第3章 机械手神经网络自适应控制

CHAPTER 3

如果被控对象的数学模型已知，滑模控制器可以使系统输出直接跟踪期望指令，但较大的建模不确定性需要较大的切换增益，这就造成抖振，抖振是滑模控制中难以避免的问题。

将滑模控制结合神经网络逼近用于非线性系统的控制中，采用神经网络实现模型未知部分的自适应逼近，可有效地降低切换增益。神经网络自适应律通过 Lyapunov 方法导出，通过自适应权重的调节保证整个闭环系统的稳定性和收敛性。

RBF(Radial Basis Function，径向基函数)神经网络于 1988 年提出。相比多层前馈 BP(Back Propagation，反向传播)网络，RBF 网络由于具有良好的泛化能力，网络结构简单，避免不必要和冗长的计算而备受关注。关于 RBF 网络的研究表明 RBF 神经网络能在一个紧凑集和任意精度下，逼近任何非线性函数[1]。目前，已经有许多针对非线性系统的 RBF 神经网络控制研究成果发表。

3.1 一种简单的 RBF 网络自适应滑模控制

3.1.1 问题描述

考虑一种简单的动力学系统：

$$\ddot{\theta}=f(\theta,\dot{\theta})+u \tag{3.1}$$

其中，θ 为转动角度，u 为控制输入。

写成状态方程形式为

$$\begin{cases}\dot{x}_1=x_2\\ \dot{x}_2=f(x)+u\end{cases} \tag{3.2}$$

其中，$x_1=\theta$，$x_2=\dot{\theta}$，$f(x)$为未知。

角度指令为 x_d，则误差及其导数为

$$e=x_1-x_d,\quad \dot{e}=x_2-\dot{x}_d$$

定义滑模函数

$$s=ce+\dot{e},\quad c>0 \tag{3.3}$$

则

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

由式(3.3)可见，如果 $s\to 0$，则 $e\to 0$ 且 $\dot{e}\to 0$。

3.1.2　RBF 网络原理

由于 RBF 网络具有万能逼近特性[1]，采用 RBF 神经网络逼近 $f(x)$，网络算法为

$$h_j=\exp\left(\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{2b_j^2}\right) \tag{3.4}$$

$$f=\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{h}(\boldsymbol{x})+\varepsilon \tag{3.5}$$

其中，$\boldsymbol{x}$ 为网络的输入，j 为网络隐含层第 j 个节点，$\boldsymbol{h}=[h_j]^{\mathrm{T}}$ 为网络的高斯基函数输出，$\boldsymbol{W}^*$ 为网络的理想权值，ε 为网络的逼近误差，$|\varepsilon|\leqslant\varepsilon_N$。

网络输入取 $\boldsymbol{x}=[x_1\quad x_2]^{\mathrm{T}}$，则网络输出为

$$\hat{f}(\boldsymbol{x})=\hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{x}) \tag{3.6}$$

3.1.3　控制算法设计与分析

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

定义 Lyapunov 函数为

$$V=\frac{1}{2}s^2+\frac{1}{2\gamma}\tilde{\boldsymbol{W}}^{\mathrm{T}}\tilde{\boldsymbol{W}} \tag{3.7}$$

其中，$\gamma>0$，$\tilde{\boldsymbol{W}}=\hat{\boldsymbol{W}}-\boldsymbol{W}^*$。

则

$$\begin{aligned}\dot{V}&=s\dot{s}\ \frac{1}{\gamma}\tilde{\boldsymbol{W}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{W}}}\\&=s(c\dot{e}+f(x)+u-\ddot{x}_d)+\frac{1}{\gamma}\tilde{\boldsymbol{W}}^{\mathrm{T}}\dot{\tilde{\boldsymbol{W}}}\end{aligned}$$

设计控制律为

$$u=-c\dot{e}-\hat{f}(x)+\ddot{x}_d-\eta\mathrm{sgn}(s) \tag{3.8}$$

则

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

取 $\eta>\varepsilon_N$，自适应律为

$$\dot{\tilde{\boldsymbol{W}}}=\gamma s\boldsymbol{h}(\boldsymbol{x}) \tag{3.9}$$

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

可见，控制律中的鲁棒项 $\eta\mathrm{sgn}(s)$的作用是克服神经网络的逼近误差，以保证系统稳定。

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 $V\geqslant 0$，$\dot{V}\leqslant 0$，则当 $t\to\infty$时，V 有界，因此，可以证明 $\hat{\boldsymbol{W}}$ 有界，但无法保证 $\hat{\boldsymbol{W}}$ 收敛于 $\boldsymbol{W}^*$。

3.1.4 仿真实例

考虑如下被控对象

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

其中，$f(x)=10x_1x_2$。

被控对象的初始状态取[0.15　0]，位置指令为 $x_d=\sin t$，控制律采用式(3.8)，自适应律采用式(3.9)，取 $\gamma=1500$，$\eta=1.5$。根据网络输入 x_1 和 x_2 的实际范围设计高斯基函数的参数[2]，参数 $\boldsymbol{c}_j$ 和 b_j 取值分别为 $0.5\times$[−2　−1　0　1　2]和 3.0。仿真程序中为了避免混淆，将 $s=ce+\dot{e}$ 中的 c 写为 λ，取 $\lambda=10$。网络权值中各个元素的初始值取 0.10。仿真结果如图 3-1 和图 3-2 所示。

