Adaptively Controlling Nonlinear Continuous-Time Systems Using Multilayer Neural Networks

Fu-Chuang Chen and Chen-Chung Liu

***Abstract*—Multilayer neural networks are used in a nonlinear adaptive control problem. The plant is an unknown feedback-linearizable continuous-time system. The control law is defined in terms of the neural network models of system nonlinearities to control the plant to track a reference command. The network parameters are updated on-line according to a gradient learning rule with dead zone. A local convergence result is provided, which says that if the initial parameter errors are small enough, then the tracking error will converge to a bounded area. Simulations are designed to demonstrate various aspects of theoretical results.**

I. Introduction

Adaptive control of linear systems has been an active research area in the past two decades. It is only recently that issues related to adaptive control of feedback-linearizable nonlinear systems are addressed, e.g., [1], [2]. An important assumption in previous works on nonlinear adaptive control is the linear dependence on the unknown parameters, i.e., the unknown nonlinear functions in the plant have the form

$$f(\cdot) = \sum_{i=1}^{n} \theta_i f_i(\cdot) \tag{1}$$

where f_i's can be some known special nonlinear functions, e.g., [1], [2], or certain basis functions, e.g., Gaussian basis functions [9] and polynomial basis functions.

Multilayer neural networks [3] are general tools for modeling nonlinear functions since they can approximate any nonlinear function to any desired accuracy [6]. An apparent feature that multilayer networks are different from the linearly parameterizing modeling methods (1) is that their parameters appear nonlinearly. In contrast to the local nature of the Gaussian networks (i.e., each network parameter can only locally affect the network output) [9], the global nature of multilayer networks (i.e., the fact that all network weights play significant roles in determining the network output due to an input) may significantly reduce the number of parameters required. Compared with interpolating polynomials, multilayer networks are usually much well-behaved since they employ sigmoid-type nonlinearities. A unique advantage associated with multilayer networks is that when the number of neuron layers is fixed (in practice, less than four layers are used), the computation time of the network is independent of the network complexity (i.e., the number of neurons used in each layer, which is related to the complexity of the unknown function to be approximated), provided appropriate computing hardware is available. On the other hand, multilayer networks have some disadvantages: First, they require significantly longer training time (compared with Gaussian networks). Second, their nonlinear parameters make related mathematical analyses very difficult.

The idea of applying multilayer neural networks to adaptive control of feedback linearizable systems appeared in [4], [5]. Some in-depth developments on discrete-time problems are available in [7], [8].

Manuscript received January 12, 1993; revised June 28, 1993. This work was supported in part by the National Science Council of the Republic of China under Grant NSC 81-0404-E-009-005.

The authors are with the Department of Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, ROC.

IEEE Log Number 9401653.

In this note, we study the continuous-time problems. Multilayer networks are used to model unknown nonlinear functions in the plant to generate cancellation controls. Networks weights are updated on-line according to a gradient type learning law which makes use of the popular back-propagation algorithm. The plant considered is a relative-degree-one single-input/single-output (SISO) system with stable zero dynamics. The stability and tracking result provided in this note is regional in system states, but local in network parameters. The local result may not be a conservative one, since a simulation in Section IV shows that the closed-loop control system indeed can go unstable if the initial parameter error is too large.

II. Linearizing Feedback Control

Consider the single-input/single-output system

$$\dot{\boldsymbol{x}} = f_0(\boldsymbol{x}) + g_0(\boldsymbol{x})u$$

$$y = h(\boldsymbol{x}) \tag{2}$$

with $\boldsymbol{x} \in R^n$; f_0, g_0, h smooth (i.e., infinitely differentiable). The states $\boldsymbol{x}$ are assumed available. Differentiating y with respect to time, one obtains

$$\dot{y} = \frac{\partial h}{\partial \boldsymbol{x}} f_0(\boldsymbol{x}) + \frac{\partial h}{\partial \boldsymbol{x}} g_0(\boldsymbol{x})u = f_1(\boldsymbol{x}) + g_1(\boldsymbol{x})u. \tag{3}$$

