9.1 概述

神经网络是一种具有高度非线性的连续时间动力系统，它有着很强的自学习功能和对非线性系统的强大映射能力，已广泛应用于复杂对象的控制中。神经网络所具有的大规模并行性、冗余性、容错性、本质的非线性及自组织、自学习、自适应能力，给不断面临挑战的控制理论带来生机。

从控制角度来看，神经网络用于控制的优越性主要表现为：

（1）神经网络可是处理那些难以用模型或规则描述的对象；

（2）神经网络采用并行分布式信息处理方式，具有很强的容错性；

（3）神经网络在本质上是非线性系统，可以实现任意非线性映射。神经网络在非线性控制系统中具有很大的发展前途；

（4）神经网络具有很强的信息综合能力，它能够同时处理大量不同类型的输入，能够很好地解决输入信息之间的互补性和冗余性问题；

（5）神经网络的硬件实现愈趋方便。大规模集成电路技术的发展为神经网络的硬件实现提供了技术手段，为神经网络在控制中的应用开辟了广阔的前景。

神经网络控制所取得的进展为：

(1) 基于神经网络的系统辨识：可在已知常规模型结构的情况下，估计模型的参数；或利用神经网络的线性、非线性特性，建立线性、非线性系统的静态、动态、逆动态及预测模型；

(2) 神经网络控制器：神经网络作为控制器，可实现对不确定系统或未知系统进行有效的控制，使控制系统达到所要求的动态、静态特性；

(3) 神经网络与其他算法相结合：神经网络与专家系统、模糊逻辑、遗传算法等相结合可构成新型控制器；

(4) 优化计算：在常规控制系统的设计中，常遇到求解约束优化问题，神经网络为这类问题提供了有效的途径；

(5) 控制系统的故障诊断：利用神经网络的逼近特性，可对控制系统的各种故障进行模式识别，从而实现控制系统的故障诊断。

神经网络控制在理论和实践上，以下问题是研究的重点：

(1) 神经网络的稳定性与收敛性问题；

(2) 神经网络控制系统的稳定性与收敛性问题；

(3) 神经网络学习算法的实时性；

(4) 神经网络控制器和辨识器的模型和结构。

9.2 神经网络控制结构

根据神经网络在控制器中的作用不同，神经网络控制器可分为两类，一类为神经控制，它是以神经网络为基础而形成的独立智能控制系统；另一类为混合神经网络控制，它是指利用神经网络学习和优化能力来改善传统控制的智能控制方法，如自适应神经网络控制等。

综合目前的各种分类方法，可将神经网络控制的结构归结为以下七类。

9.2.1 神经网络监督控制

通过对传统控制器进行学习，然后用神经网络控制器逐渐取代传统控制器的方法，称为神经网络监督控制。神经网络监督控制的结构如图9-1所示。

神经网络控制器实际上是一个前馈控制器，它建立的是被控对象的逆模型。神经网络控制器通过对传统控制器的输出进行学习，在线调整网络的权值，使反馈控制输入趋近于零，从而使神经网络控制器逐渐在控制作用中占据主导地位，最终取消反馈控制器的作用。一旦系统出现干扰，反馈控制器重新起作用。这种前馈加反馈的监督控制方法，不仅可以确保控制系统的稳定性和鲁棒性，而且可有效地提高系统的精度和自适应能力。

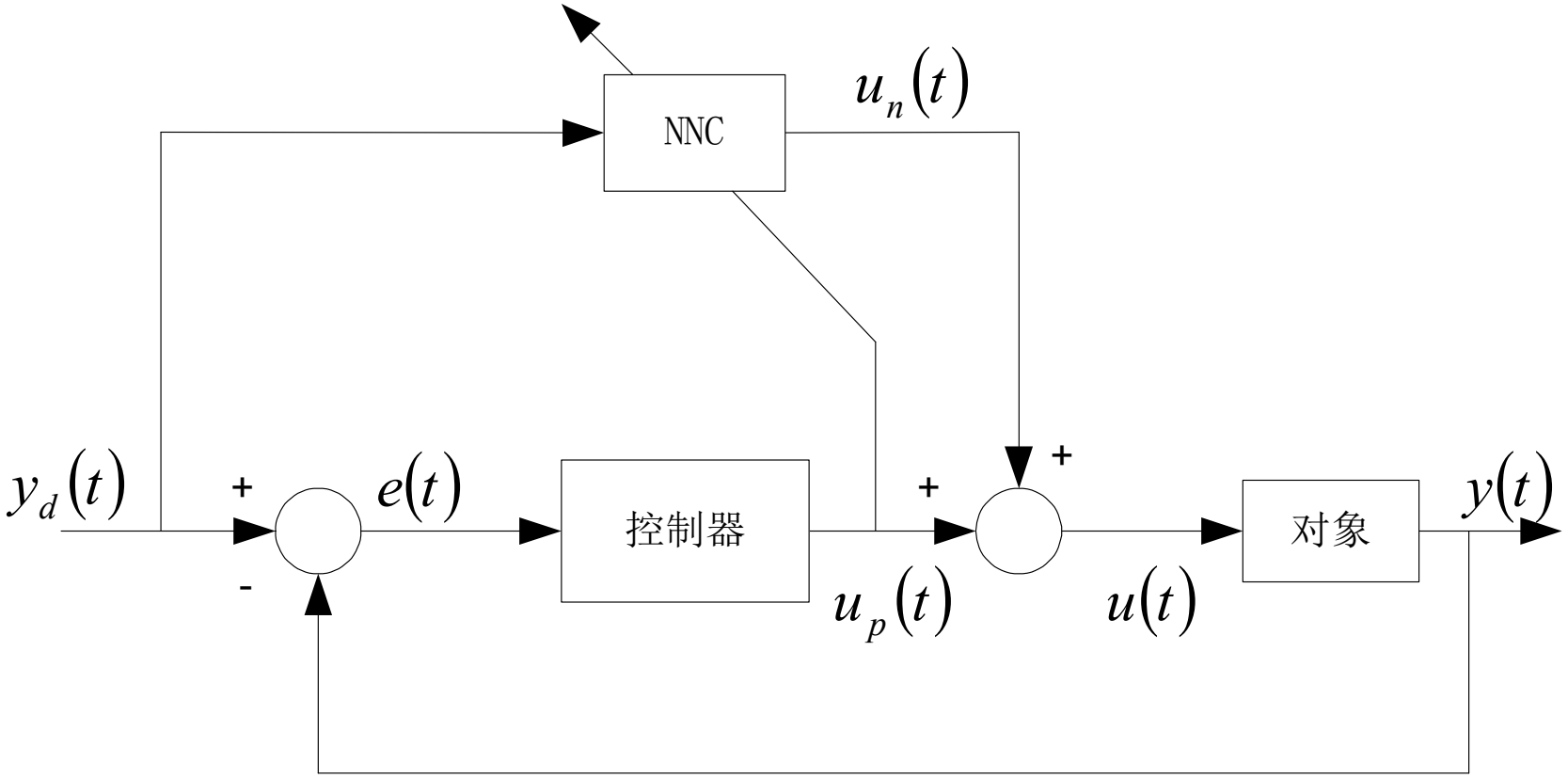


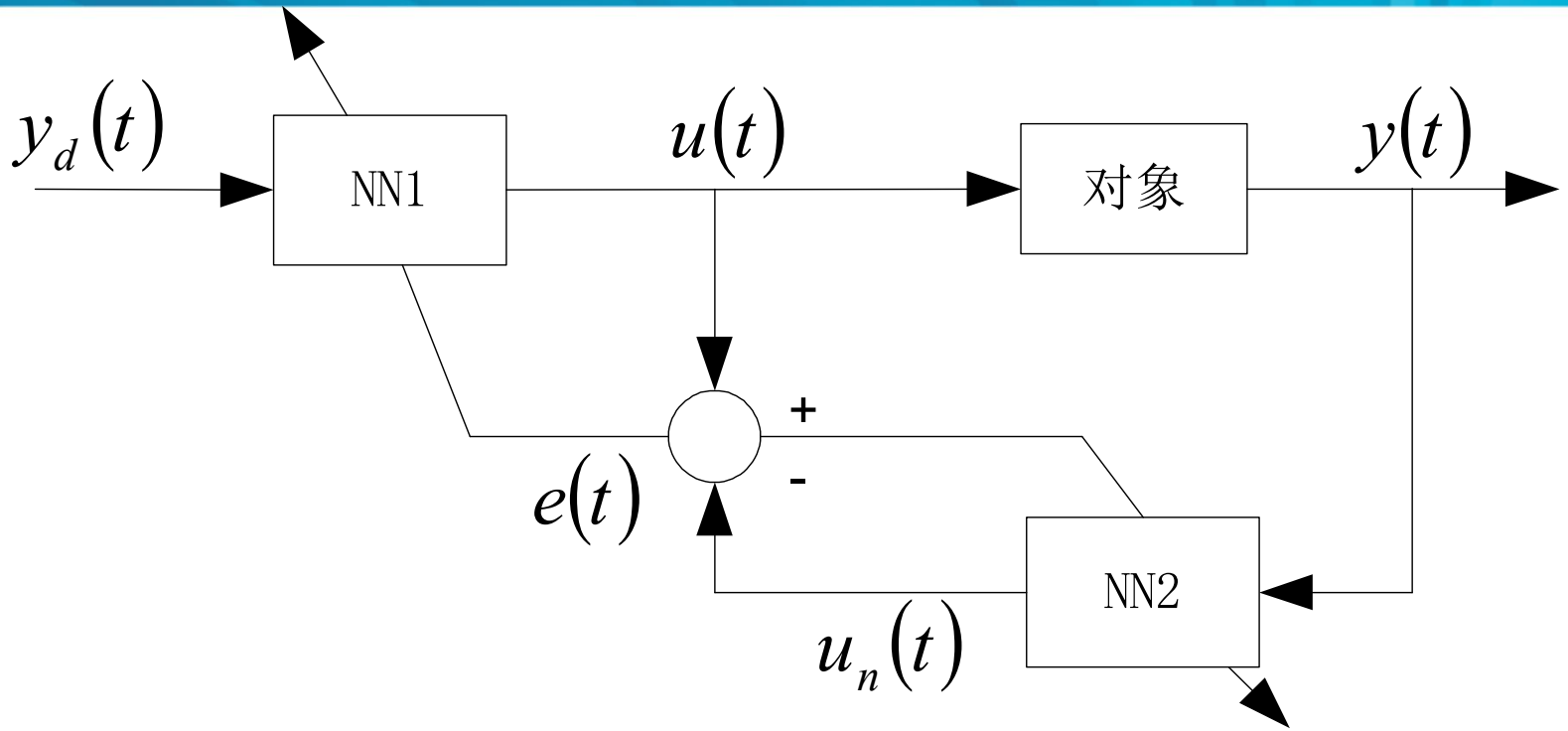
图9-1 神经网络监督控制

9.2.2 神经网络直接逆动态控制

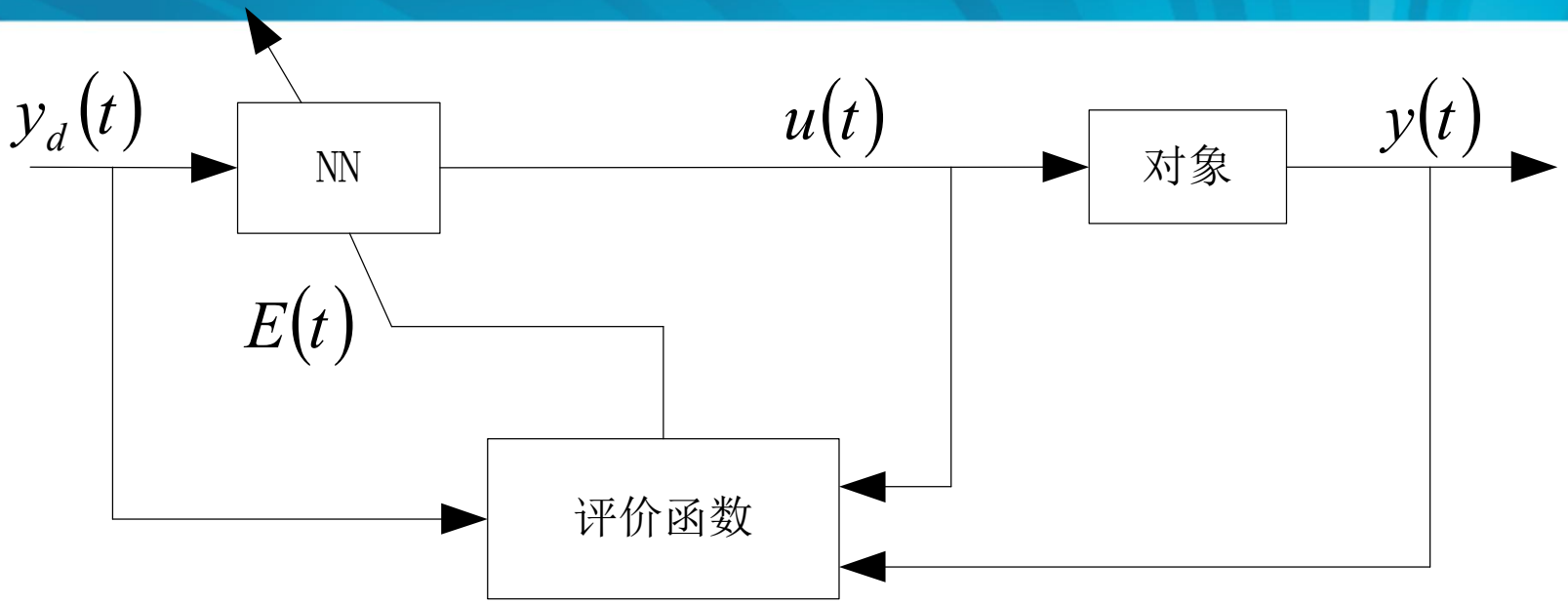
神经网络直接逆控制就是将被控对象的神经网络逆模型直接与被控对象串联起来，以便使期望输出与对象实际输出之间的传递函数为1。则将此网络作为前馈控制器后，被控对象的输出为期望输出。

显然，神经网络直接逆控制的可用性在相当程度上取决于逆模型的准确精度。由于缺乏反馈，简单连接的直接逆控制缺乏鲁棒性。为此，一般应使其具有在线学习能力，即作为逆模型的神经网络连接权能够在线调整。

图9-2为神经网络直接逆控制的两种结构方案。在图9-2(a)中，NN1和NN2为具有完全相同的网络结构，并采用相同的学习算法，分别实现对象的逆。在图9-2(b)中，神经网络NN通过评价函数进行学习，实现对象的逆控制。



（a）



（b）

图9-2　神经网络直接逆控制

9.2.3 神经网络自适应控制

与传统自适应控制相同，神经网络自适应控制也分为神经网络自校正控制和神经网络模型参考自适应控制两种。自校正控制根据对系统正向或逆模型的结果调节控制器内部参数，使系统满足给定的指标，而在模型参考自适应控制中，闭环控制系统的期望性能由一个稳定的参考模型来描述。

1 神经网络自校正控制

神经网络自校正控制分为直接自校正控制和间接自校正控制。间接自校正控制使用常规控制器，神经网络估计器需要较高的建模精度。直接自校正控制同时使用神经网络控制器和神经网络估计器。

（1）神经网络直接自校正控制

在本质上同神经网络直接逆控制，其结构如图9-2所示。

（2）神经网络间接自校正控制

其结构如图9-3所示。假设被控对象为如下单变量仿射非线性系统：

$$y(t)=f\left(y_t\right)+g\left(y_t\right)u(t)$$

若利用神经网络对非线性函数 和(y_t) 进行逼近，得到$\hat{f}(y_t)$ 和 $\hat{g}(y_t)$ ，则控制器为：

$$u\left(t\right)=\left[r(\mathrm{t})-\hat{f}\left(y_t\right)\right]/\hat{g}\left(y_t\right)$$

其中 $r(t)$ 为t时刻的期望输出值。

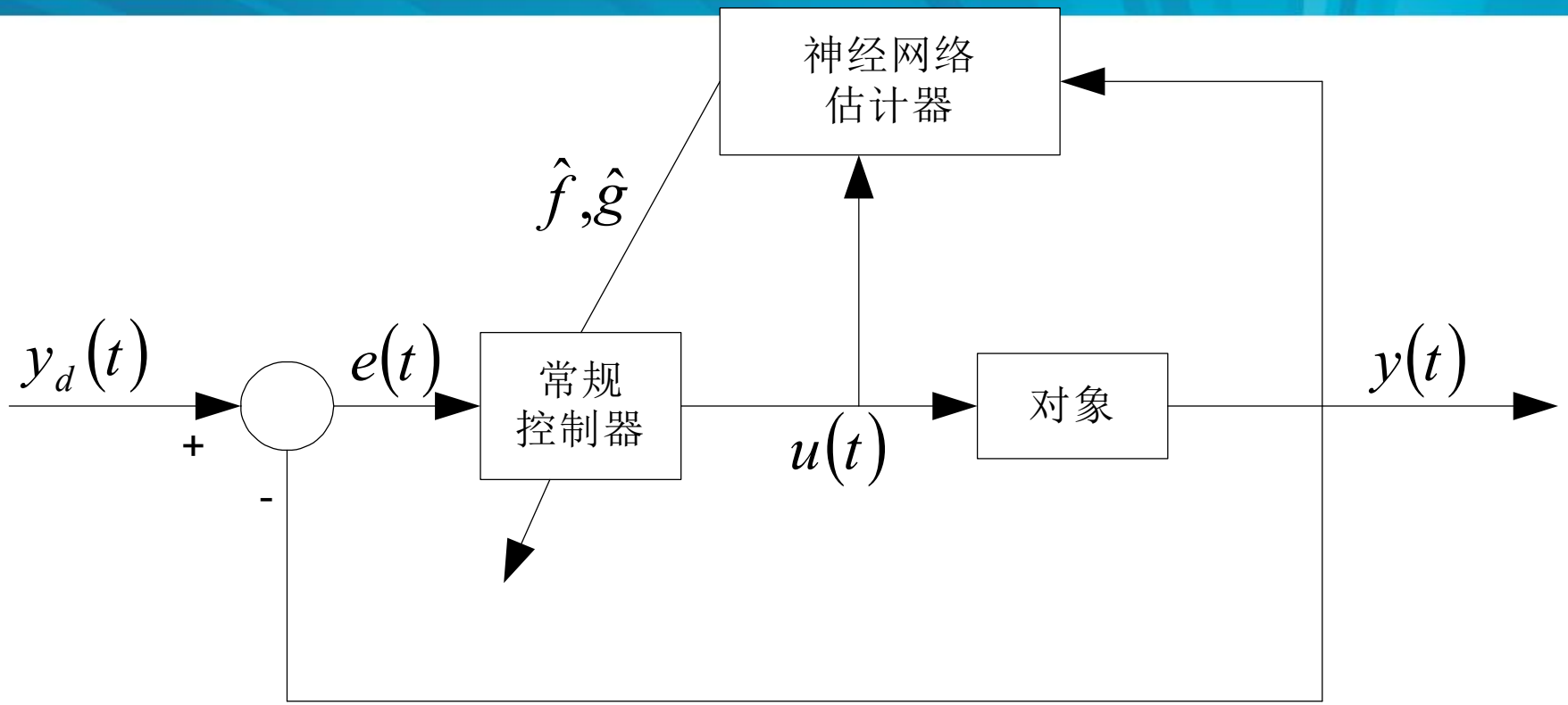


图9-3　神经网络间接自校正控制

2. 神经网络模型参考自适应控制

分为直接模型参考自适应控制和间接模型参考自适应控制两种。

（1）直接模型参考自适应控制

如图9-4所示。神经网络控制器的作用是使被控对象与参考模型输出之差为最小。但该方法需要知道对象的 $Jacobian$ 信息 $\frac{\partial y}{\partial u}$ 。