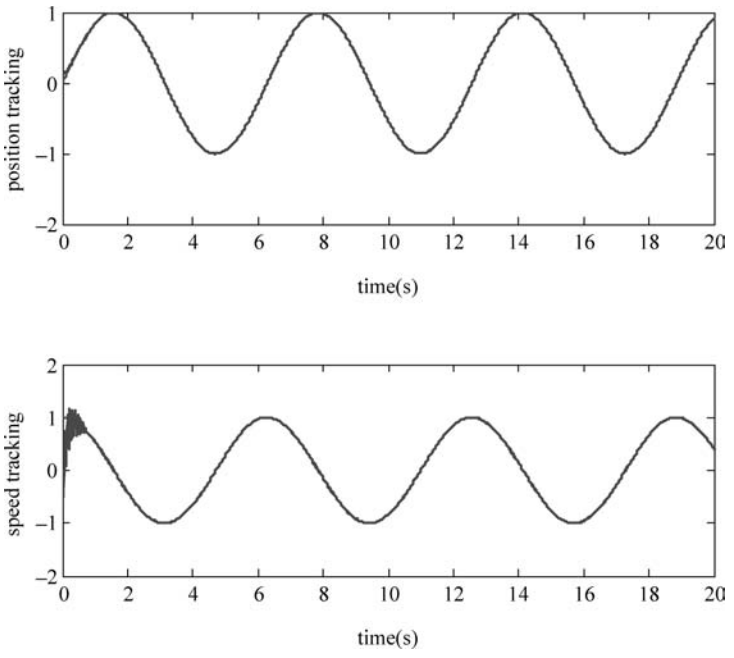


图 3-1　角度和角速度跟踪仿真结果

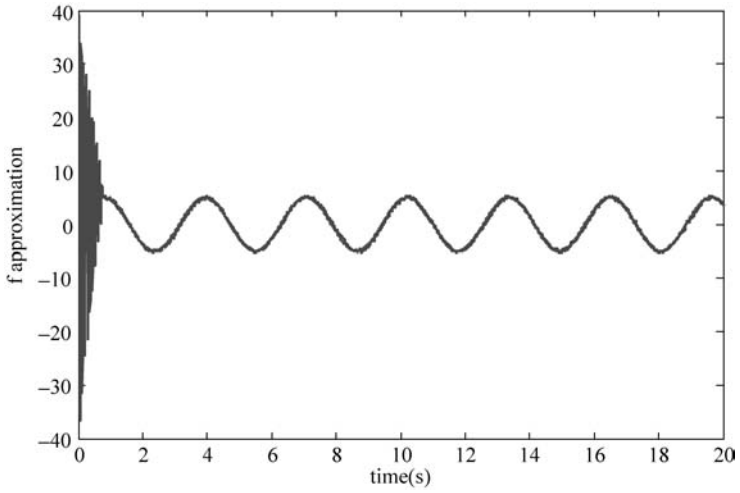
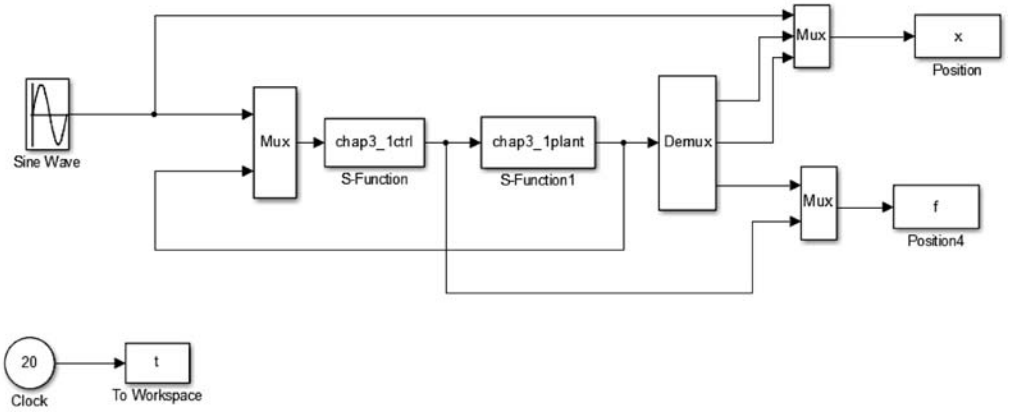


图 3-2　$f(x)$及逼近的仿真结果

仿真程序如下：

(1) Simulink 主程序：chap3_1sim. mdl。



(2) 控制律及自适应律 S 函数：chap3_1ctrl. m。

```
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;
```

(3) 被控对象 S 函数：chap3_1plant. m。

```
function [sys,x0,str,ts] = s_function(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
sizes = simsizes;
sizes.NumContStates  = 2;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 3;
sizes.NumInputs      = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 0;
sys = simsizes(sizes);
x0 = [0.15;0];
str = [];
ts = [];
function sys = mdlDerivatives(t,x,u)
ut = u(1);

f = 10 * x(1) * x(2);
sys(1) = x(2);
sys(2) = f + ut;
function sys = mdlOutputs(t,x,u)
f = 10 * x(1) * x(2);
```

```
sys(1) = x(1);
sys(2) = x(2);
sys(3) = f;
```

(4) 作图程序：chap3_1plot. m。

```
close all;

figure(1);
subplot(211);
plot(t,x(:,1),'r',t,x(:,2),'b','linewidth',2);
xlabel('time(s)');ylabel('position tracking');
subplot(212);
plot(t,cos(t),'r',t,x(:,3),'b','linewidth',2);
xlabel('time(s)');ylabel('speed tracking');

figure(2);
plot(t,f(:,1),'r',t,f(:,3),'b','linewidth',2);
xlabel('time(s)');ylabel('f approximation');
```

3.2 基于 RBF 网络逼近的机械手自适应控制

通过对文献[3]的控制方法进行详细推导及仿真分析，研究一类机械臂神经网络自适应控制的设计方法。

3.2.1 问题的提出

设 n 关节机械手方程为

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{G}(\boldsymbol{q})+\boldsymbol{F}(\dot{\boldsymbol{q}})+\boldsymbol{\tau}_{\mathrm{d}}=\boldsymbol{\tau} \tag{3.10}$$

其中，$\boldsymbol{M}(\boldsymbol{q})$为 $n\times n$ 阶正定惯性矩阵，$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$为 $n\times n$ 阶惯性矩阵，$\boldsymbol{G}(\boldsymbol{q})$为 $n\times 1$ 阶惯性向量，$\boldsymbol{F}(\dot{\boldsymbol{q}})$为摩擦力，$\boldsymbol{\tau}_{\mathrm{d}}$ 为未知外加干扰，$\boldsymbol{\tau}$ 为控制输入。

跟踪误差为

$$\boldsymbol{e}(t)=\boldsymbol{q}_{\mathrm{d}}(t)-\boldsymbol{q}(t)$$

定义滑模函数为

$$\boldsymbol{r}=\dot{\boldsymbol{e}}+\boldsymbol{\Lambda}\boldsymbol{e} \tag{3.11}$$

其中，$\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^{\mathrm{T}}>0$，则

$$\begin{aligned}
\dot{\boldsymbol{q}}&=-\boldsymbol{r}+\dot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\boldsymbol{e}\\
\boldsymbol{M}\dot{\boldsymbol{r}}&=\boldsymbol{M}(\ddot{\boldsymbol{q}}_{\mathrm{d}}-\ddot{\boldsymbol{q}}+\boldsymbol{\Lambda}\dot{\boldsymbol{e}})=\boldsymbol{M}(\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\dot{\boldsymbol{e}})-\boldsymbol{M}\ddot{\boldsymbol{q}}\\
&=\boldsymbol{M}(\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\dot{\boldsymbol{e}})+\boldsymbol{C}\dot{\boldsymbol{q}}+\boldsymbol{G}+\boldsymbol{F}+\boldsymbol{\tau}_{\mathrm{d}}-\boldsymbol{\tau}\\
&=\boldsymbol{M}(\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\dot{\boldsymbol{e}})-\boldsymbol{C}\boldsymbol{r}+\boldsymbol{C}(\dot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\boldsymbol{e})+\boldsymbol{G}+\boldsymbol{F}+\boldsymbol{\tau}_{\mathrm{d}}-\boldsymbol{\tau}\\
&=-\boldsymbol{C}\boldsymbol{r}-\boldsymbol{\tau}+\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{\tau}_{\mathrm{d}}
\end{aligned} \tag{3.12}$$

其中，$\boldsymbol{f}(\boldsymbol{x})=\boldsymbol{M}(\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\dot{\boldsymbol{e}})+\boldsymbol{C}(\boldsymbol{q}_{\mathrm{d}}+\boldsymbol{\Lambda}\boldsymbol{e})+\boldsymbol{G}+\boldsymbol{F}$。

在实际工程中，模型不确定项 $\boldsymbol{f}(\boldsymbol{x})$为未知，为此，需要对不确定项 $\boldsymbol{f}(\boldsymbol{x})$进行逼近。

采用 RBF 网络逼近 $\boldsymbol{f}(\boldsymbol{x})$，根据 $\boldsymbol{f}(\boldsymbol{x})$的表达式，网络输入取

$$\boldsymbol{x}=[\boldsymbol{e}^{\mathrm{T}} \quad \dot{\boldsymbol{e}}^{\mathrm{T}} \quad \boldsymbol{q}_{\mathrm{d}}^{\mathrm{T}} \quad \dot{\boldsymbol{q}}_{\mathrm{d}}^{\mathrm{T}} \quad \ddot{\boldsymbol{q}}_{\mathrm{d}}^{\mathrm{T}}]$$

设计控制律为

$$\boldsymbol{\tau}=\hat{\boldsymbol{f}}(\boldsymbol{x})+\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r} \tag{3.13}$$

其中，$\hat{\boldsymbol{f}}(\boldsymbol{x})$为 RBF 网络的估计值。

将控制律式(3.13)代入式(3.12)，得

$$\begin{aligned}\boldsymbol{M}\dot{\boldsymbol{r}}&=-\boldsymbol{C}\boldsymbol{r}-\hat{\boldsymbol{f}}(\boldsymbol{x})-\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{\tau}_{\mathrm{d}}\\&=-(\boldsymbol{K}_{\mathrm{v}}+\boldsymbol{C})\boldsymbol{r}+\tilde{\boldsymbol{f}}(\boldsymbol{x})+\boldsymbol{\tau}_{\mathrm{d}}=-(\boldsymbol{K}_{\mathrm{v}}+\boldsymbol{C})\boldsymbol{r}+\boldsymbol{\zeta}_0\end{aligned} \tag{3.14}$$

