第 16 卷第 3 期 控制理论与应用 Vol.16, No.3
1999 年 6 月 CONTROL THEORY AND APPLICATIONS Jun., 1999

Adaptive Control of Nonlinear Discrete-Time Systems Using Neural Networks and Least Squares Algorithm with Dead-Zone

Xie Xuejun and Wang Yuan
(Institute of Systems Science, Chinese Academy of Sciences · Beijing, 100080, P. R. China)

Abstract: Multilayer neural networks are used in a nonlinear discrete-time adaptive control problem. The weights of the neural networks are updated by using least squares (LS) algorithm with dead-zone. LS algorithm has much superior rate of convergence compared with gradient algorithm and δ- modification algorithm. For the adaptive control algorithm, we prove that: 1) all signals in the closed-loop systems are bounded; and 2) the tracking error converges to a bounded ball.

Key words: Nonlinear systems; multilayer neural networks; least squares algorithm with dead-zone; adaptive control

基于神经元网和带死区的最小二乘算法的非线性离散时间系统的自适应控制

解学军 王 远
(中国科学院系统科学研究所·北京, 100080)

摘要: 针对非线性离散时间系统,提出了一种用带死区的最小二乘算法去调节神经网参数的算法,同其他算法相比,这种算法具有非常高的收敛速度.对于这种自适应控制算法,证明了闭环系统的所有信号是有界的,跟踪误差收敛到以零为原点的球中.

关键词: 非线性系统; 多层神经元网; 带死区的最小二乘算法; 自适应控制

1 Introduction

The idea of applying multilayer neural networks to adaptive control of nonlinear continuous-time systems has become a popular architecture. An important reason is that the unknown nonlinear functions in the system can be expressed as

$$f(\cdot) = \sum_{i=1}^{n} \Theta f_i(\cdot) + \varepsilon(\cdot),$$

where f_i 's are known functions, $\varepsilon(\cdot)$ is the modeling error. However, considerable research has been conducted in the continuous-time systems, little about the use of neural networks to discrete-time systems. As we know, some results about discrete-time systems first appear in [2]. However, there is no stability proof given in [2]. In [3], authors presented a convergence result for adaptive regulation using multilayer neural networks provided that the modeling error is zero. This is a very restrictive assumption. Because this may require the use of a very large, or infinite, number of nodes for neural networks, limiting the applicability of the technique. When the modeling error is not zero, the updating rule must be modified. In [4] and [5], authors use gradient algorithm with dead-zone and δ- modification algorithm respectively. However, the least squares (LS) algorithm generally has much superior rate of convergence compared with gradient algorithm and δ- modification algorithm. Therefore, the research into the nonlinear adaptive control using neural networks and least squares algorithm with dead-zone is very important.

In this paper, LS with dead-zone algorithm is used to estimate the weights of neural networks. We prove that: 1) all signals in the closed-loop systems are bounded; and 2) the tracking error between the system output and the reference output will converge to a bounded ball. We obtain similar results using fewer restrictive assumption compared with[4].

2 Problem formulation

In this section, we consider SISO nonlinear discrete-time system discussed in [4].

$$y(k+1) = f_0(\cdot) + g_0(\cdot)u(k-d+1), \tag{2.1}$$

where f_0 and g_0 are unknown smooth functions of $y(k-n+1), \cdots, y(k), u(k-d-m+1), \cdots, u(k-d)$, y is the output, u is the input, d is the relative degree of

Manuscript received Mar. 2, 1998, revised Sep. 9, 1998.

the system and g_0 is bounded away from zero, m, n, and d are known, $m \leqslant n$.

To define the zero dynamics, (2.1) is converted into a state-space form. We select the state variables

$x_i(k) = y(k-n+i),\ i = 1,2,\cdots,n,$

$x_{n+i}(k) = u(k-m-d+i),\ i = 1,2,\cdots m+d+1.$

Therefore a state space model of (2.1) is constructed as

$$\begin{aligned}
&x_i(k+1) = x_{i+1}(k),\ i = 1,2,\cdots,n-1,\\
&x_n(k+1) = f_0(x(k)) + g_0(x(k))x_{n+m+1}(k),\\
&x_{n+i}(k+1) = x_{n+i+1}(k),\ i = 1,2,\cdots,m+d-2,\\
&x_{n+m+d-1}(k+1) = u(k),\\
&y(k) = x_n(k),
\end{aligned} \tag{2.2}$$

where $x(k) = (x_1(k),\cdots,x_n(k),\cdots,x_{m+n+d-1}(k))'$. In the case of $d > 1$, the future system outputs are needed to be expressed in terms of elements of $x(k)$. Notice that $x_n(k+1) = y(k+1)$. Then

