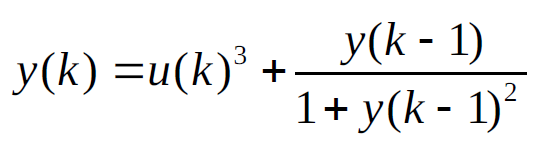
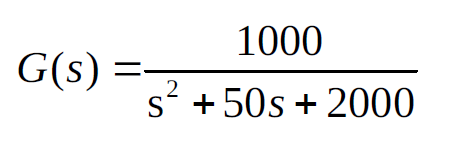
## 一、实验目的

（1）P137 采用RBF网络对如下离散模型进行逼近

（2）P185 RBF网络监督控制实例，被控对象如下：

## 二、程序代码

### 2.1、RBF网络逼近仿真实例程序代码

chap7\_6.m，具体代码如下。

%RBF identification

clear all;

close all;

alfa=0.05; %动量因子

xite=0.15; %学习速率

x=[0,1]';

b=3\*ones(5,1); %高斯基函数的宽度向量

c=[-1 -0.5 0 0.5 1; %高斯基函数的参数取值

-1 -0.5 0 0.5 1];

w=rands(5,1);

w\_1=w;w\_2=w\_1;

d\_w=0\*w;

y\_1=0;

ts=0.001;

for k=1:1:10000

time(k)=k\*ts;

u(k)=sin(k\*ts);

y(k)=u(k)^3+y\_1/(1+y\_1^2); %对象

x(1)=u(k); %x为网络输入，u为对象输入

x(2)=y\_1; %y为对象输出

for j=1:1:5

h(j)=exp(-norm(x-c(:,j))^2/(2\*b(j)\*b(j))); %隐含层神经元输出

end

ym(k)=w'\*h'; %网络输出

em(k)=y(k)-ym(k); %误差

d\_w(j)=xite\*em(k)\*h(j);%△w(j)

w=w\_1+d\_w+alfa\*(w\_1-w\_2); %按照梯度下降法调节权值

y\_1=y(k); %更新输出值

w\_2=w\_1; %更新权值

w\_1=w; %更新权值

end

figure(1);

subplot(211);

plot(time,y,'r',time,ym,'k:','linewidth',2);

xlabel('time(s)');ylabel('y and ym');

legend('ideal signal','signal approximation');

subplot(212);

plot(time,y-ym,'k','linewidth',2);

xlabel('time(s)');ylabel('error');

### 2.2、RBF网络监督控制实例程序代码

chap9\_2.m，具体代码如下。

ts=0.001;

sys=tf(1000,[1,50,2000]);

dsys=c2d(sys,ts,'z');

[num,den]=tfdata(dsys,'v');

y\_1=0;y\_2=0;%储存输出值y(k-1)，y(k-2)

u\_1=0;u\_2=0;%储存对象输入值u(k-1),u(k-2)

e\_1=0;%储存误差值e(k-1)

xi=0;

x=[0,0]';

b=0.5\*ones(4,1);%基宽参数

c=[-2 -1 1 2];%结点的中心向量

w=rands(4,1);%权值矩阵

w\_1=w;%储存权值

w\_2=w\_1;

xite=0.30; %学习速率

alfa=0.05;%动量因子

kp=25;

kd=0.3;

for k=1:1:1000

time(k)=k\*ts;

S=1;

if S==1

r(k)=0.5\*sign(sin(2\*2\*pi\*k\*ts)); %方型信号，幅值为0.5、频率为2Hz

elseif S==2

r(k)=0.5\*(sin(3\*2\*pi\*k\*ts)); %方型信号，幅值为0.5、频率为3Hz

end

y(k)=-den(2)\*y\_1-den(3)\*y\_2+num(2)\*u\_1+num(3)\*u\_2;%离散化对象

e(k)=r(k)-y(k);%理想输出与实际输出差值

xi=r(k);

for j=1:1:4

h(j)=exp(-norm(xi-c(:,j))^2/(2\*b(j)\*b(j)));%高斯基函数

end

un(k)=w'\*h';%网络输出un

%PD控制器

up(k)=kp\*x(1)+kd\*x(2);

M=2;

if M==1 %只用PD控制器

u(k)=up(k);

elseif M==2 %PD控制输出+RBF网络控制输出

u(k)=up(k)+un(k);

end

if u(k)>=10

u(k)=10;

end

if u(k)<=-10

u(k)=-10;

end

if k==400

u(k)=u(k)+6.0;

end

%更新NN权值

d\_w=-xite\*(un(k)-u(k))\*h';

w=w\_1+ d\_w+alfa\*(w\_1-w\_2);

w\_2=w\_1;

w\_1=w;

u\_2=u\_1;

u\_1=u(k);

y\_2=y\_1;

y\_1=y(k);

x(1)=e(k); %计算P

x(2)=(e(k)-e\_1)/ts; %计算D

e\_1=e(k);

end

figure(1);

plot(time,r,'r',time,y,'b');

xlabel('time(s)');ylabel('r and y');

figure(2);

subplot(311);

plot(time,un,'b');

xlabel('time(s)');ylabel('un');

subplot(312);

plot(time,up,'k');

xlabel('time(s)');ylabel('up');

subplot(313);

plot(time,u,'r');

xlabel('time(s)');ylabel('u');