其中，$\tilde{\boldsymbol{f}}(\boldsymbol{x})=\boldsymbol{f}(\boldsymbol{x})-\hat{\boldsymbol{f}}(\boldsymbol{x})$，$\boldsymbol{\zeta}_0=\tilde{\boldsymbol{f}}+\boldsymbol{\tau}_{\mathrm{d}}$。

如果定义 Lyapunov 函数

$$L=\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{r}$$

则

$$\begin{aligned}\dot{L}&=\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{r}}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\dot{\boldsymbol{M}}\boldsymbol{r}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}(\dot{\boldsymbol{M}}-2\boldsymbol{C})\boldsymbol{r}+\boldsymbol{r}^{\mathrm{T}}\boldsymbol{\zeta}_0\\&=\boldsymbol{r}^{\mathrm{T}}\boldsymbol{\zeta}_0-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}\end{aligned}$$

这说明在 $\boldsymbol{K}_{\mathrm{v}}$ 固定条件下，控制系统的稳定依赖于$\boldsymbol{\zeta}_0$，即 $\hat{\boldsymbol{f}}$ 对 $\boldsymbol{f}$ 的逼近精度及干扰$\boldsymbol{\tau}_{\mathrm{d}}$ 的大小。

采用 RBF 网络对不确定项 $\boldsymbol{f}$ 进行逼近。理想的 RBF 网络算法为

$$\phi_j=g(\|\boldsymbol{x}-\boldsymbol{c}_i\|^2/b_j^2)$$

$$y=\boldsymbol{W}\boldsymbol{\varphi}(\boldsymbol{x})$$

其中，$\boldsymbol{x}$ 为网络的输入信号，$\boldsymbol{\varphi}(\boldsymbol{x})=[\phi_1 \quad \phi_2 \quad \cdots \quad \phi_n]$，$j$ 为隐层节点的个数，i 为网络输入的个数。

3.2.2　基于 RBF 网络逼近的控制器

1. 控制器的设计

采用 RBF 网络逼近 $\boldsymbol{f}(\boldsymbol{x})$，则 RBF 神经网络的输出为

$$\hat{\boldsymbol{f}}(\boldsymbol{x})=\hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}) \tag{3.15}$$

取

$$\tilde{\boldsymbol{W}}=\boldsymbol{W}-\hat{\boldsymbol{W}}, \quad \|\boldsymbol{W}\|_{\mathrm{F}}\leqslant W_{\max}$$

设计控制律为

$$\boldsymbol{\tau}=\hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}-\boldsymbol{v} \tag{3.16}$$

其中，$\boldsymbol{v}$ 为用于克服神经网络逼近误差$\boldsymbol{\varepsilon}$ 的鲁棒项。

将控制律式(3.16)代入式(3.12)，得

$$\boldsymbol{M}\dot{\boldsymbol{r}}=-(\boldsymbol{K}_{\mathrm{v}}+\boldsymbol{C})\boldsymbol{r}+\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})+\boldsymbol{v}=-(\boldsymbol{K}_{\mathrm{v}}+\boldsymbol{C})\boldsymbol{r}+\boldsymbol{\zeta}_1 \tag{3.17}$$

其中，$\boldsymbol{\zeta}_1=\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})+\boldsymbol{v}$。

2. 稳定性及收敛性分析

根据控制律式(3.16)中是否有$v(t)$项，$\boldsymbol{\varepsilon}$和$\boldsymbol{\tau}_d$是否存在以及神经网络自适应律设计的不同，系统的收敛性不同。

1) 取$v(t)=0$，存在$\boldsymbol{\varepsilon}$和$\boldsymbol{\tau}_d$的情况

定义 Lyapunov 函数

$$L=\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{r}+\frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{F}^{-1}\tilde{\boldsymbol{W}}) \tag{3.18}$$

则

$$\dot{L}=\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{r}}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\dot{\boldsymbol{M}}\boldsymbol{r}+\mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{F}^{-1}\dot{\tilde{\boldsymbol{W}}})$$

将式(3.17)代入上式，得

$$\dot{L}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_v\boldsymbol{r}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}(\dot{\boldsymbol{M}}-2\boldsymbol{C})\boldsymbol{r}+\mathrm{tr}\tilde{\boldsymbol{W}}^{\mathrm{T}}(\boldsymbol{F}^{-1}\dot{\tilde{\boldsymbol{W}}}+\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}})+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_d) \tag{3.19}$$

考虑机械手特性，并取

$$\dot{\tilde{\boldsymbol{W}}}=-\boldsymbol{F}\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}}$$

即神经网络自适应律为

$$\dot{\hat{\boldsymbol{W}}}=\boldsymbol{F}\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}} \tag{3.20}$$

则

$$\dot{L}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_v\boldsymbol{r}+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_d)\leqslant-K_{v\min}\|\boldsymbol{r}\|^2+(\varepsilon_N+b_d)\|\boldsymbol{r}\|$$

其中，$\|\boldsymbol{\varepsilon}\|\leqslant\varepsilon_N$，$\|\boldsymbol{\tau}_d\|\leqslant b_d$。

当满足下列收敛条件时，$\dot{L}\leqslant 0$：

$$\|\boldsymbol{r}\|\geqslant(\varepsilon_N+b_d)/K_{v\min} \tag{3.21}$$

2) 取$v(t)=0$，$\boldsymbol{\varepsilon}=0$，$\boldsymbol{\tau}_d=0$的情况

Lyapunov 函数为

$$L=\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{r}+\frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{F}^{-1}\tilde{\boldsymbol{W}}) \tag{3.22}$$

此时控制律和自适应律为

$$\boldsymbol{\tau}=\hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+\boldsymbol{K}_v\boldsymbol{r} \tag{3.23}$$

$$\dot{\hat{\boldsymbol{W}}}=\boldsymbol{F}\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}} \tag{3.24}$$

由式(3.17)知

$$\boldsymbol{M}\dot{\boldsymbol{r}}=-(\boldsymbol{K}_v+\boldsymbol{C})r+\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})$$

则

$$\dot{L}=\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{r}}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\dot{\boldsymbol{M}}\boldsymbol{r}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_v\boldsymbol{r}\leqslant 0$$

3) 取$\boldsymbol{v}(t)=0$，存在$\boldsymbol{\varepsilon}$和$\boldsymbol{\tau}_d$，自适应律采取 UUB 的形式

Lyapunov 函数和控制律取式(3.22)和式(3.23)。

自适应律为

$$\dot{\hat{\boldsymbol{W}}}=\boldsymbol{F}\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}}-k\boldsymbol{F}\|\boldsymbol{r}\|\hat{\boldsymbol{W}} \tag{3.25}$$

则根据式(3.19),有

$$\dot{L}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}(\dot{\boldsymbol{M}}-2\boldsymbol{C})\boldsymbol{r}+\mathrm{tr}\tilde{\boldsymbol{W}}^{\mathrm{T}}(\boldsymbol{F}^{-1}\dot{\tilde{\boldsymbol{W}}}+\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}})+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})$$

将式(3.25)代入上式,得

$$\begin{aligned}\dot{L}&=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+\mathrm{tr}\tilde{\boldsymbol{W}}^{\mathrm{T}}(-\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}}+k\|\boldsymbol{r}\|\hat{\boldsymbol{W}}+\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}})+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})\\&=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+k\|\boldsymbol{r}\|\mathrm{tr}\tilde{\boldsymbol{W}}^{\mathrm{T}}(\boldsymbol{W}-\tilde{\boldsymbol{W}})+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})\end{aligned}$$

由于

$$\mathrm{tr}\tilde{\boldsymbol{W}}^{\mathrm{T}}(\boldsymbol{W}-\tilde{\boldsymbol{W}})=(\tilde{\boldsymbol{W}},\boldsymbol{W})_{\mathrm{F}}-\|\boldsymbol{W}\|_{\mathrm{F}}^{2}\leqslant\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}\|\boldsymbol{W}\|_{\mathrm{F}}-\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}^{2}$$

