第 3 章

CHAPTER 3

基于梯度下降法的 RBF 神经网络控制

离散神经网络控制系统中，常采用梯度下降法实现神经网络权值的学习，有代表性的研究工作如文献[1-2]。

3.1 基于 RBF 神经网络的监督控制

3.1.1 RBF 监督控制

图 3.1 为基于 RBF 神经网络的监督控制系统，其控制思想为：初始阶段采用 PD 反馈控制，然后过渡到神经网络控制。在控制过程中，如出现较大的误差，则 PD 控制起主导作用，神经网络控制起调节作用。

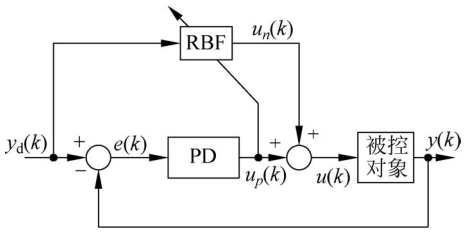

图 3.1 基于 RBF 神经网络的监督控制系统

设径向基向量为 $\boldsymbol{h}=[h_1,h_2,\cdots,h_m]^{\mathrm{T}}$，$h_j$ 为高斯函数，则

$$h_j=\exp\left(-\frac{\|\boldsymbol{x}(k)-\boldsymbol{c}_j\|^2}{2b_j^2}\right) \tag{3.1}$$

其中，$i=1$；$j=1,2,\cdots,m$；$\boldsymbol{x}(k)$ 为 RBF 神经网络的输入；$\boldsymbol{c}_j=[c_{11},c_{12},\cdots,c_{1m}]$；$\boldsymbol{b}=[b_1,b_2,\cdots,b_m]^{\mathrm{T}}$。

设权值向量为

$$\boldsymbol{w}=[w_1,w_2,\cdots,w_m]^{\mathrm{T}} \tag{3.2}$$

RBF 神经网络的输出为

$$u_n(k)=h_1w_1+\cdots+h_jw_j+\cdots+h_mw_m \tag{3.3}$$

其中，m 为隐含层节点的个数。

总控制输入为 $u(k)=u_n(k)+u_p(k)$，误差指标为

$$E(k)=\frac{1}{2}[u_n(k)-u(k)]^2 \tag{3.4}$$

采用梯度下降法，网络权值学习算法为

$$\Delta w_j(k)=-\eta\frac{\partial E(k)}{\partial w_j(k)}=\eta[u_n(k)-u(k)]h_j(k)$$

$$\boldsymbol{w}(k)=\boldsymbol{w}(k-1)+\Delta\boldsymbol{w}(k)+\alpha[\boldsymbol{w}(k-1)-\boldsymbol{w}(k-2)] \tag{3.5}$$

其中，$\eta\in[0,1]$为学习速率；$\alpha\in[0,1]$为动量因子。

3.1.2　仿真实例

设控制对象为

$$G(s)=\frac{1000}{s^3+87.35s^2+10470s}$$

取采样周期为1ms，对上述对象进行离散化，可得

$$y(k)=-\mathrm{den}(2)y(k-1)-\mathrm{den}(3)y(k-2)+\mathrm{num}(2)u(k-1)+\mathrm{num}(3)u(k-2)$$

取神经网络的结构为1-4-1，理想跟踪指令$y_d(k)$取幅值为0.5的大波信号，网络输入为$y_d(k)$，网络的初始权值取[0,1]的随机数，根据网络的输入范围，高斯函数参数取$\boldsymbol{c}=[-2\quad -1\quad 1\quad 2]^{\mathrm{T}}$，$b_j=0.5$。取学习速率$\eta=0.30$，动量因子$\alpha=0.05$。

