10}{2s^2+5s+4}$ ，在 Simulink 模块中搭建电解槽液温度 PID 控制回路，如图 4 所示。

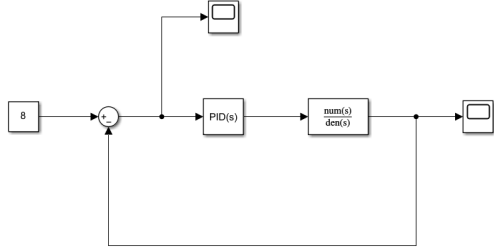


图 4 电解槽液温度 PID 控制回路

首先设置初始值，输入为定值 8、 $K_p=5$ 、 $K_i=5$ 、 $K_d=1$ ，观察控制效果。

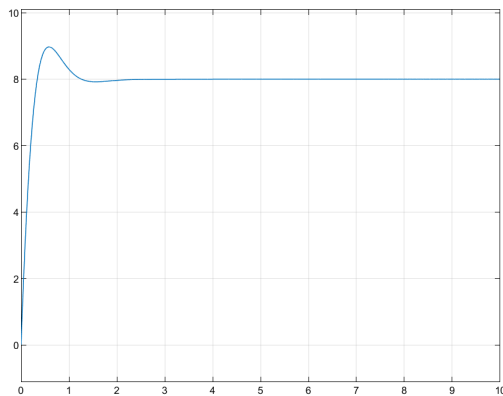


图 5 初始值下的控制效果

从图 5 可以看出，在上述参数下，温度控制的响应速度还不错，但是有超调量，在本文的电镀背景下，应当使温度的超调量尽可能小，以达到最佳的反应速率。接下来对 PID 控制器的参数进行调节。

调整 PID 参数前，最重要的是了解每个参数调节了系统响应的哪些属性，通过观察响应从而调节参数改变属性。PID 的作用概述如下：

(1) P 产生响应速度和力度，过小响应慢，过大会产生振荡，是 I 和 D 的基础。

(2) I 在有系统误差和外力作用时消除偏差，提高精度，同时也会增加响应速度，产生过冲，过大会产生振荡。

(3) D 抑制过冲和振荡，过小时系统会过冲，过大会减慢响应速度。D 的另一个作用是抵抗外界产生的突变干扰，阻止系统的突变。

PID 整定的方法一般有临界比例法、反应曲线法和衰减法。三种方法各有特点，其共同特点是都

通过试验，按照工程经验公式对控制器的参数进行整定，顺序一般是先比例再积分最后微分。

但是在 Simulink 中，PID 模块提供了参数调节器，使得 PID 的调节非常便捷，并且比手动测试计算的效果更好。双击 PID 模块，会出现下图中的界面。



图 6 PID 模块属性

点击图 6 的“调节”后，会进入 PID 调节器，我们可以改变响应时间的快慢和瞬态特性的稳健性来调整响应曲线，并且可以马上将调整后的曲线与原始曲线进行对比，如图 7 所示。

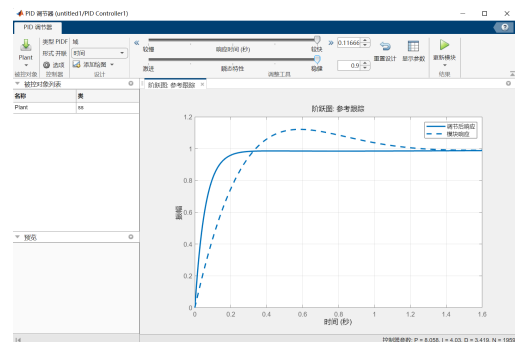


图 7 PID 调节器界面

通过 PID 调节器调节响应曲线，选择最快的响应时间和最稳健的瞬态特性，得到调整后的参数，最终  $K_p=8.0583588961179$ ， $K_i=4.03019751752646$ ， $K_d=3.41911669521807$ ，调整后的响应曲线如图 8 所示。

从图 8 中可以看出，温度最终达到了期望值，且无超调，基本达到要求。接下来对 PID 的控制性能进行评价，包括超调量、调节时间、上升时间和稳态误差，各自的含义如下。

(1) 超调量

响应的最大偏移量与终值的差与终值比的百分数，计算公式如下：

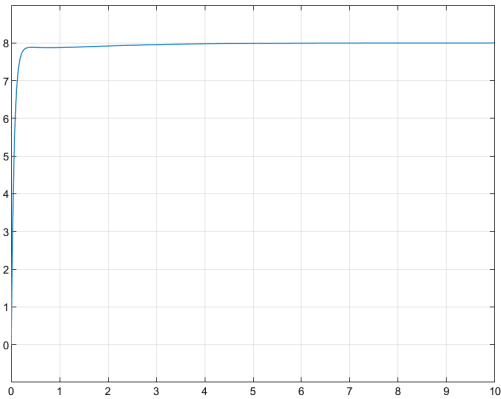


图 8 参数调整后的响应曲线

$$\delta\% = \frac{c(t_p) - c(\infty)}{c(\infty)} \cdot 100\% \tag{3}$$

(2) 调节时间

响应到达并保持在终值 5%或 2%内所需要的最短时间。

(3) 上升时间

响应从终值 10%上升到终值 90%所需的时间；对于有振荡的系统，定义为响应从零第一次上升到终值所需的时间；上升时间越短，响应速度越快。

(4) 稳态误差

稳态误差是控制系统在达到稳定状态后，输出值与期望值之间的偏差，计算公式如下：

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) \tag{4}$$

对上述 5 个指标进行计算，结果如下表所示。

表 1 PID 控制效果指标

指标	超调量	调节时间	上升时间	稳态误差
值	0	0.186s	0.1313s	0

从上表可以看出 PID 控制效果良好。

1.5 电镀槽液模糊 PID 控制

本小节对电解槽液控制对象设计模糊控制器，实现 PID 参数的调整。输入为误差和误差变化率，输出为 PID 三个参数的变化值。

1.5.1 模糊化

本文采用 Matlab 软件中的 FuzzyLogic Designer 工具箱，如下图所示，首先确定模糊控制器的输入和输出，输入为误差 E 和误差 EC，输出为 Kp、Ki 和 Kd 的变化值，实现初始值基础上的调整。

输入变量需要先经模糊控制器进行模糊化，即将实数输入量转换为模糊集合和对应的隶属度。本

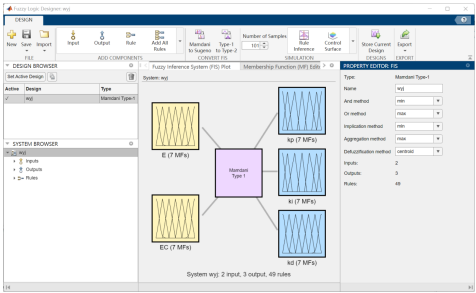


图 9 FuzzyLogic Designer 工具箱

文模糊控制器的模糊化就是将输入的三个参数的偏差 E 和偏差变化率 EC 执行模糊化的流程，即建立隶属函数得到隶属度。这里主要对误差的模糊化进行说明，本系统中其他输入输出的模糊化基本一样，以此类推。

根据实际要求，温度的控制范围为 [-6,6]，所以误差 E 的基本论域为 [-6,6]，依据控制精度将它离散为 7 个等级，对应 7 个模糊子集；EC 的基本论域为 [-3,3]，将它离散为 7 个等级，NB-负大、NM-负中、NS-负小、Z-零、PS-正小、PM-正中、PB-正大，每个模糊子集通过隶属度函数来定义。基于控制对象的特点，偏差、偏差变化率、输出变量选用三角形和高斯相结合的隶属度函数，输入变量和输出变量的隶属度函数分别如图 10-14 所示。