则

$$\begin{aligned}\dot{L}&\leqslant-K_{\mathrm{vmin}}\|\boldsymbol{r}\|^{2}+k\|\boldsymbol{r}\|\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}(W_{\max}-\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}})+(\varepsilon_{\mathrm{N}}+b_{\mathrm{d}})\|\boldsymbol{r}\|\\&=-\|\boldsymbol{r}\|(K_{\mathrm{vmin}}\|\boldsymbol{r}\|+k\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}(\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}-W_{\max})-(\varepsilon_{\mathrm{N}}+b_{\mathrm{d}}))\end{aligned}$$

由于

$$\begin{aligned}&K_{\mathrm{vmin}}\|\boldsymbol{r}\|+k\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}(\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}-W_{\max})-(\varepsilon_{\mathrm{N}}+b_{\mathrm{d}})\\&=k(\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}-W_{\max}/2)^{2}-kW_{\max}^{2}/4+K_{\mathrm{vmin}}\|\boldsymbol{r}\|-(\varepsilon_{\mathrm{N}}+b_{\mathrm{d}})\end{aligned}$$

则要使 $\dot{L}\leqslant0$,需要

$$\|\boldsymbol{r}\|\geqslant\frac{kW_{\max}^{2}/4+(\varepsilon_{\mathrm{N}}+b_{\mathrm{d}})}{K_{\mathrm{vmin}}}\tag{3.26}$$

或

$$\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}\geqslant W_{\max}/2+\sqrt{W_{\max}^{2}/4+(\varepsilon_{\mathrm{N}}+b_{\mathrm{d}})/k}\tag{3.27}$$

4) 存在$\boldsymbol{\varepsilon}$ 和$\boldsymbol{\tau}_{\mathrm{d}}$,考虑鲁棒项$\boldsymbol{v}(t)$设计的情况

将鲁棒项$\boldsymbol{v}$ 设计为

$$\boldsymbol{v}=-(\varepsilon_{\mathrm{N}}+b_{\mathrm{d}})\mathrm{sgn}(\boldsymbol{r})\tag{3.28}$$

控制律取式(3.16),神经网络自适应律取式(3.20)。

定义 Lyapunov 函数为

$$L=\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{r}+\frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{F}^{-1}\tilde{\boldsymbol{W}})$$

则

$$\dot{L}=\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{r}}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\dot{\boldsymbol{M}}\boldsymbol{r}+\mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{F}^{-1}\dot{\tilde{\boldsymbol{W}}})$$

将(3.17)式代入上式,得

$$\dot{L}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}(\dot{\boldsymbol{M}}-2\boldsymbol{C})\boldsymbol{r}+\mathrm{tr}\tilde{\boldsymbol{W}}^{\mathrm{T}}(\boldsymbol{F}^{-1}\dot{\tilde{\boldsymbol{W}}}+\boldsymbol{\varphi}\boldsymbol{r}^{\mathrm{T}})+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}}+\boldsymbol{v})$$

则

$$\dot{L}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}}+\boldsymbol{v})$$

由于

$$\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}}+\boldsymbol{v})=\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})+\boldsymbol{r}^{\mathrm{T}}\boldsymbol{v}=\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})-\|\boldsymbol{r}\|(\boldsymbol{\varepsilon}_{\mathrm{N}}+\boldsymbol{b}_{\mathrm{d}})\leqslant0$$

则

$$\dot{L}\leqslant 0$$

针对以上 4 种情况，由于当 $\dot{L}\equiv 0$ 时，$r\equiv 0$，根据 LaSalle 不变性原理，闭环系统渐进稳定，$t\to\infty$时，$r\to 0$。由于 $L\geqslant 0$，$\dot{L}\leqslant 0$，则 L 有界，从而$\widetilde{\boldsymbol{W}}$ 有界，但无法保证$\widetilde{\boldsymbol{W}}$ 收敛于 0。

3.2.3 针对 $f(x)$中各项分别进行神经网络逼近

控制律为

$$\boldsymbol{\tau}=\hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}-\boldsymbol{v} \tag{3.29}$$

鲁棒项$\boldsymbol{v}$ 取式(3.28)。由式(3.12)知，被控对象中的 $\boldsymbol{f}(\boldsymbol{x})$项可写为

$$\boldsymbol{f}(\boldsymbol{x})=\boldsymbol{M}(\boldsymbol{q})\boldsymbol{\zeta}_1(t)+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\zeta}_2(t)+\boldsymbol{G}(\boldsymbol{q})+\boldsymbol{F}(\dot{\boldsymbol{q}})$$

其中，$\boldsymbol{\zeta}_1(t)=\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\dot{\boldsymbol{e}}$，$\boldsymbol{\zeta}_2(t)=\dot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}\boldsymbol{e}$。

采用 RBF 神经网络，可以对 $\boldsymbol{f}(\boldsymbol{x})$中的各项分别进行逼近：

$$\hat{\boldsymbol{M}}(\boldsymbol{q})=\boldsymbol{W}_{\mathrm{M}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathrm{M}}(\boldsymbol{q})$$

$$\hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})=\boldsymbol{W}_{\mathrm{V}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathrm{V}}(\boldsymbol{q},\dot{\boldsymbol{q}})$$

$$\hat{\boldsymbol{G}}(\boldsymbol{q})=\boldsymbol{W}_{\mathrm{G}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathrm{G}}(\boldsymbol{q})$$

$$\hat{\boldsymbol{F}}(\dot{\boldsymbol{q}})=\boldsymbol{W}_{\mathrm{F}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathrm{F}}(\dot{\boldsymbol{q}})$$

则

$$\hat{\boldsymbol{f}}(\boldsymbol{x})=\begin{bmatrix}\boldsymbol{W}_{\mathrm{M}}^{\mathrm{T}}\boldsymbol{\zeta}_1(t) & \boldsymbol{W}_{\mathrm{V}}^{\mathrm{T}}\boldsymbol{\zeta}_2(t) & \boldsymbol{W}_{\mathrm{G}}^{\mathrm{T}} & \boldsymbol{W}_{\mathrm{F}}^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\varphi}_{\mathrm{M}}\\ \boldsymbol{\varphi}_{\mathrm{V}}\\ \boldsymbol{\varphi}_{\mathrm{G}}\\ \boldsymbol{\varphi}_{\mathrm{F}}\end{bmatrix} \tag{3.30}$$

其中，$\boldsymbol{\varphi}(x)=\begin{bmatrix}\boldsymbol{\varphi}_{\mathrm{M}}\\ \boldsymbol{\varphi}_{\mathrm{V}}\\ \boldsymbol{\varphi}_{\mathrm{G}}\\ \boldsymbol{\varphi}_{\mathrm{F}}\end{bmatrix}$，$\boldsymbol{W}^{\mathrm{T}}=\begin{bmatrix}\boldsymbol{W}_{\mathrm{M}}^{\mathrm{T}} & \boldsymbol{W}_{\mathrm{V}}^{\mathrm{T}} & \boldsymbol{W}_{\mathrm{G}}^{\mathrm{T}} & \boldsymbol{W}_{\mathrm{F}}^{\mathrm{T}}\end{bmatrix}$。

自适应律为

$$\dot{\hat{\boldsymbol{W}}}_{\mathrm{M}}=\boldsymbol{F}_{\mathrm{M}}\boldsymbol{\varphi}_{\mathrm{M}}\boldsymbol{r}^{\mathrm{T}}-k_{\mathrm{M}}\boldsymbol{F}_{\mathrm{M}}\|\boldsymbol{r}\|\hat{\boldsymbol{W}}_{\mathrm{M}} \tag{3.31}$$

$$\dot{\hat{\boldsymbol{W}}}_{\mathrm{V}}=\boldsymbol{F}_{\mathrm{V}}\boldsymbol{\varphi}_{\mathrm{V}}\boldsymbol{r}^{\mathrm{T}}-k_{\mathrm{V}}\boldsymbol{F}_{\mathrm{V}}\|\boldsymbol{r}\|\hat{\boldsymbol{W}}_{\mathrm{V}} \tag{3.32}$$

$$\dot{\hat{\boldsymbol{W}}}_{\mathrm{G}}=\boldsymbol{F}_{\mathrm{G}}\boldsymbol{\varphi}_{\mathrm{G}}\boldsymbol{r}^{\mathrm{T}}-k_{\mathrm{G}}\boldsymbol{F}_{\mathrm{G}}\|\boldsymbol{r}\|\hat{\boldsymbol{W}}_{\mathrm{G}} \tag{3.33}$$

$$\dot{\hat{\boldsymbol{W}}}_{\mathrm{F}}=\boldsymbol{F}_{\mathrm{F}}\boldsymbol{\varphi}_{\mathrm{F}}\boldsymbol{r}^{\mathrm{T}}-k_{\mathrm{F}}\boldsymbol{F}_{\mathrm{F}}\|\boldsymbol{r}\|\hat{\boldsymbol{W}}_{\mathrm{F}} \tag{3.34}$$