## 三、运行实例及分析

### 3.1、RBF网络逼近仿真实例运行结果



图1 基于权值调节的RBF网络逼近（原cjbj）



图2 改动cj中的第二行0.5为3



图3 改动cj中的第二行0.5为9

****

图4 改动b为0.5

****

图5 改动b为1

****

图6 改动b为10



图7 修改学习步长为0.01



图8 修改学习步长为0.1



图9 修改学习步长为0.9

### 3.2、RBF网络监督控制实例运行结果

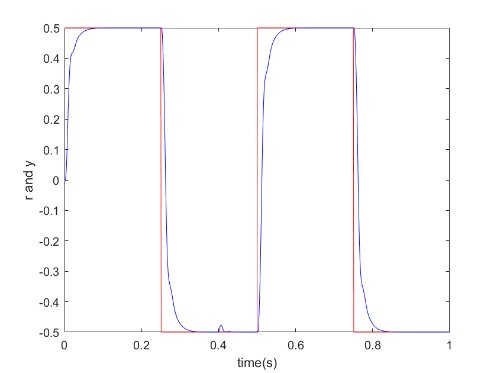


图10 方波位置跟踪

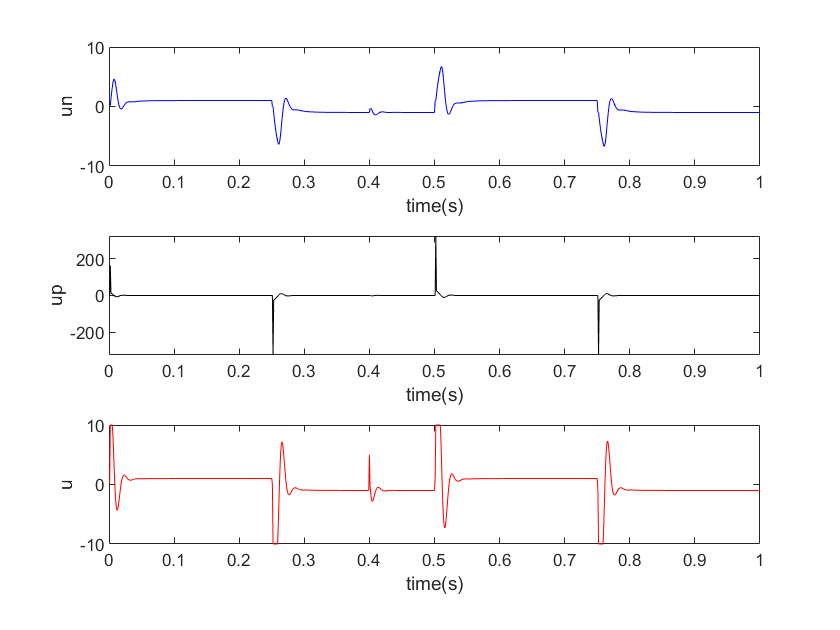


图11 神经网络、PD及总控制器输出的比较



图12 方波位置跟踪（kp改为50）



图13 神经网络、PD及总控制器输出的比较（kp改为50）



图14 方波位置跟踪（kp改为10）



图15 神经网络、PD及总控制器输出的比较（kp改为10）



图16 方波位置跟踪（kd改为0.6）



图17 神经网络、PD及总控制器输出的比较（kd改为0.6）

## 四、结果分析及总结

### 4.1、RBF网络逼近仿真实例结果分析

1) 由图中可以看出，使用RBF网络逼近，若高斯基函数参数cj和bj设置合理，学习步长不过于小，其输出结果与理想输出曲线几乎重合。

2) 仅改动cj矩阵的第二行中心点的值，发现改动值偏离原设定值越大，输出结果曲线与理想输出曲线偏离越多，误差曲线波动越明显，这也印证了cj值离输入越近，高斯函数对输入越敏感。

3) 仅改动bj即第j个神经元高斯基函数的宽度，可以看出宽度越窄，网络对输入的映射范围越小，误差波动范围越小，宽度越宽，网络对输入的映射范围越大，当宽度在合适范围内时，误差在0周围细微波动。在适当范围内，输出结果与理想输出曲线几乎重合。

4) 仅改动学习步长，可以看出，学习步长的值较小时， 输出结果与理想输出曲线偏离较大，误差波动明显，随着学习步长逐渐增大，输出结果与理想输出曲线偏离逐渐减小，直到重合，误差波动也逐渐减小，不断向0靠拢。

### 4.2、RBF网络监督控制实例结果分析

由实验结果可得：

1) 当kp，kd选择适当时，方波位置跟踪较为精准且平稳，un以及up较为稳定，up在0附近细微波动。

2) 当调大kp时，位置跟踪响应变快，但是系统产生了超调，un及up波动增加导致对象输入u波动明显。

3) 当调小kp时，位置跟踪相应缓慢，系统动态，静态特性变坏，但是un及up波动较为稳定。

4) 当调大kd时，位置跟踪响应提前制动，延长调节时间，un波动明显，导致u波动明显，系统的抗干扰能力下降。