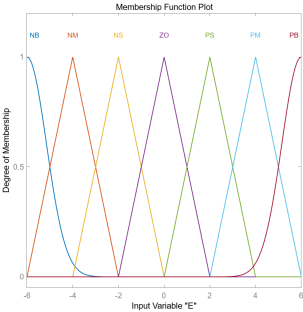


图 10 E 隶属度函数

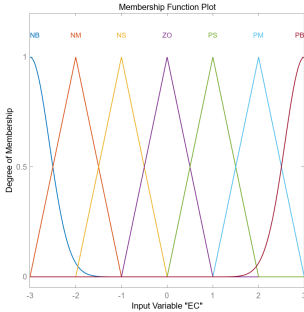


图 11 EC 隶属度函数

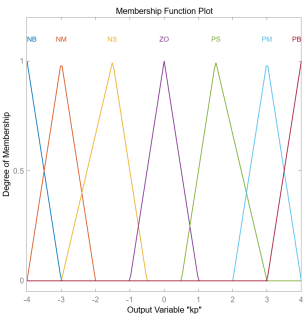


图 12 Kp 隶属度函数

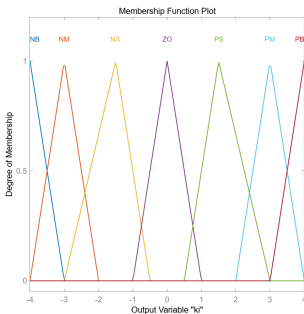


图 13 Ki 隶属度函数

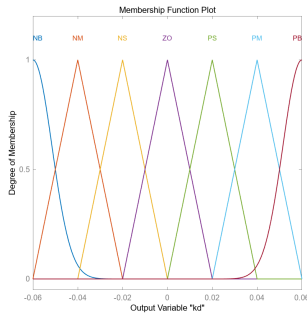


图 14 Kd 隶属度函数

### 1.5.2 规则库与推理

规则库集合了专家知识和技术熟练的操作工经验,是根据人类思维进行推理的语言表现方法。输入到输出的映射,经常使用“如果...,那么...”的规则,即 if...then... 语句来表示,或者 if...is...and... is... then...is...。本系统用如下语句,即 If  $E(t)$  is RB and  $EC(t)$  is RB then  $Kp(t)$  is RB  $Ki(t)$  is NB  $Kd(t)$  is PS 表示控制规则。

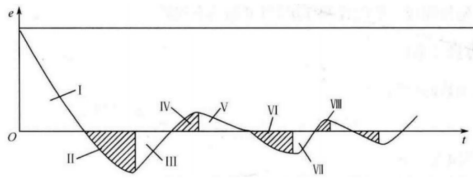


图 15 典型二阶系统单位阶跃响应的误差曲线

不同的偏差  $E$  和偏差变化率  $EC$  对 PID 控制器的参数  $Kp$ 、 $Ki$  和  $Kd$  有不同的要求。以典型二阶系统单位阶跃响应的误差曲线为例进行分析,如图 15。

(1) 误差  $|e|$  较大时,应该使控制器的  $Kp$  取较大值,以提高响应的快速性;而为了防止因为  $|EC|$  瞬时过大,  $Kd$  应该取较小的值;为控制超调,  $Ki$  也应该取值很小。

(2) 误差  $|e|$  中等时,为保证系统的相应速度,并控制超调,应减小  $Kp$ ,  $Ki$  值应适当增大,  $Kd$  应适中。

(3) 误差  $|e|$  较小时,为保证系统具有良好的稳态特性,应适当加大  $Kp$ 、 $Ki$  的取值,同时为避免产生振荡,  $Kd$  的取值应该和  $|EC|$  联系起来。此处要注意的是,对  $Kp$ 、 $Ki$ 、 $Kd$  三个参数来说,很大、很小的形容所指的具体值并不相同,100 对  $Kp$  较小,但对  $Ki$  与  $Kd$  却过大,要适当选择三个参数的范围。具体表现为利用工具箱进行隶属度函数设置时,论域设置的范围大小,要参考三个参数自身的特性。

模糊 PID 控制根据电解槽液系统运行的不同状

态考虑  $Kp$ 、 $Kd$ 、 $Ki$  三者的关联,根据工程经验设计模糊整定这三个参数,选择输入语言变量为误差  $E$  和偏差变化率  $EC$ ,语言变量值取 NB, NM, NS, ZO, PS, PM, PB 七个模糊值;选择输出语言变量为  $Kp$ 、 $Ki$ 、 $Kd$ ,语言变量值也取七个模糊值,建立  $Kp$ 、 $Ki$ 、 $Kd$  的模糊规则表如表 2、表 3、表 4 所示。

表 2 比例系数  $Kp$  模糊规则表

	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	ZO
NM	PB	PB	PM	PM	PS	ZO	ZO
NS	PM	PM	PM	PS	ZO	NS	NM
ZO	PM	PS	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	ZO	ZO	NS	NM	NM	NM	NB
PB	ZO	NS	NS	NM	NM	NB	NB

表 3 积分系数  $Ki$  模糊规则表

	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	ZO
NM	PB	PB	PM	PM	PS	ZO	ZO
NS	PM	PM	PM	PS	ZO	NS	NM
ZO	PM	PS	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	ZO	ZO	NS	NM	NM	NM	NB
PB	ZO	NS	NS	NM	NM	NB	NB

表 4 微分系数  $Kd$  模糊规则表

	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	ZO
NM	PB	PB	PM	PM	PS	ZO	ZO
NS	PM	PM	PM	PS	ZO	NS	NM
ZO	PM	PS	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	ZO	ZO	NS	NM	NM	NM	NB
PB	ZO	NS	NS	NM	NM	NB	NB

本文的模糊控制器有两个输入,三个输出,每个变量均有 7 个模糊子集,设计的规则共有 49 条, FuzzyLogic Designer 中的设计界面如下所示。

从图 17 可以看出,在空间中,模糊 PID 控制器



Rule	Weight	Name
1 If E is NB and EC is NB then Kp is PS, Ki is PS, Kd is PS	1	1.14001
2 If E is NB and EC is NB then Kp is PS, Ki is NB, Kd is NS	1	1.14002
3 If E is NB and EC is NS then Kp is PS, Ki is NB, Kd is NS	1	1.14003
4 If E is NB and EC is NS then Kp is PS, Ki is NB, Kd is PS	1	1.14004
5 If E is NB and EC is PS then Kp is PS, Ki is NS, Kd is NB	1	1.14005
6 If E is NB and EC is PS then Kp is PS, Ki is NS, Kd is PS	1	1.14006
7 If E is NB and EC is PS then Kp is PS, Ki is PS, Kd is NB	1	1.14007
8 If E is NB and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14008
9 If E is NM and EC is NB then Kp is PS, Ki is NB, Kd is NS	1	1.14009
10 If E is NM and EC is NB then Kp is PS, Ki is NB, Kd is PS	1	1.14010
11 If E is NM and EC is NS then Kp is PS, Ki is NB, Kd is NS	1	1.14011
12 If E is NM and EC is NS then Kp is PS, Ki is NB, Kd is PS	1	1.14012
13 If E is NM and EC is PS then Kp is PS, Ki is NS, Kd is NB	1	1.14013
14 If E is NM and EC is PS then Kp is PS, Ki is NS, Kd is PS	1	1.14014
15 If E is NM and EC is PS then Kp is PS, Ki is PS, Kd is NB	1	1.14015
16 If E is NM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14016
17 If E is NS and EC is NB then Kp is PS, Ki is NB, Kd is NS	1	1.14017
18 If E is NS and EC is NB then Kp is PS, Ki is NB, Kd is PS	1	1.14018
19 If E is NS and EC is NS then Kp is PS, Ki is NB, Kd is NS	1	1.14019
20 If E is NS and EC is NS then Kp is PS, Ki is NB, Kd is PS	1	1.14020
21 If E is NS and EC is PS then Kp is PS, Ki is NS, Kd is NB	1	1.14021
22 If E is NS and EC is PS then Kp is PS, Ki is NS, Kd is PS	1	1.14022
23 If E is NS and EC is PS then Kp is PS, Ki is PS, Kd is NB	1	1.14023
24 If E is NS and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14024
25 If E is Z and EC is NB then Kp is PS, Ki is NB, Kd is NS	1	1.14025
26 If E is Z and EC is NB then Kp is PS, Ki is NB, Kd is PS	1	1.14026
27 If E is Z and EC is NS then Kp is PS, Ki is NB, Kd is NS	1	1.14027
28 If E is Z and EC is NS then Kp is PS, Ki is NB, Kd is PS	1	1.14028
29 If E is Z and EC is PS then Kp is PS, Ki is NS, Kd is NB	1	1.14029
30 If E is Z and EC is PS then Kp is PS, Ki is NS, Kd is PS	1	1.14030
31 If E is Z and EC is PS then Kp is PS, Ki is PS, Kd is NB	1	1.14031
32 If E is Z and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14032
33 If E is PM and EC is NB then Kp is PS, Ki is NB, Kd is NS	1	1.14033
34 If E is PM and EC is NB then Kp is PS, Ki is NB, Kd is PS	1	1.14034
35 If E is PM and EC is NS then Kp is PS, Ki is NB, Kd is NS	1	1.14035
36 If E is PM and EC is NS then Kp is PS, Ki is NB, Kd is PS	1	1.14036
37 If E is PM and EC is PS then Kp is PS, Ki is NS, Kd is NB	1	1.14037
38 If E is PM and EC is PS then Kp is PS, Ki is NS, Kd is PS	1	1.14038
39 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is NB	1	1.14039
40 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14040
41 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14041
42 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14042
43 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14043
44 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14044
45 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14045
46 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14046
47 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14047
48 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14048
49 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14049
50 If E is PM and EC is PS then Kp is PS, Ki is PS, Kd is PS	1	1.14050