其中，$k_{\mathrm{M}}>0$，$k_{\mathrm{V}}>0$，$k_{\mathrm{G}}>0$，$k_{\mathrm{F}}>0$。

稳定性分析如下：

定义 Lyapunov 函数为

$$L=\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{r}+\frac{1}{2}\mathrm{tr}(\widetilde{\boldsymbol{W}}_{\mathrm{M}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{M}}^{-1}\widetilde{\boldsymbol{W}}_{\mathrm{M}})+\frac{1}{2}\mathrm{tr}(\widetilde{\boldsymbol{W}}_{\mathrm{V}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{V}}^{-1}\widetilde{\boldsymbol{W}}_{\mathrm{V}})+$$

$$\frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{W}}_{\mathrm{G}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{G}}^{-1}\tilde{\boldsymbol{W}}_{\mathrm{G}})+\frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{W}}_{\mathrm{F}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{F}}^{-1}\tilde{\boldsymbol{W}}_{\mathrm{F}})$$

则

$$\dot{L}=\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}\dot{\boldsymbol{r}}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}\dot{\boldsymbol{M}}\boldsymbol{r}+\mathrm{tr}(\tilde{\boldsymbol{W}}_{\mathrm{M}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{M}}^{-1}\dot{\tilde{\boldsymbol{W}}}_{\mathrm{M}})+\mathrm{tr}(\tilde{\boldsymbol{W}}_{\mathrm{V}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{V}}^{-1}\dot{\tilde{\boldsymbol{W}}}_{\mathrm{V}})+$$

$$\mathrm{tr}(\tilde{\boldsymbol{W}}_{\mathrm{G}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{G}}^{-1}\dot{\tilde{\boldsymbol{W}}}_{\mathrm{G}})+\mathrm{tr}(\tilde{\boldsymbol{W}}_{\mathrm{F}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{F}}^{-1}\dot{\tilde{\boldsymbol{W}}}_{\mathrm{F}})$$

将式(3.17)代入上式，得

$$\dot{L}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+\frac{1}{2}\boldsymbol{r}^{\mathrm{T}}(\dot{\boldsymbol{M}}-2\boldsymbol{V}_{\mathrm{m}})\boldsymbol{r}+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})+\boldsymbol{r}^{\mathrm{T}}\boldsymbol{v}+\mathrm{tr}\tilde{\boldsymbol{W}}_{\mathrm{M}}^{\mathrm{T}}(\boldsymbol{F}_{\mathrm{M}}^{-1}\dot{\tilde{\boldsymbol{W}}}_{\mathrm{M}}+\boldsymbol{\varphi}_{\mathrm{M}}\boldsymbol{r}^{\mathrm{T}})+$$

$$\mathrm{tr}\tilde{\boldsymbol{W}}_{\mathrm{V}}^{\mathrm{T}}(\boldsymbol{F}_{\mathrm{V}}^{-1}\dot{\tilde{\boldsymbol{W}}}_{\mathrm{V}}+\boldsymbol{\varphi}_{\mathrm{V}}\boldsymbol{r}^{\mathrm{T}})+\mathrm{tr}\tilde{\boldsymbol{W}}_{\mathrm{G}}^{\mathrm{T}}(\boldsymbol{F}_{\mathrm{G}}^{-1}\dot{\tilde{\boldsymbol{W}}}_{\mathrm{G}}+\boldsymbol{\varphi}_{M}\boldsymbol{r}^{\mathrm{T}})+\mathrm{tr}\tilde{\boldsymbol{W}}_{\mathrm{F}}^{\mathrm{T}}(\boldsymbol{F}_{\mathrm{F}}^{-1}\dot{\tilde{\boldsymbol{W}}}_{\mathrm{F}}+\boldsymbol{\varphi}_{\mathrm{F}}\boldsymbol{r}^{\mathrm{T}}) \tag{3.35}$$

考虑机械手特性，并将神经网络自适应律式(3.31)～式(3.34)代入上式，得

$$\dot{L}=-\boldsymbol{r}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{v}}\boldsymbol{r}+k_{\mathrm{M}}\|\boldsymbol{r}\|\,\mathrm{tr}\tilde{\boldsymbol{W}}_{\mathrm{M}}^{\mathrm{T}}(\boldsymbol{W}_{\mathrm{M}}-\tilde{\boldsymbol{W}}_{\mathrm{M}})+k_{\mathrm{V}}\|\boldsymbol{r}\|\,\mathrm{tr}\tilde{\boldsymbol{W}}_{\mathrm{V}}^{\mathrm{T}}(\boldsymbol{W}_{\mathrm{V}}-\tilde{\boldsymbol{W}}_{\mathrm{V}})+$$

$$k_{\mathrm{G}}\|\boldsymbol{r}\|\,\mathrm{tr}\tilde{\boldsymbol{W}}_{\mathrm{G}}^{\mathrm{T}}(\boldsymbol{W}_{\mathrm{G}}-\tilde{\boldsymbol{W}}_{G})+k_{\mathrm{F}}\|\boldsymbol{r}\|\,\mathrm{tr}\tilde{\boldsymbol{W}}_{\mathrm{F}}^{\mathrm{T}}(\boldsymbol{W}_{\mathrm{F}}-\tilde{\boldsymbol{W}}_{\mathrm{F}})+\boldsymbol{r}^{\mathrm{T}}(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})+\boldsymbol{r}^{\mathrm{T}}\boldsymbol{v}$$

由于

$$\mathrm{tr}\tilde{\boldsymbol{W}}^{\mathrm{T}}(\boldsymbol{W}-\tilde{\boldsymbol{W}})=(\tilde{\boldsymbol{W}},\boldsymbol{W})_{\mathrm{F}}-\|\boldsymbol{W}\|_{\mathrm{F}}^{2}\leqslant\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}\|\boldsymbol{W}\|_{\mathrm{F}}-\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}^{2}$$

考虑鲁棒项(3.28)，则

$$\begin{aligned}\dot{L}\leqslant&-K_{\mathrm{vmin}}\|\boldsymbol{r}\|^{2}+k_{\mathrm{M}}\|\boldsymbol{r}\|\,\|\tilde{\boldsymbol{W}}_{\mathrm{M}}\|_{\mathrm{F}}(W_{\mathrm{Mmax}}-\|\tilde{\boldsymbol{W}}_{\mathrm{M}}\|_{\mathrm{F}})+\\&k_{\mathrm{V}}\|\boldsymbol{r}\|\,\|\tilde{\boldsymbol{W}}_{\mathrm{V}}\|_{\mathrm{F}}(W_{\mathrm{Vmax}}-\|\tilde{\boldsymbol{W}}_{\mathrm{V}}\|_{\mathrm{F}})+\\&k_{\mathrm{G}}\|\boldsymbol{r}\|\,\|\tilde{\boldsymbol{W}}_{\mathrm{G}}\|_{\mathrm{F}}(W_{\mathrm{Gmax}}-\|\tilde{\boldsymbol{W}}_{\mathrm{G}}\|_{\mathrm{F}})+k_{\mathrm{M}}\|\boldsymbol{r}\|\,\|\tilde{\boldsymbol{W}}_{\mathrm{F}}\|_{\mathrm{F}}(W_{\mathrm{Fmax}}-\|\tilde{\boldsymbol{W}}_{\mathrm{F}}\|_{\mathrm{F}})\\=&-\|\boldsymbol{r}\|[K_{\mathrm{vmin}}\|\boldsymbol{r}\|+k_{\mathrm{M}}\|\tilde{\boldsymbol{W}}_{\mathrm{M}}\|_{\mathrm{F}}(\|\tilde{\boldsymbol{W}}_{\mathrm{M}}\|_{\mathrm{F}}-W_{\mathrm{Mmax}})+\\&k_{\mathrm{V}}\|\tilde{\boldsymbol{W}}_{\mathrm{V}}\|_{\mathrm{F}}(\|\tilde{\boldsymbol{W}}_{\mathrm{V}}\|_{\mathrm{F}}-W_{\mathrm{Vmax}})+k_{\mathrm{G}}\|\tilde{\boldsymbol{W}}_{\mathrm{G}}\|_{\mathrm{F}}(\|\tilde{\boldsymbol{W}}_{\mathrm{G}}\|_{\mathrm{F}}-W_{\mathrm{Gmax}})+\\&k_{\mathrm{F}}\|\tilde{\boldsymbol{W}}_{\mathrm{F}}\|_{\mathrm{F}}(\|\tilde{\boldsymbol{W}}_{\mathrm{F}}\|_{\mathrm{F}}-W_{\mathrm{Fmax}})]\end{aligned}$$

