# 现代控制工程课程实验报告

姓名：刘志成 班级：人工智能82班 学号：2183511589

实验时间：2020.11.20

**硬盘读/写碰头组件控制设计与仿真**

## 实验目的

（1）理解热处理温度控制的过程及其性能指标；

（2）通过实验，了解简单的单神经元PID控制。

## 实验原理

1. 单神经元PID控制

单神经元作为构成神经网络的基本单位，具有自学习和自适应能力，且结构简单响应速度快，单神经元PID就是将单个神经元与PID控制器结合起来，一定程度上解决了传统PID调节器不易在线实时整定参数，难以对复杂过程和时变系统参数进行有效控制等问题。单神经元PID控制的结构框图如下图所示：

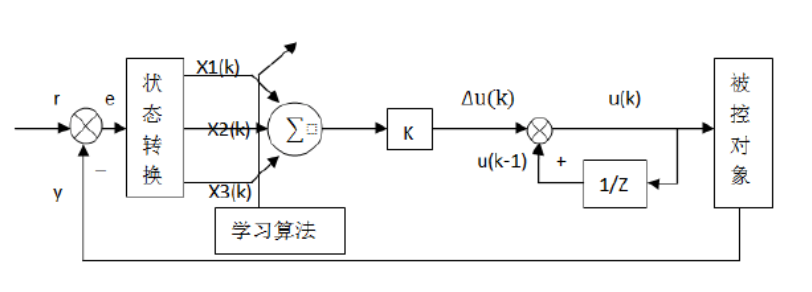


图1 单神经元自适应控制器原理图

其中输入为,输出为，误差为，控制信号为。

设神经元的输入层有三个节点，分别为：

输出节点为输出的控制量的变化：

其中，为比例系数，为加权系数。采用规范化的学习算法，有：

其中为学习率。

1. 单神经元PID控制的参数调整方法

关于权值的初始值选择，可以通过下述方法计算：

其中, Kp, Ki, Kd为PID控制器参数的最优整定值。本次实验可以采用Ziegler-Nichols整定的方法来确定Kp, Ki, Kd。

关于学习率的调整方法，有以下规则：

规则一 :对于K,若系统超调过大,则减小,若过渡过程时间太长,则增大。K的调节改变对响应影响很大，对于大延时系统,应该足够小来满足稳定性和减少震荡。

规则二 :对于,若过程从超调趋向平稳时间太长,则可增大;若超调迅速下降至低于给定值,此后,又缓慢上升稳态时间太长,则可减少,增加积分项作用;对于大延时系统,为减少超调,,应该适当大些。

1. 实验内容

如下图所示是热处理温度控制的示意图。在生产和工业中 ,对温度的控制是十分普遍的。但由于温度对象种类繁多,各自内在机理不同,数学模型存在一定的局限性,而且普遍具有时间常数大,纯滞后时间长,时变性较明显等特点,本次实验将利用单神经元PID控制来控制温度系统。探究单神经元PID控制在温度控制系统的优缺点。



在实际控制过程中系统是多变量的，但在这里我们将其简化为单输入单输出的系统进行研究。

1. 建立离散化模型

根据已有的经验,普通电加热炉可以得知被控对象可近似为一阶加滞后环节。本次实验种设被控对象的传递函数为：，若取采样时间为，利用Z变换，建立离散化模型，即建立被控对象输出Y和控制量U的差分方程。

1. 单神经元控制系统仿真

根据上述实验原理和建立的离散化模型，编写matlab程序，仿真单神经元PID温度控制系统，其中被控对象为上述普通电加热炉。程序可视化要求有，P,I,D对应权重的变化随时间的曲线，输出Y随时间变换的曲线，控制量U随时间变化的曲线。

设控制量初始值为1，输入值,权重初始值按照实验原理中计算，已知对于系统,Ziegler-Nichols整定的方法公式如下：

结合采样时间Ts,有：

（3）参数调节

根据学习率调整规则，调节学习率和比例系数，使系统的超调量小于20%，,。

## 实验结果

MATLAB代码如下：

clc;

clear;

% 超参数

param\_K = 0.01;

param\_gamma\_p = 0.9;

param\_gamma\_i = 0.00001;

param\_gamma\_d = 0.3;

iteration = 10000;

K = 94.75;

tao = 530;

T = 2900.6;

Ts = 10;

Kp = 1.2\*T/K/tao;

Ki = Kp\*Ts/(2\*tao);