图 16 模糊规则设计

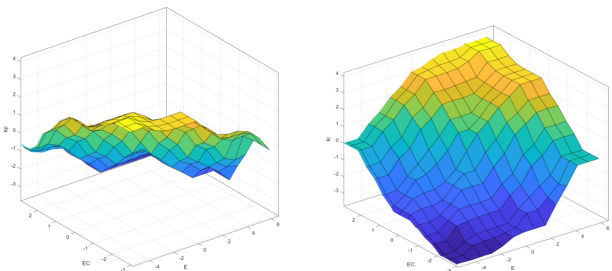


图 17 模糊控制 Kp 和 Ki 的非线性对应关系

是一个通过原点的分片曲面，具有非线性调节规律，但在每一个模糊分段里，都具有分段线性调节规律。

1.5.3 反模糊化

反模糊化是将模糊控制器的输出量转为执行器能接受的精确量。反模糊常用的方法较多，本文使用重心法，公式如下。

u\_n(t) = \frac{\sum u\_i \int A(u\_i)}{\sum \int A(u\_i)} \tag{5}

1.5.4 实验结果分析

在 FuzzyLogic Designer 中设计好模糊控制器后，将模糊控制器导出，文件尾部为 fis，之后开始再在 simulink 上搭建完整控制流程的模型如图 18 所示。

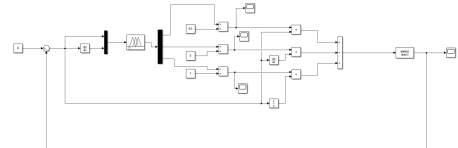


图 18 模糊 PID 控制回路

Simulink 建模结束后，把上文建立的 wangyu-jie.fis 输入到 Fuzzy Logic Controller 中，代码如下，随后观察跟随情况。clc; clear; close all; warning off; addpath(genpath(pwd)); dd=readfis('wangyu-jie.fis');

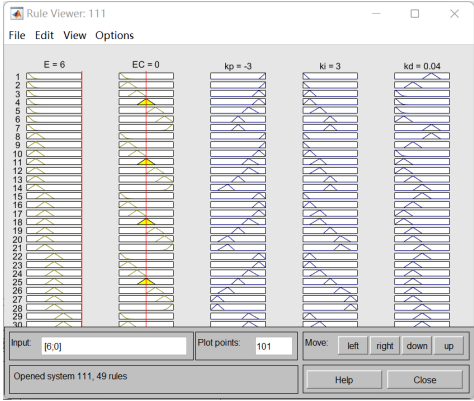


图 19 模糊 PID 调节界面

给定初始值 Kp、Ki、Kd 的初始值，启动模糊控制后，会出现图 19 所示的调节界面。

给定初始的 Kp=290，Kd=10，Ki=1，PID 参数模糊自整定在此基础上进行调整。最终 Kp=300，Ki=12，Kd=0.98。从图 20 可以看到输出信号最终稳定到了 10，无超调量。

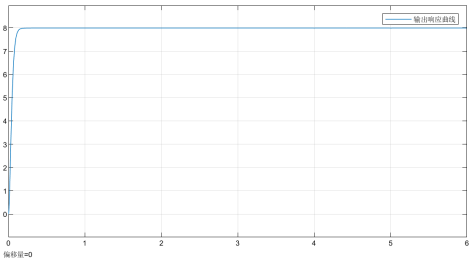


图 20 输出响应曲线

选择 1.4 节中的简单 PID 控制作为对比方法，进行了对比实验。将超调量、调节时间等指标作为衡量电镀槽液温度控制方法的性能指标，测试两种控制方法对突然的温度变化响应的快速性和稳定性，如下表所示。

表 5 模糊 PID 控制性能指标

指标	超调量	调节时间	上升时间	稳态误差
值	0	0.100s	0.07s	0

结合表 1 和表 5 可以发现，无论是简单的 PID 控制还是改进的 PID 参数模糊自整定，都实现了对给定温度的跟踪，但 PID 参数模糊自整定的跟踪效果更好，具体体现为在无超调量的基础上，调节时间和上升时间明显缩短，系统的动态性能更好。

实验中的不足之处是模糊控制器具有一定的主观性，适用的温度范围有限。对电解槽液温度控制而言，不同的温度期望论域需要设计不同的控制器，

如果温度范围较小,隶属度函数和模糊规则的设计需要细致到小区间;如果温度范围较大,则需要扩大模糊子集的区域。在本文中,设计的模糊控制器适合低温范围。

综上,相比于传统方法,本文所用方法能够在更短时间和更稳定的状态下准确达到目标控制温度,且可以有效消除控制过程中的静态误差。使得控制器在面对电镀过程中复杂的非线性干扰时表现出更强的控制能力,从而达到更精确、更稳定的温度控制效果。

## 2 神经网络控制

### 2.1 研究背景

钢铁工业,自工业革命以来,一直是全球经济发展的基石。作为建筑、交通、制造、国防等众多领域不可或缺的材料,钢铁的生产和应用直接关系到一个国家的工业水平和综合国力。随着技术的进步和对可持续发展的追求,钢铁生产过程中的效率和环保性成为行业发展的指标。

在钢铁生产的多个环节中,烧结过程是高炉炼铁工艺提供合格烧结矿的第一过程和能耗第二高的工序,烧结矿的质量直接影响着高炉的生产。因此对烧结过程进行优化和技术创新,提前预测烧结过程质量指标并根据预测结果做出相应调整,可降低生产成本,提高烧结矿的质量,对于钢铁生产的绿色可持续发展具有重要意义。

然而,由于设备操作复杂、环境恶劣以及安全性要求高,直接在现场对控制对象进行阶跃信号测试常常不方便且存在安全隐患,对于大型设备或高温高压环境,直接进行阶跃信号测试可能需要对设备进行停机或者大幅度的操作调整,增加了操作人员的风险和负担。因此,通过仿真测试来模拟和评估系统响应变得至关重要,能够有效减少操作风险,优化控制策略,提升生产效率和质量,确保冶金生产的持续稳定运行和安全性。

### 2.2 研究对象

控制烧结炉内的温度是烧结过程关键的一部分。烧结炉的温度动态特性可以用一个二阶传递函数来描述,这个传递函数表示了考虑到热量传导和热惯性等因素。假设工业炉的温度控制系统的传递函数为:

$$G(s) = \frac{64}{5s^2 + 16s + 64} \quad (6)$$

这个传递函数表示了加热设备和散热设备到炉内温度的关系。实验中将设计一个神经网络控制器,使其可以作为系统动态特性的模型器,用于快速生成系统的响应曲线。这在工程开发和预测性分析中

有广泛应用,例如在设计阶段评估控制策略的效果。在设计新的控制器之前,可以使用神经网络来验证不同控制策略对系统响应的影响,以便在实际应用中进行更有效的调试和优化。

### 2.3 研究原理

本实验中使用的是 BP 神经网络 (Backpropagation Neural Network),它是一种基本的人工神经网络结构,包括输入层、隐藏层和输出层三部分。

