人工智能学院实验报告

姓名：周杨 学号：2112030064 专业：人工智能 年级：2021 级

课程：机器人学及其应用导论 主讲教师：付乐

辅导教师：付乐

实验时间：2024 年 5 月 28 日 下 午 6 时至 8 时

实验地点：九章 B407

实验题目：RBF 网络自适应控制仿真实验

实验目的：通过该实验，熟悉机器人在 MATLAB 开发环境下采用 RBF 网络进行自适应控制的基本操作，掌握神经网络自适应控制方法。

实验环境（硬件和软件）计算机，matlab

实验内容及步骤：

1、RBF 网络自适应控制原理

考虑如下的动力学系统：

$$\ddot{\theta} = f(\theta, \dot{\theta}) + u$$

式中，θ 为转动角度，u 为控制输入。

写成状态方程形式为

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = f(x) + u$$

式中，f(x)为未知函数。

位置指令为 xd，则误差及其导数为 $e = x_1 - x_d$， $\dot{e} = x_2 - \dot{x}_d$

定义误差函数为 $s = \lambda e + \dot{e}$，$\lambda > 0$

则，$\dot{s} = \lambda \dot{e} + \ddot{e} = \lambda \dot{e} + \dot{x}_2 - \ddot{x}_d = \lambda \dot{e} + f(x) + u - \ddot{x}_d$

由此可见，如果 $s \to 0$，则 $e \to 0$ 且 $\dot{e} \to 0$。

采用 RBF 网络逼近 $f(x)$，算法为

$$h_j = \exp\left(\frac{\left\| x - c_j^T \right\|^2}{2b_j^2}\right)$$

$$f(x) = W^{*T} h + \varepsilon$$

式中，x 为网络的输入，j 为网络隐含层的第 j 个节点；$h = \left[h_1 \cdots h_j\right]^T$ 为网络的高斯基函数输出；c_j 为网络的第 j 个节点的中心向量；W^* 为网络的理想权值；ε 为网络的逼近误差，$\varepsilon < \varepsilon_N$。

网络输入取 $x = \left[x_1\ x_2\right]^T$，则网络输出为 $\hat{f}(x) = \hat{W}^T h$。

由于

$$f(x) - \hat{f}(x) = W^{*T} h + \varepsilon - \hat{W}^T h = -\tilde{W}^T h + \varepsilon$$

定义 Lyapunov 函数为 $V=\frac{1}{2}s^2+\frac{1}{2\gamma}\tilde{W}^T\tilde{W}$

式中，$\gamma>0$，$\tilde{W}=\hat{W}-W^*$，则

$$\dot{V}=s\dot{s}+\frac{1}{\gamma}\tilde{W}^T\dot{\hat{W}}=s\left(\lambda\dot{e}+f(x)+u-\ddot{x}_d\right)+\frac{1}{\gamma}\tilde{W}^T\dot{\hat{W}}$$

设计如下的控制律：

$$u=-\lambda\dot{e}-\hat{f}(x)+\ddot{x}_d-\eta\,\mathrm{sgn}(s)$$

则

$$\begin{aligned}\dot{V}&=s\left(f(x)-\hat{f}(x)-\eta\,\mathrm{sgn}(s)\right)+\frac{1}{\gamma}\tilde{W}^T\dot{\hat{W}}\\&=s\left(-\tilde{W}^Th+\varepsilon-\eta\,\mathrm{sgn}(s)\right)+\frac{1}{\gamma}\tilde{W}^T\dot{\hat{W}}\\&=\varepsilon s-\eta|s|+\tilde{W}^T\left(\frac{1}{\gamma}\dot{\hat{W}}-sh\right)\end{aligned}$$

取 $\eta>\varepsilon_N$，自适应控制律为

$$\dot{\hat{W}}=\gamma sh$$

则 $\dot{V}=\varepsilon s-\eta|s|\le 0$。

由于当且仅当 $s=0$ 时，$\dot{V}=0$，即当 $\dot{V}\equiv 0$ 时，$s\equiv 0$。根据 LaSalle 不变性原理，闭环系统为渐进稳定，即当 $t\to\infty$ 时，$s\to 0$，系统的收敛速度取决于 η。