Kd = Kp\*(0.5\*tao)/Ts;

w1\_0 = Kp/(Kp+Ki+Kd);

w2\_0 = Ki/(Kp+Ki+Kd);

w3\_0 = Kd/(Kp+Ki+Kd);

disp(w1\_0);disp(w2\_0);disp(w3\_0)

w1(1) = w1\_0;

w2(1) = w2\_0;

w3(1) = w3\_0;

r = 80;

u(1) = 1;

y(1) = 0;

for k = 1:iteration

e(k) = r - y(k);

if k == 1

x1(k) = e(k);

x2(k) = e(k);

x3(k) = e(k);

elseif k == 2

x1(k) = e(k) - e(k-1);

x2(k) = e(k);

x3(k) = e(k) - 2\*e(k-1);

elseif k > 2

x1(k) = e(k) - e(k-1);

x2(k) = e(k);

x3(k) = e(k) - 2\*e(k-1) + e(k-2);

end

deltau(k) = param\_K \* (w1(k)\*x1(k) + w2(k)\*x2(k) + w3(k)\*x3(k))/(abs(w1(k)) + abs(w2(k)) + abs(w3(k)));

if k == 1

u(k) = deltau(k) + 1;

elseif k > 1

u(k) = deltau(k) + u(k-1);

end

if k < 53

y(k+1) = 0.9966\*y(k) + 0.3221;

elseif k >= 53

y(k+1) = 0.9966\*y(k) + 0.3221\*u(k-52);

end

w1(k+1) = w1(k)+param\_gamma\_p\*e(k)\*u(k)\*x1(k);

w2(k+1) = w2(k)+param\_gamma\_i\*e(k)\*u(k)\*x2(k);

w3(k+1) = w3(k)+param\_gamma\_d\*e(k)\*u(k)\*x3(k);

end

y = y(1:iteration);

u = u(1:iteration);

w1 = w1(1:iteration);

w2 = w2(1:iteration);

w3 = w3(1:iteration);

n = (1:10:10\*iteration);

figure(1)

plot(n,y);

figure(2)

plot(n,u);

figure(3)

plot(n,w1);

figure(4)

plot(n,w2);

figure(5)

plot(n,w3);

超参数为

param\_K = 0.01;

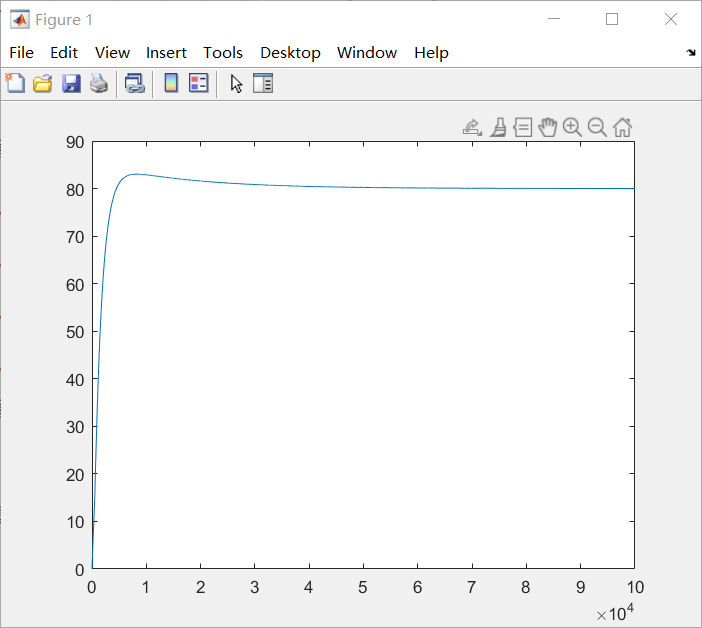
param\_gamma\_p = 0.9;

param\_gamma\_i = 0.00001;

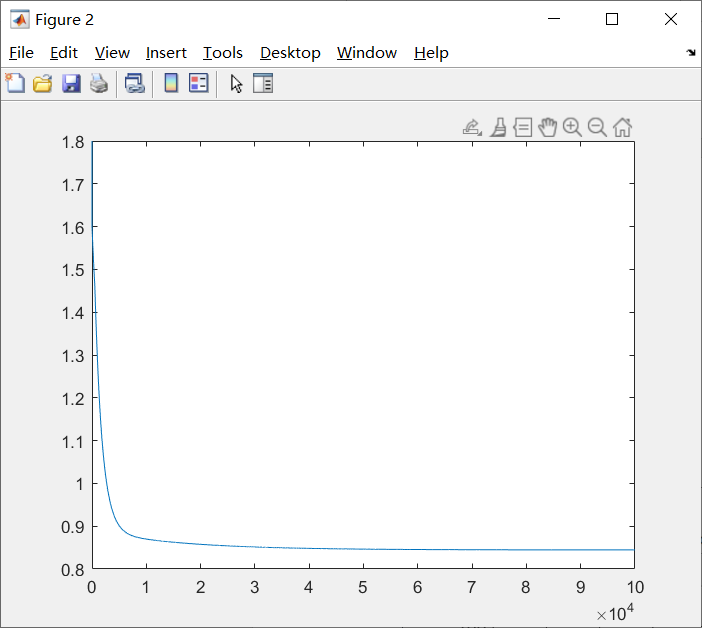
param\_gamma\_d = 0.3;