(1) 输入层 (Input Layer): 接收输入数据的层。每个输入特征对应输入层的一个节点,输入层节点数目等于输入特征的维度。

(2) 隐藏层 (Hidden Layers): 接收来自输入层或前一隐藏层的输入信号,并进行加权求和后通过激活函数进行非线性变换。隐藏层可以有一个或多个,每个隐藏层包含多个神经元(节点),隐藏层的神经元数目是用户预先设定的参数。

(3) 输出层 (Output Layer): 接收来自最后一个隐藏层的输入,负责生成网络的输出。输出层的节点数通常等于任务中需要预测或分类的类别数目。

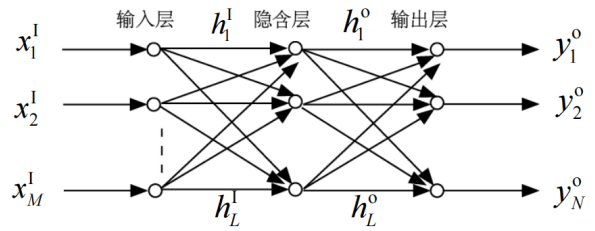


图 21 BP 神经网络示意图

前馈神经网络的原理包括两大部分,前向传播和反向传播。

#### (1) 前向传播

输入数据通过输入层传递至第一个隐藏层,每个节点接收到来自输入层的输入信号,并根据连接权重进行加权求和。

每个隐藏层节点将加权和作为输入,通过激活函数(如 sigmoid、ReLU 等)进行非线性转换,产生输出。

隐藏层的输出再传递到下一个隐藏层,直至传递到输出层。输出层根据任务类型(回归、分类等)决定使用不同的激活函数和损失函数。

#### (2) 反向传播

训练过程中,神经网络通过反向传播算法调整连接权重,以最小化输出与实际目标之间的误差(损失函数)。

反向传播利用梯度下降法沿着损失函数梯度的反方向更新权重,使网络能够逐步优化学习。

## 2.4 实验内容

### 2.4.1 训练神经网络

在训练之前, 首先进行数据准备。用 MATLAB Control System Toolbox 中的 `tf` 函数定义了一个二阶系统模型 `sys`, 其具体参数为给定的阻尼比 `zeta` 和自然频率 `wn`。然后创建了一个时间向量 `t`, 并生成了阶跃输入信号 `u`。使用 `lsim` 函数计算了系统对阶跃输入的响应 `y`, 并将输入数据 `inputs` 设计为 `[u; y']`, 目标输出数据 `targets` 设计为 `y'`, 这样就准备好了用于阶跃跟踪的输入输出数据。

BP 神经网络 (Backpropagation Neural Network) 是一种基于梯度下降优化算法的前馈神经网络 (Feedforward Neural Network), 用于训练和调整神经网络的权重和偏置, 以最小化预测输出与真实目标之间的误差。它主要包括以下几个部分。

(1) 网络结构: BP 神经网络通常由多层神经元组成, 包括输入层、若干隐藏层和输出层。每个神经元接收上一层的输出作为输入, 并通过激活函数 (如 sigmoid 函数、tanh 函数等) 将输入信号转换为输出信号。

(2) 损失函数 (Loss Function): 训练 BP 神经网络的第一步是定义损失函数, 它衡量了网络预测输出与真实目标之间的差异。常见的损失函数包括均方误差 (Mean Squared Error, MSE):

$$E = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (7)$$

其中,  $N$  是训练样本数量,  $y_i$  是真实输出,  $\hat{y}_i$  是神经网络预测输出。

(3) 梯度下降优化算法: BP 神经网络使用梯度下降 (Gradient Descent) 算法来最小化损失函数。梯度下降的基本思想是沿着损失函数的负梯度方向更新网络参数, 以减小损失函数的值。

(4) 反向传播算法 (Backpropagation): 反向传播算法是梯度下降在神经网络中的具体实现。它通过链式法则计算损失函数对每个参数 (权重和偏置) 的梯度, 并根据梯度方向更新参数。以下是反向传播算法的具体步骤:

#### 1) 前向传播 (Forward Propagation)

将输入样本  $X$  输入神经网络, 计算每一层的输出

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)} \quad (8)$$

$$a^{(l)} = f(z^{(l)}) \quad (9)$$

其中,  $W^{(l)}$  是第层的权重矩阵,  $b^{(l)}$  是第  $l$  层的偏置变量,  $f$  是激活函数,  $a^{(l)}$  是第层的激活输出。

#### 2) 计算输出层的误差 (Output Layer Error)

计算输出层的误差

$$\delta^{(l)} = \nabla_a E \cdot f'(z^{(l)}) \quad (10)$$

其中,  $\nabla_a E$  是损失函数对输出  $a^{(l)}$  的梯度,  $f'$  是输出层的激活函数的导数。

#### 3) 反向传播误差 (Backward Error Propagation)

从输出层开始, 向前逐层传播误差。

$$\delta^{(l)} = ((W^{(l+1)})^T \delta^{(l+1)}) \cdot f'(z^{(l)}) \quad (11)$$

其中,  $\delta^{(l)}$  是第  $l$  层的误差,  $W^{(l+1)}$  是从第  $l+1$  层到第  $l$  层的权重矩阵,  $f'$  是第  $l$  层的激活函数的导数。

#### 4) 计算梯度 (Compute Gradients)

计算损失函数对权重和偏置的梯度

$$\frac{\partial E}{\partial W^{(l)}} = \delta^{(l)} (a^{(l-1)})^T \quad (12)$$

$$\frac{\partial E}{\partial b^{(l)}} = \delta^{(l)} \quad (13)$$

#### 5) 参数更新 (Update Parameters)

根据梯度下降算法更新权重和偏置, 其中,  $\eta$  是学习率, 控制参数更新的步长。

$$W^{(l)} \leftarrow W^{(l)} - \eta \frac{\partial E}{\partial W^{(l)}} \quad (14)$$

$$b^{(l)} \leftarrow b^{(l)} - \eta \frac{\partial E}{\partial b^{(l)}} \quad (15)$$

实验中使用 `feedforwardnet` 函数创建了一个具有 10 个隐藏层神经元的前馈神经网络 `net`。使用 `train` 函数对神经网络 `net` 进行训练, 输入数据为 `inputs`, 目标输出数据为 `targets`。神经网络训练状态、最终模型性能和回归曲线分别如图 22-24 所示。

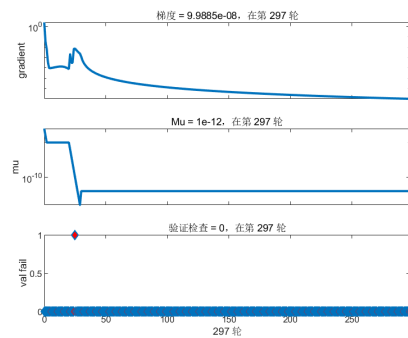


图 22 神经网络训练状态

从上图可以看出, 训练效果良好, 均方误差最后趋于 0, 回归曲线的拟合度也较好。



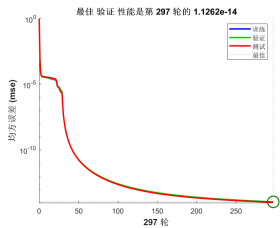


图 23 模型性能

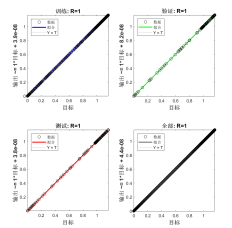


图 24 回归曲线

#### 2.4.2 阶跃信号跟踪测试

训练好模型后，对新的二阶系统进行阶跃信号跟踪测试，实验中设置工业炉温控制对象的传递函数为：

$$G(s) = \frac{64}{5s^2 + 16s + 64} \quad (16)$$

实际单位阶跃响应与神经网络预测值的对比图如下所示。

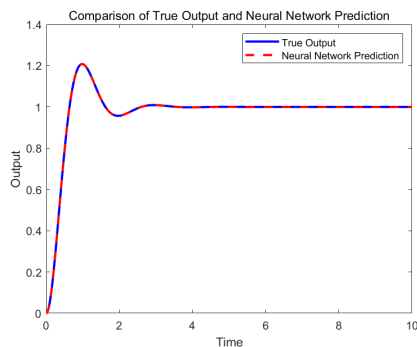


图 25 真实值与预测值对比图

计算预测值与真实值的 MSE 和 MAE，结果如下。

表 6 模型性能

均方误差 (MSE)	平均绝对值误差 (MAE)
2.4142e-11	7.9137e-07

从图 25 和表 6 中可以看到，神经网络的预测性能良好，误差小，几乎与真实曲线相吻合。

综上，实验中成功设计出一个神经网络模型，使其可以作为工业炉二阶系统动态特性的模型器，用于快速生成系统的响应曲线。这在工程开发和预测性分析中有广泛应用。在设计新的控制器之前，可以使用该神经网络来验证不同控制策略对系统响应的影响，以便在实际应用中进行更有效的调试和优化。