仿真结果如图3.2和图3.3所示。RBF监督控制的仿真程序为chap3_1.m，详见附录。

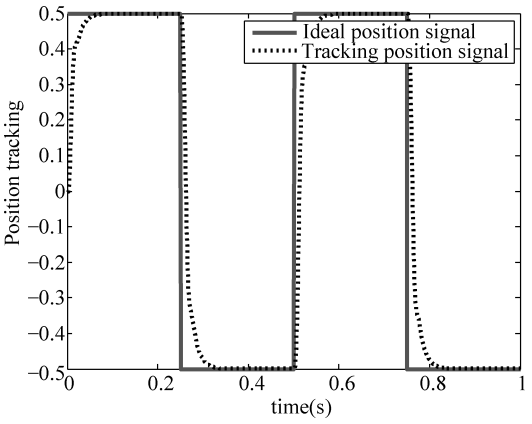

图3.2　方波跟踪效果

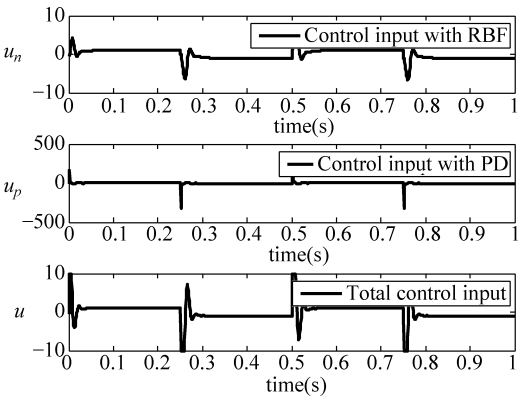

图3.3　神经网络输入、PD控制输入及总控制输入

3.2　基于RBF神经网络的模型参考自适应控制

3.2.1　控制系统设计

图3.4为基于RBF神经网络的模型参考自适应控制系统框图。

设理想跟踪指令为$y_m(k)$，则定义跟踪误差为

$$e(k)=y_m(k)-y(k)\tag{3.6}$$

网络权值学习误差指标为

$$E(k)=\frac{1}{2}e(k)^2\tag{3.7}$$

控制输入为RBF神经网络的输出：

$$u(k)=h_1w_1+\cdots+h_jw_j+\cdots+h_mw_m\tag{3.8}$$

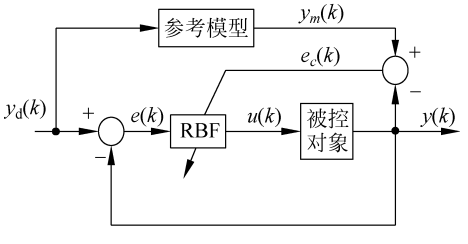

图3.4　基于RBF神经网络的模型参考自适应控制系统

其中，m 为隐含层的节点个数；w_j 为节点的权值；h_j 为高斯函数的输出。

在 RBF 神经网络中，$\boldsymbol{x}=[x_1,x_2,\cdots,x_n]^{\mathrm{T}}$ 为网络输入，$\boldsymbol{h}=[h_1,h_2,\cdots,h_m]^{\mathrm{T}}$，$h_j$ 为高斯函数：

$$h_j=\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{2b_j^2}\right) \tag{3.9}$$

其中，$i=1,2,\cdots,n$；$j=1,2,\cdots,m$。$b_j>0$，$\boldsymbol{c}_j=[c_{j1},\cdots,c_{ji},\cdots,c_{jn}]$，$\boldsymbol{b}=[b_1,b_2,\cdots,b_m]^{\mathrm{T}}$。

设权值向量为

$$\boldsymbol{w}=[w_1,w_2,\cdots,w_m]^{\mathrm{T}} \tag{3.10}$$

由梯度下降法，网络的学习算法为

$$\Delta w_j(k)=-\eta\frac{\partial E(k)}{\partial w}=\eta e_c(k)\frac{\partial y(k)}{\partial u(k)}h_j$$

$$w_j(k)=w_j(k-1)+\Delta w_j(k)+\alpha\Delta w_j(k) \tag{3.11}$$

其中，η 为学习速率；α 为动量因子；$\eta\in[0,1]$，$\alpha\in[0,1]$。

同理可得

$$\Delta b_j(k)=-\eta\frac{\partial E(k)}{\partial b_j}=\eta e_c(k)\frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial b_j}$$

$$=\eta e_c(k)\frac{\partial y(k)}{\partial u(k)}w_j h_j\frac{\|\boldsymbol{x}-\boldsymbol{c}_{ij}\|^2}{b_j^3} \tag{3.12}$$

$$b_j(k)=b_j(k-1)+\eta\Delta b_j(k)+\alpha[b_j(k-1)-b_j(k-2)] \tag{3.13}$$

$$\Delta c_{ij}(k)=-\eta\frac{\partial E(k)}{\partial c_{ij}}=\eta e_c(k)\frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial c_{ij}}$$

$$=\eta e_c(k)\frac{\partial y(k)}{\partial u(k)}w_j h_j\frac{x_i-c_{ij}}{b_j^2} \tag{3.14}$$

$$c_{ij}(k)=c_{ij}(k-1)+\eta\Delta c_{ij}(k)+\alpha[c_{ij}(k-1)-c_{ij}(k-2)] \tag{3.15}$$