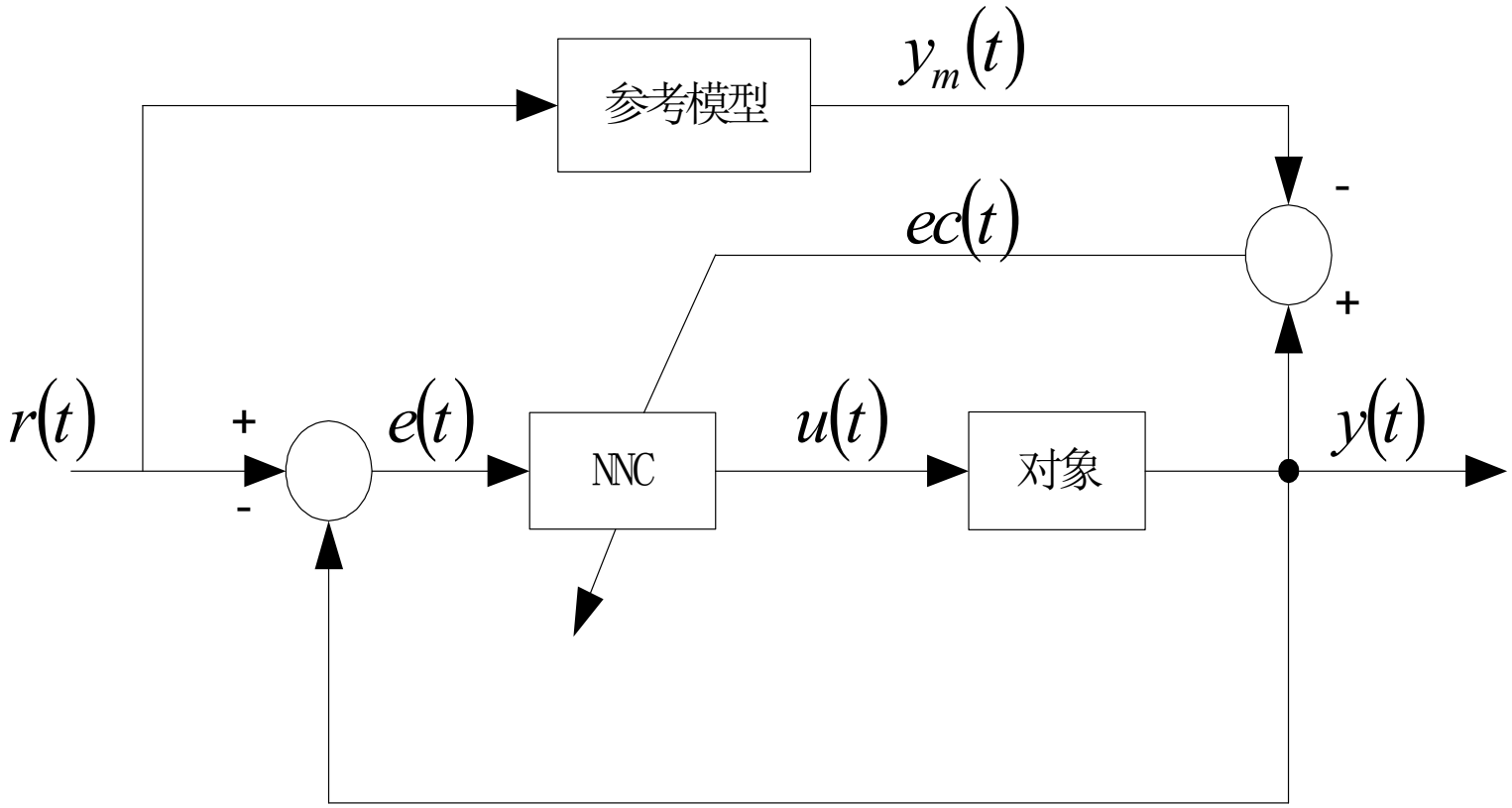


图9-4　神经网络直接模型参考自适应控制

（2）间接模型参考自适应控制

如图9-5所示。神经网络辨识器NNI向神经网络控制器NNC提供对象的信息，用于控制器NNC的学习。

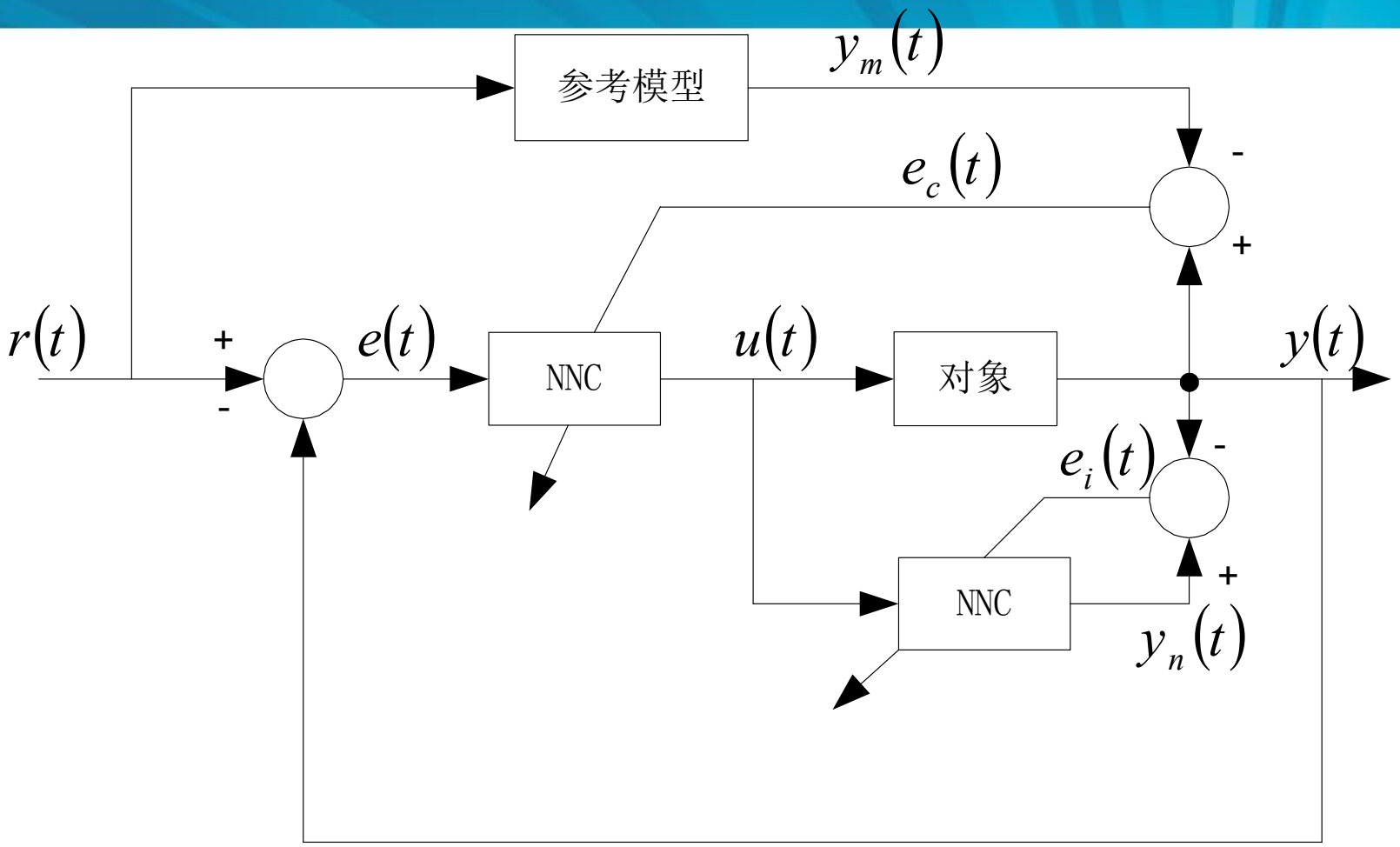


图9-5 神经网络间接模型参考自适应控制

9.2.4　神经网络内模控制

经典的内模控制将被控系统的正向模型和逆模型直接加入反馈回路，系统的正向模型作为被控对象的近似模型与实际对象并联，两者输出之差被用作反馈信号，该反馈信号又经过前向通道的滤波器及控制器进行处理。控制器直接与系统的逆有关，通过引入滤波器来提高系统的鲁棒性。

图9-6为神经网络内模控制，被控对象的正向模型及控制器均由神经网络来实现。

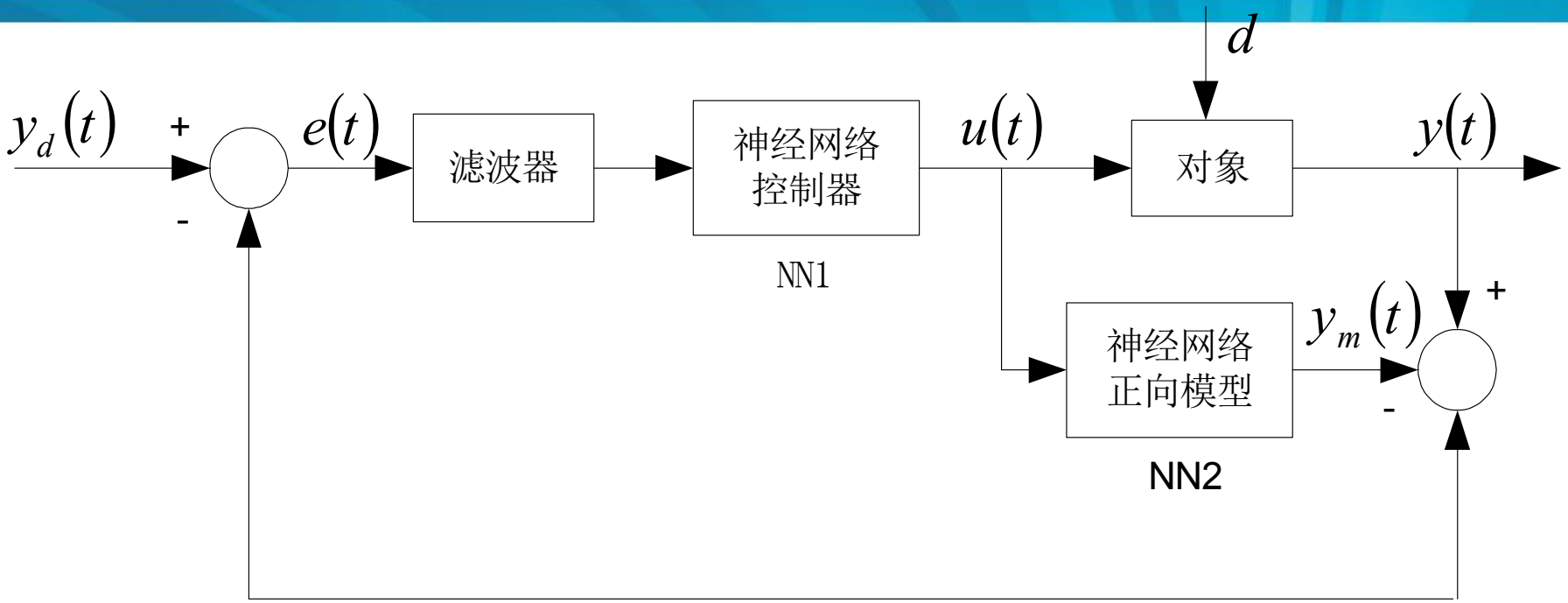


图9-6 神经网络内模控制

9.2.5 神经网络预测控制

预测控制又称为基于模型的控制，是70年代后期发展起来的新型计算机控制方法，该方法的特征是预测模型、滚动优化和反馈校正。

神经网络预测控制的结构如图9-7所示，神经网络预测器建立了非线性被控对象的预测模型，并可在线进行学习修正。

利用此预测模型，通过设计优化性能指标，利用非线性优化器可求出优化的控制作用 $u(t)$。

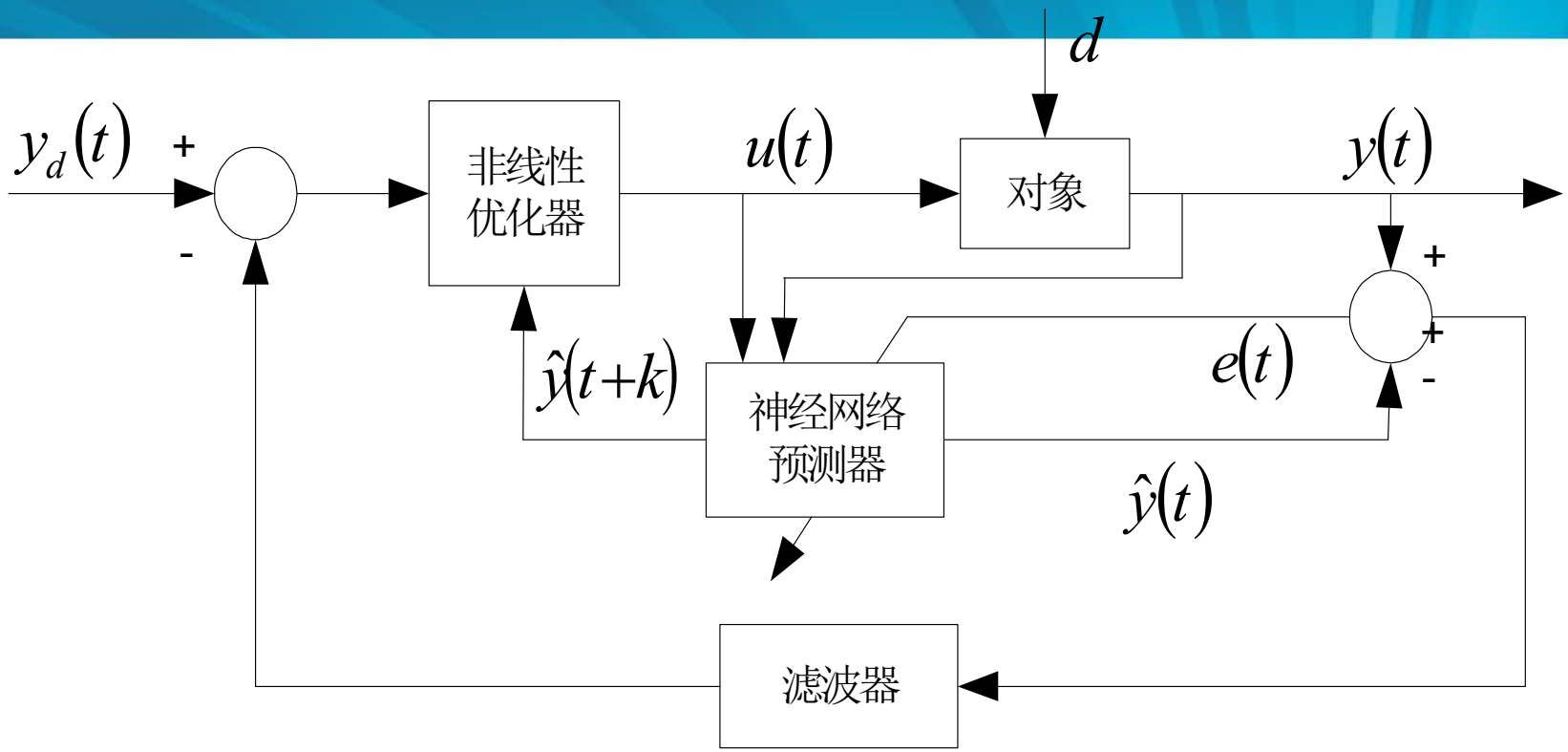


图9-7　神经网络预测控制

9.2.6 神经网络自适应评判控制

神经网络自适应评判控制通常由两个网络组成，如图9-8所示。自适应评判网络通过不断的奖励、惩罚等再励学习，使自己逐渐成为一个合格的“教师”，学习完成后，根据系统目前的状态和外部激励反馈信号r(t)产生一个内部再励信号 $\hat{r}(t)$，以对目前的控制效果作出评价。控制选择网络相当于一个在内部再励信号 $\hat{r}(t)$指导下进行学习的多层前馈神经网络控制器，该网络在进行学习后，根据编码后的系统状态，在允许控制集中选择下一步的控制作用。