### 3 专家系统控制

#### 3.1 研究背景

专家系统是一种基于知识的人工智能系统，它模拟了人类专家在特定领域中的推理和问题解决能力。专家系统的核心原理是基于知识库和推理引擎，其中产生式表示法是一种常用的知识表示方法。简而言之，专家系统是一种模拟人类专家解决领域问题的计算机程序系统。专家系统的基本功能取决于它所含有的知识，因此，有时也把专家系统称为基于知识的系统。

专家控制系统 (expert control system, ECS) 已广泛应用于故障诊断、工业设计和过程控制，为解决工业控制难题提供一种新的方法，是实现工业过程控制的重要技术。专家系统控制作为人工智能领域的重要分支，利用专家的经验 and 知识形成的规则库，能够模拟和实现复杂的决策过程，广泛应用于工业自动化、医疗诊断、金融风险管理等领域。其背景在于提高系统智能化、自动化水平，解决传统控制方法难以处理的复杂、非线性问题，从而提升生产效率、降低成本，并推动技术创新和产业发展。

#### 3.2 研究对象

在工业生产过程中，温度控制对于确保产品质量和生产效率至关重要。工业炉用于加热、熔化或热处理材料，例如在冶金、陶瓷制造和半导体加工等行业中。温度控制系统需要保持炉内温度在设定的目标值，以确保产品在最佳条件下处理。工业炉温控制系统使用专家系统控制具有多方面的优点和重要意义，主要体现在以下几个方面。

##### (1) 自适应性和灵活性

专家系统控制能够根据实时监测的数据和预设的规则，自动调整控制策略和参数，以应对不同的工作状态和环境变化。在工业炉温控制中，温度的变化可能受多种因素影响，如物料负荷、环境温度和生产速度等。专家系统可以根据这些因素动态调整控制策略，使得系统能够快速响应变化，保持稳定的温度控制效果。

##### (2) 知识的形式化和重复利用

专家系统将工业炉温度控制的知识形式化为规则集合或逻辑推理方式，这些规则包括了工艺工程师的经验 and 专业知识。这些规则不仅使得控制过程更加智能化，还可以在不同的工业炉应用中重复利用，节省了系统设计和调试的时间成本。

##### (3) 处理复杂的非线性和动态特性

工业炉温度控制系统通常具有复杂的非线性和动态特性，例如传热过程中的惯性、滞后和非线性特征。专家系统能够通过规则匹配和逻辑推理，有

效地处理这些复杂性,使得控制器能够在不同工况下保持良好的控制性能。

#### (4) 减少人为误差和提高一致性

由于专家系统控制是基于预设的规则和知识库进行决策的,因此减少了人为因素对控制决策的影响。这种自动化的控制方法能够提高控制过程的一致性和稳定性,减少由于人为误差导致的生产异常和质量问题。

综上所述,工业炉温控制系统应用专家系统控制具有明显的优势,能够有效应对复杂的工业环境和控制需求,提高生产过程的智能化、自动化水平,从而增强企业的竞争力和可持续发展能力。

### 3.3 研究原理

工业炉的温度动态特性可以用一个三阶传递函数来描述,这个传递函数表示了加热元件(如电热管)对温度的影响,包括传热延迟和温度惯性。

假设工业炉的温度控制系统的传递函数为:

$$G(s) = \frac{523542}{s^3 + 87.35s^2 + 10470s + 1} \quad (17)$$

这个传递函数表示了加热元件输入功率到炉内温度的关系,其中考虑了系统的惯性和延迟。实验中将使用专家系统 PID 控制器来控制工业炉的温度,使其能够快速响应温度设定值的变化,并保持温度稳定,减少超调和稳态误差。专家控制以专家系统为背景,其结构包括:

(1) 知识库:是领域知识的存储器。

(2) 数据库:用于存放当前处理对象用户提供的数据和推理得到的中间结果。

(3) 推理机:用于控制和协调整个专家系统的工作,它根据当前的输入数据,再利用知识库的知识,按一定推理策略去处理解决当前的问题。

(4) 解释机构:为用户解释推理结果,以使用户了解推理过程。

(5) 知识获取机构:通过设计一组程序,为修改知识库中原有的知识和扩充新知识提供手段,包括删除原有知识,将向专家获取的新知识加入到知识库。知识获取被称为专家系统的瓶颈。

专家系统 PID 控制器通过规则调整控制器参数和输出,以适应不同的工作状态,提高系统的响应速度和稳定性。控制器需要实时监测温度误差及其变化,并根据预设的控制规则调整加热元件的功率输出。

实验一共包括四大部分,初始化系统和控制器参数、实时控制过程、监控和调整,绘制和分析控制效果。具体步骤如下:

#### (1) 初始化系统和控制器参数

设定初始 PID 参数 ( $k_p$ ,  $k_i$ ,  $k_d$ ) 和控制规则,

初始化系统状态和变量。

#### (2) 实时控制过程

采样工业炉的当前温度,并计算与目标温度的误差;使用 PID 控制算法计算控制输入(加热元件的功率);根据专家系统规则调整控制输入,以应对不同的误差情况(例如,当误差过大时,减少控制输入以避免过热);更新系统状态和变量,准备下一次控制计算。

#### (3) 监控和调整

实时监控温度变化,并在必要时调整 PID 参数和控制规则,以适应生产过程中可能发生的变化。

#### (4) 绘制和分析控制效果

绘制温度跟踪曲线,分析系统的响应时间、超调量和稳态误差;绘制误差曲线,分析系统的控制精度。

### 3.4 实验内容

本文工业炉的温度控制系统的传递函数为:

$$G(s) = \frac{523542}{s^3 + 87.35s^2 + 10470s + 1} \quad (18)$$

实验中在 Matlab 中进行,设置采样时间间隔为 1ms,并用 c2d 把中  $G(s)$  离散化。采用专家 PID 设计控制器,在仿真过程中,误差上限为 0.001,程序中 5 条规则与控制算法的五种情况相对应。

(1) 当  $|e(k)| > M1$  时,说明误差的绝对值已经很大。不论误差变化趋势如何,都应考虑控制器的输出按定值输出,以达到迅速调整误差,使误差绝对值以最大速度减小,同时避免超调。此时,它相当于实施开环控制。

(2) 当  $e(k) e(k) > 0$  或  $e(k) = 0$  时,说明误差在朝误差绝对值增大方向变化,或误差为某一常值,未发生变化。

- 如果  $|e(k)| > M2$ ,说明误差较大,可考虑由控制器实施较强的控制作用,使误差绝对值朝减小方向变化,迅速减小误差的绝对值。

- 如果  $|e(k)| < M2$ ,说明尽管误差朝绝对值增大方向变化,但误差绝对值本身并不是很大,可考虑实施一般的控制作用,扭转误差的变化趋势,使其朝误差绝对值减小方向变化。

(3) 当  $e(k) e(k) < 0$ ,  $e(k) e(k-1) > 0$  或者  $e(k) = 0$  时,说明误差的绝对值朝减小的方向变化,或者已经达到平衡状态。此时,可考虑采取保持控制器输出不变。

