人工智能学院实验报告

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

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：



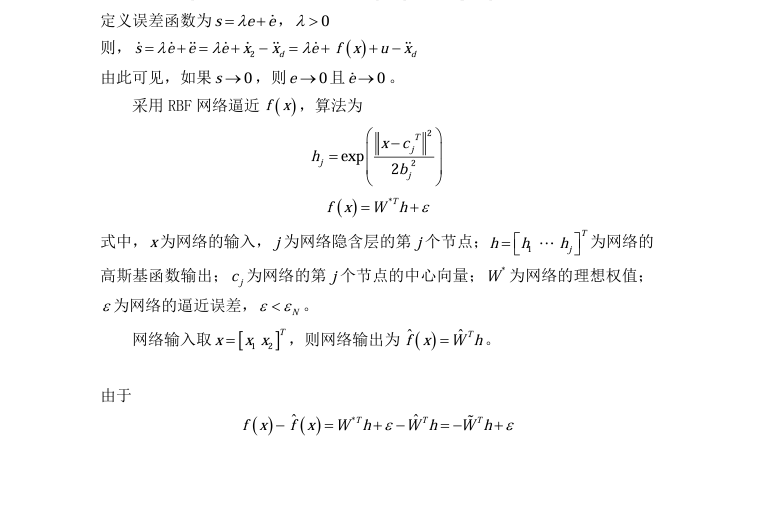
式中，θ为转动角度，u为控制输入。

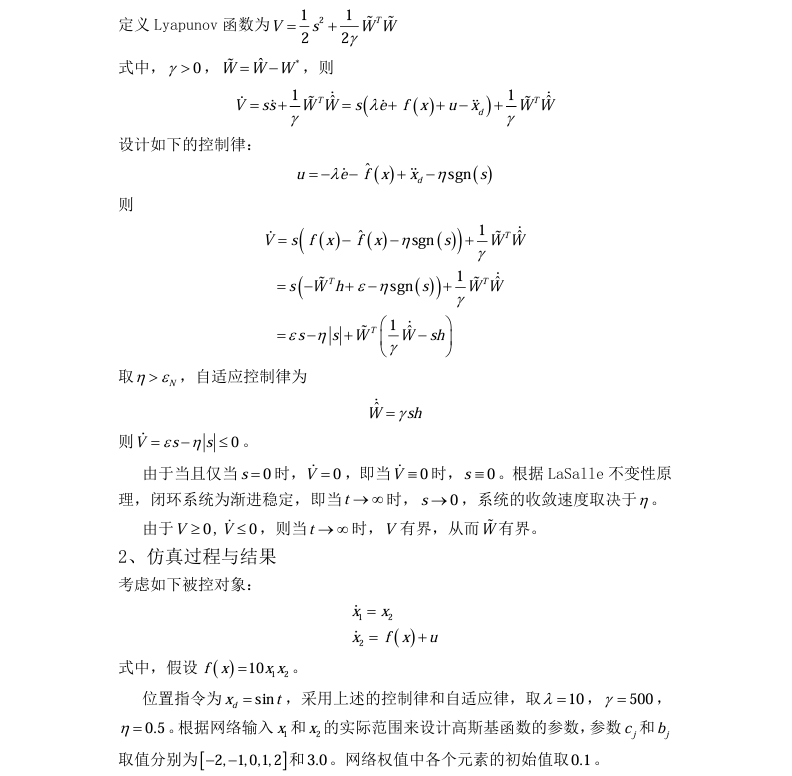
写成状态方程形式为



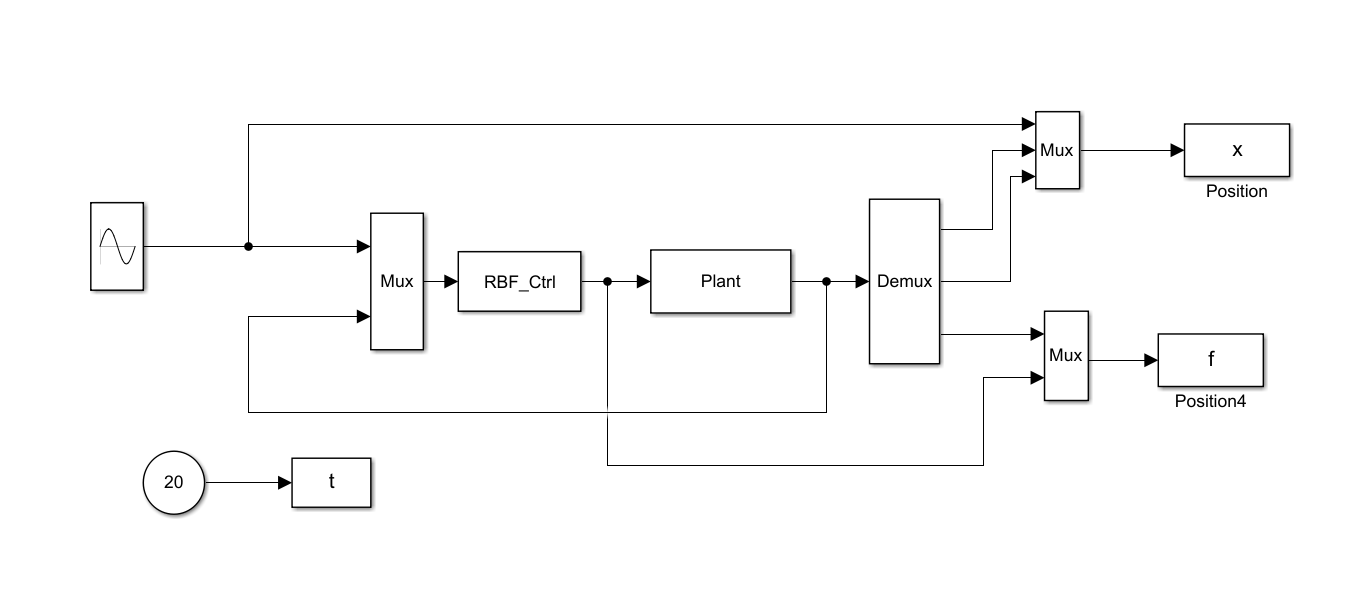