$$x_n(k+2) = f_0(x(k+1)) + g_0(x(k+1))x_{n+m+1}(k+1). \tag{2.3}$$

Replacing $x(k+1)$ in (2.3) by the right-hand side of (2.2), we have

$$x_n(k+2) = f_1(x(k)) + g_1(x(k))x_{n+m+2}(k).$$

By applying the same technique recursively, one gets

$$\begin{aligned}
&x_n(k+3) = f_2(x(k)) + g_2(x(k))x_{n+m+3}(k),\\
&\qquad\vdots\\
&x_n(k+d-1) = f_{d-2}(x(k)) + g_{d-2}(x(k))x_{n+m+d-1}(k).
\end{aligned}$$

Consider the state transformation

$$z(k) = \begin{bmatrix} z_{11}(k) \\ \vdots \\ z_{1,n+d-1}(k) \\ z_{21}(k) \\ \vdots \\ z_{2m}(k) \end{bmatrix} = \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k+d-1) \\ x_{n+1}(k) \\ \vdots \\ x_{n+m}(k) \end{bmatrix} = T(x(k)). \tag{2.4}$$

It can be shown that inverse of (2.4), i.e., $x = T^{-1}(z)$, exists provided $g_0(x),\cdots,g_{d-2}(x)$ are bounded away from zero over the domain of interest. After application of the transformation (2.3), (2.2) becomes

$$\begin{aligned}
&z_{1i}(k+1) = z_{1,i+1}(k),\ i = 1,2,\cdots,n+d-2,\\
&z_{1,n+d-1}(k+1) =\\
&f_{d-1}(x(k)) + g_{d-1}(x(k))x_{n+m+d-1}(k+1) =\\
&f_{d-1}(T^{-1}(z(k))) + g_{d-1}(T^{-1}(z(k)))u(k) =\\
&F(z(k)) + G(z(k))u(k),\\
&z_{2i}(k+1) = z_{2,i+1}(k),\ i = 1,2,\cdots,m-1,\\
&z_{2m}(k+1) = \frac{z_{1,n+1}(k) - f_0(x(k))}{g_0(x(k))} =\\
&\frac{z_{1,n+1}(k) - f_0(T^{-1}(z(k)))}{g_0(T^{-1}(z(k)))},\\
&y(k) = z_{1n}(k).
\end{aligned} \tag{2.5}$$

Since the nonlinear functions $F(\cdot)$ and $G(\cdot)$ in (2.5) are unknown, we use multilayer neural networks to model the nonlinear functions. Now, we are in a position to formulate the problems under study as follows.

Control Objectives Determine a feedback control $u = u(x \mid \Theta)$ and the updating rule for adjusting the parameter Θ (where Θ is the weights of a neural network) such that:

i) all signals in the closed-loop system are all bounded;

ii) the tracking error between the system output and the reference output will converge to a bounded ball.

3 Adaptive control using multilayer neural networks and LS algorithm with dead-zone

In this section, we first state our assumption on the system, and then develop an adaptive control law based on the neural networks.

Assumption 1 (The Minimum Phase Assumption)

The change of variables $e_{2i}(k) = z_{2i}(k) - c$ transforms (2.5) into

$$\begin{aligned}
&e_{2i}(k+1) = e_{2,i+1}(k),\ i = 1,2,\cdots,m-1,\\
&e_{2m}(k+1) = \frac{-f_0(T^{-1}(0,e_2(k)+C))}{g_0(T^{-1}(0,e_2(k)+C))} - c.
\end{aligned} \tag{3.1}$$

We assume that the origin of (3.1) is exponentially stable, and there is a Lyapunov function $V(e_2)$ which satisfies

$$\begin{aligned}
&m_1(e_2(k))^2 \leqslant V(e_2(k)) \leqslant m_2(e_2(k))^2,\\
&V(e_2(k+1)) - V(e_2(k)) \leqslant -a \mid e_2(k) \mid^2,\\
&\left|\frac{\partial V(e_2(k))}{\partial e_2(k)}\right| \leqslant L \mid e_2(k) \mid
\end{aligned} \tag{3.2}$$

in some ball $B \subset \mathbb{R}^m$ (see [4]).

In this paper, we consider three-layer neural networks with p hidden neurons.

$$\hat{h}(x(k), w) =$$

$$\sum_{i=1}^{p} w_i H\Big(\sum_{j=1}^{m+n+d-1} w_{ij}x_j + \hat{w}_i\Big) = w'\phi(x), \quad (3.3)$$

where w_i, w_{ij} and $\hat{w}_i$ are unknown parameters to be estimated, w depends on w_i, w_{ij} and $\hat{w}_i$. In addition, in our updating rule and convergence analysis, we need the following assumption: H is differentiable and there exists constants $M > 0$ and $N > 0$ such that

$$\begin{aligned} &| H(x,w) | \leq M, \\ &\left|\frac{\partial H(x,w)}{\partial w}\right| \leq N, \quad \forall x \in (-\infty, +\infty). \end{aligned} \quad (3.4)$$

For example, $H(x,w)$ is often taken as Gaussian function, hyperbolic tangent function etc.