(4) 当  $e(k) e(k) < 0$ ,  $e(k) e(k-1) < 0$  时,说明误差处于极值状态。如果此时误差的绝对值较大,即  $|e(k)| > M2$ ,可考虑实施较强的控制作用。如果此时

误差的绝对值较小, 即 $|e(k)| < M_2$ , 可考虑实施较弱的控制作用。

(5) 当 $|e(k)| < e$  (精度) 时, 说明误差的绝对值很小, 此时加入积分环节, 减小稳态误差。以上各式中,  $e_m(k)$  为误差  $e$  的第  $k$  个极值;  $u(k)$  为第  $k$  次控制器的输出;  $u(k-1)$  为第  $k-1$  次控制器的输出;  $k_1$  为增益放大系数,  $k > 1$ ;  $k_2$  为抑制系数,  $0 < k_2 < 1$ ;  $M_1, M_2$  为设定的误差界限,  $M_1 > M_2 > 0$ ;  $k$  为控制周期的序号 (自然数);  $e$  为任意小的正实数。

设计完专家系统控制后, 进行仿真, 给定阶跃信号作为输入, 跟踪曲线和误差曲线如下图所示。

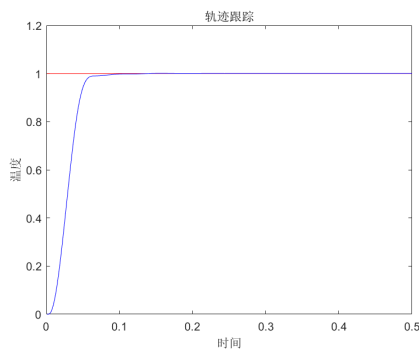


图 26 轨迹跟踪曲线

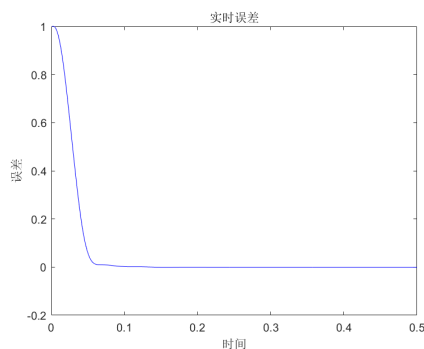


图 27 实时误差曲线

表 7 专家系统控制性能指标

指标	超调量	调节时间	上升时间	稳态误差
值	0	0.0520	0.033s	0

从上表可以发现, 专家系统 PID 温度控制实现了对给定温度的跟踪, 无超调量, 调节时间和上升时间也很短, 系统的动态性能和稳态性能都不错。

## 4 总结与感悟

### 4.1 总结

本文的《智能控制》课程报告共分为模糊控制、神经网络控制和专家系统控制三大部分, 各自独立, 有不同的控制对象。实验内容与智能控制课程的课内理论知识关系密切, 大大加深了我对理论部分的理解。

在模糊控制章节, 实验分为两部分。第一个是设计电镀槽液温度 PID 控制器: 以固定幅值的信号作为激励信号, 表示电镀槽液期望的温度, 通过 PID 控制器实现对信号的跟踪控制, 使得电镀槽液达到期望值。第二个是设计电镀槽液温度模糊 PID 控制器, 任务是找出 PID 的三个参数  $K_p$ 、 $K_i$ 、 $K_d$  与误差  $E$  和误差变化率  $E_c$  之间的模糊关系, 在运行中不断检测  $E$  和  $E_c$ , 根据确定的模糊控制规则来对三个参数进行在线调整。通过 MATLAB 自带的 fuzzy 工具箱, 设计不同语言变量的隶属度函数和对应的规则, 最后在 Simulink 中运行结果, 将两种控制策略的效果进行对比分析, 最后分析得出设计的模糊控制器性能良好。

在神经网络控制章节, 成功设计出一个神经网络模型, 使其可以作为工业炉二阶系统动态特性的模型器, 用于快速生成系统的响应曲线。这在工程开发和预测性分析中有广泛应用。在设计新的控制器之前, 可以使用该神经网络来验证不同控制策略对系统响应的影响, 以便在实际应用中进行更有效的调试和优化。在最后的阶跃信号跟踪测试中模型表现优秀。

在专家系统控制章节, 回想起在上学期做过一个根据动物特征识别动物的专家系统, 但与本次实验所做的有些不同, 其差别在于动物识别系统中的知识库是用数组形式存储的, 而本次实验中的知识库基于对误差的分析, 变得更加复杂。本文设计了炉温专家系统 PID 控制器, 通过规则调整控制器参数和输出, 以适应不同的工作状态, 提高系统的响应速度和稳定性。控制器需要实时监测温度误差及其变化, 并根据预设的控制规则调整加热元件的功率输出。一共包括四部分, 初始化系统和控制器参数、实时控制过程、监控和调整, 绘制和分析控制效果。

综上所述, 实验内容包含了智能控制课程几乎所有重点知识, 对智能控制方法相关知识的理解和应用有很大的帮助。

### 4.2 感悟

安老师的《智能控制》课程让我受益匪浅, 与以往的课程非常不同, 安老师上课风格幽默风趣, 生

动形象,并不是一味地填鸭式教学和读ppt,而是真正注重这门课的重点知识,以及培养我们的主观能动性。

我印象深刻是上课期间安老师经常会提到别的课程的知识,例如自控原理、过控、运控等。宏观上,老师都会把之前的知识串联起来,使其环环相扣,避免我们稀里糊涂地学完,却不知道学了什么。微观上,安老师偶尔会聚焦到某一个小方法上,比如问我们为什么进行拉氏变换、 $z$ 变化等,很多时候我们知道这个小知识点,也知道怎么用,但是不知道它本质是用来干什么。就像安老师说的,你一定要知道自己在做什么和为什么要这么做。细细想来,的确如此,在前几个学期我基本上每学完和考完一门课就把那些知识抛之脑后了,以至于很多时候不知道自己学了什么,也不去思考每门课之间的关联。这样迷迷糊糊地学习并没有很大的成效。在之后,我将努力践行安老师教给我的理念,思考不同课程、不同章节、不同方法之间的关联,将知识串联起来。

此外,上课期间我的自学能力也得到了提升,无论是安老师指导的PBL,还是Latex版本的报告要求,都使我的自学能力得到了进一步的提升。

行文至此,《智能控制》课程也迎来结束。未来在控制科学与工程的道路,课程蕴含的知识和老师认真严谨的工作态度永远会引领我前行,带给我启迪!

## 致谢

感谢安老师在《智能控制》课程中的教导,不论是幽默风趣的上课风格,还是灵活轻松的学习氛围,亦或是自学能力的着重培养,都是这门课程的亮点,也带给我非常多的启发。

## References

- 王丽娜,曹建安,王莲花,等. 基于模糊控制的温室气候控制器设计[J]. 中国农机化学报, 2024, 45(07): 75-80. DOI: 10.13733/j.jcam. issn. 2095-5553. 2024. 07. 012.
- 钱贾伟,李梦婕,成梁. 基于模糊自整定PID的水箱液位控制[J]. 青海电力, 2024, 43(02): 53-57. DOI: 10.15919/j.cnki.qhep. 2024. 02. 009.
- 李钢. 基于模糊PID控制的恒压供水系统的研究[J]. 电子设计工程, 2024, 32(13): 83-87. DOI: 10.14022/j.issn1674-6236. 2024. 13. 018.
- 田亚平,杨江辉,王瑞邦,等. 弧齿锥齿轮动力特性分析及其神经网络控制[J]. 振动与冲击, 2024, 43(12): 166-172. DOI: 10.13465/j.cnki.jvs. 2024. 12. 019.
- 陈志勇,李攀,叶明旭,等. 自动驾驶电动车辆基于参数预测的径向基函数神经网络自适应控制[J]. 中国机械工程, 2024, 35(06): 982-992.
- 朱凤磊,张立新,胡雪,等. 基于BP神经网络PID的节水灌溉施肥系统研究[J]. 农机化研究, 2024, 46(11): 53-58. DOI: 10.13427/j.cnki.njyi. 2024. 11. 007.
- 杨盛泉,龚晨,杨洪波,等. 电瓷梭式窑PID专家系统的设计与实现[J]. 西安工业大学学报, 2019, 39(01): 86-92. DOI: 10.16185/j.jxatu.edu.cn. 2019. 01. 014.
- 付雪奇. 输液配药机器人设计及其专家系统控制[D]. 哈尔滨工业大学, 2015.