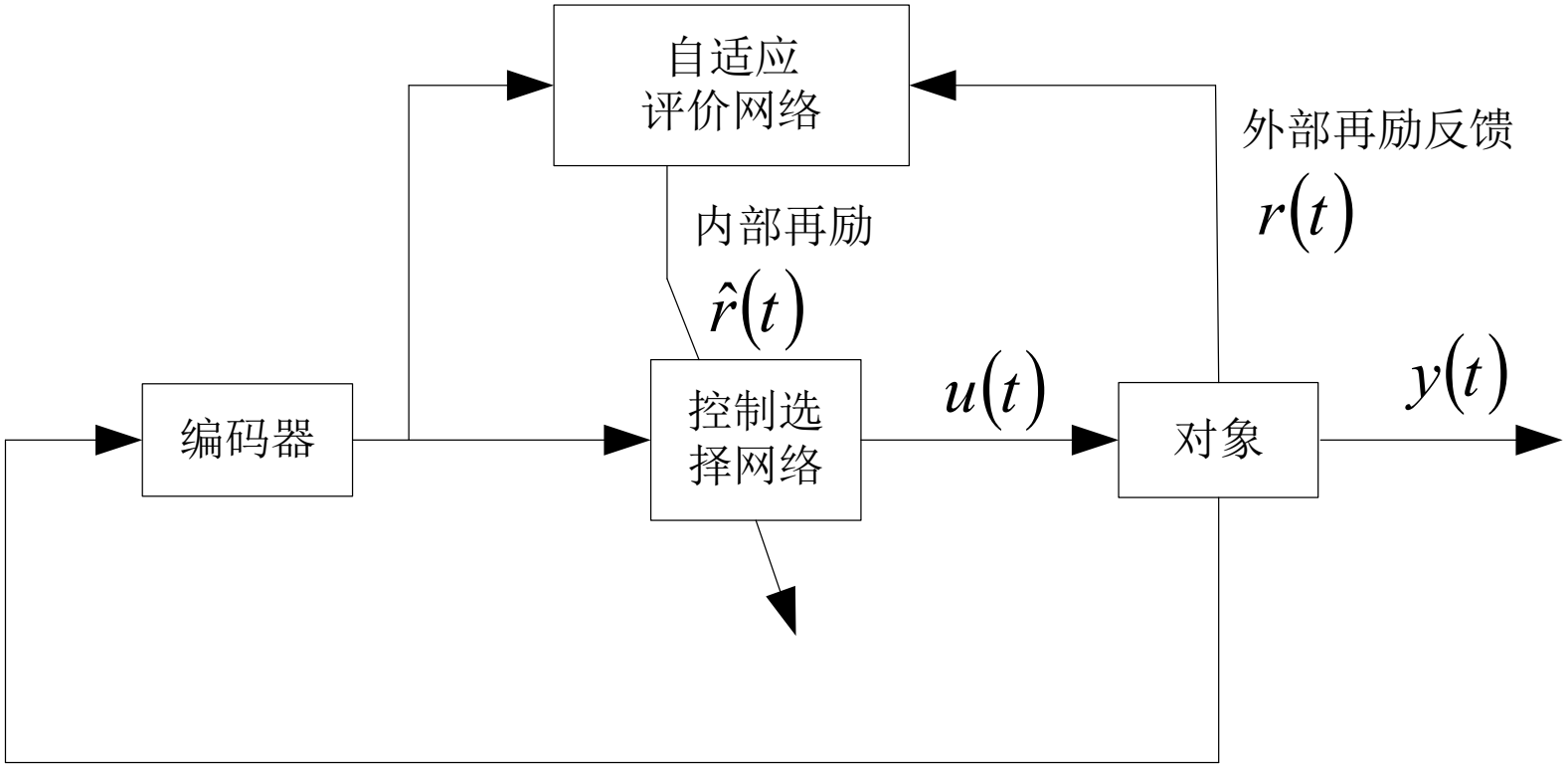


图9-8 神经网络自适应评判控制

9.2.7　神经网络混合控制

该控制方法是集成人工智能各分支的优点，由神经网络技术与模糊控制、专家系统等相结合而形成的一种具有很强学习能力的智能控制系统。

由神经网络和模糊控制相结合构成模糊神经网络，由神经网络和专家系统相结合构成神经网络专家系统。神经网络混合控制可使控制系统同时具有学习、推理和决策能力。

9.3 单神经元网络控制

9.3.1 单神经元自适应控制算法

单神经元自适应控制的结构如图9-9所示。

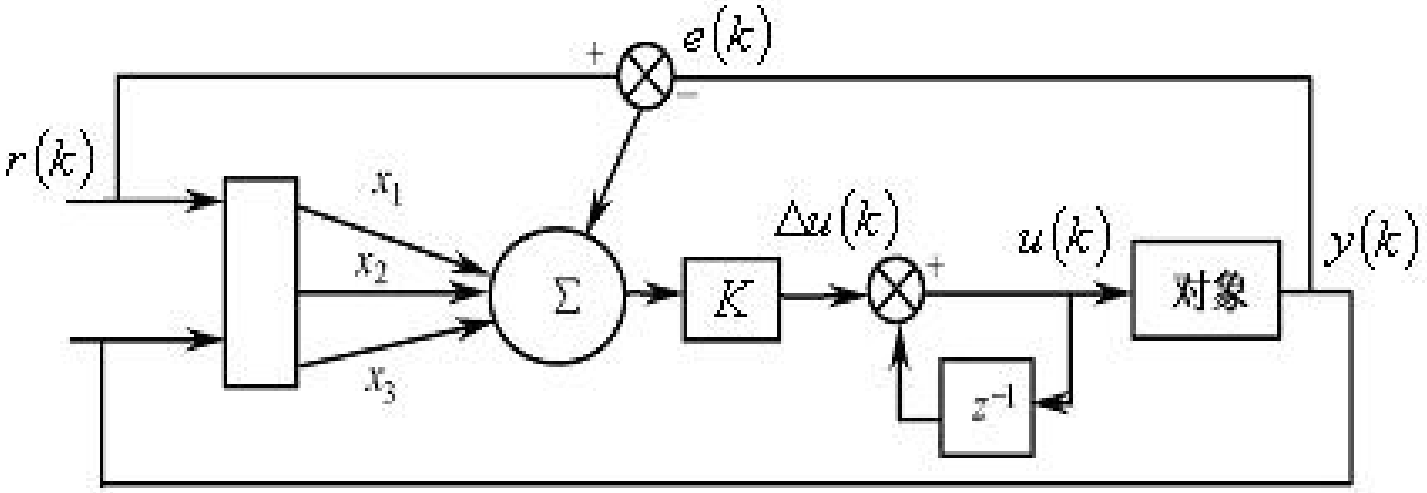
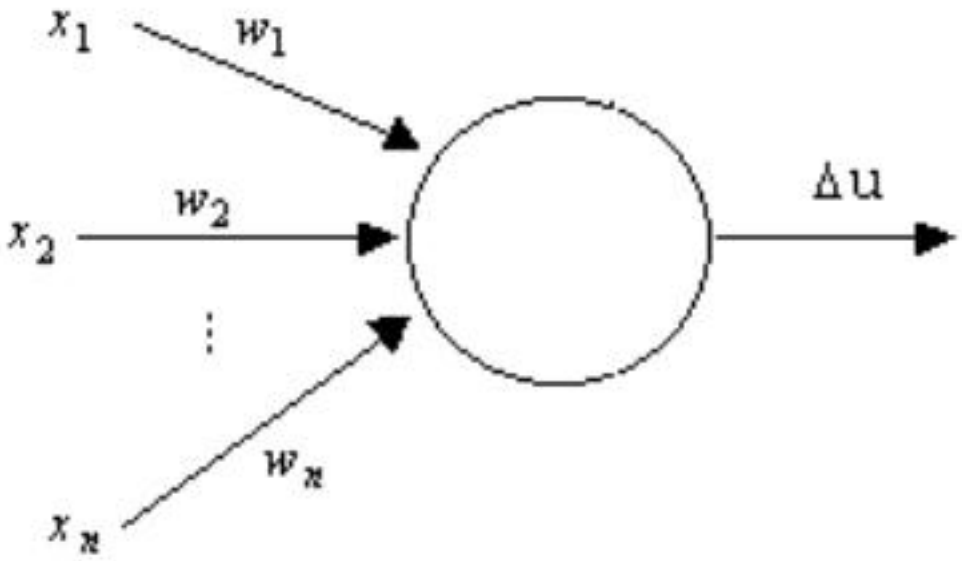


图9-9 单神经元自适应PID控制结构



单神经网络控制器

单神经元自适应控制器是通过对加权系数的调整来实现自适应、自组织功能，控制算法为

$$u(k)=u(k-1)+K\sum_{i=1}^{3}w_i(k)x_i(k)$$

如果权系数的调整按有监督的**Hebb**学习规则实现，即在学习算法中加入监督项$z(k)$，则神经网络权值学习算法为：

$$w_1(k) = w_1(k-1) + \eta z(k) u(k) x_1(k)$$
$$w_2(k) = w_2(k-1) + \eta z(k) u(k) x_2(k)$$
$$w_3(k) = w_3(k-1) + \eta z(k) u(k) x_3(k)$$

式中，

$$z(k) = e(k)$$

$$x_1(k) = e(k)$$

$$x_2(k) = e(k) - e(k-1)$$

$$x_3(k) = \Delta^2 e(k) = e(k) - 2e(k-1) + e(k-2)$$

η 为学习速率，K 为神经元的比例系数，$K > 0$ 。

K值的选择非常重要。K越大，则快速性越好，但超调量大，甚至可能使系统不稳定。当被控对象时延增大时，K值必须减少，以保证系统稳定。K值选择过小，会使系统的快速性变差。

9.3.2 仿真实例

被控对象为

$$y(k)=0.368y(k-1)+0.26y(k-2)+0.10u(k-1)+0.632u(k-2)$$

输入指令为一方波信号

$$r(k)=0.5\mathrm{sgn}(\sin(4\pi t))$$

采样时间为1ms，采用单神经元自适应控制律进行控制。

仿真程序：chap9_1.m

9.4　RBF网络监督控制

9.4.1　RBF网络监督控制算法

基于RBF网络的监督控制系统结构如图9-14所示。

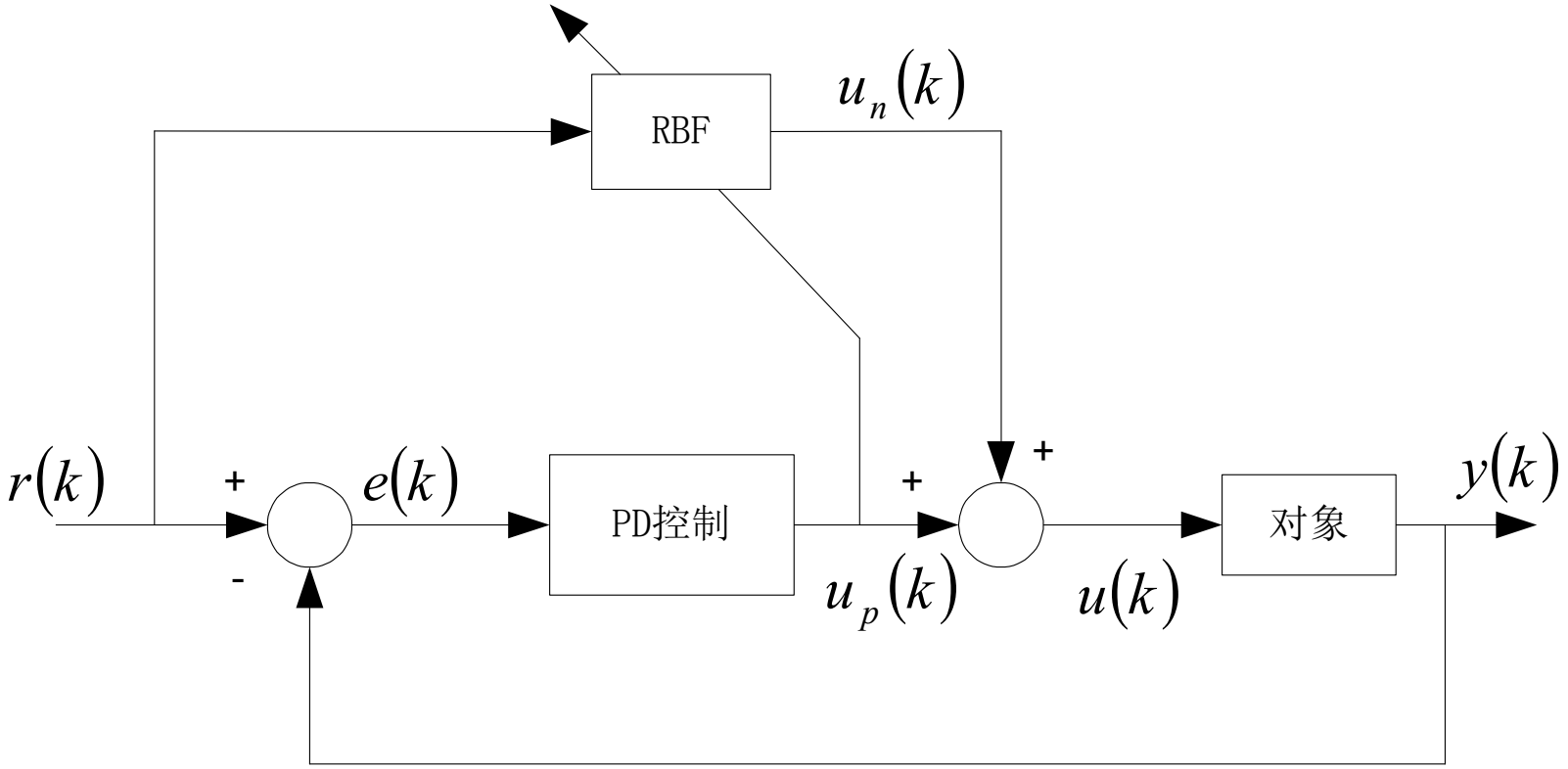


图9-14　神经网络监督控制

在RBF网络结构中，取网络的输入为 $r(k)$，网络的径向基向量为 $\mathbf{H}=[h_1, \cdots, h_m]^T$ h_j为高斯基函数：

$$h_j = \exp\left(-\frac{\left\|r(k)-\mathbf{C}_j\right\|^2}{2b_j^2}\right)$$

网络的权向量为：　　$\mathbf{W}=[w_1,\ \cdots,\ w_m]^{\mathrm{T}}$

RBF网络的输出为：

$$u_n(k)=h_1w_1+\cdots h_jw_j\cdots+h_mw_m$$

其中 m 为RBF网络隐层神经元的个数。

控制律为：

$$u(k)=u_p(k)+u_n(k)$$

设神经网络调整的性能指标为：

$$E(k)=\frac{1}{2}\left(u_n(k)-u(k)\right)^2$$

近似地取 $$\frac{\partial u_p(k)}{\partial w_j(k)}=\frac{\partial u_n(k)}{\partial w_j(k)}$$

由此所产生的不精确通过权值调节来补偿。

采用梯度下降法调整网络的权值：

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

$$= \eta(u_n(k) - u(k)) h_j(k)$$

神经网络权值的调整过程为：

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \Delta\mathbf{W}(k) + \alpha(\mathbf{W}(k-1) - \mathbf{W}(k-2))$$

其中 η为学习速率，α为动量因子。

9.4.2 仿真实例

被控对象为：

$$G(s)=\frac{1000}{s^2+50s+2000}$$

RBF网络监督控制程序为chap9_2.m

9.5 RBF网络自校正控制

9.5.1 神经网络自校正控制原理

自校正控制有两种结构：直接型与间接型。直接型自校正控制也称直接逆动态控制，是前馈控制。间接自校正控制是一种由辨识器将对象参数进行在线估计，用调节器（或控制器）实现参数的自动整定相结合的自适应控制技术，可用于结构已知而参数未知但恒定的随机系统，也可用于结构已知而参数缓慢时变的随机系统。

神经间接自校正控制结构如图9-17所示，它由两个回路组成：

（1）自校正控制器与被控对象构成的反馈回路。

（2）神经网络辩识器与控制器设计，以得到控制器的参数。

辩识器与自校正控制器的在线设计是自校正控制实现的关键。

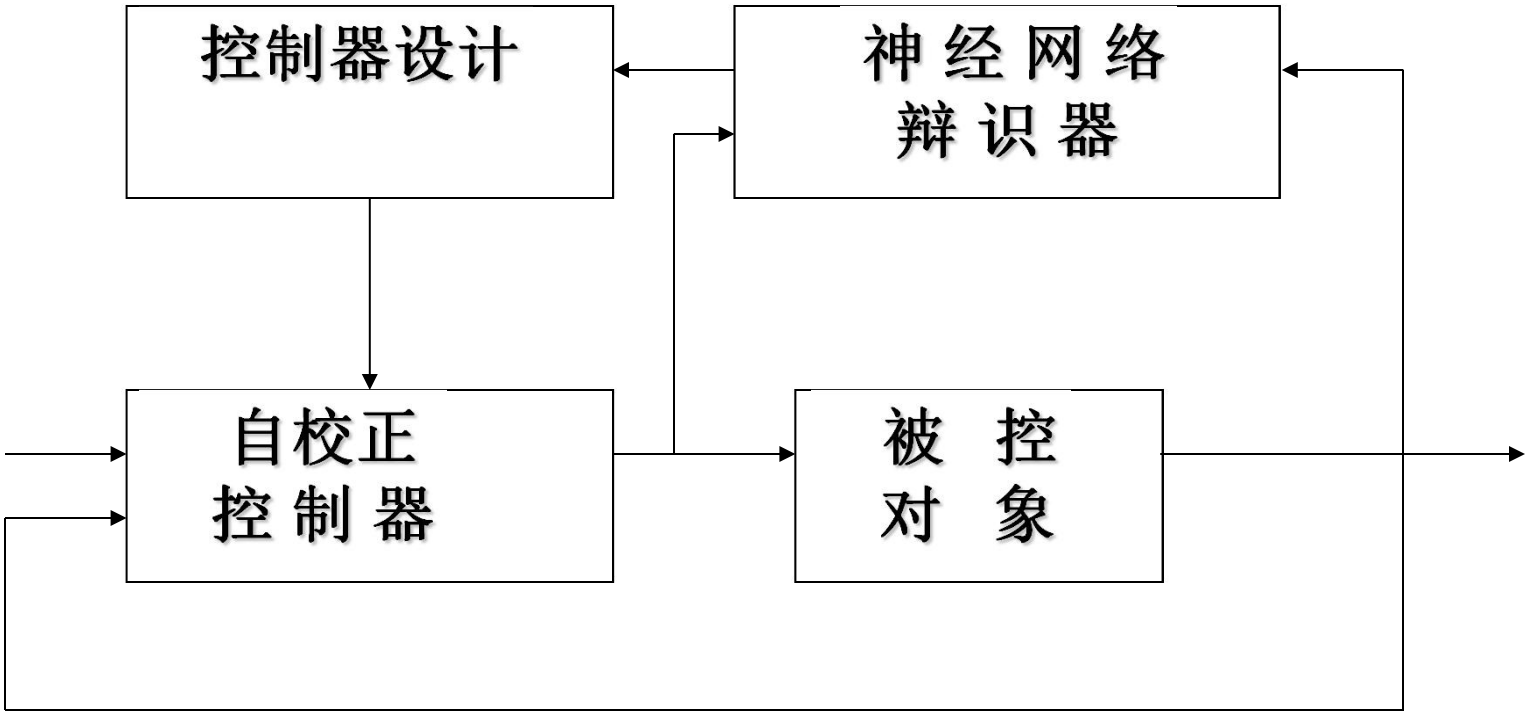


图9-17　神经网络间接自校正控制框图

9.5.2 自校正控制算法

考虑被控对象：

$$y(k+1)=g[y(k)]+\varphi[y(k)]u(k)$$

其中 u，y 分别为对象的输入、输出，$\varphi[\bullet]$ 为非零函数。

若 $g[\bullet]$, $\varphi[\bullet]$ 已知，根据“确定性等价原则”，控制器的控制算法为：

$$u(k)=\frac{-g[\bullet]}{\varphi[\bullet]}+\frac{r(k+1)}{\varphi[\bullet]}$$

若$g[\bullet]$，$\varphi[\bullet]$未知，则通过在线训练神经网络辨识器，由辨识器结果$Ng[\bullet]$、$N\varphi[\bullet]$代替$g[\bullet]$、$\varphi[\bullet]$，控制器的控制算法为：

$$u(k)=\frac{-Ng[\bullet]}{N\varphi[\bullet]}+\frac{r(k+1)}{N\varphi[\bullet]}$$

9.5.3　RBF网络自校正控制算法

采用两个RBF网络分别实现未知项 $g[\bullet]$ 、$\varphi[\bullet]$的辨识。RBF网络辨识器的结构如图9-18所示，

$\mathbf{w}$ 和 $\mathbf{v}$ 分别为两个神经网络的权值向量。

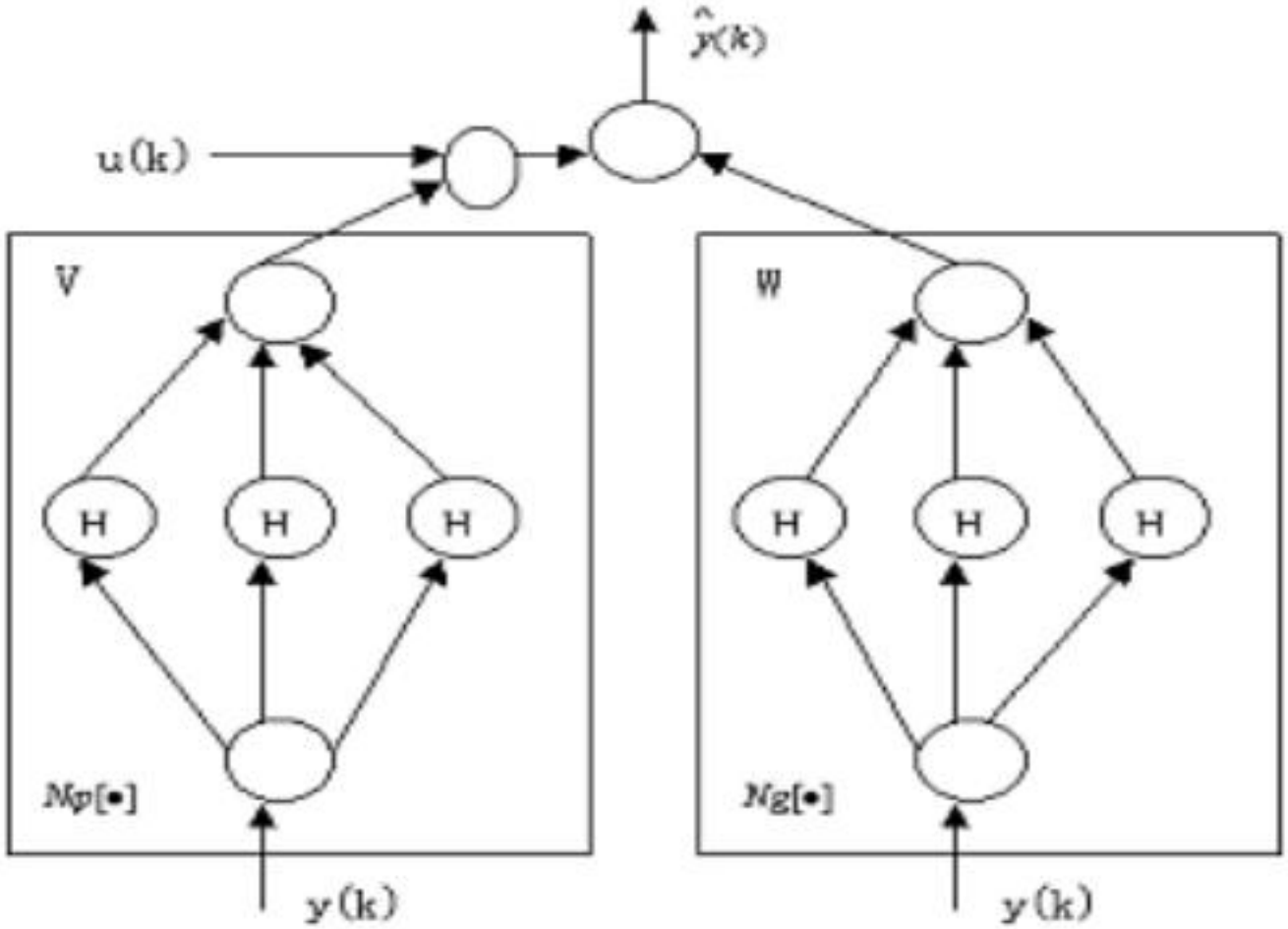


图9-18　神经网络辨识器

在RBF网络结构中，取网络的输入为 $y(k)$ ，网络的径向基向量为 $\mathbf{H}=[h_1, \cdots, h_m]^T$ ， h_j 为高斯基函数：

$$h_j = \exp(-\frac{\|y(k)-\mathbf{C}_j\|^2}{2b_j^2})$$

其中 $j=1,\cdots,m$ 。b_j 为节点 j 的基宽度参数，，$\mathbf{C}_j$ 为网络第 j 个结点的中心矢量，$b_j>0$

$\mathbf{C}_j=[c_{11},\cdots,c_{1m}]$ ，$\mathbf{B}=[b_1,\cdots,b_m]^T$ 。

网络的权向量为：

$$\mathbf{W}=\left[w_1,\quad \cdots,\quad w_m\right]^{\mathrm{T}}$$

$$\mathbf{V}=\left[v_1,\quad \cdots,\quad v_m\right]^{\mathrm{T}}$$

两个RBF网络的输出分别为：

$$Ng(k)=h_1w_1+\cdots h_jw_j\cdots+h_mw_m$$

$$N\varphi(k)=h_1v_1+\cdots h_jv_j\cdots+h_mv_m$$

其中 m 为RBF网络隐层神经元的个数。

辨识后，对象的输出为：

$$y_m(k)=Ng\left[y(k-1);W(k)\right]+N\varphi\left[y(k-1);V(k)\right]u(k-1)$$

设神经网络调整的性能指标为：

$$E(k)=\frac{1}{2}\left(y(k)-y_m(k)\right)^2$$

采用梯度下降法调整网络的权值：

$$\Delta w_j(k)=-\eta_w\frac{\partial E(k)}{\partial w_j(k)}=\eta_w\left(y(k)-y_m(k)\right)h_j(k)$$

$$\Delta v_j(k)=-\eta_v\frac{\partial E(k)}{\partial v_j(k)}=\eta_v\left(y(k)-y_m(k)\right)h_j(k)$$