Rewrite the system in an input-output form as

$$y(k+d) = f_{d-1}(x(k)) + g_{d-1}(x(k))u(k). \quad (3.5)$$

If f and g are known, then the feedback control is defined as

$$u(k) = \frac{-f_{d-1}(x(k)) + r(k)}{g_{d-1}(x(k))}, \quad (3.6)$$

where $r(k)$ is the desired output. Since f and g are unknown, we replace them by the neural networks defined by (3.3) respectively. System (3.5) is modeled by the neural networks

$$\hat{y}(k+d) = \hat{f}_{d-1}(x(k),w) + \hat{g}_{d-1}(x(k),v)u(k), \quad (3.7)$$

where $\hat{f}(\cdot,\cdot)$ and $\hat{g}(\cdot,\cdot)$ are three-layer neural networks with p and q hidden neurons respectively. The resulting control law is

$$u(k) = \frac{-\hat{f}_{d-1}(x(k),w(k)) + r(k)}{\hat{g}_{d-1}(x(k),v(k))}, \quad (3.8)$$

where $w(k)$ and $v(k)$ denote the estimates of w and v at time k, $f_{d-1}(x) = F(z)$ and $g_{d-1}(x) = G(z)$. Because the set $\{\hat{g}_{d-1}(x(k),v(k)) = 0\}$ may have a positive probability, in order to avoid zero divisor, we use the following control law

$$u(k) = \frac{-\hat{f}_{d-1}(x(k),w(k)) + r(k)}{\hat{g}_{d-1}(x(k),v(k)) + \delta\,\mathrm{sgn}(\hat{g}_{d-1}(x(k),v(k)))}, \quad (3.9)$$

where δ is an arbitrarily small positive constant, and

$$\mathrm{sgn}(x) = \begin{cases} 1, & x \geq 0, \\ -1, & x < 0. \end{cases}$$

To better define the error, rewrite (3.5) and (3.6) as

$$y(k+1) = f_{d-1}(x(k-d+1)) + g_{d-1}(x(k-d+1))u(k-d+1) \quad (3.10)$$

and

$$\hat{y}(k+1) = \hat{f}_{d-1}(x(k-d+1),w) + \hat{g}_{d-1}(x(k-d+1),v)u(k-d+1). \quad (3.11)$$

The estimated system output is

$$y(k+1)^* = \hat{f}_{d-1}(x(k-d+1),w(k)) + (\hat{g}_{d-1}(x(k-d+1),v(k)) + \delta\,\mathrm{sgn}(\hat{g}_{d-1}(x(k-d+1),v(k))))u(k-d+1). \quad (3.12)$$

The error $e(k+1)^*$ is defined as

$$e(k+1)^* = y(k+1)^* - y(k+1). \quad (3.13)$$

The estimate $\Theta(k)$ for $\Theta = [w^{\mathrm{T}}, v^{\mathrm{T}}]^{\mathrm{T}}$ is updated by the recursive LS algorithm with dead-zone.

$$\Theta(k+1) = \Theta(k) + a(k)b(k)P(k)\varphi(k-d+1)\cdot (y(k+1) - y(k+1)^*), \quad (3.14)$$

$$\begin{cases} P(k+1) = P(k) - a(k)b(k)\cdot P(k)\varphi(k)\varphi(k)^{\mathrm{T}}P(k), \\ P(0) = a_0 I, \end{cases} \quad (3.15)$$

$$a(k) = \frac{1}{1 + b(k)\varphi'(k-d+1)P(k)\varphi(k-d+1)}, \quad (3.16)$$

$$b(k) = \begin{cases} \beta, & \text{if } a(k)\, | e(k+1)^* | \geq \sqrt{\alpha}d_0, \\ 0, & \text{if } a(k)\, | e(k+1)^* | < \sqrt{\alpha}d_0, \end{cases} \quad (3.17)$$

$$g(k-d+1) = \left[\frac{\partial y(k+1)^*}{\partial \Theta}\Big|_{\Theta(k)}\right]' = \begin{bmatrix} \left(\dfrac{\partial \hat{f}_{d-1}(x(k-d+1),w(k))}{\partial w(k)}\right)' \\ \left(\dfrac{\partial(\hat{g}_{d-1}(x(k-d+1),v(k)) + \delta\,\mathrm{sgn}(\hat{g}_{d-1}(x(k-d+1),v(k))))}{\partial v(k)}\right)' u(k-d+1) \end{bmatrix}, \quad (3.18)$$

where a_0 is a positive constant, the positive constants d_0, $\alpha > 1$, and β are properly chosen.