由于

$$k\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}(\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}-W_{\max})=k(\|\tilde{\boldsymbol{W}}\|_{\mathrm{F}}-W_{\max}/2)^{2}-kW_{\max}^{2}/4$$

要使 $\dot{L}\leqslant 0$，需要

$$\|\boldsymbol{r}\|\geqslant\frac{k_{\mathrm{M}}W_{\mathrm{Mmax}}^{2}/4+k_{\mathrm{V}}W_{\mathrm{Vmax}}^{2}/4+k_{\mathrm{G}}W_{\mathrm{Gmax}}^{2}/4+k_{\mathrm{F}}W_{\mathrm{Fmax}}^{2}/4}{K_{\mathrm{vmin}}} \tag{3.36}$$

或

$$\|\tilde{\boldsymbol{W}}_{\mathrm{M}}\|_{\mathrm{F}}\leqslant W_{\mathrm{Mmax}}\text{ 且 }\|\tilde{\boldsymbol{W}}_{\mathrm{V}}\|_{\mathrm{F}}\leqslant W_{\mathrm{Vmax}}\text{ 且 }\|\tilde{\boldsymbol{W}}_{\mathrm{G}}\|_{\mathrm{F}}\leqslant W_{\mathrm{Gmax}}\text{ 且 }\|\tilde{\boldsymbol{W}}_{\mathrm{F}}\|_{\mathrm{F}}\leqslant W_{\mathrm{Fmax}} \tag{3.37}$$

考虑 $\dot{L}\leqslant 0$ 时，如取 k_{V} 足够大，当 $t\to\infty$ 时，$\boldsymbol{r}\to 0$，从而 $\boldsymbol{e}\to 0$，$\dot{\boldsymbol{e}}\to 0$。由于 $L\geqslant 0$，$\dot{L}\leqslant 0$，则 L 有界，从而 $\boldsymbol{r}$ 和 $\tilde{W}_{\mathrm{M}}$、$\tilde{W}_{\mathrm{V}}$、$\tilde{W}_{\mathrm{G}}$、$\tilde{W}_{\mathrm{F}}$ 有界，但无法保证 $\tilde{W}_{\mathrm{M}}$、$\tilde{W}_{\mathrm{V}}$、$\tilde{W}_{\mathrm{G}}$、$\tilde{W}_{\mathrm{F}}$ 收敛于零。

3.2.4　仿真实例

选择二关节机机械臂系统，其动力学模型为

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{G}(\boldsymbol{q})+\boldsymbol{F}(\dot{\boldsymbol{q}})+\boldsymbol{\tau}_{\mathrm{d}}=\boldsymbol{\tau}$$

其中，

$$\boldsymbol{M}(\boldsymbol{q})=\begin{bmatrix}p_1+p_2+2p_3\cos q_2 & p_2+p_3\cos q_2\\ p_2+p_3\cos q_2 & p_2\end{bmatrix}$$

$$\boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}})=\begin{bmatrix}-p_3\dot{q}_2\sin q_2 & -p_3(\dot{q}_1+\dot{q}_2)\sin q_2\\ p_3\dot{q}_1\sin q_2 & 0\end{bmatrix}$$

$$\boldsymbol{G}(\boldsymbol{q})=\begin{bmatrix}p_4 g\cos q_1+p_5 g\cos(q_1+q_2)\\ p_5 g\cos(q_1+q_2)\end{bmatrix}$$

$$\boldsymbol{F}(\dot{\boldsymbol{q}})=0.2\mathrm{sgn}(\dot{\boldsymbol{q}}),\quad \boldsymbol{\tau}_d=[0.1\sin(t)\quad 0.1\sin(t)]^{\mathrm{T}}$$

取 $\boldsymbol{p}=[p_1,p_2,p_3,p_4,p_5]=[2.9,0.76,0.87,3.04,0.87]$。RBF 网络高斯基函数参数的取值对神经网络控制的作用很重要，如果参数取值不合适，将使高斯基函数无法得到有效的映射，从而导致 RBF 网络无效。故 c 按网络输入值的范围取值，取 $b_j=0.20$，

$$\boldsymbol{c}=0.1\times\begin{bmatrix}-1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5\\ -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5\\ -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5\\ -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5\\ -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5\end{bmatrix}$$

，网络的初始权值取零，网络输入取 $\boldsymbol{z}=[\boldsymbol{e}\quad\dot{\boldsymbol{e}}\quad\boldsymbol{q}_d\quad\dot{\boldsymbol{q}}_d\quad\ddot{\boldsymbol{q}}_d]$。

系统的初始状态为[0.09　0　−0.09　0]，两个关节的角度指令分别为 $q_{1d}=0.1\sin t$，$q_{2d}=0.1\sin t$，控制参数取 $\boldsymbol{K}_v=\mathrm{diag}\{20,20\}$，$\boldsymbol{F}=\mathrm{diag}\{1.5,1.5\}$，$\boldsymbol{\Lambda}=\mathrm{diag}\{5,5\}$，在鲁棒项中，取 $\varepsilon_N=0.20$，$b_d=0.10$。

采用 Simulink 和 S 函数进行控制系统的设计，神经网络权值矩阵中任意元素初值取 0.10。总体逼近控制器子程序 chap3_9ctrl.m，按 3.2.2 节第 4 种情况设计控制律，控制律取式(3.16)，v 取式(3.28)，自适应律取式(3.20)。采用总体逼近控制器，仿真结果如图 3-3～图 3-6 所示。