神经网络权值的调整过程为：

$$\mathbf{W}(k)=\mathbf{W}(k-1)+\Delta\mathbf{W}(k)+\alpha(\mathbf{W}(k-1)-\mathbf{W}(k-2))$$

$$\mathbf{V}(k)=\mathbf{V}(k-1)+\Delta\mathbf{V}(k)+\alpha(\mathbf{V}(k-1)-\mathbf{V}(k-2))$$

其中η_w和η_v 为学习速率，α为动量因子。

神经网络自校正控制系统的结构如图9-19所示。

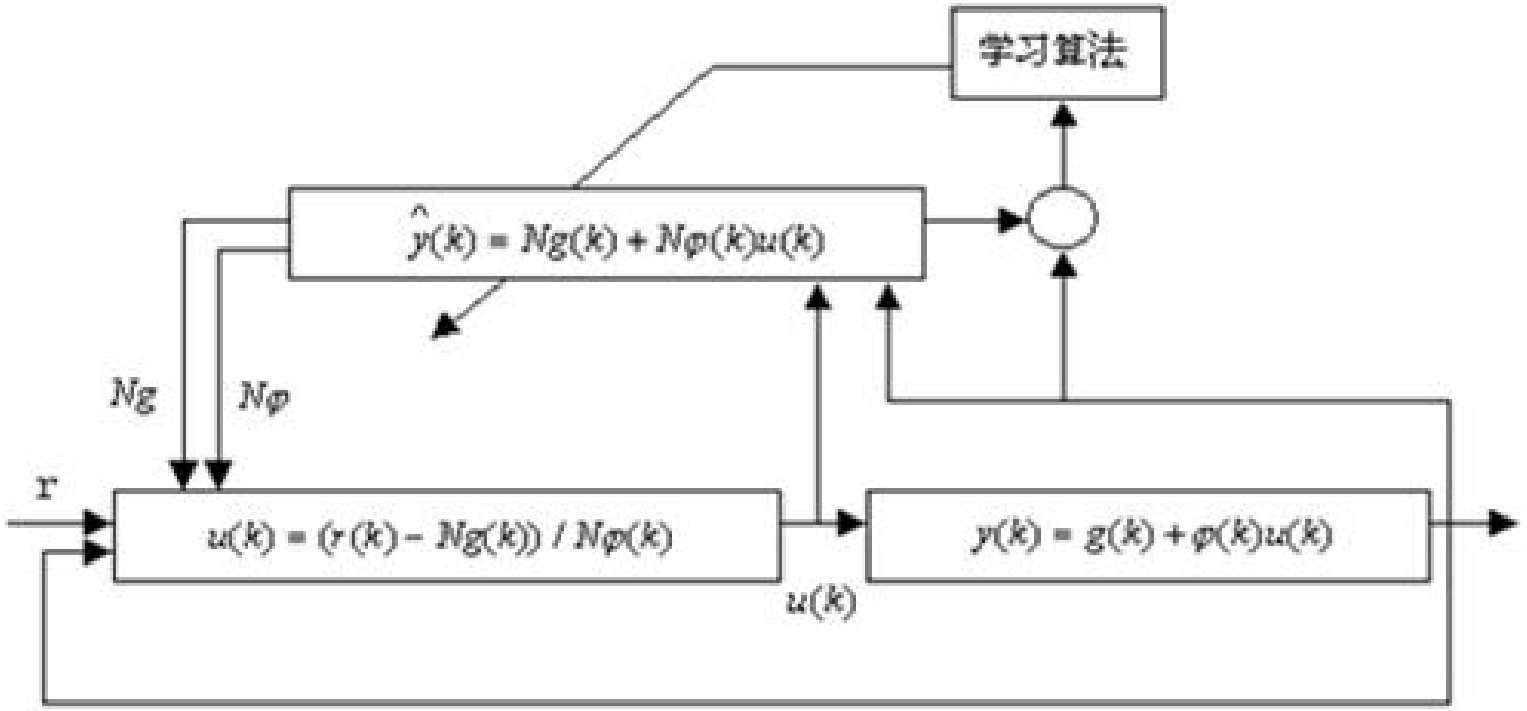


图9-19　神经网络自校正控制框图

9.5.4 仿真实例

被控对象为：

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

其中 $g[y(k)]=0.8\sin(y(k-1))$ ， $\varphi[y(k)]=15$ 。

RBF网络自校正控制程序为chap9_3.m。

9.6 基于RBF网络直接模型参考自适应控制

9.6.1 基于RBF网络的控制器设计

控制系统的结构如图9-23所示。

设参考模型输出为 $y_m(k)$ ，控制系统要求对象的输出$y(k)$ 能够跟踪参考模型的输出 $y_m(k)$ 。

则跟踪误差为：

$$ec(k)=y_m(k)-y(k)$$

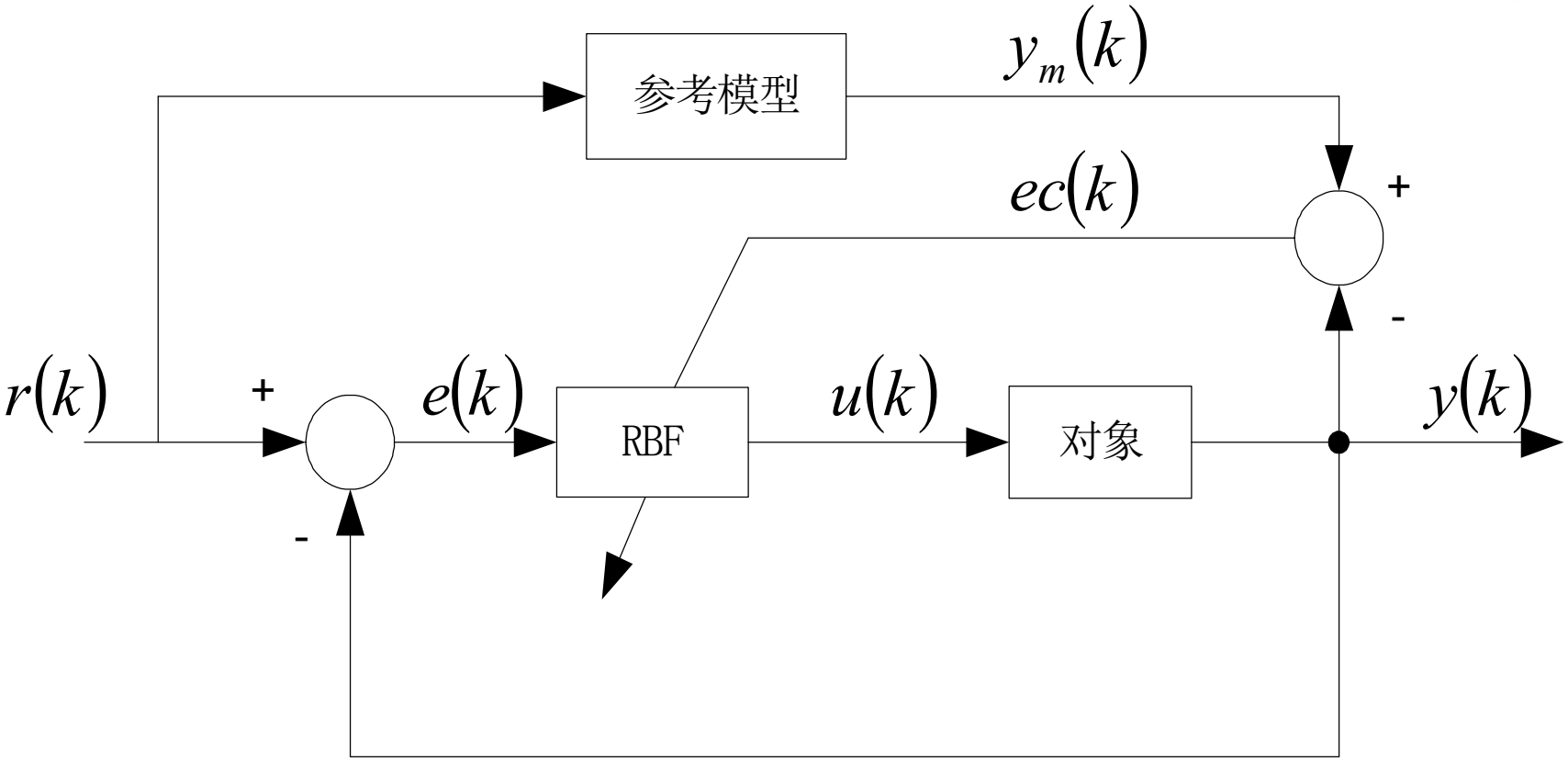


图9-23 基于RBF网络的直接模型参考自适应控制

控制目标函数为：

$$E(k)=\frac{1}{2}ec(\mathrm{k})^2$$

控制器为RBF网络的输出：

$$u(k)=h_1w_1+\cdots h_jw_j\cdots+h_mw_m$$

其中 m 为RBF网络隐层神经元的个数，w_j 为第 j 个网络隐层神经元与输出层之间的连接权，h_j 为第个 j 隐层神经元的输出。

在RBF网络结构中，$\mathbf{X}=[x_1,\cdots,x_n]^T$为网络的输入向量。RBF网络的径向基向量为 $\mathbf{H}=[h_1,\ \cdots,\ h_m]^T$，$h_j$为高斯基函数：

$$h_j=\exp\left(-\frac{\|\mathbf{X}-\mathbf{C}_j\|^2}{2b_j^2}\right)$$

其中 $j=1,\cdots,m$，b_j为节点 j 的基宽度参数，$b_j>0$ $\mathbf{C}_j$为网络第 j 个结点的中心矢量，$\mathbf{C}_j=[c_{j1},\cdots,c_{ji},\cdots,c_{jn}]$，

$\mathbf{B}=[b_1,\cdots,b_m]^T$。

网络的权向量为：

$$\mathbf{W}=\left[w_1,\quad \cdots,\quad w_m\right]^{\mathrm{T}}$$

按梯度下降法及链式法则，可得权值的学习算法如下：

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

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

其中η为学习速率，α为动量因子。

同理，可得RBF网络隐层神经元的高斯函数的中心参数及基宽的学习算法如下：

$$\Delta b_j(k) = -\eta \frac{\partial E(k)}{\partial b_j} = \eta ec(k) \frac{\partial y(\mathrm{k})}{\partial u(k)} \frac{\partial u(\mathrm{k})}{\partial b_j} = \eta ec(k) \frac{\partial y(\mathrm{k})}{\partial u(k)} w_j h_j \frac{\left\| \mathbf{x} - c_{ij} \right\|^2}{b_j^{\,3}}$$

$$b_j(\mathrm{k}) = b_j(\mathrm{k}-1) + \eta \Delta b_j(k) + \alpha (b_j(\mathrm{k}-1) - b_j(\mathrm{k}-2))$$

$$\Delta c_{ij}(k) = -\eta \frac{\partial E(k)}{\partial c_{ij}} = \eta ec(k) \frac{\partial y(\mathrm{k})}{\partial u(k)} \frac{\partial u(\mathrm{k})}{\partial c_{ij}} = \eta ec(k) \frac{\partial y(\mathrm{k})}{\partial u(k)} w_j h_j \frac{x_i - c_{ij}}{b_j^{\,2}}$$

$$c_{ij}(\mathrm{k}) = c_{ij}(\mathrm{k}-1) + \eta \Delta c_{ij}(k) + \alpha (\mathrm{c}_{ij}(\mathrm{k}-1) - \mathrm{c}_{ij}(\mathrm{k}-2))$$

在学习算法中，$\frac{\partial y(\mathrm{k})}{\partial u(k)}$ 称为*Jacobian*信息，表示系统的输出对控制输入的敏感性，其值可由神经网络辨识而得。在神经网络算法中，对 $\frac{\partial y(\mathrm{k})}{\partial u(k)}$的值的精确度要求不是很高，不精确部分可通过网络参数及权值的调整来修正，关键是其符号，因此可用$\frac{\partial y(\mathrm{k})}{\partial u(k)}$ 的正负号来代替，这样可使算法更加简单。

9.6.2 仿真实例

被控对象为一非线性模型：

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

取采样周期为 $ts=1ms$，参考模型为

$y_m(k)=0.6y_m(k-1)+r(k)$，其中$r(k)$为正弦信号，

$$r(k)=0.50\sin(2\pi k\times ts)$$

RBF网络直接模型参考自适应控制程序为chap9_4.m

9.7 一种简单的RBF网络自适应控制

9.7.1 问题描述

考虑一种简单的动力学系统：

$$\ddot{\theta} = f\left(\theta, \dot{\theta}\right) + u \tag{9.30}$$

其中 θ 为转动角度，u 为控制输入。

写成状态方程形式为：

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f\left(x\right) + u \end{aligned} \tag{9.31}$$

其中 f(x)为未知。

位置指令为 x_d，则误差及其导数为

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

定义误差函数为

$$s = \lambda e + \dot{e} \qquad \lambda > 0 \tag{9.32}$$

则

$$\dot{s} = \lambda \dot{e} + \ddot{e} = \lambda \dot{e} + \dot{x}_2 - \ddot{x}_{\mathrm{d}} = \lambda \dot{e} + f(x) + u - \ddot{x}_{\mathrm{d}}$$

由式（9.32）可见，如果$s \to 0$，则 $e \to 0$ 且 $\dot{e} \to 0$。

9.7.2 RBF网络原理

由于RBF网络具有万能逼近特性[20]，采用RBF神经网络逼近f(x)，网络算法为：

$$h_j = \exp\left(\frac{\left\|\boldsymbol{x}-\boldsymbol{c}_j\right\|^2}{2b_j^2}\right) \tag{9.33}$$

$$f = \boldsymbol{W}^{*\mathrm{T}}\boldsymbol{h}(\boldsymbol{x}) + \varepsilon \tag{9.34}$$

其中 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，$\varepsilon \le \varepsilon_{\mathrm{N}}$。

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

$$\hat{f}(x) = \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{h}(x) \tag{9.35}$$

9.7.3 控制算法设计与分析

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

定义Lyapunov函数为

$$V=\frac{1}{2}s^2+\frac{1}{2\gamma}\tilde{\boldsymbol{W}}^{\mathrm{T}}\tilde{\boldsymbol{W}} \tag{9.36}$$

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

则

$$\dot{V}=s\dot{s}+\frac{1}{\gamma}\tilde{W}^{\mathrm{T}}\dot{\hat{W}}=s\left(c\dot{e}+f\left(x\right)+u-\ddot{x}_{\mathrm{d}}\right)+\frac{1}{\gamma}\tilde{\boldsymbol{W}}^{\mathrm{T}}\dot{\hat{\boldsymbol{W}}}$$

设计控制律为

$$u=-c\dot{e}-\hat{f}\left(x\right)+\ddot{x}_{\mathrm{d}}-\eta\,\mathrm{sgn}\left(s\right) \tag{9.37}$$

则

$$\dot{V} = s\left(f(x) - \hat{f}(x) - \eta \operatorname{sgn}(s)\right) + \frac{1}{\gamma}\tilde{\boldsymbol{W}}^{\mathrm{T}}\dot{\hat{\boldsymbol{W}}}$$

$$= s\left(-\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{x}) + \varepsilon - \eta \operatorname{sgn}(s)\right) + \frac{1}{\gamma}\tilde{\boldsymbol{W}}^{\mathrm{T}}\dot{\hat{\boldsymbol{W}}}$$

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

取 $\eta > |\varepsilon|_{\max}$ ，自适应律为

$$\dot{\hat{\boldsymbol{W}}} = \gamma s \boldsymbol{h}(\boldsymbol{x}) \qquad (9.38)$$

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

9.7.4 仿真实例

考虑如下被控对象

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

式中，$f(x)=10x_1x_2$

位置指令为 $x_d=\sin t$，控制律采用式（9.37），自适应律采用式（9.38），取 $\gamma=500$，$\eta=0.50$。根据网络输入x1 和x2 的实际范围来设计高斯基函数的参数，参数 ci和 bi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0.10。仿真结果如图9.25和图9.26所示。