式中，f(x)为未知函数。

位置指令为xd，则误差及其导数为





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;

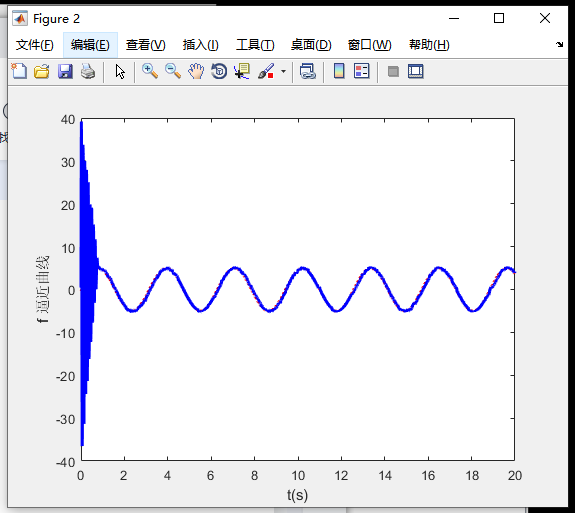
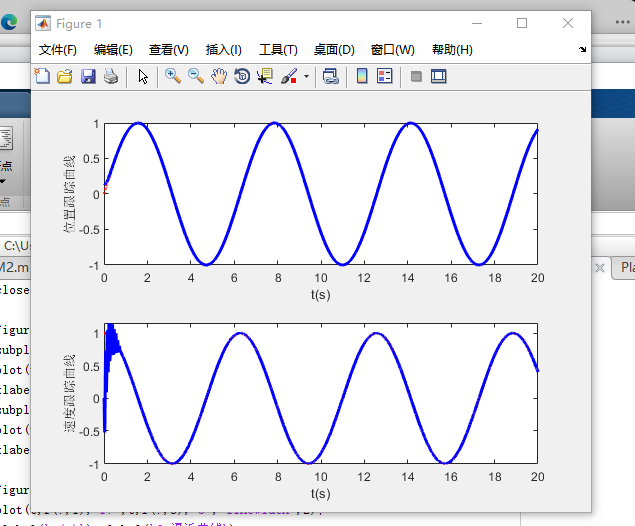
%fn=10\*x1+x2; %Precise f

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