Assumption 1: The function $g_1(\boldsymbol{x})$ is bounded away from zero over the compact set $S_1 \in R^n$, that is

$$|g_1(\boldsymbol{x})| \geq b > 0, \qquad \forall \boldsymbol{x} \in S_1. \qquad \square \tag{4}$$

Then for the linearizing feedback control law

$$u = \frac{r - f_1(\boldsymbol{x})}{g_1(\boldsymbol{x})} \tag{5}$$

where

$$r = \dot{y}_m + \alpha(y_m - y), \qquad \alpha > 0 \tag{6}$$

$y_m(t)$ being the reference trajectory, the tracking error is described by the following error equation

$$\dot{e} + \alpha e = 0 \tag{7}$$

where $e = y - y_m$. It is clear that e will approach zero.

Then $(n-1)$ other system states are associated with the zero dynamics. With additional assumptions (which are assumed accordingly here) [1], there exists a diffeomorphism

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = T(\boldsymbol{x}) \tag{8}$$

such that T transforms the system (2) into its global normal form

$$\dot{z}_1 = f(z_1, z_2) + g(z_1, z_2)u$$

$$\dot{z}_2 = \psi(z_1, z_2)$$

$$y = z_1. \tag{9}$$

Notice that $f(z) = f(T(\boldsymbol{x})) = f_1(\boldsymbol{x})$ and $g(z) = g(T(\boldsymbol{x})) = g_1(\boldsymbol{x})$. If $\boldsymbol{x} = \mathbf{0}$ is the equilibrium point of the undriven system and $h(0) = 0$ (without loss of generality), the zero dynamics are defined to be

$$\dot{z}_2 = \psi(0, z_2). \tag{10}$$

Assumption 2: The system (9) is globally exponentially minimum phase. By that we mean the zero dynamics

$$\dot{z}_2 = \psi(0, z_2)$$

are globally exponentially stable. Further, ψ is assumed to be Lipschitz in z_1 and z_2. Then a converse Lyanpunov theorem [10] implies that $\exists V_2(z_2)$ such that

$$c_1|z_2|^2 \le V_2(z_2) \le c_2|z_2|^2,$$

$$\frac{dV_2}{dz_2} \cdot \psi(0, z_2) \le -\beta|z_2|^2, \quad \text{and}$$

$$\left|\frac{dV_2(z_2)}{dz_2}\right| \le L|z_2|. \tag{11}$$

in some ball $B_{\gamma 1} \subset R^{n-1}$. □

III. Adaptive Control Using Neural Networks

Suppose $f_1(\boldsymbol{x})$ and $g_1(\boldsymbol{x})$ are unknown, and they are modeled by the two multilayer networks $\hat{f}_1(\boldsymbol{x}, \boldsymbol{w})$ and $\hat{g}_1(\boldsymbol{x}, \boldsymbol{v})$ respectively, where $\boldsymbol{w}$ and $\boldsymbol{v}$ are vectors of network parameters. The functions $\hat{f}_1(\cdot, \cdot)$ and $\hat{g}_1(\cdot, \cdot)$ depend on the structure of the neural network and the number of neurons. For example, if $\hat{f}_1(\cdot, \cdot)$ is a three-layer neural network with p neurons in the hidden layer, it can be expressed as

$$\hat{f}_1(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{p} w_i H\left(\sum_{j=1}^{n} w_{ij}x_j + \hat{w}_i\right) \tag{12}$$

where w_i's are weights between the output and the hidden layer, w_{ij}'s are weights between the hidden and the input layer, and $\hat{w}_i$'s are the bias weights of hidden neurons. In this research, we use the hyperbolic tangent function $H(x) = (e^x - e^{-x})/(e^x + e^{-x})$.