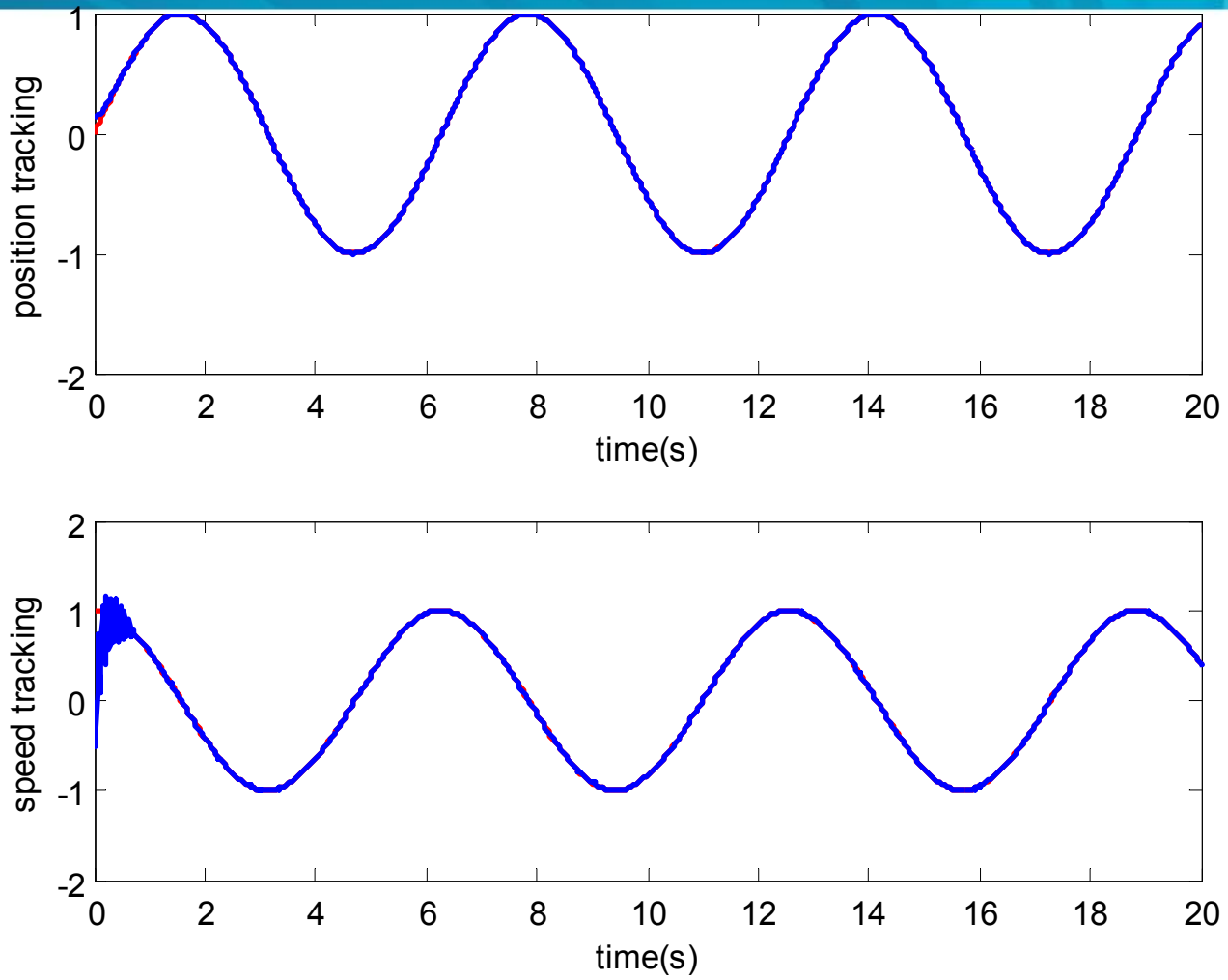


图9.25 位置和速度跟踪

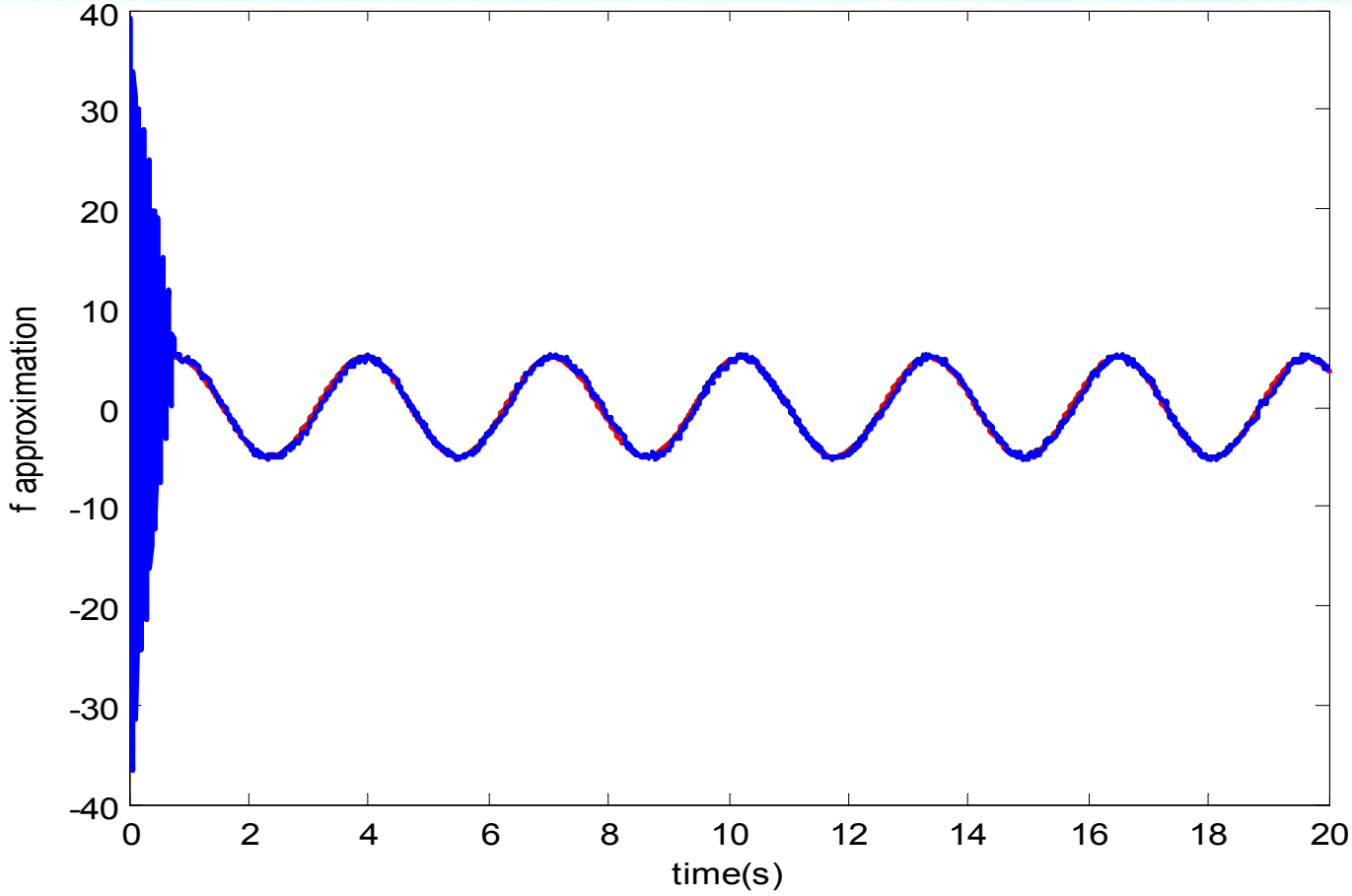

40
30
20
10
0
-10
-20
-30
-40
f approximation
0
2
4
6
8
10
12
14
16
18
20
time(s)

图9.26 f(x)及逼近

9.8 基于不确定逼近的RBF网络自适应控制

通过对文献[18]的控制方法进行详细推导及仿真分析，研究基于模型不确定逼近的RBF网络机器人自适应控制的设计方法。

9.8.1 问题的提出

设n关节机械手的动力学方程为：

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{d} \qquad (9.39)$$

其中 $\mathbf{D}(\mathbf{q})$ 为 $n\times n$ 阶正定惯性矩阵，$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$为 $n\times n$ 阶惯性矩阵，$\mathbf{G}(\mathbf{q})$为 $n\times 1$ 阶惯性向量。

如果模型建模精确，且 $\mathbf{d}=0$，则控制律可设计为：

$$\boldsymbol{\tau} = \mathbf{D}(\mathbf{q})\left(\ddot{\mathbf{q}}_d - \mathbf{k}_v\dot{\mathbf{e}} - \mathbf{k}_p\mathbf{e}\right) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \qquad (9.40)$$

将控制律式（9.40）代入式（9.39）中，得到稳定的闭环系统为

$$\ddot{e}+k_v\dot{e}+k_p e=0 \qquad (9.41)$$

其中 q_d 为理想的角度，$e=q-q_d$ ，$\dot{e}=\dot{q}-\dot{q}_d$ 。

在实际工程中，对象的实际模型很难得到，即无法得到精的 $D(q)$、$C(q,\dot{q})$、$G(q)$ ，只能建立理想的名义模型。

将机器人名义模型表示为 $D_0(q)$ 、$C_0(q,\dot{q})$、$G_0(q)$ ，针对名义模型的控制律设计为：

$$\tau=D_0(q)\left(\ddot{q}_d-k_v\dot{e}-k_p e\right)+C_0(q,\dot{q})\dot{q}+G_0(q) \qquad (9.42)$$

将控制律式（9.42）代入式（9.39）中，得

$$D(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=D_0(q)\left(\ddot{q}_d-k_v\dot{e}-k_p e\right)+C_0(q,\dot{q})\dot{q}+G_0(q)+d \qquad (9.43)$$

取 $\Delta D=D_0-D$ ，$\Delta C=C_0-C$ ，$\Delta G=G_0-G$ ，则

$$\ddot{e}+k_v\dot{e}+k_p e=D_0^{-1}(\Delta D\ddot{q}+\Delta C\dot{q}+\Delta G+d) \qquad (9.44)$$

由式（9.44）可见，由于模型建模的不精确会导致控制性能的下降。因此，需要对建模不精确部分进行逼近。

式（9.44）中，取建模不精确部分 $\mathbf{f}(\mathbf{x})=\mathbf{D}_0^{-1}(\Delta\mathbf{D}\ddot{\mathbf{q}}+\Delta\mathbf{C}\dot{\mathbf{q}}+\Delta\mathbf{G}+\mathbf{d})$ 则可将式（9.44）转化为如下误差状态方程

$$\dot{x}=Ax+Bf \qquad (9.45)$$

式中，$A=\begin{pmatrix}0 & I\\ -k_p & -k_v\end{pmatrix}$，$B=\begin{pmatrix}0\\ I\end{pmatrix}$，I为单位阵。

假设模型不确定项f为已知，则可设计修正的控制律为：

$$\boldsymbol{\tau}=\mathbf{D}_0(\mathbf{q})\left(\ddot{\mathbf{q}}_d-\mathbf{k}_v\dot{\mathbf{e}}-\mathbf{k}_p\mathbf{e}\right)+\mathbf{C}_0(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{G}_0(\mathbf{q})-D_0(q)f \qquad (9.46)$$

将控制律式（9.46）代入式（9.39）中，则可得到稳定的闭环系统式（9.41）。

在实际工程中，模型不确定项 f 为未知，为此，需要对不确定项f进行逼近，从而在控制律中实现对不确定项 f的补偿。

9.8.2 模型不确定部分的RBF网络逼近

采用RBF网络对不确定项 $\mathbf{f}(\mathbf{x})$ 进行自适应逼近。

RBF网络算法为：

$$\phi_i = g\left(\left\|\mathbf{x}-\mathbf{c}_i\right\|^2 / b_i^2\right) \quad i=1,2\cdots,n$$

$$\mathbf{y}=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x})$$

其中X为网络的输入信号，Y为网络的输出信号，$\boldsymbol{\varphi}=[\phi_1,\phi_2,\cdots,\phi_n]$，为高斯基函数的输出，$\theta$ 为神经网络权值。

由已知证明可知[19,20]，在下述假设条件下，RBF网络针对连续函数在紧集范围内具有任意精度的逼近能力。

假设：

- （1）神经网络输出 $\hat{f}(x,\theta)$ 为连续；
- （2）存在理想逼近的神经网络输出 $\hat{f}(x,\theta^*)$ ，针对一个非常小的正实数 ε_0 ，有 $\max\left\|\hat{\mathbf{f}}\left(\mathbf{x},\boldsymbol{\theta}^*\right)-\mathbf{f}\left(\mathbf{x}\right)\right\|\le\varepsilon_0$

误差状态方程式（9.45）可写为

$$\dot{x} = Ax + B\left\{\hat{f}(x,\theta^*) + [f(x) - \hat{f}(x,\theta^*)]\right\} \tag{9.47}$$

其中，$\boldsymbol{\theta}^* = \arg\min_{\theta\in\beta(M_\theta)}\left\{\sup_{x\in\varphi(M_x)}\left\|\mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta})\right\|\right\}$，$\boldsymbol{\theta}^*$为 $n\times n$阶矩阵，表示对 $\mathbf{f}(\mathbf{x})$最佳逼近的神经网络权值。

式(9.47)可写为

$$\dot{x} = Ax + B\left\{\hat{f}(x,\theta^*) + \eta\right\} \tag{9.48}$$

取 $\boldsymbol{\eta}$ 为理想神经网络的逼近误差，即

$$\boldsymbol{\eta} = \mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta}^*) \tag{9.49}$$

逼近误差 η 为有界，其界为 η_0，即

$$\eta_0=\sup\left\|f(x)-\hat{f}(x,\theta^*)\right\| \tag{9.50}$$

神经网络输出 $\hat{f}(\bullet)$ 的最佳估计值为

$$\hat{f}(x,\theta^*)=\theta^{*T}\varphi(x) \tag{9.51}$$

则式（9.48）可写为

$$\dot{x}=Ax+B\left\{\theta^{*T}\varphi(x)+\eta\right\} \tag{9.52}$$

9.8.3 控制器的设计及分析

控制器设计为

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2 \tag{9.53}$$

其中

$$\boldsymbol{\tau}_1 = \mathbf{D}_0(\mathbf{q})\left(\ddot{\mathbf{q}}_d - \mathbf{k}_v\dot{\mathbf{e}} - \mathbf{k}_p\mathbf{e}\right) + \mathbf{C}_0(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_0(\mathbf{q}) \tag{9.54}$$

$$\boldsymbol{\tau}_2 = -\mathbf{D}_0(\mathbf{q})\hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta}) \tag{9.55}$$

其中 $\hat{\theta}$ 为 θ^* 的估计值，$\hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta}) = \hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x})$ 。

将控制律式（9.53）代入式（9.39）中，有

$$\begin{aligned}&\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})\\ &= \mathbf{D}_0(\mathbf{q})\left(\ddot{\mathbf{q}}_d - \mathbf{k}_v\dot{\mathbf{e}} - \mathbf{k}_p\mathbf{e}\right) + \mathbf{C}_0(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_0(\mathbf{q}) - \mathbf{D}_0(\mathbf{q})\hat{\mathbf{f}} + \mathbf{d}\end{aligned}$$

$\mathbf{D}_0(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_0(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_0(\mathbf{q})$ 分别减去上式左右两边，得：

即

$$\begin{aligned}&\Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \Delta\mathbf{G}(\mathbf{q}) + \mathbf{d}\\ &= \mathbf{D}_0(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{D}_0(\mathbf{q})\left(\ddot{\mathbf{q}}_d - \mathbf{k}_v\dot{\mathbf{e}} - \mathbf{k}_p\mathbf{e}\right) + \mathbf{D}_0(\mathbf{q})\hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta})\end{aligned}$$

$$\Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \Delta\mathbf{G}(\mathbf{q}) + \mathbf{d} = \mathbf{D}_0(\mathbf{q})\left(\ddot{\mathbf{e}} + \mathbf{k}_v\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta})\right)$$

$$\ddot{\mathbf{e}} + \mathbf{k}_v\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta}) = \mathbf{D}_0^{-1}(\mathbf{q})\left(\Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \Delta\mathbf{G}(\mathbf{q}) + \mathbf{d}\right)$$

$$\ddot{\mathbf{e}} + \mathbf{k}_v\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta}) = \mathbf{f}(\mathbf{x})$$

式中，$\mathbf{f}(\mathbf{x}) = \mathbf{D}_0^{-1}\left(\Delta\mathbf{D}\ddot{\mathbf{q}} + \Delta\mathbf{C}\dot{\mathbf{q}} + \Delta\mathbf{G} + \mathbf{d}\right)$

上式可写为：$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\left\{\mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta})\right\}$

其中 $\mathbf{A} = \begin{pmatrix} 0 & \mathbf{I} \\ -\mathbf{k}_p & -\mathbf{k}_v \end{pmatrix}$，$\mathbf{B} = \begin{pmatrix} 0 \\ \mathbf{I} \end{pmatrix}$

由于

$$\mathbf{f}(\mathbf{x})-\hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta})=\mathbf{f}(\mathbf{x})-\hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta}^*)+\hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta}^*)-\hat{\mathbf{f}}(\mathbf{x},\boldsymbol{\theta})=\boldsymbol{\eta}+\boldsymbol{\theta}^{*\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x})-\hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x})=\boldsymbol{\eta}+\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x})$$

则

$$\dot{\mathbf{x}}=\mathbf{A}\mathbf{x}+\mathbf{B}\left(\boldsymbol{\eta}-\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x})\right) \qquad (9.56)$$

其中 $\tilde{\boldsymbol{\theta}}=\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*$ 。

定义Lyapunov函数为

$$V=\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}+\frac{1}{2\gamma}\left\|\tilde{\boldsymbol{\theta}}\right\|^2 \qquad (9.57)$$

其中 $\gamma>0$ 。

存在正定矩P和Q，并满足如下Lyapunov方程

$$\mathbf{P}\mathbf{A}+\mathbf{A}^{\mathrm{T}}\mathbf{P}=-\mathbf{Q} \qquad (9.58)$$

定义

$$\|\mathbf{R}\|^2=\sum_{i,j}\left|r_{ij}\right|^2=\mathrm{tr}\left(\mathbf{R}\mathbf{R}^{\mathrm{T}}\right)=\mathrm{tr}\left(\mathbf{R}^{\mathrm{T}}\mathbf{R}\right)$$

其中 tr(□)为矩阵的迹，则

$$\left\|\tilde{\boldsymbol{\theta}}\right\|_F^{\ 2}=\mathrm{tr}\left(\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\tilde{\boldsymbol{\theta}}\right)$$

$$\begin{aligned}\dot{V}&=\frac{1}{2}\left[\mathbf{x}^{\mathrm{T}}\mathbf{P}\dot{\mathbf{x}}+\dot{\mathbf{x}}^{\mathrm{T}}\mathbf{P}\mathbf{x}\right]+\frac{1}{\gamma}\mathrm{tr}\left(\dot{\tilde{\boldsymbol{\theta}}}^{\mathrm{T}}\tilde{\boldsymbol{\theta}}\right)\\&=\frac{1}{2}\left[\mathbf{x}^{\mathrm{T}}\mathbf{P}\left(\mathbf{A}\mathbf{x}+\mathbf{B}\left(-\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\varphi}\left(\mathbf{x}\right)+\boldsymbol{\eta}\right)\right)+\left(\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}+\left(-\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\varphi}\left(\mathbf{x}\right)+\boldsymbol{\eta}\right)^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\right)\mathbf{P}\mathbf{x}\right]+\frac{1}{\gamma}\mathrm{tr}\left(\dot{\tilde{\boldsymbol{\theta}}}^{\mathrm{T}}\tilde{\boldsymbol{\theta}}\right)\\&=\frac{1}{2}\left[\mathbf{x}^{\mathrm{T}}\left(\mathbf{P}\mathbf{A}+\mathbf{A}^{\mathrm{T}}\mathbf{P}\right)\mathbf{x}+\left(-\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{B}\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\varphi}\left(\mathbf{x}\right)+\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{B}\boldsymbol{\eta}-\boldsymbol{\varphi}^{\mathrm{T}}\left(\mathbf{x}\right)\tilde{\boldsymbol{\theta}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}+\boldsymbol{\eta}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}\right)\right]+\frac{1}{\gamma}\mathrm{tr}\left(\dot{\tilde{\boldsymbol{\theta}}}^{\mathrm{T}}\tilde{\boldsymbol{\theta}}\right)\\&=-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}-\boldsymbol{\varphi}^{\mathrm{T}}\left(\mathbf{x}\right)\tilde{\boldsymbol{\theta}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}+\boldsymbol{\eta}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}+\frac{1}{\gamma}\mathrm{tr}\left(\dot{\tilde{\boldsymbol{\theta}}}^{\mathrm{T}}\tilde{\boldsymbol{\theta}}\right)\end{aligned}$$

其中$\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{B}\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\varphi}\left(\mathbf{x}\right)=\boldsymbol{\varphi}^{\mathrm{T}}\left(\mathbf{x}\right)\tilde{\boldsymbol{\theta}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}$，$\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{B}\boldsymbol{\eta}=\boldsymbol{\eta}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}$，$P^T=P$。

由于
$$\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x})\tilde{\boldsymbol{\theta}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}=\mathrm{tr}\left[\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x})\tilde{\boldsymbol{\theta}}\right] \tag{9.60}$$

则
$$\dot{V}=-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}+\frac{1}{\gamma}\mathrm{tr}\left(-\gamma\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x})\tilde{\boldsymbol{\theta}}+\dot{\tilde{\boldsymbol{\theta}}}^{\mathrm{T}}\tilde{\boldsymbol{\theta}}\right)+\boldsymbol{\eta}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x} \tag{9.61}$$

可以采用以下两种自适应律设计方法。

（1）自适应律一

取自适应律为：
$$\dot{\hat{\boldsymbol{\theta}}}^{\mathrm{T}}=\gamma\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x}) \tag{9.62}$$

即
$$\dot{\hat{\boldsymbol{\theta}}}=\gamma\boldsymbol{\varphi}(\mathbf{x})\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{B}$$

则

$$\dot{V} = -\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \boldsymbol{\eta}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}$$

由已知

$$\left\|\boldsymbol{\eta}^{\mathrm{T}}\right\| \le \left\|\boldsymbol{\eta}_0\right\| \quad \left\|\mathbf{B}\right\| = 1$$

设 $\lambda_{\min}(\mathbf{Q})$ 为矩阵Q特征值的最小值，$\lambda_{\max}(P)$为矩阵P特征值的最大值，则

$$\dot{V} \le -\frac{1}{2}\lambda_{\min}(\mathbf{Q})\|\mathbf{x}\|^2 + \|\boldsymbol{\eta}_0\|\lambda_{\max}(\mathbf{P})\|\mathbf{x}\|$$

$$= -\frac{1}{2}\|\mathbf{x}\|\left[\lambda_{\min}(\mathbf{Q})\|\mathbf{x}\| - 2\|\boldsymbol{\eta}_0\|\lambda_{\max}(\mathbf{P})\right] \qquad (9.63)$$

要使 $\dot{V} \le 0$，需要 $\lambda_{\min}(\mathbf{Q}) \ge \frac{2\lambda_{\max}(\mathbf{P})}{\|\mathbf{x}\|}\eta_0$，即X的收敛半径为

$$\|\mathbf{x}\| = \frac{2\lambda_{\max}(\mathbf{P})}{\lambda_{\min}(\mathbf{Q})}\boldsymbol{\eta}_0$$

可见，当Q的特征值越大、P的特征值越小、神经网络建模误差 η 的上界 η_0 越小时，X的收敛半径越小，跟踪效果越好。但该方法不能保证权值 $\tilde{\theta}=\theta^*-\hat{\theta}$的有界性。为此，可采用自适应律二。

（2）自适应律二

取自适应律为：

$$\dot{\hat{\theta}} = \gamma \varphi \mathbf{x}^{\mathrm{T}} \mathbf{P} \mathbf{B} + k_1 \gamma \|\mathbf{x}\| \hat{\theta} \qquad (9.65)$$