Remark 3.1 Assumption 1 in [4] is not needed by properly modifying the control law. δ can be chosen an any small constant.

Remark 3.2 Assumption 3 in [4] is justified by the approximation results of [6]. Therefore given a positive constant ε and a compact set $S \subset \mathbb{R}^{m+n+d-1}$, there exist coefficients w, v such that

$$\begin{aligned} &\max_{x\in S} | \hat{f}_{d-1}(x,w) - f_{d-1}(x) | \leqslant \varepsilon, \\ &\max_{x\in S} | \hat{g}_{d-1}(x,v) - g_{d-1}(x) | \leqslant \varepsilon. \end{aligned} \tag{3.19}$$

4 Performance analysis

In this section, we will give main results.

Theorem 4.1 Let $| r(k) | \leqslant r^*$ for all $k \geqslant 0$. Given any constant $\rho > 0$ and any small constant d_0, there exist positive constants $\rho_1 = \rho_1(\rho, r^*), \rho_2 = \rho_2(\rho, r^*), \varepsilon^* = \varepsilon^*(\rho, d_0, r^*)$ and $\lambda^* = \lambda^*(\rho, d_0, r^*)$ such that if (3.19) holds on $S \supset B_{\rho_1}$ with $\varepsilon < \varepsilon^*$, Assumption 1 is satisfied on B_{ρ_2}, $| x(0) | \leqslant \rho$, and $| \Theta(0) | \leqslant \lambda < \lambda^*$. Then, the adaptive controller consisting of (3.9), (3.10) and (3.12) ~ (3.18) guarantees the following properties:

i) $| \Theta(k) |$ are bounded, and $| \Theta(k+1) - \Theta(k) |$ will converge to zero, where $| \cdot |$ denotes Euclidean norm;

ii) $\{y(k)\}$ and $\{u(k)\}$ are bounded, for all k;

iii) The tracking error between the system output and the reference output will converge to a ball of radius $\sqrt{\alpha} d_0$ centered at origin.

Proof We prove Theorem 4.1 by 5 steps.

Step 1 The dynamics associated with z_1 are

$$\begin{aligned} z_{1i}(k+1) &= z_{1,i+1}(k), \ i = 1,2,\cdots,n+d-2, \\ z_{1,n+d+1}(k) &= F(z(k)) + G(z(k))u(k). \end{aligned} \tag{4.1}$$

The last equation can be rewritten as

$$\begin{aligned} &z_{1,n+d+1}(k) = \\ &F(z(k)) + G(z(k))u(k) = \\ &\hat{F}(z(k),w) + (\hat{G}(z(k),v) + \\ &\delta \mathrm{sgn}(\hat{G}(z(k),v)))u(k) + w_1(k) = \\ &\hat{F}(z(k),w) - \hat{F}(z(k),w(k)) + (\hat{G}(z(k),v) + \\ &\delta \mathrm{sgn}(\hat{G}(z(k),v)))u(k) - \hat{G}(z(k),v(k)) + \\ &\delta \mathrm{sgn}(\hat{G}(z(k),v(k))))u(k) + w_1(k) + r(k) = \\ &r(k) + w_1(k) + w_2(k), \end{aligned} \tag{4.2}$$

where $\hat{F}(z(k),w) = \hat{f}_{d-1}(T^{-1}(z(k)),w)$ and

$$\hat{G}(z(k),v) = \hat{g}_{d-1}(T^{-1}(z(k)),v),$$

$$\begin{aligned} w_1(k) = &(F(z(k)) - \hat{F}(z(k),w)) + \\ &(G(z(k)) - (\hat{G}(z(k),v) + \\ &\delta \mathrm{sng}(\hat{G}(z(k),v))))u(k), \end{aligned} \tag{4.3}$$

$$\begin{aligned} w_2(k) = &(\hat{F}(z(k),w) - \hat{F}(z(k),w(k))) + \\ &((\hat{G}(z(k),v) + \delta \mathrm{sng}(\hat{G}(z(k),v))) - \\ &(\hat{G}(z(k),v(k)) + \delta \mathrm{sng}(\hat{G}(z(k),v(k)))))u(k). \end{aligned} \tag{4.4}$$