Assumption 3: There exist coefficients $\boldsymbol{w}$ and $\boldsymbol{v}$ such that $\hat{f}_1$ and $\hat{g}_1$ approximate the continuous functions f_1 and g_1, with accuracy ϵ over a compact set $S_1 \in R^n$, that is,

$$\exists \boldsymbol{w}, \boldsymbol{v} \ \text{s.t.} \quad \max|\hat{f}_1(\boldsymbol{x}, \boldsymbol{w}) - f_1(\boldsymbol{x})| \le \epsilon, \quad \text{and}$$
$$\max|\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}) - g_1(\boldsymbol{x})| \le \epsilon, \quad \forall \boldsymbol{x} \in S_1. \quad \square \tag{13}$$

For convenience, denote $\Theta = [\boldsymbol{w} \cdot \boldsymbol{v}]^T$. Assumption 3 is justified by the approximation results of [6]. In our work we assume that the structure of the network and the number of neurons have been already specified, and (13) holds for the plant under consideration, but we do not assume that we know the weights $\boldsymbol{w}$ and $\boldsymbol{v}$ for which (13) is satisfied. Let $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ denote the estimates of $\boldsymbol{w}$ and $\boldsymbol{v}$ at time t. Then the control $u(t)$ is defined as the following.

Control Law

$$u(t) = \frac{-\hat{f}_1(\boldsymbol{x}(t), \boldsymbol{w}_t) + r(t)}{\hat{g}_1(\boldsymbol{x}(t), \boldsymbol{v}_t)} \tag{14}$$

Define the error $e = y - y_m$, and denote the parameter vector at time t as $\Theta_t = [\boldsymbol{w}_t \quad \boldsymbol{v}_t]^T$. The network weights are updated according to the following.

Updating Law

$$\dot{\Theta}_t = \mu D(e)\boldsymbol{J} \tag{15}$$

where μ is a positive number representing the learning rate, D is a dead-zone function defined as

$$D(e) = e, \qquad \text{if}|e| > d_0$$

$$D(e) = 0, \qquad \text{if}|e| \le d_0 \tag{16}$$

and

$$\boldsymbol{J} = \begin{bmatrix} \left(\left.\frac{\partial \hat{f}_1(\boldsymbol{x}, \boldsymbol{w})}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}_t}\right)^T \\ \left(\left.\frac{\partial \hat{g}_1(\boldsymbol{x}, \boldsymbol{v})}{\partial \boldsymbol{v}}\right|_{\boldsymbol{v}_t}\right)^T u \end{bmatrix}. \tag{17}$$

The Jacobian matrix $\boldsymbol{J}$ can be calculated using the routines of the back-propagation algorithm [3]. Define $\tilde{\Theta} = \Theta_t - \Theta$. Then the updating law (15) can be rewritten as

$$\dot{\tilde{\Theta}} = \mu D(e)\boldsymbol{J}. \tag{18}$$

Theorem: Suppose $|y_m(t)| \le d_1$ for all $t \ge 0$, and $y_m(t)$ has bounded derivatives. Given any constant $\rho > 0$, any small constant $d_0 > 0$, and any positive α, there exist positive constants $\gamma_1 = \gamma_1(\rho, d_1)$, $\gamma_2 = \gamma_2(\rho, d_1)$, $\epsilon^* = \epsilon^*(\rho, d_0, d_1, \alpha)$, $\delta^* = \delta^*(\rho, d_0, d_1, \alpha)$, and $\mu^* = \mu^*(\rho, d_0, d_1, \alpha)$ such that if $T^{-1}(B_{\gamma 2}) \subset S_1$ and Assumptions 1 and 3 are satisfied on S_1, Assumption 2 is satisfied on $B_{\gamma 1}$, and

$$|z(0)| \le \rho,$$

$$|\tilde{\Theta}(0)| = |\Theta_0 - \Theta| \le \frac{\delta^*}{\sqrt{2}},$$

$$\epsilon \le \epsilon^*$$

$$0 < \mu \le \mu^*$$

then the tracking error $e = y - y_m$ will converge to a ball of radius d_0 centered at the origin.