将式（9.65）代入式（9.61）得：

$$\dot{V} = -\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \frac{1}{\gamma}\mathrm{tr}\left(k_1\gamma\|\mathbf{x}\|\hat{\theta}^{\mathrm{T}}\tilde{\theta}\right) + \boldsymbol{\eta}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}$$

$$= -\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + k_1\|\mathbf{x}\|\mathrm{tr}\left(\hat{\theta}^{\mathrm{T}}\tilde{\theta}\right) + \boldsymbol{\eta}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}$$

根据F范数的性质，有 $\mathrm{tr}\left[\tilde{\mathbf{x}}^{\mathrm{T}}\left(\mathbf{x}-\tilde{\mathbf{x}}\right)\right]\le\left\|\tilde{\mathbf{x}}\right\|_{\mathrm{F}}\left\|\mathbf{x}\right\|_{\mathrm{F}}-\left\|\tilde{\mathbf{x}}\right\|_{\mathrm{F}}^{2}$ ，则

$$\mathrm{tr}\left[\hat{\boldsymbol{\theta}}^{\mathrm{T}}\tilde{\boldsymbol{\theta}}\right]=\mathrm{tr}\left[\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\hat{\boldsymbol{\theta}}\right]=\mathrm{tr}\left[\tilde{\boldsymbol{\theta}}^{\mathrm{T}}\left(\boldsymbol{\theta}^{*}+\tilde{\boldsymbol{\theta}}\right)\right]\le\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}}\left\|\boldsymbol{\theta}^{*}\right\|_{\mathrm{F}}-\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}}^{2}$$

则

$$\begin{aligned}\dot{V}&\le-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}+k_1\left\|\mathbf{x}\right\|\left(\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}}\left\|\boldsymbol{\theta}^{*}\right\|_{\mathrm{F}}-\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}}^{2}\right)+\boldsymbol{\eta}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{P}\mathbf{x}\\&\le-\frac{1}{2}\lambda_{\min}\left(\mathbf{Q}\right)\left\|\mathbf{x}\right\|^{2}+k_1\left\|\mathbf{x}\right\|\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}}\left\|\boldsymbol{\theta}^{*}\right\|_{\mathrm{F}}-k_1\left\|\mathbf{x}\right\|\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}}^{2}+\left\|\boldsymbol{\eta}_0\right\|\lambda_{\max}\left(\mathbf{P}\right)\left\|\mathbf{x}\right\|\\&\le-\left\|\mathbf{x}\right\|\left(\frac{1}{2}\lambda_{\min}\left(\mathbf{Q}\right)\left\|\mathbf{x}\right\|-k_1\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}}\theta_{\max}+k_1\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}}^{2}-\left\|\boldsymbol{\eta}_0\right\|\lambda_{\max}\left(\mathbf{P}\right)\right)\\&=-\left\|\mathbf{x}\right\|\left(\frac{1}{2}\lambda_{\min}\left(\mathbf{Q}\right)\left\|\mathbf{x}\right\|+k_1\left(\left\|\tilde{\boldsymbol{\theta}}\right\|_{F}-\frac{\theta_{\max}}{2}\right)^{2}-\frac{k_1}{4}\theta_{\max}^{2}-\left\|\boldsymbol{\eta}_0\right\|\lambda_{\max}\left(\mathbf{P}\right)\right)\end{aligned}$$

要使 $\dot{V}\le0$，需要满足以下条件：$\frac{1}{2}\lambda_{\min}\left(\mathbf{Q}\right)\left\|\mathbf{x}\right\|\ge\left\|\boldsymbol{\eta}_0\right\|\lambda_{\max}\left(\mathbf{P}\right)+\frac{k_1}{4}\theta_{\max}^{2}$

或

$$\mathbf{k}_1\left(\left\|\tilde{\boldsymbol{\theta}}\right\|_{F}-\frac{\theta_{\max}}{2}\right)^{2}\ge\left\|\boldsymbol{\eta}_0\right\|\lambda_{\max}\left(\mathbf{P}\right)+\frac{k_1}{4}\theta_{\max}^{2}$$

即收敛条件为：

$$\|\mathbf{x}\| \geq \frac{2}{\lambda_{\min}(\mathbf{Q})}\left(\|\boldsymbol{\eta}_0\|\lambda_{\max}(\mathbf{P})+\frac{k_1}{4}\boldsymbol{\theta}_{\max}^2\right)$$

或

$$\left\|\tilde{\boldsymbol{\theta}}\right\|_{\mathrm{F}} \geq \frac{\theta_{\max}}{2}+\sqrt{\frac{1}{k_1}\left(\|\boldsymbol{\eta}_0\|\lambda_{\max}(\mathbf{P})+\frac{k_1}{4}\theta_{\max}^2\right)}$$

因此，采用自适应律二，可保证权值的有界性，即解决神经网络权值的收敛问题。

从 $\|x\|$ 的收敛情况可知：当Q的特征值越大、P的特征值越小、神经网络建模误差 η 的上界 η_0 越小、$\theta_{\max}$ 越小时，则X的收敛半径越小，跟踪效果越好。

9.8.4 仿真实例

选二关节机器人系统（不考虑摩擦力），其动力学模型为：

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{G}(\mathbf{q})=\boldsymbol{\tau}+\mathbf{d}$$

其中

$$\mathbf{D}(\mathbf{q})=\begin{bmatrix} v+q_{01}+2\gamma\cos(q_2) & q_{01}+q_{02}\cos(q_2) \\ q_{01}+q_{02}\cos(q_2) & q_{01} \end{bmatrix}$$

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})=\begin{bmatrix}-q_{02}\dot{q}_2\sin(q_2) & -q_{02}(\dot{q}_1+\dot{q}_2)\sin(q_2)\\ q_{02}\dot{q}_1\sin(q_2) & 0\end{bmatrix}$$

$$\mathbf{G}(\mathbf{q})=\begin{bmatrix}15g\cos q_1+8.75g\cos(q_1+q_2)\\ 8.75g\cos(q_1+q_2)\end{bmatrix}$$

其中 $v=13.33$，$q_{01}=8.98$，$q_{02}=8.75$，$g=9.8$ 。

误差扰动、位置指令和系统的初始状态分别为：

$$d_1=2,\ \ d_2=3, d_3=6$$

$$\omega=d_1+d_2\|\mathbf{e}\|+d_3\|\dot{\mathbf{e}}\|$$

位置指令为：

$$\begin{cases}q_{1\mathrm{d}}=1+0.2\sin(0.5\pi t)\\ q_{2\mathrm{d}}=1-0.2\cos(0.5\pi t)\end{cases}$$

被控对象的初值为

$$[q_1\quad q_2\quad q_3\quad q_4]^{\mathrm{T}}=[0.6\quad 0.3\quad 0.5\quad 0.5]^{\mathrm{T}}$$

控制参数取：

$$\mathbf{Q}=\begin{bmatrix}50 & 0 & 0 & 0\\ 0 & 50 & 0 & 0\\ 0 & 0 & 50 & 0\\ 0 & 0 & 0 & 50\end{bmatrix} \quad \alpha=3 \quad k_p=\begin{bmatrix}\alpha^2 & 0\\ 0 & \alpha^2\end{bmatrix} \quad k_v=\begin{bmatrix}2\alpha & 0\\ 0 & 2\alpha\end{bmatrix}$$

采用控制律式(9.53)和自适应律式(9.62)，采用Simulink和S函数进行控制系统的设计，仿真结果如图9-27至图9-30所示。

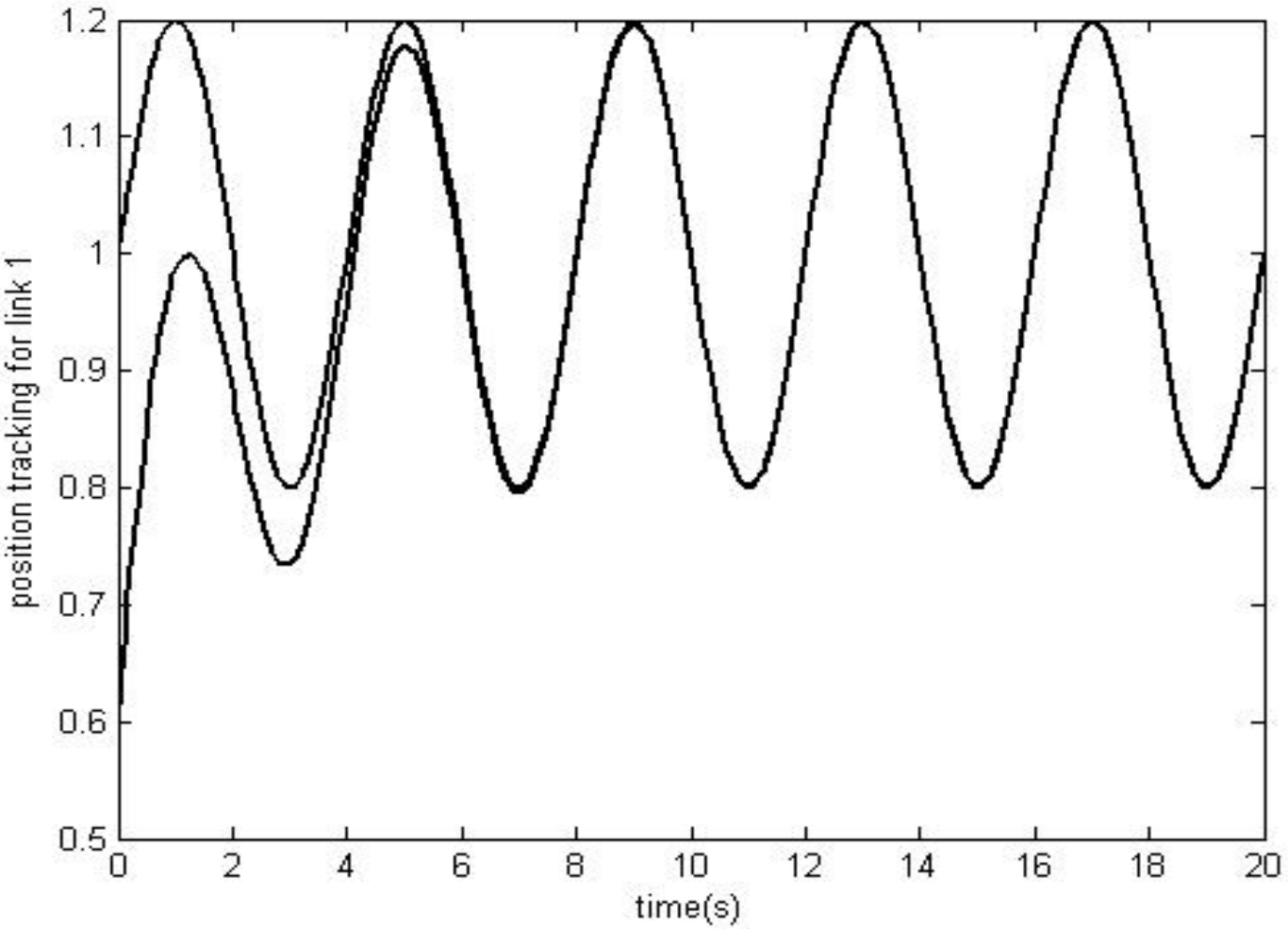


图9-27　关节1的位置跟踪

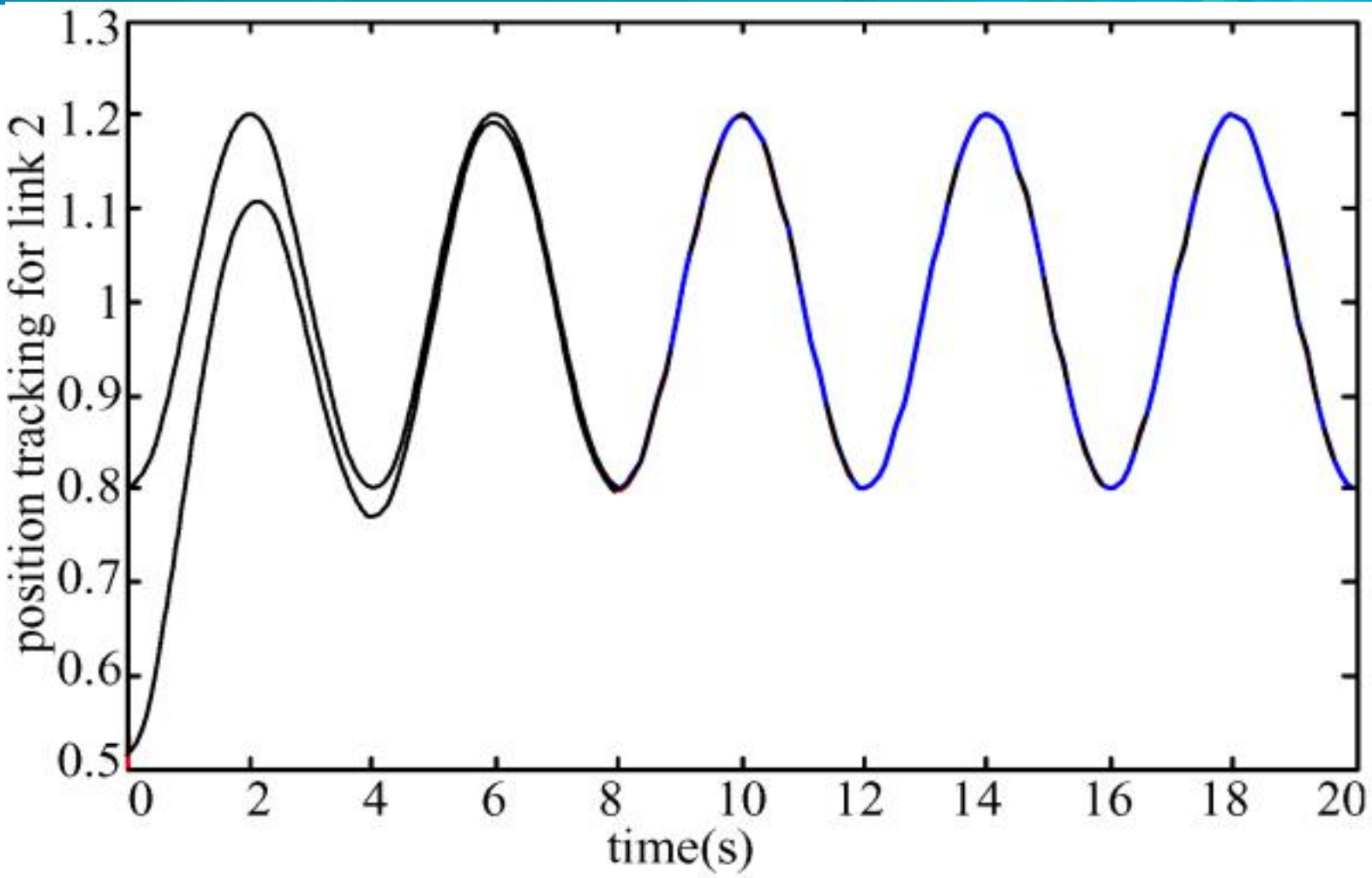


图9-28　关节2的位置跟踪

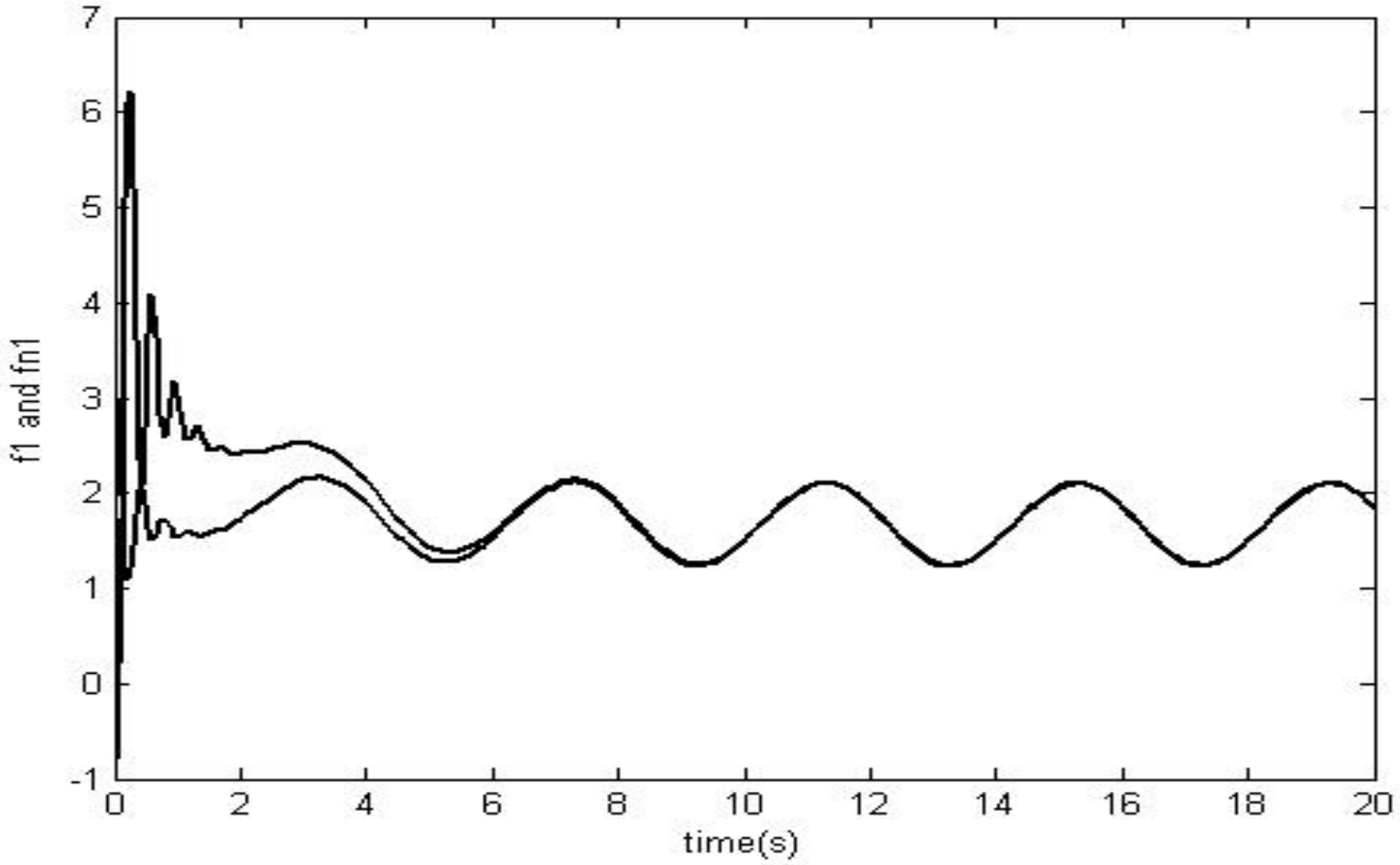


图9-29关节1的模型不确定项及其估计

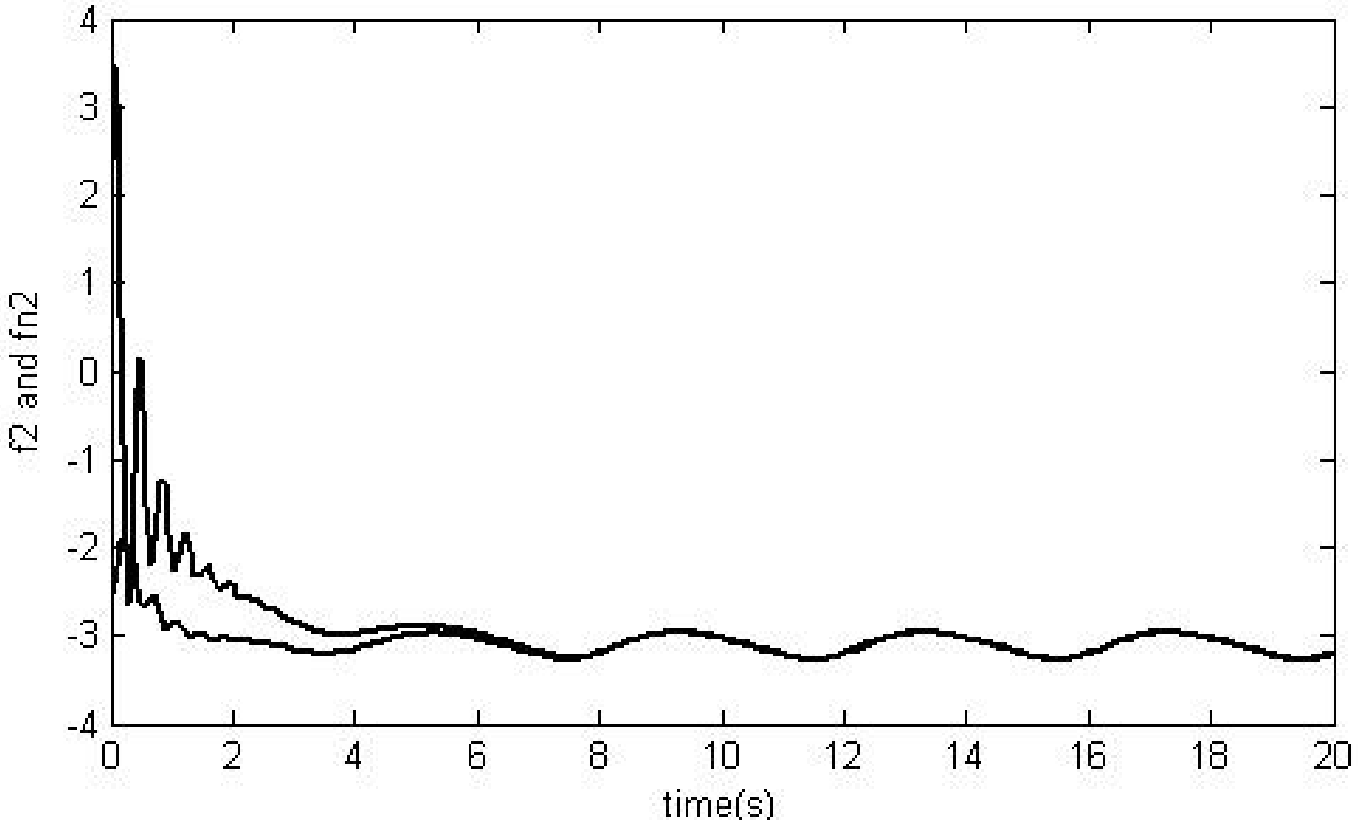


图9-30 关节2的模型不确定项及其逼近

9.9 基于模型整体逼近的机器人RBF网络自适应控制

通过对文献[21]的控制方法进行详细推导及仿真分析，研究基于模型整体逼近的机器人RBF网络自适应控制设计方法。

9.9.1 问题的提出

设n关节机械手方程为：

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{G}(\mathbf{q})+\mathbf{F}(\dot{\mathbf{q}})+\boldsymbol{\tau}_d=\boldsymbol{\tau} \qquad (9.66)$$