**王愉杰** 中国地质大学(武汉)自动化学院自动化专业本科生. 主要研究兴趣为机器学习.

E-mail: wangyujie@cug.edu.cn

(WANG Yu-Jie Undergraduate, Automation major, School of Automation, China University of Geosciences (Wuhan). His research interest covers

machine learning.)

## 附录

- 获取模糊控制器文件的属性

代码: % 获取 FIS 的详细属性

```
fis_info = getfis(fis);
disp(fis_info)
命令行显示:
fis_info = getfis(fis);
disp(fis_info)
name: '111'
type: 'mamdani'
andMethod: 'min'
orMethod: 'max'
defuzzMethod: 'centroid'
impMethod: 'min'
aggMethod: 'max'
input: [1×2 struct] output: [1×3 struct]
rule: [1×49 struct]
```

- 显示 FIS 文件的模糊规则

代码:

% 显示 FIS 的模糊规则

showrule(fis)

命令行显示:

ans =

49×71 char 数组

'1. If (E is NB) and (EC is NB) then (kp is PB)(ki is NB)(kd is PS) (1) '

'2. If (E is NB) and (EC is NM) then (kp is PB)(ki is NB)(kd is NS) (1) '



- '3. If (E is NB) and (EC is NS) then (kp is PM)(ki is NM)(kd is NB) (1) '
- '4. If (E is NB) and (EC is ZO) then (kp is PM)(ki is NM)(kd is NB) (1) '
- '5. If (E is NB) and (EC is PS) then (kp is PS)(ki is NS)(kd is NB) (1) '
- '6. If (E is NB) and (EC is PM) then (kp is ZO)(ki is ZO)(kd is NM) (1) '
- '7. If (E is NB) and (EC is PB) then (kp is ZO)(ki is ZO)(kd is PS) (1) '
- '8. If (E is NM) and (EC is NB) then (kp is PB)(ki is NB)(kd is PS) (1) '
- '9. If (E is NM) and (EC is NM) then (kp is PB)(ki is NB)(kd is NS) (1) '
- '10. If (E is NM) and (EC is NS) then (kp is PM)(ki is NM)(kd is NB) (1) '
- '11. If (E is NM) and (EC is ZO) then (kp is PS)(ki is NS)(kd is NM) (1) '
- '12. If (E is NM) and (EC is PS) then (kp is PS)(ki is NS)(kd is NM) (1) '
- '13. If (E is NM) and (EC is PM) then (kp is ZO)(ki is ZO)(kd is NS) (1) '
- '14. If (E is NM) and (EC is PB) then (kp is NS)(ki is ZO)(kd is ZO) (1) '
- '15. If (E is NS) and (EC is NB) then (kp is PM)(ki is NB)(kd is ZO) (1) '
- '16. If (E is NS) and (EC is NM) then (kp is PM)(ki is NM)(kd is NS) (1) '
- '17. If (E is NS) and (EC is NS) then (kp is PM)(ki is NS)(kd is NM) (1) '
- '18. If (E is NS) and (EC is ZO) then (kp is PS)(ki is NS)(kd is NM) (1) '
- '19. If (E is NS) and (EC is PS) then (kp is ZO)(ki is ZO)(kd is NS) (1) '
- '20. If (E is NS) and (EC is PM) then (kp is NS)(ki is PS)(kd is NS) (1) '
- '21. If (E is NS) and (EC is PB) then (kp is NS)(ki is PS)(kd is ZO) (1) '
- '22. If (E is ZO) and (EC is NB) then (kp is PM)(ki is NM)(kd is ZO) (1) '
- '23. If (E is ZO) and (EC is NM) then (kp is PM)(ki is NM)(kd is NS) (1) '
- '24. If (E is ZO) and (EC is NS) then (kp is PS)(ki is NS)(kd is NS) (1) '
- '25. If (E is ZO) and (EC is ZO) then (kp is ZO)(ki is ZO)(kd is NS) (1) '
- '26. If (E is ZO) and (EC is PS) then (kp is NS)(ki is PS)(kd is NS) (1) '
- '27. If (E is ZO) and (EC is PM) then (kp is NM)(ki is PM)(kd is NS) (1) '
- '28. If (E is ZO) and (EC is PB) then (kp is NM)(ki is PM)(kd is ZO) (1) '
- '29. If (E is PS) and (EC is NB) then (kp is PS)(ki is NM)(kd is ZO) (1) '
- '30. If (E is PS) and (EC is NM) then (kp is PS)(ki is NS)(kd is ZO) (1) '
- '31. If (E is PS) and (EC is NS) then (kp is ZO)(ki is ZO)(kd is ZO) (1) '
- '32. If (E is PS) and (EC is ZO) then (kp is NS)(ki is PS)(kd is ZO) (1) '
- '33. If (E is PS) and (EC is PS) then (kp is NS)(ki is PS)(kd is ZO) (1) '
- '34. If (E is PS) and (EC is PM) then (kp is NM)(ki is PM)(kd is ZO) (1) '
- '35. If (E is PS) and (EC is PB) then (kp is NM)(ki is PB)(kd is ZO) (1) '
- '36. If (E is PM) and (EC is NB) then (kp is PS)(ki is ZO)(kd is PB) (1) '
- '37. If (E is PM) and (EC is NM) then (kp is ZO)(ki is ZO)(kd is NS) (1) '
- '38. If (E is PM) and (EC is NS) then (kp is NS)(ki is PS)(kd is PS) (1) '
- '39. If (E is PM) and (EC is ZO) then (kp is NM)(ki is PM)(kd is PS) (1) '
- '40. If (E is PM) and (EC is PS) then (kp is NM)(ki is PM)(kd is PS) (1) '
- '41. If (E is PM) and (EC is PM) then (kp is NM)(ki is PB)(kd is PS) (1) '
- '42. If (E is PM) and (EC is PB) then (kp is NB)(ki is PB)(kd is PB) (1) '
- '43. If (E is PB) and (EC is NB) then (kp is ZO)(ki is ZO)(kd is PB) (1) '
- '44. If (E is PB) and (EC is NM) then (kp is ZO)(ki is ZO)(kd is PM) (1) '
- '45. If (E is PB) and (EC is NS) then (kp is NM)(ki is PS)(kd is PM) (1) '
- '46. If (E is PB) and (EC is ZO) then (kp is NM)(ki is PM)(kd is PM) (1) '
- '47. If (E is PB) and (EC is PS) then (kp is NM)(ki is PM)(kd is PS) (1) '
- '48. If (E is PB) and (EC is PM) then (kp is NB)(ki is PB)(kd is PS) (1) '
- '49. If (E is PB) and (EC is PB) then (kp is NB)(ki is PB)(kd is PB) (1) '
- 输出输入变量的隶属函数设置内容
- 代码:

```

% 2. 输出输入变量的隶属函数设置内容
for i = 1:numel(fis.Inputs)
    fprintf('Input %d: %s\n', i, fis.Inputs(i).Name);
    for j = 1:numel(fis.Inputs(i).MembershipFunctions)
        mf = fis.Inputs(i).MembershipFunctions(j);
        fprintf(' MF%d: %s, Type: %s, Params: [%s]\n', ... j, mf.Name, mf.Type, num2str(mf.Parameters));
    end
end
% 3. 输出输出变量的隶属函数设置内容
for i = 1:numel(fis.Outputs)
    fprintf('Output %d: %s\n', i, fis.Outputs(i).Name);
    for j = 1:numel(fis.Outputs(i).MembershipFunctions)
        mf = fis.Outputs(i).MembershipFunctions(j);
        fprintf(' MF%d: %s, Type: %s, Params: [%s]\n', ... j, mf.Name, mf.Type, num2str(mf.Parameters));
    end
end
命令行显示:
Input 1: E
MF1: NB, Type: gaussmf, Params: [0.8493 -6]
MF2: NM, Type: trimf, Params: [-6 -4 -2]
MF3: NS, Type: trimf, Params: [-4 -2 0]
MF4: ZO, Type: trimf, Params: [-2 0 2]
MF5: PS, Type: trimf, Params: [0 2 4]
MF6: PM, Type: trimf, Params: [2 4 6]
MF7: PB, Type: gaussmf, Params: [0.8493 6]
Input 2: EC
MF1: NB, Type: gaussmf, Params: [0.4247 -3]
MF2: NM, Type: trimf, Params: [-3 -2 -1]
MF3: NS, Type: trimf, Params: [-2 -1 0]
MF4: ZO, Type: trimf, Params: [-1 0 1]
MF5: PS, Type: trimf, Params: [0 1 2]
MF6: PM, Type: trimf, Params: [1 2 3]
MF7: PB, Type: gaussmf, Params: [0.4254 3]
Output 1: kp
MF1: NB, Type: trimf, Params: [-4 -4 -3]