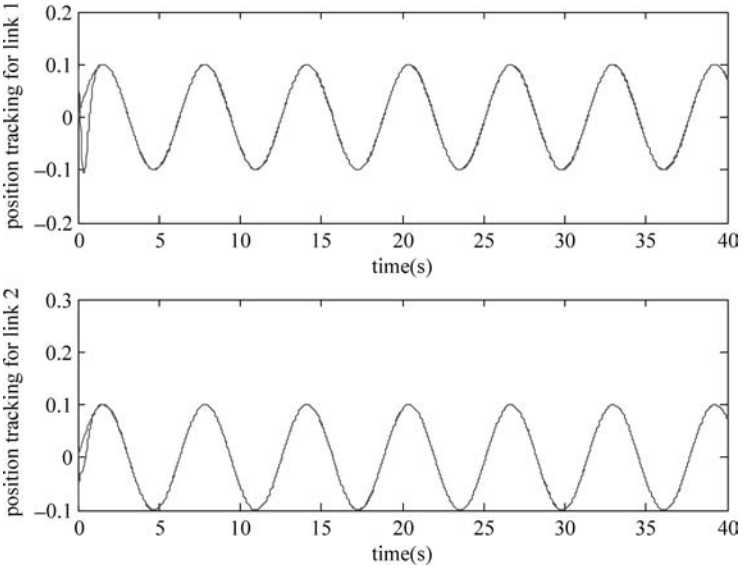


图 3-3　关节 1 及关节 2 的角度跟踪仿真结果

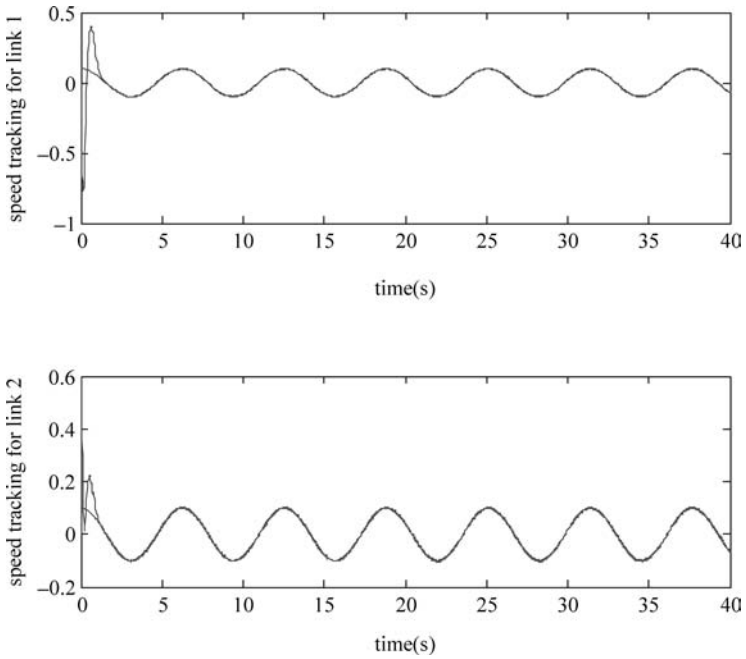


图 3-4　关节 1 及关节 2 的角速度跟踪仿真结果

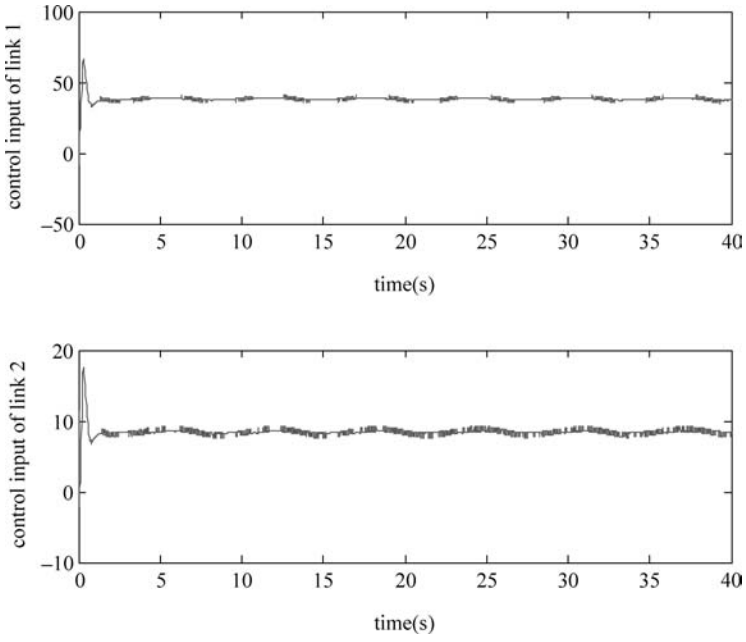


图 3-5　关节 1 及关节 2 的控制输入仿真结果

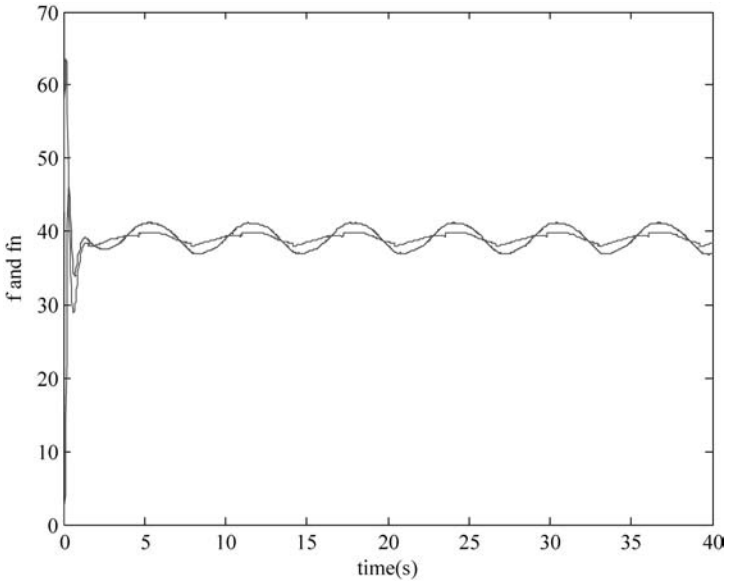
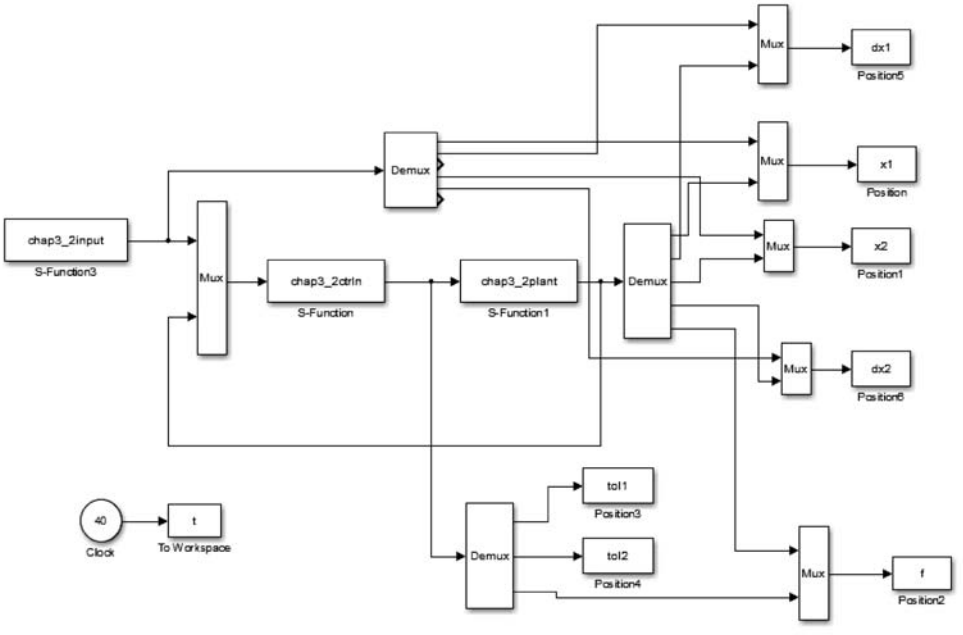


图 3-6　关节 1 及关节 2 的 $\|f(x)\|$ 及其逼近 $\|\hat{f}(x)\|$ 的仿真结果

仿真程序如下：

(1) Simulink 主程序：chap3_2sim. mdl。



(2) 位置指令子程序：chap3_2input. m。

```
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
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 6;
sizes.NumInputs      = 0;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
qd1 = 0.1 * sin(t);
d_qd1 = 0.1 * cos(t);
dd_qd1 = -0.1 * sin(t);
qd2 = 0.1 * sin(t);
d_qd2 = 0.1 * cos(t);
dd_qd2 = -0.1 * sin(t);

sys(1) = qd1;
sys(2) = d_qd1;
sys(3) = dd_qd1;
sys(4) = qd2;
sys(5) = d_qd2;
sys(6) = dd_qd2;
```