Define

$$e_{1i}(k) = z_{1i}(k) - r(k-n-d+i). \tag{4.5}$$

Then (4.2) can be reexpressed as

$$\begin{aligned} e_{1i}(k+1) &= e_{1,i+1}(k), \ i = 1,2,\cdots,n+d-2, \\ e_{1,n+d-1}(k) &= w_1(k) + w_2(k). \end{aligned} \tag{4.6}$$

Define

$$e_{2i}(k) = z_{2i}(k) - c. \tag{4.7}$$

Then the dynamics associated with z_2 are transformed into

$$\begin{aligned} e_{2i}(k+1) &= e_{2,i+1}(k), \ i = 1,2,\cdots,m-1, \\ e_{2,m}(k+1) &= u_{k-d+1}(k) - c. \end{aligned} \tag{4.8}$$

Thus, (4.6) and (4.8) is the new state space representation of the closed-loop system. Denote

$$\begin{aligned} e_1(k) &= (e_{11}(k),\cdots,e_{1,n+d-1}(k))', \\ e_2(k) &= (e_{21}(k),\cdots,e_{2,m}(k))', \\ R(k) &= (r(k-1),\cdots,r(k-n-d+1))'. \end{aligned}$$

This step is mainly used in the proof of Step 4.

Step 2 Consider the set

$$\Omega = \left\{ \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} : | e_1 | \leqslant \mu_1, \ | e_2 | \leqslant \mu_2 \right\}, \tag{4.9}$$

where μ_1 and μ_2 will be chosen in step 3. Since $z(k) = e(k) + [R(k) \ C]'$ and $x(k) = T^{-1}(z(k))$, we can conclude that as long as $e(k) \in \Omega$, then $x(k) \in B_{\rho_1}$, where B_{ρ_1} is a ball depending on μ_1, μ_2, d_1 and $| C |$.

Consider the set

$$\Omega_1 = \{\tilde{\theta} : | \tilde{\theta} | \leqslant \lambda\}. \tag{4.10}$$

In the following, we will show that as long as $e(k) \in \Omega$, Ω_1 will be an invariant set, provided ε and λ are sufficiently small. We consider (3.10)

$$\begin{aligned} &y(k+1) = \\ &f_{d-1}(x(k-d+1)) + \\ &g_{d-1}(x(k-d+1))u(k-d+1) = \\ &\hat{f}_{d-1}(x(k-d+1),w) + (\hat{g}_{d-1}(x(k-d+1),v) + \\ &\delta \mathrm{sgn}(\hat{g}_{d-1}(x(k-d+1),v)))u(k-d+1) + O(\varepsilon+\delta). \end{aligned} \tag{4.11}$$

By $e(k) \in \Omega$ and Remark 3.2, the last term on the right-hand side of (4.11) is $O(\varepsilon + \delta)$.

The estimate of the system output is

$$y(k+1)^* = \hat{f}_{d-1}(x(k-d+1), w(k)) + (\hat{g}_{d-1}(x(k-d+1), v(k)) + \delta \operatorname{sgn}(\hat{g}_{d-1}(x(k-d+1), v(k))))u(k-d+1). \tag{4.12}$$

By Taylor formula, it follows that

$$\begin{aligned}
&e(k+1) = y(k+1)^* - y(k+1) = \\
&\hat{f}_{d-1}(x(k-d+1), w(k)) - \hat{f}_{d-1}(x(k-d+1), w)) + \\
&(\hat{g}_{d-1}(x(k-d+1), v(k)) + \delta \operatorname{sgn}(\hat{g}_{d-1}(x(k-d+1), v(k))))u(k-d+1) - \\
&(\hat{g}_{d-1}(x(k-d+1), v) + \delta \operatorname{sgn}(\hat{g}_{d-1}(x(k-d+1), v)))u(k-d+1) + O(\varepsilon + \delta) = \\
&\tilde{\Theta}(k)' \begin{bmatrix} \left(\dfrac{\partial \hat{f}_{d-1}(x(k-d+1), w(k))}{\partial w(k)}\right)' \\ \left(\dfrac{\partial(\hat{g}_{d-1}(x(k-d+1), v(k)) + \delta \operatorname{sgn}(\hat{g}_{d-1}(x(k-d+1), v(k))))}{\partial v(k)}\right)' u(k-d+1) \end{bmatrix} + d(k) = \\
&\tilde{\Theta}'(k)\varphi(k-d+1) + d(k),
\end{aligned} \tag{4.13}$$

where $d(k) = O(|\tilde{\Theta}(k)|^2) + O(\varepsilon + \delta)$, $\tilde{\Theta}(k) = \Theta(k) - \Theta$. Since $x(k)$ is bounded, from Remark 3.2, there exist c_1 and c_2 such that

$$|d(k)| \leqslant c_1 |\tilde{\Theta}(k)|^2 + c_2(\varepsilon + \delta),$$

therefore, we can choose λ, δ and ε which are small enough such that

$$|d(k)| \leqslant c_1 |\tilde{\Theta}(k)|^2 + c_2(\varepsilon + \delta) \leqslant d_0. \tag{4.14}$$