各参数的随时间变化图如下

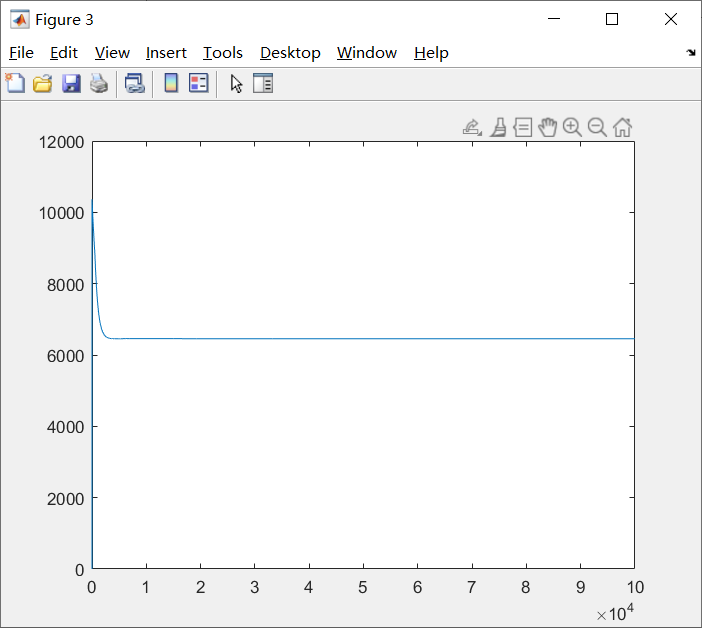
Y随时间变化图



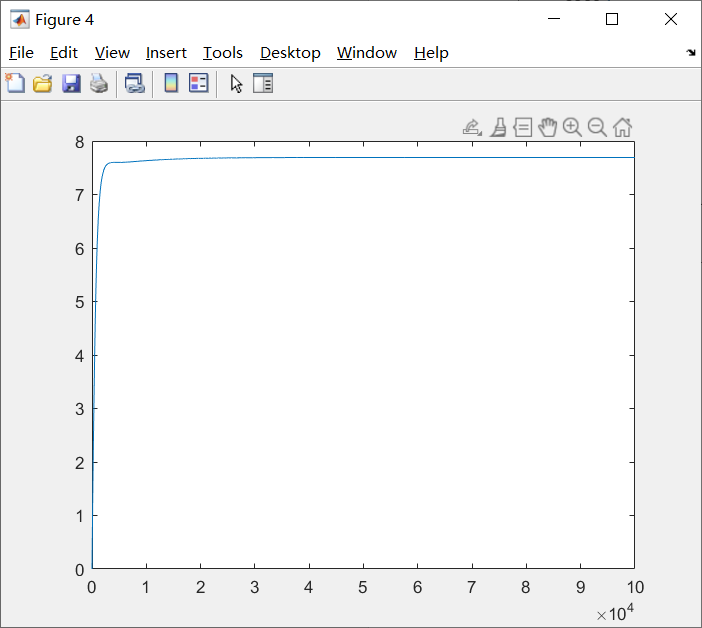
U随时间变化图



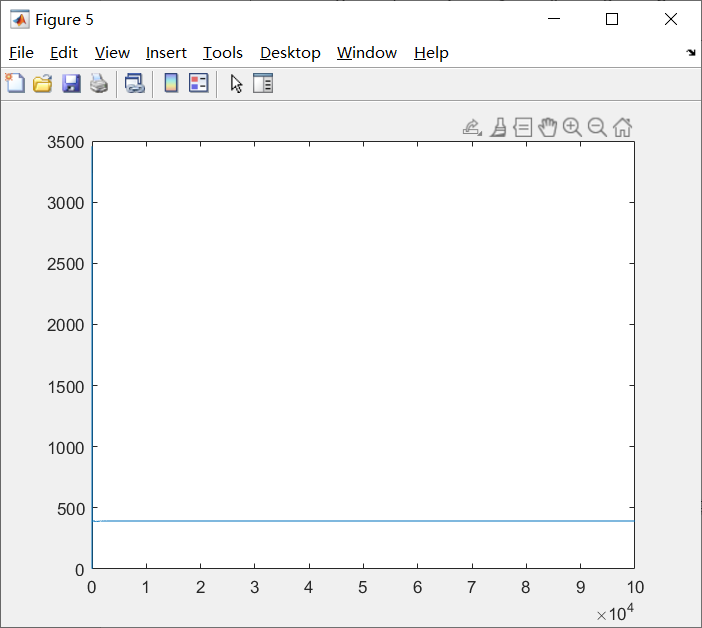
W1随时间变化图



W2随时间变化图



W3随时间变化图



从图中可以看到，系统的上升时间，超调量，调节时间基本上满足题目要求

## 分析与讨论

本次实验在调整超参数方面略有困难，需要多次尝试才能满足题目要求的各个时域特征。

## 总结与体会

在这次实验中，我使用MATLAB软件仿真了自适应的PID控制器的单层神经元的过程，锻炼了我的编程能力。