由于 $V\ge 0$，$\dot{V}\le 0$，则当 $t\to\infty$ 时，V 有界，从而 $\tilde{W}$ 有界。

2、仿真过程与结果

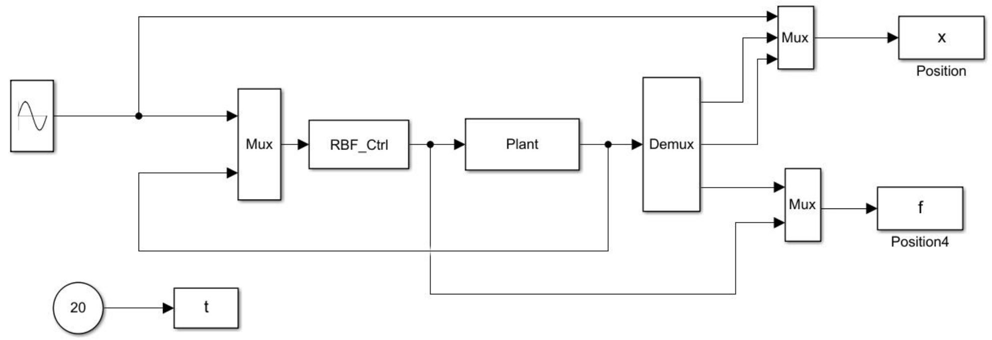
考虑如下被控对象：

$$\begin{aligned}\dot{x}_1&=x_2\\\dot{x}_2&=f(x)+u\end{aligned}$$

式中，假设 $f(x)=10x_1x_2$。

位置指令为 $x_d=\sin t$，采用上述的控制律和自适应律，取 $\lambda=10$，$\gamma=500$，$\eta=0.5$。根据网络输入 x_1 和 x_2 的实际范围来设计高斯基函数的参数，参数 c_j 和 b_j 取值分别为 $[-2,-1,0,1,2]$ 和 3.0。网络权值中各个元素的初始值取 0.1。