From (3.14), (3.16) and (4.13), it is easy to see that

$$\begin{aligned}
&\tilde{\Theta}'(k+1)\varphi(k-d+1) + d(k) = \\
&a(k)(\tilde{\Theta}'(k)\varphi(k-d+1) + d(k)) = \\
&a(k)(y^*(k+1) - y(k+1)).
\end{aligned} \tag{4.15}$$

Using (3.14), (3.15) and (4.15), we see

$$\begin{aligned}
\tilde{\Theta}(k+1) = \tilde{\Theta}(k) - b(k)P(k)\varphi(k-d+1) \cdot \\
(\tilde{\Theta}'(k+1)\varphi(k-d+1) + d(k))
\end{aligned} \tag{4.16}$$

and

$$\begin{aligned}
&P(k+1)^{-1} = \\
&P(k)^{-1} + b(k)\varphi(k-d+1)\varphi(k-d+1)'.
\end{aligned} \tag{4.17}$$

It follows from (4.15) ~ (4.17) that

$$\begin{aligned}
&V(k+1) = \\
&\tilde{\Theta}'(k+1)P(k+1)^{-1}\tilde{\Theta}(k+1) = \\
&V(k) + b(k)(\tilde{\Theta}'(k+1)\varphi(k-d+1))^2 - \\
&2b(k)\tilde{\Theta}'(k)\varphi(k-d+1)(\tilde{\Theta}'(k+1)\varphi(k- \\
&d+1) + d(k)) + b(k)^2\varphi'(k-d+1)P(k)\varphi(k- \\
&d+1)(\tilde{\Theta}'(k+1)\varphi(k-d+1) + d(k))^2 = \\
&V(k) - b(k)(\tilde{\Theta}'(k+1)\varphi(k-d+1))^2 - \\
&2b(k)\tilde{\Theta}'(k+1)\varphi(k-d+1)d(k) - \\
&b(k)^2\varphi'(k-d+1)P(k)\varphi(k-d+1) \cdot \\
&(\tilde{\Theta}'(k+1)\varphi(k-d+1) + d(k))^2 \leqslant \\
&V(k) - b(k)(\tilde{\Theta}'(k+1)\varphi(k-d+1))^2 - \\
&2b(k)\tilde{\Theta}'(k+1)\varphi(k-d+1)d(k) = \\
&V(k) - b(k)(a(k)^2(y(k+1) - \\
&y(k+1)^*)^2 - d(k)^2).
\end{aligned} \tag{4.18}$$

By (3.17) and (4.14), we have

$$V(k+1) \leqslant V(k), \quad \forall k. \tag{4.19}$$

Hence, it follows from (4.17) and (4.19) that

$$\begin{aligned}
&a_0^{-1} |\tilde{\Theta}(k)|^2 = \\
&\tilde{\Theta}'(k)P(0)^{-1}\tilde{\Theta}(k) \leqslant \tilde{\Theta}'(k)P(k)^{-1}\tilde{\Theta}(k) \leqslant \\
&\tilde{\Theta}'(0)P(0)^{-1}\tilde{\Theta}(0) = a_0^{-1} |\tilde{\Theta}(0)|^2,
\end{aligned} \tag{4.20}$$

which leads to

$$|\tilde{\Theta}(k)| \leqslant |\tilde{\Theta}(0)|. \tag{4.21}$$

Consequently, we show that Ω_1 is a positively invariant set.

Step 3 It is similar to the proof of step 3 ~ 5 in [4], we can obtain that $e(k)$ remains in Ω for all $k \geqslant 0$.

Step 4 From (3.4), (3.9) and (4.21), we obtain

$$\begin{aligned}
&|u(k)| \leqslant \frac{1}{\delta}(r^* + \sqrt{p}M |\hat{\Theta}(k)|) \leqslant \\
&\frac{1}{\delta}(r^* + \sqrt{p}M |(\tilde{\Theta}(0)| + |\Theta|)) = \sigma.
\end{aligned} \tag{4.22}$$

From (4.12), (4.13) and (4.14), we have

$$\begin{aligned}
&|y(k+1)| = \\
&|y(k+1)^* + e(k+1)| = \\
&|\tilde{\Theta}'(k)\varphi(k-d+1) + d(k) + \\
&\hat{f}_{d-1}(x(k-d+1), w(k)) + \\
&(\hat{g}_{d-1}(x(k-d+1), v(k)) +
\end{aligned}$$

$\delta \mathrm{sgn}(\hat{g}_{d-1}(x(k-d+1), v(k))))u(k-d+1)| \leqslant \sqrt{p+q}N|\tilde{\Theta}(0)| + \sqrt{\alpha}d_0 + \sqrt{p}M(|\tilde{\Theta}(0)| + |\Theta|) + \sigma(\sqrt{q}M(|\tilde{\Theta}(0)| + |\Theta|) + \delta)$. (4.23)

Obviously, the conclusion ii) is true.