Proof: Consider the sets

$$I_e = \{e \,|\, |e| \le \gamma\} \tag{19}$$

and

$$I_{\tilde{\Theta}} = \{\tilde{\Theta} \,|\, |\tilde{\Theta}| \le \delta\} \tag{20}$$

where γ and δ are positive constants. In the forthcoming analysis, it is assumed that

$$e(t) \in I_e, \quad \text{and} \quad \tilde{\Theta}(t) \in I_{\tilde{\Theta}}, \qquad \forall t \ge 0. \tag{21}$$

Later on, we will show that the assumption (21) will never be violated from the beginning of the control process, based on an invariant set argument and some conditions imposed on γ, δ and ϵ. Since $e = z_1 - y_m$, we have $|z_1| \le \gamma + d_1 \triangleq k_1$. Using the function $V_2(z_2)$ defined in (11), one obtains

$$\begin{aligned} \dot{V}_2 &= \frac{dV_2}{dz_2}\psi(z_1, z_2) \\ &\le -\beta|z_2|^2 + \frac{dV_2}{dz_2}(\psi(z_1, z_2) - \psi(0, z_2)) \\ &\le -\beta|z_2|^2 + Lk_1 l|z_2| \end{aligned} \tag{22}$$

where l is the Lipschitz constant of $\psi(z_1, z_2)$ in z_1. Therefore,

$$\dot{V}_2 \le 0 \quad \text{for} \quad |z_2| \ge \frac{Lk_1 l}{\beta}$$

which, together with the bounds $c_1|z_2|^2 \le V_2(z_2) \le c_2|z_2|^2$ in (11), shows that

$$|z_2| \le \sqrt{\frac{c_2}{c_1}}\left(\frac{Lk_1 l}{\beta}\right) \triangleq k_2, \qquad \forall t \ge 0. \tag{23}$$

Let $\gamma_2 = \sqrt{(k_1^2 + k_2^2)}$ and define the set $B_{\gamma 2} = \{z \,|\, |z| \le \gamma_2\}$. Now we have

$$z \in B_{\gamma 2}, \qquad \forall t \ge 0. \tag{24}$$

Therefore,

$$\boldsymbol{x} \in T^{-1}(B_{\gamma 2}), \qquad \forall t \geq 0. \tag{25}$$

By (9), (13), and (14), one obtains

$$\begin{aligned}
\dot{z} &= f(z) + g(z)u \\
&= f_1(\boldsymbol{x}) + g_1(\boldsymbol{x})\left[\frac{-\hat{f}_1(\boldsymbol{x}, \boldsymbol{w}_t) + r}{\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t)}\right] \\
&\quad - \hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t)\left[\frac{-\hat{f}_1(\boldsymbol{x}, \boldsymbol{w}_t) + r}{\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t)}\right] \\
&\quad + \hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t)\left[\frac{-\hat{f}_1(\boldsymbol{x}, \boldsymbol{w}_t) + r}{\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t)}\right] \\
&= r + [f_1(\boldsymbol{x}) - \hat{f}_1(\boldsymbol{x}, \boldsymbol{w}_t)] + [g_1(\boldsymbol{x}) - \hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t)]u \\
&= r + [(\hat{f}_1(\boldsymbol{x}, \boldsymbol{w}) - \hat{f}_1(\boldsymbol{x}, \boldsymbol{w}_t)) + (\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}) - \hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t))u] \\
&\quad + [(f_1(\boldsymbol{x}) - \hat{f}_1(\boldsymbol{x}, \boldsymbol{w})) + (g_1(\boldsymbol{x}) - \hat{g}_1(\boldsymbol{x}, \boldsymbol{v}))u] \\
&= r + [-\tilde{\Theta}^T \boldsymbol{J} + O(|\tilde{\Theta}|^2)] + [O(\epsilon)].
\end{aligned} \tag{26}$$

Since $r = \dot{y}_m + \alpha(y_m - y)$, (26) becomes

$$\dot{e} + \alpha e = -\tilde{\Theta}^T \boldsymbol{J} + O(|\tilde{\Theta}|^2) + O(\epsilon) = -\tilde{\Theta}^T \boldsymbol{J} + \eta(t). \tag{27}$$