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

跟踪误差为：

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

定义误差函数为：

$$\mathbf{r}=\dot{\mathbf{e}}+\Lambda\mathbf{e} \qquad (9.67)$$

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

$$\dot{\mathbf{q}}=-\mathbf{r}+\dot{\mathbf{q}}_{\mathrm{d}}+\Lambda\mathbf{e}$$

$$\begin{aligned}
\mathbf{D}\dot{\mathbf{r}} &= \mathbf{D}\left(\ddot{\mathbf{q}}_{\mathrm{d}}-\ddot{\mathbf{q}}+\Lambda\dot{\mathbf{e}}\right)=\mathbf{D}\left(\ddot{\mathbf{q}}_{\mathrm{d}}+\Lambda\dot{\mathbf{e}}\right)-\mathbf{D}\ddot{\mathbf{q}} \\
&= \mathbf{D}\left(\ddot{\mathbf{q}}_{\mathrm{d}}+\Lambda\dot{\mathbf{e}}\right)+\mathbf{C}\dot{\mathbf{q}}+\mathbf{G}+\mathbf{F}+\boldsymbol{\tau}_{\mathrm{d}}-\boldsymbol{\tau} \\
&= \mathbf{D}\left(\ddot{\mathbf{q}}_{\mathrm{d}}+\Lambda\dot{\mathbf{e}}\right)-\mathbf{C}\mathbf{r}+\mathbf{C}\left(\dot{\mathbf{q}}_{\mathrm{d}}+\Lambda\mathbf{e}\right)+\mathbf{G}+\mathbf{F}+\boldsymbol{\tau}_{\mathrm{d}}-\boldsymbol{\tau} \\
&= -\mathbf{C}\mathbf{r}-\boldsymbol{\tau}+\mathbf{f}+\boldsymbol{\tau}_{\mathrm{d}}
\end{aligned} \qquad (9.68)$$

其中 $\mathbf{f}(\mathbf{x})=\mathbf{D}\left(\ddot{\mathbf{q}}_{\mathrm{d}}+\Lambda\dot{\mathbf{e}}\right)+\mathbf{C}\left(\mathbf{q}_{\mathrm{d}}+\Lambda\mathbf{e}\right)+\mathbf{G}+\mathbf{F}$ 。

在实际工程中，模型不确定项f为未知，为此，需要对不确定项f进行逼近。采用RBF网络逼近f，根据 $\mathbf{f}(\mathbf{x})$ 的表达式，网络输入取 $\mathbf{x}=\begin{bmatrix}\mathbf{e}^{\mathrm{T}} & \dot{\mathbf{e}}^{\mathrm{T}} & \mathbf{q}_{\mathrm{d}}^{\mathrm{T}} & \dot{\mathbf{q}}_{\mathrm{d}}^{\mathrm{T}} & \ddot{\mathbf{q}}_{\mathrm{d}}^{\mathrm{T}}\end{bmatrix}$

设计控制律为：

$$\boldsymbol{\tau}=\hat{\mathbf{f}}+\mathbf{K}_{\mathrm{v}}\mathbf{r} \tag{9.69}$$

其中 $\hat{\mathbf{f}}(\mathbf{x})$ 为针对f进行逼近的RBF网络输出值。

将控制律式（9.69）代入式（9.68），得：

$$\begin{aligned}\mathbf{D}\dot{\mathbf{r}} &= -\mathbf{C}\mathbf{r}-\hat{\mathbf{f}}-\mathbf{K}_{\mathrm{v}}\mathbf{r}+\mathbf{f}+\boldsymbol{\tau}_{\mathrm{d}}\\ &= -\left(\mathbf{K}_{\mathrm{v}}+\mathbf{C}\right)\mathbf{r}+\tilde{\mathbf{f}}+\boldsymbol{\tau}_{\mathrm{d}}=-\left(\mathbf{K}_{\mathrm{v}}+\mathbf{C}\right)\mathbf{r}+\varsigma_{0}\end{aligned} \tag{9.70}$$

如果定义Lyapunov函数

$$L=\frac{1}{2}\mathbf{r}^{\mathrm{T}}\mathbf{D}\mathbf{r}$$

则

$$\dot{L}=\mathbf{r}^{\mathrm{T}}\mathbf{D}\dot{\mathbf{r}}+\frac{1}{2}\mathbf{r}^{\mathrm{T}}\dot{\mathbf{D}}\mathbf{r}=-\mathbf{r}^{\mathrm{T}}\mathbf{K}_{\mathrm{v}}\mathbf{r}+\frac{1}{2}\mathbf{r}^{\mathrm{T}}\left(\dot{\mathbf{D}}-2\mathbf{C}\right)\mathbf{r}+\mathbf{r}^{\mathrm{T}}\varsigma_0$$

$$\dot{L}=\mathbf{r}^{\mathrm{T}}\varsigma_0-\mathbf{r}^{\mathrm{T}}\mathbf{K}_{\mathrm{v}}\mathbf{r}$$

这说明在 $\mathbf{K}_{\mathrm{v}}$ 固定条件下，控制系统的稳定依赖于 ς_0 ，即 $\hat{\mathbf{f}}$ 对 $\mathbf{f}$ 的逼近精度及干扰 τ_{d} 的大小。

采用RBF网络对不确定项f进行逼近。理想的RBF网络算法为：

$$\phi_i=g\left(\left\|\mathbf{x}-\mathbf{c}_i\right\|^2/\sigma_i^2\right)\qquad i=1,2,\cdots,n$$

$$y=\mathbf{W}^{*T}\boldsymbol{\varphi}\left(\mathbf{x}\right)\qquad f(x)=w^{*T}\boldsymbol{\varphi}\left(\mathbf{x}\right)+\varepsilon$$

其中x为网络的输入信号， $\boldsymbol{\varphi}=[\phi_1\ \phi_2\ \cdots\ \phi_n]$

9.9.2 针对进行逼近的控制

1．控制器的设计

采用RBF网络逼近，则RBF神经网络的输出为：

$$\hat{\mathbf{f}}(\mathbf{x})=\hat{\mathbf{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x}) \tag{9.71}$$

取 $\tilde{\mathbf{W}}=\mathbf{W}-\hat{\mathbf{W}}$，$\|\mathbf{W}\|_{\mathrm{F}}\leq W_{\max}$

设计控制律为：

$$\boldsymbol{\tau}=\hat{\mathbf{f}}(\mathbf{x})+\mathbf{K}_{\mathrm{v}}\mathbf{r}-\mathbf{v} \tag{9.72}$$

其中 v 为用于克服神经网络逼近误差 ε 的鲁棒项。

将控制律式（9.72）代入式（9.68），得：

$$\mathbf{D}\dot{\mathbf{r}}=-(\mathbf{K}_{\mathrm{v}}+\mathbf{C})\mathbf{r}+\tilde{\mathbf{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x})+(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})+\mathbf{v}=-(\mathbf{K}_{\mathrm{v}}+\mathbf{C})\mathbf{r}+\boldsymbol{\varsigma}_{1} \tag{9.73}$$

其中 $\boldsymbol{\varsigma}_{1}=\tilde{\mathbf{W}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x})+(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}})+\mathbf{v}$ 。

将鲁棒项v设计为：

$$\mathbf{v}=-\left(\varepsilon_{\mathrm{N}}+b_{\mathrm{d}}\right)\mathrm{sgn}\left(\mathbf{r}\right) \tag{9.74}$$

其中 $\|\boldsymbol{\varepsilon}\|\leq e_{\mathrm{N}}$ ，$\|\boldsymbol{\tau}_{\mathrm{d}}\|\leq b_{\mathrm{d}}$ 。

2．稳定性及收敛性分析

定义Lyapunov函数：$L=\frac{1}{2}\mathbf{r}^{\mathrm{T}}\mathbf{D}\mathbf{r}+\frac{1}{2}\mathrm{tr}\left(\tilde{\mathbf{W}}^{\mathrm{T}}\mathbf{F}^{-1}\tilde{\mathbf{W}}\right)$

则 $$\dot{L}=\mathbf{r}^{\mathrm{T}}\mathbf{D}\dot{\mathbf{r}}+\frac{1}{2}\mathbf{r}^{\mathrm{T}}\dot{\mathbf{D}}\mathbf{r}+\mathrm{tr}\left(\tilde{\mathbf{W}}^{\mathrm{T}}\mathbf{F}^{-1}\dot{\tilde{\mathbf{W}}}\right)$$

将（9.73）式代入上式，得

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

根据机器人特性有 $\mathbf{r}^{\mathrm{T}}\left(\dot{\mathbf{D}}-2\mathbf{C}\right)\mathbf{r}=0$。取 $\dot{\tilde{\mathbf{W}}}=-\mathbf{F}\varphi\mathbf{r}^{\mathrm{T}}$，即神经网络自适应律为：

$$\dot{\hat{\mathbf{W}}}=\mathbf{F}\varphi\mathbf{r}^{\mathrm{T}} \tag{9.75}$$

则

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

由于

$$\mathbf{r}^{\mathrm{T}}\left(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}}+\mathbf{v}\right)=\mathbf{r}^{\mathrm{T}}\left(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}}\right)+\mathbf{r}^{\mathrm{T}}\mathbf{v}=\mathbf{r}^{\mathrm{T}}\left(\boldsymbol{\varepsilon}+\boldsymbol{\tau}_{\mathrm{d}}\right)-\|\mathbf{r}\|\left(\boldsymbol{\varepsilon}_{\mathrm{N}}+\mathbf{b}_{\mathrm{d}}\right)\leq 0$$

则

$$\dot{L}\leq 0$$

9.9.3 仿真实例

选二关节机器人力臂系统，其动力学模型为：

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

其中

$$\mathbf{D}(\mathbf{q})=\begin{bmatrix} p_1+p_2+2p_3\cos q_2 & p_2+p_3\cos q_2 \\ p_2+p_3\cos q_2 & p_2 \end{bmatrix},\quad \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})=\begin{bmatrix} -p_3\dot{q}_2\sin q_2 & -p_3(\dot{q}_1+\dot{q}_2)\sin q_2 \\ p_3\dot{q}_1\sin q_2 & 0 \end{bmatrix}$$

$$\mathbf{G}(\mathbf{q})=\begin{bmatrix} p_4 g\cos q_1+p_5 g\cos(q_1+q_2) \\ p_5 g\cos(q_1+q_2) \end{bmatrix},\quad \mathbf{F}(\dot{\mathbf{q}})=0.02\,\mathrm{sgn}(\dot{\mathbf{q}}),\quad \boldsymbol{\tau}_{\mathrm{d}}=\begin{bmatrix} 0.2\sin(t) & 0.2\sin(t) \end{bmatrix}^{\mathrm{T}}$$

取 $\mathbf{p}=[p_1,p_2,p_3,p_4,p_5]=[2.9,0.76,0.87,3.04,0.87]$。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 $b=10$ ，网络的初始权值取零，网络输入 $\mathbf{z}=[\mathbf{e}\quad \dot{\mathbf{e}}\quad \mathbf{q}_d\quad \dot{\mathbf{q}}_d\quad \ddot{\mathbf{q}}_d]$。

系统的初始状态为 $[0.09\quad 0\quad -0.09\quad 0]$，两个关节的位置指令分别为 $q_{1d}=0.1\sin t$ ，$q_{2d}=0.1\sin t$ ，控制参数取 $\mathbf{K}_v=\mathrm{diag}\{20,20\}$ ，$\mathbf{F}=\mathrm{diag}\{15,15\}$ ，$\Lambda=\mathrm{diag}\{5,5\}$ ，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仿真。首先采用针对 $f(x)$ 进行逼近的控制器子程序chap9_6ctrl.m，控制律取式（9.72），自适应律取式（9.75），仿真结果如图9-31至图9-33所示。

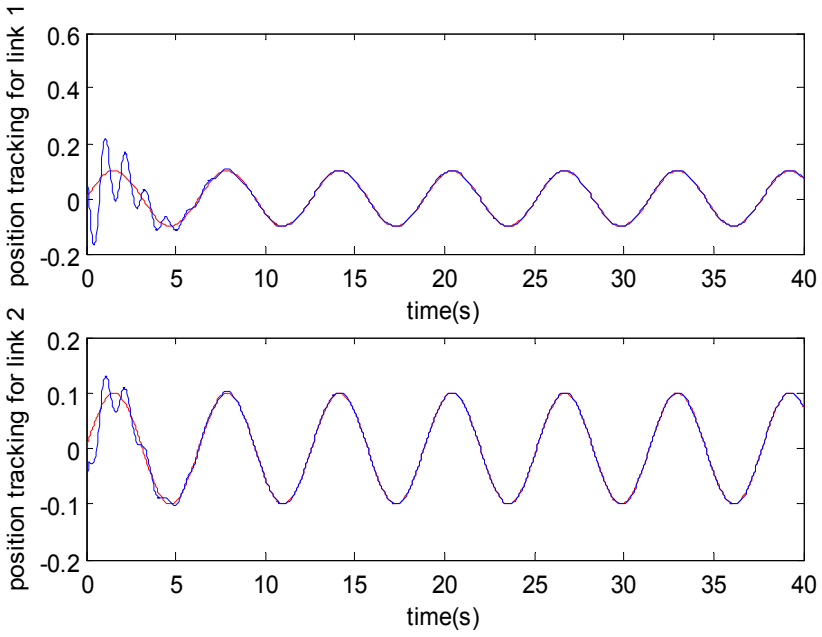


图9-31 关节1及关节2的位置跟踪

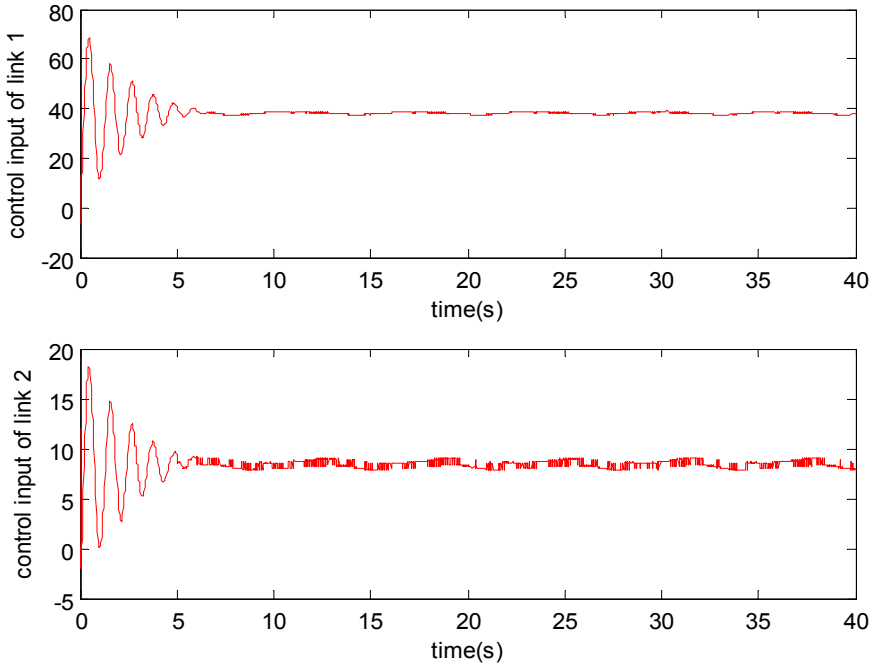
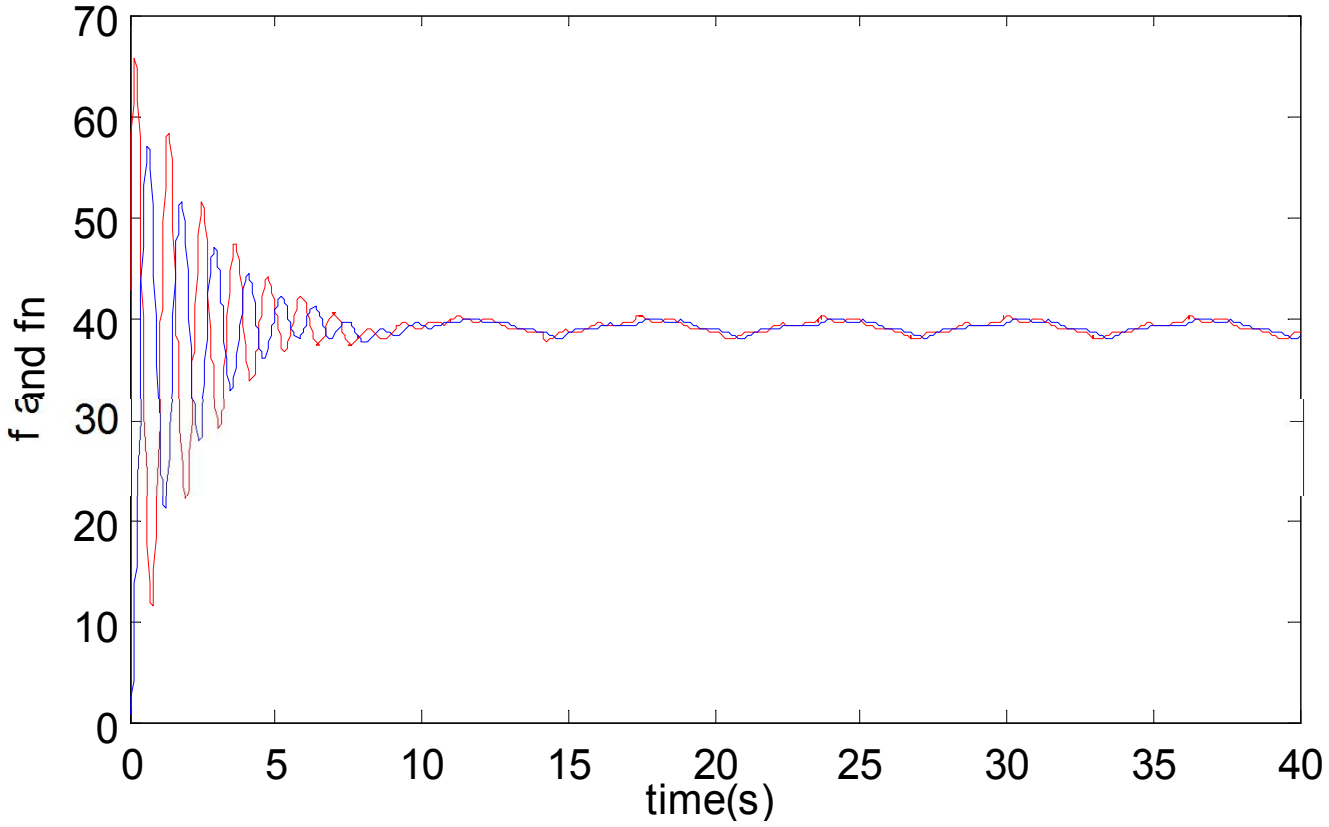