Step 5 Now we prove conclusion i) and iii). From (3.17), (4.14) and (4.18), it is easy to see that

$$(1-\frac{1}{\alpha})a(k)^2b(k)(y(k+1)-y(k+1)^*)^2 \leqslant V(k) - V(k+1) - b(k)(\frac{1}{\alpha}a(k)^2(y(k+1) - y(k+1)^*)^2 - d(k)^2) \leqslant V(k) - V(k+1). \quad (4.24)$$

Summing up both sides of this expression from 0 to ∞ leads to

$$(1-\frac{1}{\alpha})\sum_{k=0}^{\infty} a(k)^2b(k)(y(k+1)-y(k+1)^*)^2 \leqslant V(0) = \tilde{\Theta}(0)'P(0)^{-1}\tilde{\Theta}(0) < \infty, \quad (4.25)$$

which implies

$$a(k)^2b(k)(y(k+1)-y(k+1)^*)^2 \to 0. \quad (4.26)$$

By (3.4) and (3.15), we have

$$\varphi'(k+d-1)P(k)\varphi(k+d-1) \leqslant \varphi'(k+d-1)P(0)\varphi(k+d-1) \leqslant a_0|\varphi(k+d-1)|^2 \leqslant M_1,$$

where M_1 is a positive constant. From (3.14) and (4.26), we obtain

$$|\Theta(k+1) - \Theta(k)|^2 \leqslant a_0\varphi'(k+d-1)P(k)\varphi(k+d-1)a(k)^2b(k)^2 \cdot (y(k+1) - y(k+1)^*)^2 \to 0, \ k \to \infty.$$

Thus we get

$$\Theta(k+1) - \Theta(k) \to 0, \ k \to 0. \quad (4.27)$$

It follows from (4.26) that there exists an integer $k_0 > 0$ such that $\forall k \geqslant k_0$, we have

$$a(k)|y(k+1) - y(k+1)^*| < \sqrt{\alpha}d_0. \quad (4.28)$$

Suppose that the assertion (4.28) is not true, then there is a subsequence $\{k_i : i = 1,2,\cdots\}$ such that

$$a(k_i)|y(k_i+1) - y(k_i+1)^*| \geqslant \sqrt{\alpha}d_0.$$

So, by (3.17) we obtain

$$a(k_i)\sqrt{b(k_i)}|y(k_i+1) - y(k_i+1)^*| \geqslant \sqrt{\alpha\beta}d_0.$$

Set $i \to \infty$, then $\{a(k_i)^2b(K_i)(y(k_i+1) - y(k_i+1)^*)^2\}$ does not converge to zero. It contradicts (4.27). Therefore, (4.28) is true. By (3.17), $a(k) = 1$ for all $k \geqslant k_0$. Hence, form (4.28)

$$|y(k+1) - y(k+1)^*| < \sqrt{\alpha}d_0, \ \forall k \geqslant k_0. \quad (4.29)$$

Recall that

$$y(k+d)^* = \hat{f}_{d-1}(x(k), w(k+d-1)) + (\hat{g}_{d-1}(x(k), v(k+d-1)) + \delta \mathrm{sgn}(\hat{g}_{d-1}(x(k), v(k+d-1))))u(k), \quad (4.30)$$

while the control $u(k)$ is generated from

$$r(k) = \hat{f}_{d-1}(x(k), w(k)) + (\hat{g}_{d-1}(x(k), v(k)) + \delta \mathrm{sgn}(\hat{g}_{d-1}(x(k), v(k))))u(k). \quad (4.31)$$

Since $\hat{f}(\cdot,\cdot)$ and $\hat{g}(\cdot,\cdot)$ satisfy Lipschitz condition in the compact set S, and notice that δ can be chosen arbitrarily small, therefore

$$|y(k+d)^* - r(k)| \leqslant |\hat{f}_{d-1}(x(k), w(k+d-1)) - \hat{f}_{d-1}(x(k), w(k))| + |(\hat{g}_{d-1}(x(k), v(k+d-1)) + \delta \mathrm{sgn}(\hat{g}_{d-1}(x(k), v(k+d-1))))u(k) - (\hat{g}_{d-1}(x(k), v(k)) + \delta \mathrm{sgn}(\hat{g}_{d-1}(x(k), v(k))))u(k)| \leqslant K|\Theta(k+d-1) - \Theta(k)| + o(1) \to 0, \ k \to \infty. \quad (4.32)$$

It follows from (4.29) and (4.32) that

$$|y(k+d) - r(k)| \leqslant |y(k+d) - y(k+d)^*| + |y(k+d)^* - r(k)| < \sqrt{\alpha}d_0, \ k \to \infty. \quad (4.33)$$

Hence the proof is complete.