其中，$\frac{\partial y(k)}{\partial u(k)}$为 Jacobian 阵，表征系统输出对控制输入的灵敏度。在学习算法中，$\frac{\partial y(k)}{\partial u(k)}$值可采用差分求得，不精确部分可通过权值的调整来修正。可用$\frac{\Delta y(k)}{\Delta u(k)}$的正负号来代替$\frac{\partial y(k)}{\partial u(k)}$，这样可使算法更加简单。

3.2.2 仿真实例

取离散被控对象为

$$y(k)=[-0.10y(k-1)+u(k-1)]/[1+y(k-1)^2]$$

其中，采样周期为 $t_s=1\mathrm{ms}$，参考模型为 $y_m(k)=0.6y_m(k-1)+y_{\mathrm{d}}(k)$，理想跟踪指令为 $y_{\mathrm{d}}(k)=0.50\sin(2\pi k\times t_s)$。

取 RBF 神经网络的输入为 $y_{\mathrm{d}}(k)$、$e_c(k)$ 和 $y(k)$，学习速率为 $\eta=0.35$，动量因子为 $\alpha=0.05$。

根据网络的输入范围，高斯函数参数值为 $\boldsymbol{c}=\begin{bmatrix}-3&-2&-1&1&2&3\\-3&-2&-1&1&2&3\\-3&-2&-1&1&2&3\end{bmatrix}^{\mathrm{T}}$，$\boldsymbol{b}=[2,2,$

2,2,2,2]^{T},网络初始权值取[0,1]的随机值。

仿真结果如图 3.5 和图 3.6 所示。基于 RBF 神经网络的模型参考自适应控制程序为 chap3_2.m。

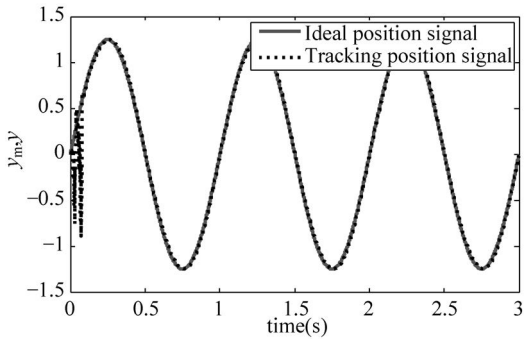

图 3.5　正弦跟踪

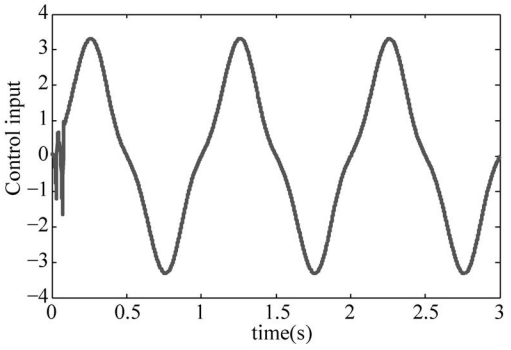

图 3.6　控制输入

3.3　RBF 神经网络自校正控制

3.3.1　系统描述

考虑被控对象：

$$y(k+1)=f[y(k)]+bu(k) \tag{3.16}$$

其中，u 和 y 分别为输入和输出，b 为已知实数。

若 $f(\cdot)$已知，则根据确定性等价原则，自校正控制算法为

$$u(k)=\frac{-f(\cdot)+y_d(k+1)}{b} \tag{3.17}$$

采用控制律式(3.17)，系统的输出 $y(k)$能精确地跟踪输入 $y_d(k)$。

3.3.2　RBF 控制算法设计

若 $f(\cdot)$未知，则可通过在线训练神经网络辨识器，得到 $f(\cdot)$的辨识值 $N\varphi(\cdot)$，此时自校正控制算法为

$$u(k)=\frac{-N\varphi(\cdot)+y_d(k+1)}{b} \tag{3.18}$$

采用 RBF 神经网络可实现未知函数项 $f(\cdot)$的辨识。在 RBF 神经网络结构中，取网络的输入为 $y(k)$，h_j 为网络第 i 个输入第 j 个结点高斯函数输出：