```

```

MF2: NM, Type: trimf, Params: [-4 -3 -2]
MF3: NS, Type: trimf, Params: [-3 -0.5]
MF4: ZO, Type: trimf, Params: [-1 0 1]
MF5: PS, Type: trimf, Params: [0.5 1.5 3]
MF6: PM, Type: trimf, Params: [2 3 4]
MF7: PB, Type: trimf, Params: [3 4 4]
Output 2: ki
MF1: NB, Type: trimf, Params: [-4 -4 -3]
MF2: NM, Type: trimf, Params: [-4 -3 -2]
MF3: NS, Type: trimf, Params: [-3 -0.5]
MF4: ZO, Type: trimf, Params: [-1 0 1]
MF5: PS, Type: trimf, Params: [0.5 1.5 3]
MF6: PM, Type: trimf, Params: [2 3 4]
MF7: PB, Type: trimf, Params: [3 4 4]
Output 3: kd
MF1: NB, Type: gaussmf, Params: [0.008493 -0.06]
MF2: NM, Type: trimf, Params: [-0.06 -0.04 -0.02]
MF3: NS, Type: trimf, Params: [-0.04 -0.02 3.469e-18]
MF4: ZO, Type: trimf, Params: [-0.02 6.505e-19 0.02]
MF5: PS, Type: trimf, Params: [3.469e-18 0.02 0.04]
MF6: PM, Type: trimf, Params: [0.02 0.04 0.06]
MF7: PB, Type: gaussmf, Params: [0.008493 0.06]

```

#### • 神经网络训练

```

clc;clear
s = tf('s');
zeta = 0.5; % 阻尼比
wn = 1; % 自然频率
sys = wn2 / (s2 + 2*zeta*wn*s + wn2);
t = 0:0.01:10; % 时间向量
u = ones(size(t)); % 阶跃输入
[y, ~] = lsim(sys, u, t); % 系统的阶跃响应
% 输入数据为系统响应和输入信号
inputs = [u; y]; % 输入数据的维度应为 2 x N
targets = y; % 目标输出数据应为 1 x N
net = feedforwardnet(10); % 创建一个具有 10 个隐藏层神经元的前馈神经网络

```

```

net = train(net, inputs, targets); % 使用生成
的训练数据进行训练
save('trained_net.mat', 'net');
• 阶跃响应测试
clc;
clear;
% 加载训练好的神经网络模型
load('trained_net.mat', 'net');
% 定义待测试的二阶系统模型（如果不同于训
练时的模型）
s = tf('s');
zeta = 1; % 阻尼比
wn = 8; % 自然频率
sys = wn^2 / (5*s^2 + 2*zeta*wn*s + wn^2);
% 生成阶跃输入信号
t = 0:0.01:10; % 时间向量
u = ones(size(t)); % 阶跃输入
% 获取系统的阶跃响应（作为真实输出，用于
比较）
[y_true, ~] = lsim(sys, u, t);
% 使用神经网络进行预测
inputs = [u; y_true]; % 输入数据的维度应为
2 x N
y_pred = net(inputs); % 使用神经网络进行预
测，y_pred 为预测的输出
% 绘制真实输出和神经网络预测输出的比较
figure;
plot(t, y_true, 'b', 'LineWidth', 2); % 真实输
出
hold on;
plot(t, y_pred, 'r-', 'LineWidth', 2); % 神经网
络预测输出
xlabel('Time');
ylabel('Output');
legend('True Output', 'Neural Network Predic-
tion');
title('Comparison of True Output and Neural
Network Prediction');
% 可以进一步分析误差等其他指标
error = y_true - y_pred; % 计算误差
mse = (sum(error.^2))/1001
mean_error = mean(abs(error)); % 平均绝对
误差
disp(['Mean absolute error: ',
num2str(mean_error)]);
• 专家系统
clc;clear

```

```

% 采样时间间隔
ts = 0.001;
% 三阶传递函数
sys = tf(5.23542e005, [1, 87.35, 1.047e004, 1]);
% 离散化传递函数
dsys = c2d(sys, ts, 'z');
% 建立模型
[num, den] = tfdata(dsys, 'v');
% 设置对象
u_1 = 0; u_2 = 0; u_3 = 0;
y_1 = 0; y_2 = 0; y_3 = 0;
x = [0, 0, 0]';
x2_1 = 0;
% 初始化
kp = 0.6;
ki = 0.03;
kd = 0.01;
error_1 = 0;
for k = 1:1:500
time(k) = k * ts;
r(k) = 1.0; % 目标轨迹
u(k) = kp * x(1) + kd * x(2) + ki * x(3); %
PID 控制器
% 专家控制规则
% 规则 1
if abs(x(1)) > 0.8 % Rule1: 打开控制规则
u(k) = 0.45;
elseif abs(x(1)) > 0.40
u(k) = 0.40;
elseif abs(x(1)) > 0.20
u(k) = 0.12;
elseif abs(x(1)) > 0.01
u(k) = 0.10;
end
% 规则 2
if x(1) * x(2) > 0 || (x(2) == 0) % Rule2
if abs(x(1)) >= 0.05
u(k) = u_1 + 2 * kp * x(1);
else
u(k) = u_1 + 0.4 * kp * x(1);
end
end
% 规则 3
if (x(1) * x(2) < 0 & x(2) * x2_1 > 0) || (x(1)
== 0) % Rule3
u(k) = u(k);
end
% 规则 4

```

```

if x(1) * x(2) < 0 & x(2) * x2_1 < 0 % Rule4
if abs(x(1)) >= 0.05
u(k) = u_1 + 2 * kp * error_1;
else
u(k) = u_1 + 0.6 * kp * error_1;
end
end
% 规则 5
if abs(x(1)) <= 0.001 % Rule5: 积分分离 PI
控制
u(k) = 0.5 * x(1) + 0.010 * x(3);
end
% 限制控制器的输出
if u(k) >= 10
u(k) = 10;
end
if u(k) <= -10
u(k) = -10;
end
% 线性模型
y(k) = -den(2) * y_1 - den(3) * y_2 - den(4)
* y_3 + num(1) * u(k) + num(2) * u_1 + num(3)
* u_2 + num(4) * u_3;
error(k) = r(k) - y(k);
% 反馈
% 修改每次循环返回的参数
u_3 = u_2; u_2 = u_1; u_1 = u(k);

```

```

y_3 = y_2; y_2 = y_1; y_1 = y(k);
x(1) = error(k); % 计算 P
x2_1 = x(2);
x(2) = (error(k) - error_1) / ts; % 计算 D
x(3) = x(3) + error(k) * ts; % 计算 I
error_1 = error(k);
end
% 输出
figure(1);
plot(time, r, 'r', time, y, 'b');
xlabel('时间');
ylabel('温度');
title('轨迹跟踪');
figure(2);
plot(time, r - y, 'b');
xlabel('时间');
ylabel('误差');
title('实时误差');
%
% 指定要查找的 y 值
y_to_find = 0.95;
% 使用线性插值查找对应的 x 值
x_value = interp1(y, time, y_to_find, 'linear');
disp(['对应于 y = ', num2str(y_to_find), ' 的
x 值是 ', num2str(x_value)]);

```