Remark 4.1 Assumption 1 is used in the proof of Step 3.

5 Conclusion

In this paper, we give theoretical analysis of the use of multilayer neural networks and least squares algorithm with dead-zone in the control of nonlinear discrete-time systems with relative degree possible higher than one. The convergence result of the paper is local with respect to the initial parameters but not local with respect to the initial states of the system. We adopt LS algorithm with dead-zone to update the weights of the neural networks. The algorithm has much superior rate of convergence compared with gradient algorithm and ε –

(Continued on page 379)

Contr., 1989, 34(11): 1203 – 1204

2 Xu B. and Liu Y. An improved Razumikhin-type theorem and its applications. IEEE Trans. Automat. Contr., 1994, 39(4): 839 – 841

3 Xu B. Comments "Robust stability of delay dependence for linear uncertain systems." IEEE Trans. Automat. Contr., 1994, 39(11): 2365

4 Su J H. Further results on the robust stability of linear systems with a single time delay. Systmes & Control Letters, 1994, 23(5): 375 – 379

5 Wu H and Mizukami K. Robust stability criteria for dynamical systems including delayed perturbations. IEEE Trans. Automat. Contr., 1995, 40(3): 487 – 490

6 Xu B. On delay-independent stability of large-scale systems with time delays. IEEE Trans. Automat. Contr., 1995, 40(5): 930 – 933

7 Chen J, Xu D and Shafai B. On sufficient conditions for stability independent of delay. IEEE Trans. Automat. Contr., 1995, 40(9): 1675 – 1680

8 Trinh H. and Aldeen M. Stability robustness bounds for linear systems with delayed perturbations. IEE Proc. Control Theory and Applications, 1995, 142(4): 345 – 350

9 Xu B, Fu Y and Li B. Further results on robust bounds for large-scale time-delay systems with structured and unstructured uncertainties. Int. J. Systems Sci., 1996, 27(12): 1491 – 1495

10 Xu B. Stability robustness bounds for linear systems with multiple time-varying delayed perturbations. Int. J. Systems Sci., 1997, 28 (12): 1311 – 1317

11 Kim J H. Robust stability of linear systems with delayed perturbations. IEEE Trans. Automat. Contr., 1996, 41(12): 1820 – 1822

12 Hale J K and Lunel S M V. Introduction to Functional Differential Equations. New York: Springer-Verlag, 1993

本文作者简介

胥布工 见本刊 1999 年第 2 期第 305 页.

(Continued on page 360)

δ- modification algorithm. The similar results are obtained, using fewer restrictive assumptions compared with [4].

References

1 Sanner R M and Slotine J J E. Gaussian networks for direct adaptive control. IEEE Trans. Neural Networks, 1992, 3(6): 837 – 863

2 Narendra K S and Parthasarathy K. Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Networks, 1990, 1 (1): 4 – 37

3 Chen F C and Khalil H K. Adaptive control of nonlinear systems using neural networks. Int. J. Control, 1992, 55: 1299 – 1317

4 Chen F C and Khalil H K. Adaptive control of a class of nonlinear discrete-time systems using neural networks. IEEE Trans. Automat. Contr., 1995, 40(5): 791 – 801

5 Jagannathan S and Lewis F L. Multilayer discrete-time neural-net controller with guaranteed performance. IEEE Trans. Neural Networks, 1996, 7(1): 107 – 131

6 Hornik K, Stichcombe M and White H. Multilayer feedforward networks are universal approximators. Neural Networks, 1989, 2(2): 359 – 366

7 Ioannou P A and Sun. J. Stable and Robust Adaptive Control. New Jersey: Prentice-Hall, 1995

本文作者简介

解学军 1968 年生. 分别于 1991 年, 1994 年在曲阜师范大学获学士和硕士学位. 现为中科院系统科学研究所博士生. 主要研究方向是自适应控制与信号处理.

王 远 1972 年生. 1993 年毕业于哈尔滨师范大学数学系, 1996 年在黑龙江大学应用数学研究所获硕士学位. 现为中科院系统科学研究所博士生. 主要研究方向是自适应控制理论.