$$h_j=\exp\left[-\frac{\|y(k)-c_{1j}\|^2}{2b_j^2}\right] \tag{3.19}$$

其中，$i=1$，$j=1,\cdots,m$，b_j 为节点 j 的宽度参数，$b_j>0$，c_{ij} 为网络第 i 个输入第 j 个结点的中心点坐标值。

网络的权向量为 $\boldsymbol{W}=[w_1,w_2,\cdots,w_m]^{T}$，RBF 神经网络的输出为

$$N\varphi(k)=\sum_{j=1}^{m}h_jw_j \tag{3.20}$$

其中，m 为 RBF 网络隐层神经元的个数。

辨识后，对象的输出为

$$y_n(k)=N\varphi[y(k-1)]+bu(k-1) \tag{3.21}$$

神经网络调整的性能指标为

$$E(k)=\frac{1}{2}[y(k)-y_n(k)]^2 \tag{3.22}$$

采用梯度下降法调整网络的权值：

$$\Delta w_j(k)=-\eta\frac{\partial E(k)}{\partial w_j(k)}=\eta[y(k)-y_n(k)]h_j(k)$$

神经网络权值的调整过程为

$$\boldsymbol{W}(k)=\boldsymbol{W}(k-1)+\Delta\boldsymbol{W}(k)+\alpha[\boldsymbol{W}(k-1)-\boldsymbol{W}(k-2)] \tag{3.23}$$

其中，η 为学习速率，α 为动量因子。

由式(3.16)和式(3.21)可得

$$y(k)-y_n(k)=f[y(k-1)]-N\varphi[y(k-1)]$$

通过梯度下降法，可实现 $t\to\infty$ 时，$E(k)\to0$，$N\varphi[y(k-1)]\to f[y(k-1)]$，从而实现未知函数 $f(\cdot)$ 的辨识。

3.3.3 仿真实例

被控对象为

$$y(k)=0.8\sin[y(k-1)]+15u(k-1)$$

其中，$f[y(k)]=0.8\sin[y(k-1)]$。

输入信号为正弦信号 $y_d(k)=\sin(0.1\pi t)$。取 $y(k)$ 作为网络的输入，网络隐含层神经元个数取 $m=5$，网络结构为 1-5-1，网络的初始权值每个值取 1.0，$b_j=5.0$。

采用控制律式(3.18)，网络权值学习参数为 $\eta=0.10$，$\alpha=0.05$。RBF 神经网络自校正控制程序为 chap3_3.m。仿真结果如图 3.7～图 3.9 所示。仿真程序为 chap3_3.m，详见附录。