Before further development, we need to make it clear that the control $u(t)$ is uniformly bounded if δ and ϵ are small enough. The control $u(t)$ would go to infinity if $\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t)$ approaches zero. Since

$$\begin{aligned}
|\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t) - g_1(\boldsymbol{x})| &\leq |\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t) - \hat{g}_1(\boldsymbol{x}, \boldsymbol{v})| + |\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}) - g_1(\boldsymbol{x})| \\
&\leq \overline{c}|\tilde{\Theta}(t)|^2 + \epsilon \leq \overline{c}\delta^2 + \epsilon
\end{aligned} \tag{28}$$

the network $\hat{g}_1(\boldsymbol{x}, \boldsymbol{v}_t)$ is bounded away from zero and has the same sign as $g_1(\boldsymbol{x})$, $\forall t \geq 0$, provided δ and ϵ are small enough such that for all $\delta \leq \overline{\delta}$ and $\epsilon \leq \overline{\epsilon}$, $\overline{c}\delta^2 + \epsilon \leq \overline{c}\overline{\delta}^2 + \overline{\epsilon} \leq (b/2)$, (since $|g_1(\boldsymbol{x})| \geq b > 0$, see Assumption 1). With bounded state $\boldsymbol{x}$ and bounded control u, there exist c_3 and c_4 (depending on γ_2, $\overline{\delta}$, and $\overline{\epsilon}$) such that

$$|\eta(t)| \leq c_3|\tilde{\Theta}(t)|^2 + c_4\epsilon \leq c_3\delta^2 + c_4\epsilon. \tag{29}$$

If δ, ϵ are small enough, then there exists $\sigma > 0$ such that

$$|\eta(t)| + \sigma < \alpha d_0. \tag{30}$$

Define the function $V(e, \tilde{\Theta})$ as

$$\begin{aligned}
V(e, \tilde{\Theta}) &= \frac{1}{2}\mu d_0^2 + \frac{1}{2}\tilde{\Theta}^T\tilde{\Theta}, \quad \text{if} |e| \leq d_0 \\
&= \frac{1}{2}\mu e^2 + \frac{1}{2}\tilde{\Theta}^T\tilde{\Theta}, \quad \text{if} |e| > d_0.
\end{aligned} \tag{31}$$

Then,

$$\begin{aligned}
\text{when} \quad |e| \leq d_0, \quad \dot{V} &= \tilde{\Theta}^T\dot{\tilde{\Theta}} = 0, \\
\text{when} \quad |e| > d_0, \quad \dot{V} &= \mu e\dot{e} + \tilde{\Theta}^T\dot{\tilde{\Theta}} \\
&= \mu e(-\alpha e - \tilde{\Theta}^T\boldsymbol{J} + \eta(t)) + \tilde{\Theta}^T(\mu e\boldsymbol{J}) \\
&= \mu(-\alpha e^2 + e\eta(t)) \\
&< -\mu|e|\sigma < -\mu d_0\sigma < 0.
\end{aligned} \tag{32}$$

The result (32) is arrived at under the assumption (21). Next we focus on the assumption (21). Consider the set

$$M = \left\{ \begin{pmatrix} e \\ \tilde{\Theta} \end{pmatrix} \middle| \mu e^2 + \tilde{\Theta}^T\tilde{\Theta} \leq \delta^2 \right\}. \tag{33}$$

If μ and $\tilde{\Theta}(0)$ are chosen such that

$$\mu \leq \frac{\delta^2}{2e^2(0)}$$

$$|\tilde{\Theta}(0)| \leq \frac{\delta}{\sqrt{2}} \tag{34}$$

then $\begin{pmatrix} e(0) \\ \tilde{\Theta}(0) \end{pmatrix}$ falls into M, which, together with the derivation (32), guarantees that M is an invariant set, i.e., $\begin{pmatrix} e(t) \\ \tilde{\Theta}(t) \end{pmatrix}$ will stay in M for all $t \geq 0$. Hence the assumptions that $e(t) \in I_e$ and $\tilde{\Theta}(t) \in I_{\tilde{\Theta}}$, $\forall t \geq 0$, will never be violated if γ is large enough (when $\mu = (\delta_2/2e^2(0))$, γ can be set to be $\sqrt{2}e(0)$; γ increases as μ decreases), and if δ and ϵ are small enough such that $u(t)$ is bounded away from zero and that (30) hold.