(3) 总体逼近控制器子程序：chap3_2ctrl.m。

```
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 node c b Fai
node = 7;
c = 0.1 * [ - 1.5  - 1  - 0.5 0 0.5 1 1.5;
        - 1.5  - 1  - 0.5 0 0.5 1 1.5;
        - 1.5  - 1  - 0.5 0 0.5 1 1.5;
        - 1.5  - 1  - 0.5 0 0.5 1 1.5;
        - 1.5  - 1  - 0.5 0 0.5 1 1.5];
b = 10;
Fai = 5 * eye(2);

sizes = simsizes;
sizes.NumContStates  = 2 * node;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 3;
sizes.NumInputs      = 11;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 0;
sys = simsizes(sizes);
x0  = 0.1 * ones(1,2 * node);
str = [];
ts  = [];
function sys = mdlDerivatives(t,x,u)
global node c b Fai
qd1 = u(1);
d_qd1 = u(2);
dd_qd1 = u(3);
qd2 = u(4);
d_qd2 = u(5);
dd_qd2 = u(6);

q1 = u(7);
d_q1 = u(8);
q2 = u(9);
d_q2 = u(10);

q = [q1;q2];

e1 = qd1 - q1;
e2 = qd2 - q2;
de1 = d_qd1 - d_q1;
de2 = d_qd2 - d_q2;
e = [e1;e2];
de = [de1;de2];
r = de + Fai * e;

qd = [qd1;qd2];
dqd = [d_qd1;d_qd2];
dqr = dqd + Fai * e;
ddqd = [dd_qd1;dd_qd2];
ddqr = ddqd + Fai * de;
```

```
z1 = [e(1);de(1);qd(1);dqd(1);ddqd(1)];
z2 = [e(2);de(2);qd(2);dqd(2);ddqd(2)];
for j = 1:1:node
    h1(j) = exp( - norm(z1 - c(:,j))^2/(b * b));
    h2(j) = exp( - norm(z2 - c(:,j))^2/(b * b));
end

F = 1.5 * eye(node);

    for i = 1:1:node
        sys(i) = 1.5 * h1(i) * r(1);
        sys(i + node) = 1.5 * h2(i) * r(2);
    end

function sys = mdlOutputs(t,x,u)
global node c b Fai

qd1 = u(1);
d_qd1 = u(2);
dd_qd1 = u(3);
qd2 = u(4);
d_qd2 = u(5);
dd_qd2 = u(6);

q1 = u(7);
d_q1 = u(8);
q2 = u(9);
d_q2 = u(10);

q = [q1;q2];

e1 = qd1 - q1;
e2 = qd2 - q2;
de1 = d_qd1 - d_q1;
de2 = d_qd2 - d_q2;
e = [e1;e2];
de = [de1;de2];
r = de + Fai * e;

qd = [qd1;qd2];
dqd = [d_qd1;d_qd2];
dqr = dqd + Fai * e;
ddqd = [dd_qd1;dd_qd2];
ddqr = ddqd + Fai * de;

z = [e;de;qd;dqd;ddqd];
W_f1 = [x(1:node)]';
W_f2 = [x(node + 1:node * 2)]';

z1 = [e(1);de(1);qd(1);dqd(1);ddqd(1)];
```

```
z2 = [e(2);de(2);qd(2);dqd(2);ddqd(2)];
for j = 1:1:node
    h1(j) = exp( - norm(z1 - c(:,j))^2/(b * b));
    h2(j) = exp( - norm(z2 - c(:,j))^2/(b * b));
end

fn = [W_f1 * h1';
    W_f2 * h2'];
Kv = 20 * eye(2);

    epN = 0.20;
    bd = 0.1;
    v = - (epN + bd) * sign(r);
    tol = fn + Kv * r - v;
fn_norm = norm(fn);

sys(1) = tol(1);
sys(2) = tol(2);
sys(3) = fn_norm;
```

(4) 被控对象子程序：chap3_2plant.m。

```
function [sys,x0,str,ts] = s_function(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 p g
sizes = simsizes;
sizes.NumContStates  = 4;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 5;
sizes.NumInputs      = 3;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 0;
sys = simsizes(sizes);
x0 = [0.09 0 - 0.09 0];
str = [];
ts = [];

p = [2.9 0.76 0.87 3.04 0.87];
```

```
g = 9.8;
function sys = mdlDerivatives(t,x,u)
global p g

M = [p(1) + p(2) + 2 * p(3) * cos(x(3)) p(2) + p(3) * cos(x(3));
     p(2) + p(3) * cos(x(3)) p(2)];
V = [ - p(3) * x(4) * sin(x(3)) - p(3) * (x(2) + x(4)) * sin(x(3));
      p(3) * x(2) * sin(x(3))  0];
G = [p(4) * g * cos(x(1)) + p(5) * g * cos(x(1) + x(3));
     p(5) * g * cos(x(1) + x(3))];
dq = [x(2);x(4)];
F = 0.2 * sign(dq);
told = [0.1 * sin(t);0.1 * sin(t)];

tol = u(1:2);

S = inv(M) * (tol - V * dq - G - F - told);

sys(1) = x(2);
sys(2) = S(1);
sys(3) = x(4);
sys(4) = S(2);
function sys = mdlOutputs(t,x,u)
global p g
M = [p(1) + p(2) + 2 * p(3) * cos(x(3)) p(2) + p(3) * cos(x(3));
     p(2) + p(3) * cos(x(3)) p(2)];
V = [ - p(3) * x(4) * sin(x(3)) - p(3) * (x(2) + x(4)) * sin(x(3));
      p(3) * x(2) * sin(x(3))  0];
G = [p(4) * g * cos(x(1)) + p(5) * g * cos(x(1) + x(3));
     p(5) * g * cos(x(1) + x(3))];
dq = [x(2);x(4)];
F = 0.2 * sign(dq);
told = [0.1 * sin(t);0.1 * sin(t)];

qd1 = sin(t);
d_qd1 = cos(t);
dd_qd1 = - sin(t);
qd2 = sin(t);
d_qd2 = cos(t);
dd_qd2 = - sin(t);
qd1 = 0.1 * sin(t);
d_qd1 = 0.1 * cos(t);
dd_qd1 = - 0.1 * sin(t);
qd2 = 0.1 * sin(t);
d_qd2 = 0.1 * cos(t);
dd_qd2 = - 0.1 * sin(t);

q1 = x(1);
d_q1 = dq(1);
q2 = x(3);
d_q2 = dq(2);
```

```
q = [q1;q2];
e1 = qd1 - q1;
e2 = qd2 - q2;
de1 = d_qd1 - d_q1;
de2 = d_qd2 - d_q2;
e = [e1;e2];
de = [de1;de2];
Fai = 5 * eye(2);
dqd = [d_qd1;d_qd2];
dqr = dqd + Fai * e;
ddqd = [dd_qd1;dd_qd2];
ddqr = ddqd + Fai * de;
f = M * ddqr + V * dqr + G + F;
f_norm = norm(f);

sys(1) = x(1);
sys(2) = x(2);
sys(3) = x(3);
sys(4) = x(4);
sys(5) = f_norm;
```

(5) 绘图子程序：chap3_2plot. m。

```
close all;

figure(1);
subplot(211);
plot(t,x1(:,1),'r',t,x1(:,2),'b');
xlabel('time(s)');ylabel('position tracking for link 1');
subplot(212);
plot(t,x2(:,1),'r',t,x2(:,2),'b');
xlabel('time(s)');ylabel('position tracking for link 2');

figure(2);
subplot(211);
plot(t,dx1(:,1),'r',t,dx1(:,2),'b');
xlabel('time(s)');ylabel('speed tracking for link 1');
subplot(212);
plot(t,dx2(:,1),'r',t,dx2(:,2),'b');
xlabel('time(s)');ylabel('speed tracking for link 2');

figure(3);
subplot(211);
plot(t,tol1(:,1),'r');
xlabel('time(s)');ylabel('control input of link 1');
subplot(212);
plot(t,tol2(:,1),'r');
xlabel('time(s)');ylabel('control input of link 2');

figure(4);
plot(t,f(:,1),'r',t,f(:,2),'b');
xlabel('time(s)');ylabel('f and fn');
```

参考文献

[1] Park J, Sandberg I W. Universal approximation using radial basis function networks[J]. Neural Computation, 1991, 3(2): 246-257.

[2] 刘金琨. RBF 神经网络自适应控制 MATLAB 仿真[M]. 北京：清华大学出版社, 2014.

[3] Lewis F L, Liu K, Yesildirek A. Neural net robot controller with guaranteed tracking. performance [J]. IEEE Transactions on Neural Networks, 1995, 6(3): 703-715.