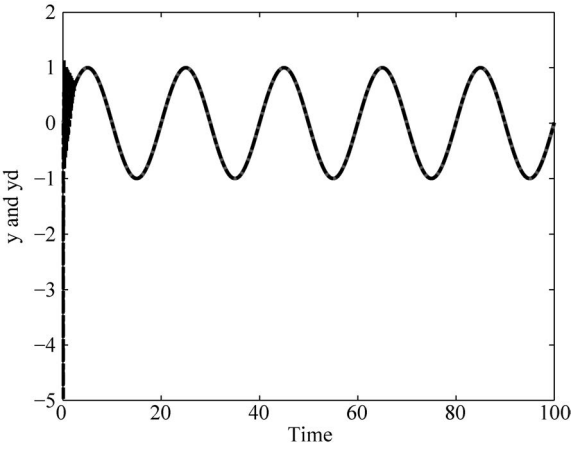

图 3.7　正弦位置跟踪

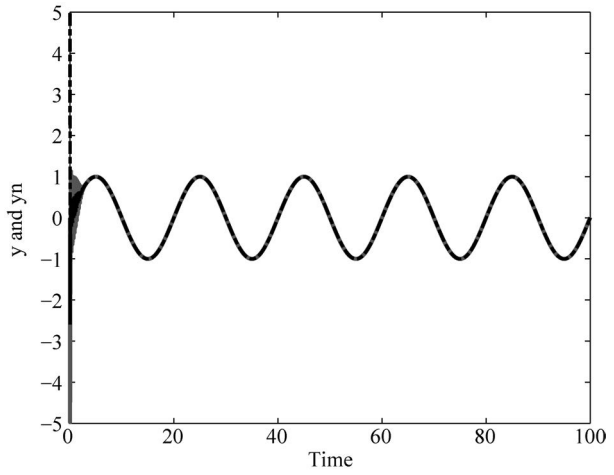

图 3.8　输出 y 及其逼近 y_n

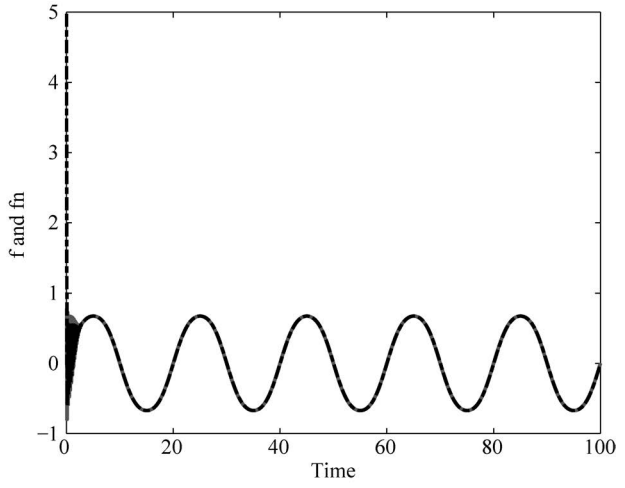

图 3.9　$f(x,t)$ 及其辨识结果 $\hat{f}(x,t)$

附录　仿真程序

3.1.2 节的程序

RBF 监督控制仿真程序：chap3_1.m

```
% RBF Supervisory Control
clear all;
close all;

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;
```

```
u_1 = 0;u_2 = 0;
e_1 = 0;

xi = 0;
x = [0,0]';

b = 0.5 * ones(4,1);
c = [ - 2  - 1 1 2];
w = rands(4,1);
w_1 = w;
w_2 = w_1;

xite = 0.30;
alfa = 0.05;

kp = 25;
kd = 0.3;
for k = 1:1:1000
    time(k) = k * ts;
S = 1;
if S == 1
   yd(k) = 0.5 * sign(sin(2 * 2 * pi * k * ts));    % Square Signal
elseif S == 2
   yd(k) = 0.5 * (sin(3 * 2 * pi * k * ts));        % Sine Signal
end

y(k) = - den(2) * y_1 - den(3) * y_2 + num(2) * u_1 + num(3) * u_2;
e(k) = yd(k) - y(k);

xi = yd(k);

for j = 1:1:4
    h(j) = exp( - norm(xi - c(:,j))^2/(2 * b(j) * b(j)));
end
un(k) = w' * h';

% PD Controller
up(k) = kp * x(1) + kd * x(2);

M = 2;
if M == 1                  % Only Using PID Control
   u(k) = up(k);
elseif M == 2              % Total control output
    u(k) = up(k) + un(k);
end

if u(k) > = 10
   u(k) = 10;
end
if u(k) < = - 10
   u(k) = - 10;
end
% Update NN Weight
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);              % Calculating P
x(2) = (e(k) - e_1)/ts;   % Calculating D
e_1 = e(k);
end
figure(1);
plot(time,yd,'r',time,y,'k:','linewidth',2);
xlabel('time(s)');ylabel('Position tracking');
legend('Ideal position signal','Tracking position signal');

figure(2);
subplot(311);
plot(time,un,'k','linewidth',2);
xlabel('time(s)');ylabel('un');
legend('Control input with RBF');
subplot(312);
plot(time,up,'k','linewidth',2);
xlabel('time(s)');ylabel('up');
legend('Control input with PD');
subplot(313);
plot(time,u,'k','linewidth',2);
xlabel('time(s)');ylabel('u');
legend('Total control input');
```