图9-32 关节1及关节2的控制输入



9-33　关节1及关节2的$\|\mathbf{f}(\mathbf{x})\|$及其逼近$\|\hat{\mathbf{f}}(\mathbf{x})\|$

9.10神经网络数字控制

9.10.1基本原理

在工程实际中，控制算法一般在计算机或DSP中实现，这就需要将控制算法离散化。本节讨论神经网络自适应控制律的数学化实现方法。

数字控制系统结构如图9-34所示，其中控制器为数字控制算法，被控对象的输入、输出为模拟信号，通过D/A和A/D与数字信号相连接。数字控制算法的程序框图如图9-35所示。

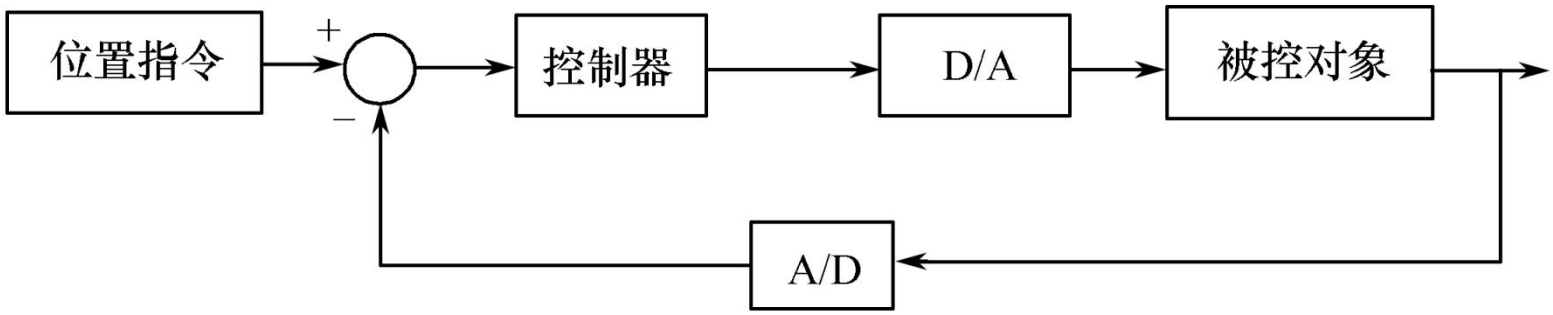


图9-34 数字控制系统结构

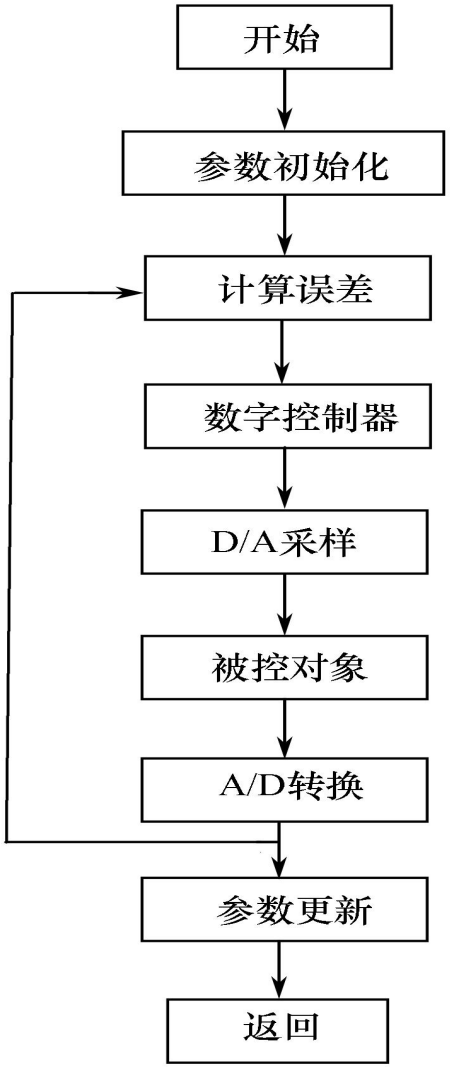


图9-35 数字控制算法程序框图

针对9.8节中的控制算法，选择被控对象为单电机模型，其动力学模型为

$$\mathbf{D}\ddot{\mathbf{q}}+\mathbf{C}\dot{\mathbf{q}}+\mathbf{G}=\boldsymbol{\tau}+d \qquad (9.76)$$

式中，$D_0=\frac{3}{4}ml^2$，$G_0=m\lg\cos q$。

取 $x_1=q$，$x_2=\dot{q}$，则方程式(9.76)可转化为动力学方程

$$\dot{x}_1=x_2$$

$$\dot{x}_2=D^{-1}(\tau+d-C\dot{q}-G) \qquad (9.77)$$

在Matlab仿真中，在每个采样时间T内，采用Runge-Kutta迭代算法求解式(9.77)，从而实现连续被控对象的离散求解，仿真中采用了Matlab函数“ode45”进行离散积分求解。

针对自适应律的离散化问题，一种方法是利用采样时间进行差分的离散化，另一种方法是利用数值迭代方法进行离散化。下面介绍一种高精度数值迭代方法——RKM(Runge-Kutta-Merson)方法。以离散化 $\dot{x} = f(t,x)$ 为例，采样时间为T，如果采用差分方法离散化，n+1时刻的x值为 $x_{n+1} = x_n + Tf(t_n, x_n)$，而采用RKM方法，则n+1时刻的x值为

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 4k_4 + k_5) \qquad (9.78)$$

其中

$$k_1 = Tf\left(t_{\mathrm{n}}, x_{\mathrm{n}}\right)$$

$$k_2 = Tf\left(t_{\mathrm{n}} + \frac{1}{3}T, x_{\mathrm{n}} + \frac{1}{3}k_1\right)$$

$$k_3 = Tf\left(t_{\mathrm{n}} + \frac{1}{3}T, x_{\mathrm{n}} + \frac{1}{6}k_1 + \frac{1}{6}k_2\right)$$

$$k_4 = Tf\left(t_{\mathrm{n}} + \frac{1}{2}T, x_{\mathrm{n}} + \frac{1}{8}k_1 + \frac{3}{8}k_3\right)$$

$$k_5 = Tf\left(t_{\mathrm{n}} + T, x_{\mathrm{n}} + \frac{1}{2}k_1 - \frac{3}{2}k_3 + 2k_4\right)$$

针对自适应律式(9.65)的权值的表达式 $\dot{\hat{\omega}} = \gamma h \mathbf{x}^{\mathrm{T}} \mathbf{P} \mathbf{B} - k_1 \gamma \left\| \mathbf{x} \right\| \hat{\omega}$ 针对每个采样时间，采用RKM迭代算法式(9.78)进行求解，考虑到自适应律表达式中权值 $\hat{\omega}$ 相当于式(9.78)中的 x_n ，且没有出现时间变量t，采样时间为ts，故离散求解式(9.65)的Matlab程序如下：

```
w(i,1)=w_1(i,1)+1/6*(k1+4*k4+k5);

k1=ts*(gama*h(i)*xi'*P*B-k1*gama*norm(xi)*w_1(i,1));

k2=ts*(gama*h(i)*xi'*P*B-k1*gama*norm(xi)*(w_1(i,1)+1/3*k1));

k3=ts*(gama*h(i)*xi'*P*B-
k1*gama*norm(xi)*(w_1(i,1)+1/6*k1+1/6*k2));

k4=ts*(gama*h(i)*xi'*P*B-
k1*gama*norm(xi)*(w_1(i,1)+1/8*k1+3/8*k3));

k5=ts*(gama*h(i)*xi'*P*B-k1*gama*norm(xi)*(w_1(i,1)+1/2*k1-
3/2*k3+2*k4));
```

9.10.2仿真实例

仿真中，采用控制律式(9.53)和自适应律式(9.65)。在离散化自适应律式(9.65)微分方程时，取两种离散化方法：当S=1时，采用简单的差分方法进行离散化，当S=2时，采用RKM方法进行离散化。取m=1，$l=1$，g=9.8，d=0.5sin(t)，$C_0=2$，$\Delta D=0.8D_0$，$\Delta C=0.8C_0$，$\Delta G=0.8G_0$。系统的初始状态为x=[0,0]。位置指令为 $q_d=0.5\sin(k,t_s)$，采样时间取 $t_s=0.001$，取控制其参数为 $k_p=40$，$k_v=20$，$\boldsymbol{Q}=\begin{bmatrix}2000 & 0\\ 0 & 2000\end{bmatrix}$，取自适应律参数为 $\gamma=5$，$k_1=0.01$。RBF网络的隐含层节点数取10，网络权值的初始值取0，高斯基参数初始值的选取见控制器主程序chap9_8.m。取M=1，未采用神经网络补偿，仿真结果如图9-36所示。取M=2，采用神经网络补偿，仿真结果如图9-37和图9-38所示。

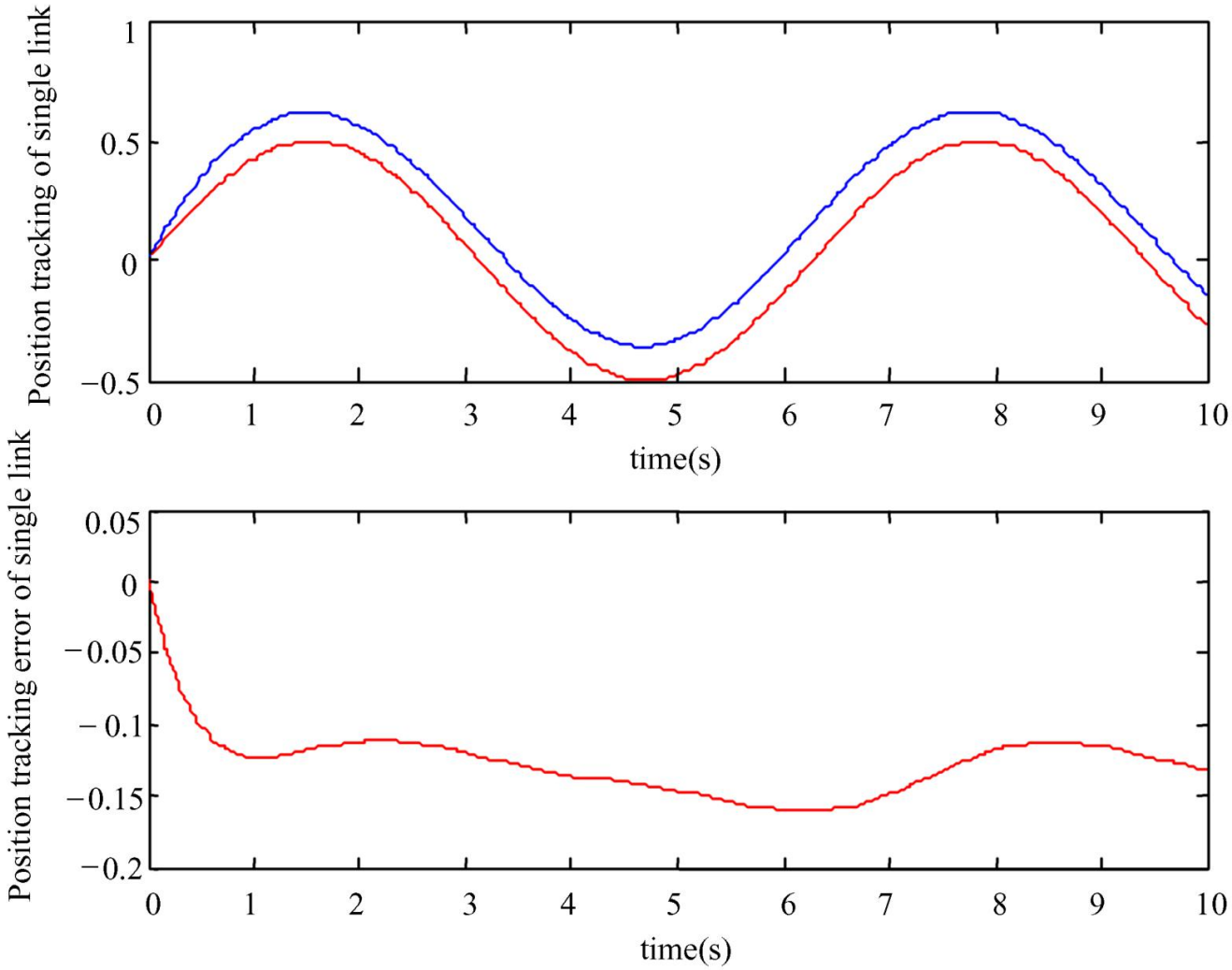


图9-36 未用神经网络补偿的位置跟踪及其误差（M=1）

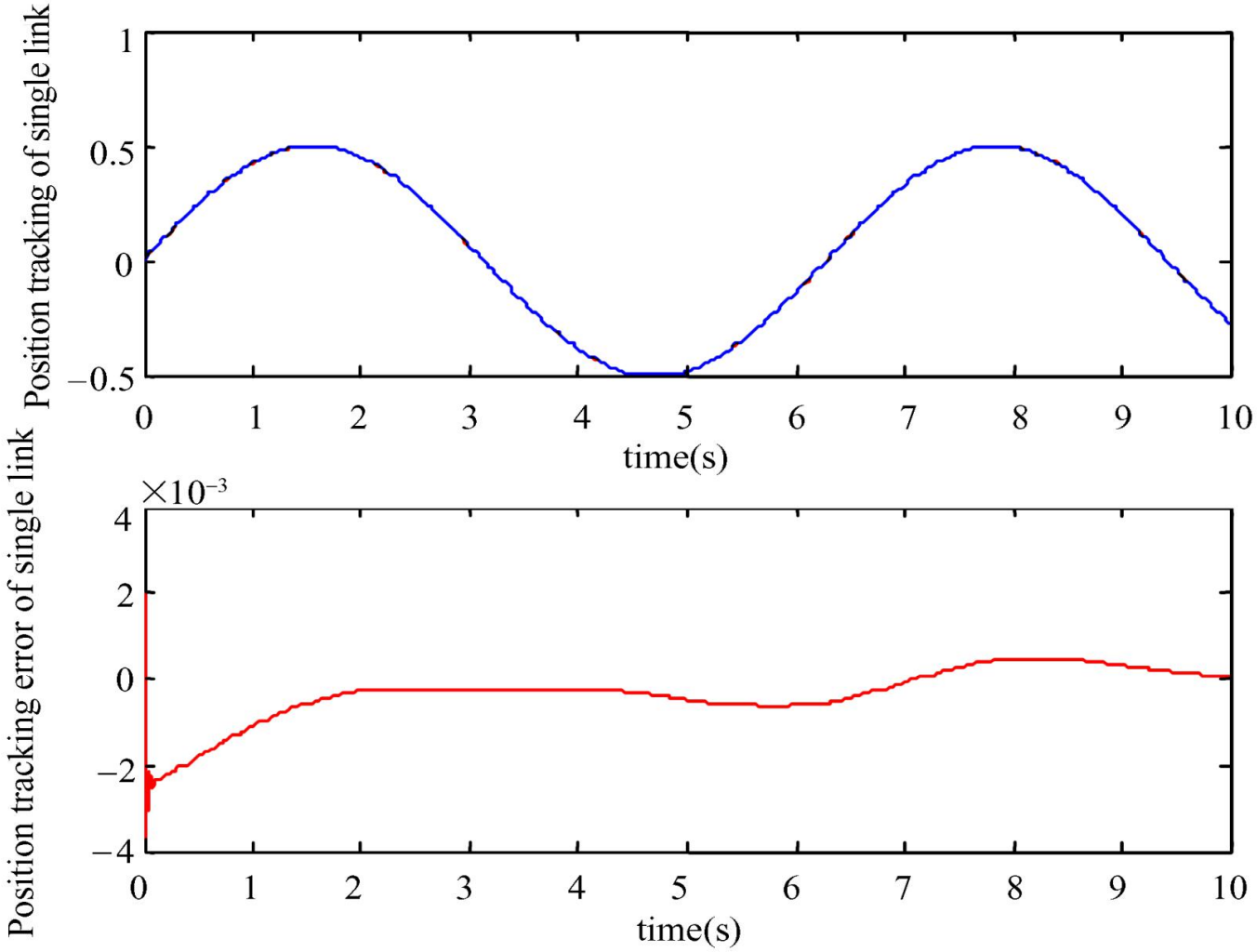


图9-37采用神经网络补偿的位置跟踪及其误差（M=2）

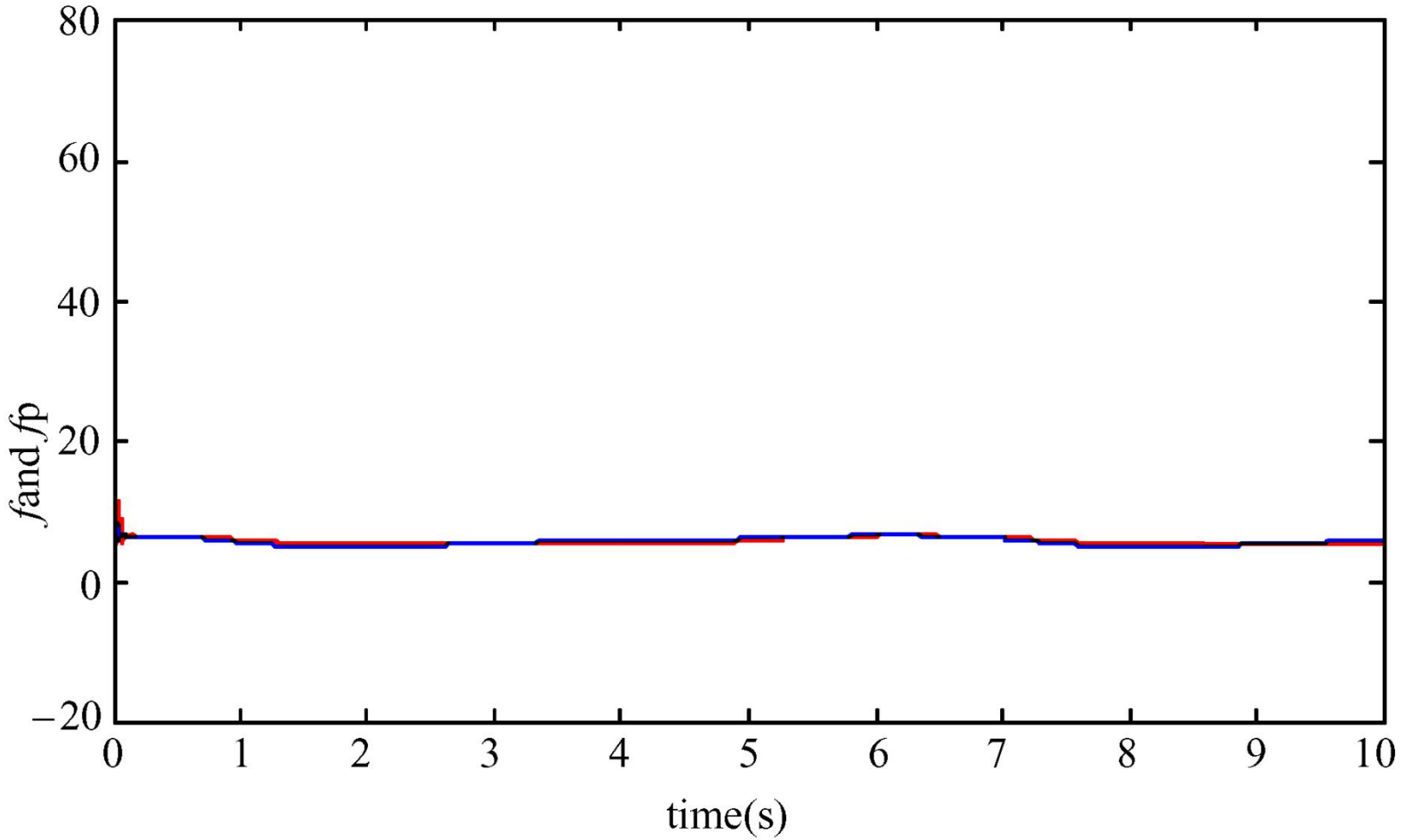


图9-38 不确定项及其神经网络逼近结果（M=2）