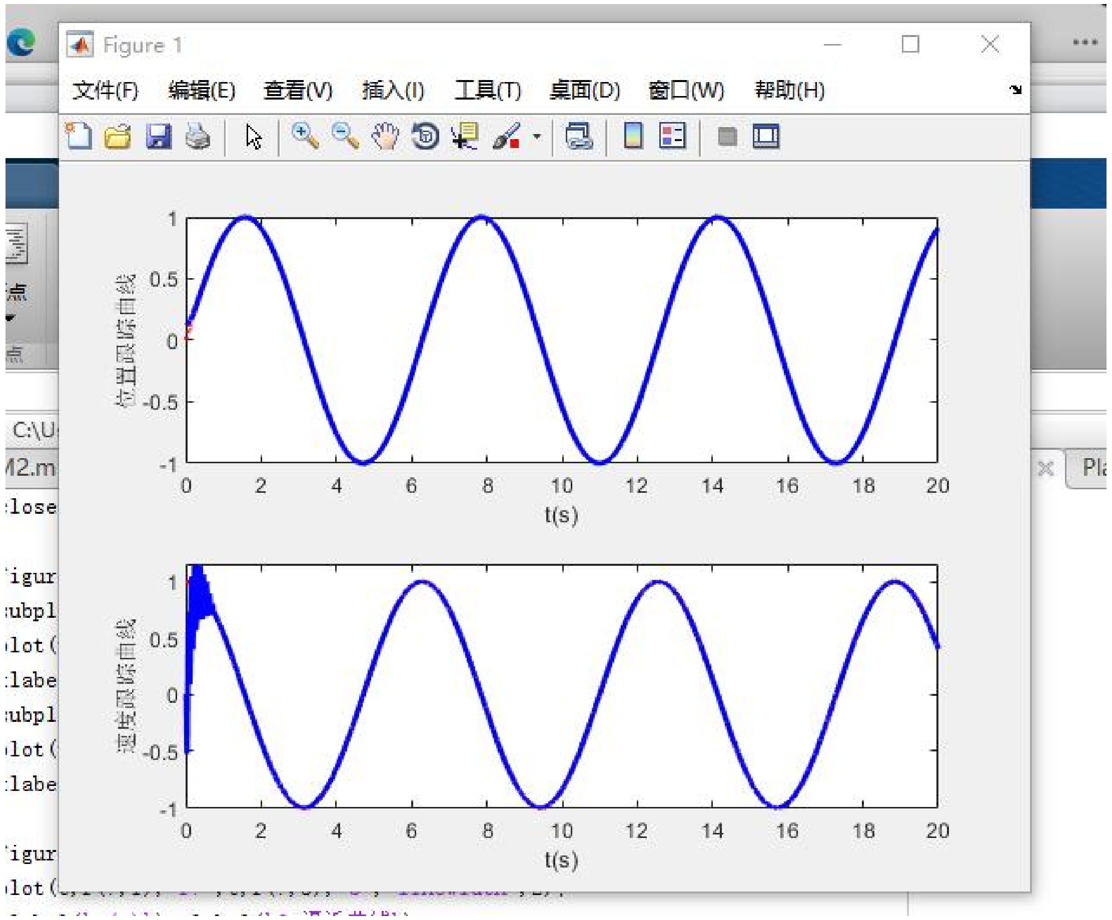
实验源程序如下：

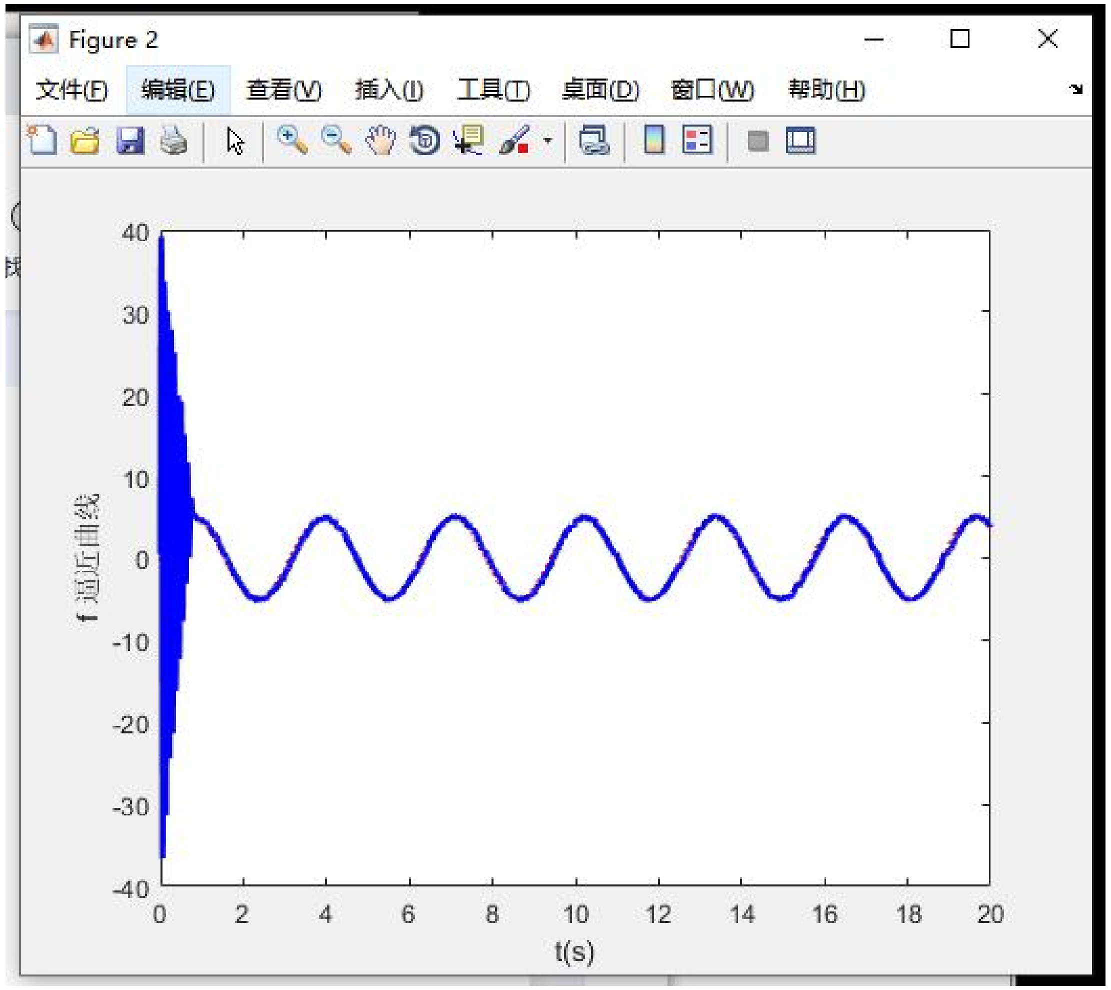



```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
   [sys,x0,str,ts]=mdlInitializeSizes;
case 1,
   sys=mdlDerivatives(t,x,u);
case 3,
   sys=mdlOutputs(t,x,u);
case {2,4,9}
   sys=[];
otherwise
   error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts]=mdlInitializeSizes
global b c lama
sizes = simsizes;
sizes.NumContStates  = 5;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 2;
sizes.NumInputs      = 4;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
```

```
x0  = 0.1*ones(1,5);
str = [];
ts  = [0 0];
c=0.5*[-2 -1 0 1 2;
      -2 -1 0 1 2];
b=3.0;
lama=10;
function sys=mdlDerivatives(t,x,u)
global b c lama
xd=sin(t);
dxd=cos(t);

x1=u(2);
x2=u(3);
e=x1-xd;
de=x2-dxd;
s=lama*e+de;

W=[x(1) x(2) x(3) x(4) x(5)]';
xi=[x1;x2];

h=zeros(5,1);
for j=1:1:5
   h(j)=exp(-norm(xi-c(:,j))^2/(2*b^2));
end

gama=1500;
for i=1:1:5
   sys(i)=gama*s*h(i);
end

function sys=mdlOutputs(t,x,u)
global b c lama
xd=sin(t);
dxd=cos(t);
ddxd=-sin(t);

x1=u(2);
x2=u(3);
e=x1-xd;
de=x2-dxd;
s=lama*e+de;

W=[x(1) x(2) x(3) x(4) x(5)];
xi=[x1;x2];

h=zeros(5,1);
for j=1:1:5
   h(j)=exp(-norm(xi-c(:,j))^2/(2*b^2));
end
fn=W*h;
xite=1.50;

%fn=10*x1+x2;  %Precise f
ut=-lama*de+ddxd-fn-xite*sign(s);

sys(1)=ut;
sys(2)=fn;
```

实验数据记录：





问题与讨论：

可以通过内置的 error 函数来打印错误信息使代码更具有处理异常的能力