3.2.2 节的程序

基于 RBF 神经网络的模型参考自适应控制：chap3_2.m

```
% Model Reference Adaptive RBF Control
clear all;
close all;

u_1 = 0;
y_1 = 0;
ym_1 = 0;

x = [0,0,0]';
c = [ -3 -2 -1 0 1 2 3;
      -3 -2 -1 0 1 2 3;
      -3 -2 -1 0 1 2 3];
b = 2;
w = rands(1,7);

xite = 0.35;
alfa = 0.05;
h = [0,0,0,0,0,0,0]';

c_1 = c;c_2 = c;
b_1 = b;b_2 = b;
w_1 = w;w_2 = w;

ts = 0.001;
for k = 1:1:3000
time(k) = k * ts;

yd(k) = 0.50 * sin(2 * pi * k * ts);
ym(k) = 0.6 * ym_1 + yd(k);
```

```
y(k) = ( - 0.1 * y_1 + u_1)/(1 + y_1^2); % Nonlinear plant

for j = 1:1:7
    h(j) = exp( - norm(x - c(:,j))^2/(2 * b^2));
end
u(k) = w * h;

ec(k) = ym(k) - y(k);
dyu(k) = sign((y(k) - y_1)/(u(k) - u_1));

d_w = 0 * w;
for j = 1:1:7
    d_w(j) = xite * ec(k) * h(j) * dyu(k);
end
w = w_1 + d_w + alfa * (w_1 - w_2);
 % Return of parameters
u_1 = u(k);
y_1 = y(k);
ym_1 = ym(k);

x(1) = yd(k);
x(2) = ec(k);
x(3) = y(k);

w_2 = w_1;w_1 = w;
end
figure(1);
plot(time,ym,'r',time,y,'k:','linewidth',2);
xlabel('time(s)');ylabel('ym,y');
legend('Ideal position signal','Tracking position signal');
figure(2);
plot(time,u,'r','linewidth',2);
xlabel('time(s)');ylabel('Control input');
```

3.3.3 节的程序

RBF 神经网络自校正控制：chap3_3.m

```
clear all;
close all;
xite = 0.10;
alfa = 0.05;

w = ones(5,1);
cij = [ - 2  - 1 0 1 2];
bj = 15;
h = zeros(5,1);

w_1 = w;w_2 = w_1;
u_1 = 0;y_1 = 0;
ts = 0.02;
for k = 1:1:5000
time(k) = k * ts;
yd(k) = sin(0.1 * pi * k * ts);

 % Practical Plant;
f(k) = 0.8 * sin(y_1);
b = 15;
```

```
y(k) = f(k) + b * u_1;

for j = 1:1:5
    h(j) = exp( - norm(y(k) - cij(:,j))^2/(2 * bj^2));
end

fn(k) = w' * h;
yn(k) = fn(k) + b * u_1;

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

d_w = 0 * w;
for j = 1:1:5
    d_w(j) = xite * e(k) * h(j);
end
w = w_1 + d_w + alfa * (w_1 - w_2);

u(k) = ( - fn(k) + yd(k))/b;

u_1 = u(k);
y_1 = y(k);

w_2 = w_1;
w_1 = w;
end
figure(1);
plot(time,yd,'r',time,y,' - .k','linewidth',2);
xlabel('Time');ylabel('y and yd');
figure(2);
plot(time,y,'r',time,yn,' - .k','linewidth',2);
xlabel('Time');ylabel('y and yn');
figure(3);
plot(time,f,'r',time,fn,' - .k','linewidth',2);
xlabel('Time');ylabel('f and fn');
```

参考文献

[1] NORIEGA J R,WANG H. A direct adaptive neural network control for unknown nonlinear systems and its application[J]. IEEE Transactions on Neural Networks,1998,9(1): 27-34.

[2] SUYKENS J A K,VANDEWALLE J P L,DEMOOR B L R. Artificial neural networks for modeling and control of nonlinear systems[M]. Boston: Kluwer Academic,1996.

[3] 刘金琨. 智能控制[M]. 2 版. 北京：电子工业出版社,2012.