Since $\dot{V} < -\mu d_0\sigma < 0$ when $|e| > d_0$, the total time during which adaptation takes place is finite. Let T_i denotes the time interval during which the tracking error e, for the ith time, stays outside the dead zone. If there are only finite times that the trajectory of the error would leave (and then reenter) the dead zone, the e will eventually stay in the dead zone. If the error may leave the dead zone for infinitely many times, still we have $\sum_{i=1}^{\infty} T_i$ being finite. It follows that

$$T_i \to 0 \quad \text{as} \quad i \to \infty. \tag{35}$$

If has been shown that e is bounded via an invariant set argument. Hence, from (27), it is clear that $\dot{e}$ is also bounded. Let $|e_i|$ denote the largest tracking error during the T_i interval. Then (35) and a bounded $\dot{e}$ imply that

$$|e_i| - d_0 \to 0 \quad \text{as} \quad i \to \infty. \tag{36}$$

The result (36) says that e will converge to the dead zone. □

Remarks:

1) The main challenge in this problem is the fact that the output of the multilayer network depends nonlinearly on the network parameters. Except a dead zone applied around the tracking error [11], the parameter updating rule (15) employed is typical of rules used to adjust linear parameters [1]. The main purpose of the dead zone is to cover the modeling error and the nonlinear effects of the parameter errors (see (30)). As a consequence, our convergence result is local with respect to the parameters; that is, the initial parameter errors are required to be small enough. This may not be a conservative result for the updating rule used, however, since Simulation Part A indicates that the closed-loop control system may go unstable if the initial parameter errors are too large. Therefore, in practice, before a multilayer network is used in the closed-loop control, an identification process involving the multilayer networks [12] is required, or the networks have to be trained to learn the nonlinearities from a (crude) model of the system (if available).
2) The theorem is regional in system states, i.e., the initial system state can start anywhere in a compact set. It is clear from the proof that the larger the compact set is (to which the initial states belong), the more restrictive the requirement on the initial parameters. Furthermore, the largest learning rate μ that can be used is inversely proportional to the square of the initial tracking error (see (34)). These features are demonstrated in Simulation Part B.

IV. Simulation

The simulation is conducted using the nonlinear plant

$$\begin{aligned}
\dot{y} = -0.2(\sin(y) + \cos(y)) - \frac{y}{1 + y^2} \\
+ (0.4\sin(y)\cos(y^2) + 0.8)u
\end{aligned} \tag{37}$$

which is a special case of the transformed plant (9) containing no zero-dynamics. There are four layers in both $\hat{f}(y, \boldsymbol{w})$ and $\hat{g}(y, \boldsymbol{v})$:

TABLE I
RELATIONS BETWEEN d_0 AND δ

	δ							
	0.0	0.05	0.1	0.3	0.7	1.0	1.5	2.0
d_0 0.008	★	★	★	★	★	★	★	×
0.01	2040	360	480	★	★	★	★	×
0.02	120	240	360	720	★	★	★	×
0.04	120	240	240	360	840	★	★	×
0.08	120	120	240	240	360	600	3840	×

TABLE II
RELATIONS BETWEEN α AND δ

	δ							
	0.0	0.05	0.1	0.3	0.5	1.0	1.5	2.0
α 1.0	2760	★	★	★	★	★	×	×
2.0	240	240	360	★	★	★	×	×
4.0	120	240	360	600	4080	★	★	×
8.0	120	240	460	480	1200	★	★	×
16.0	120	120	360	480	480	1560	4200	×

the input and output layers contain only one linear neuron, and the two hidden layers both contain five nonlinear neurons. In practice, the modeling error ϵ is determined once the structure of the neural network is determined. To estimate the size of ϵ, we have the neural network undergo an off-line training until the maximum output errors between f and $\hat{f}$, and between g and $\hat{g}$ reduce to 0.001297 and 0.001304, respectively. The network parameters after training are treated as the optimal ones. Since the updating laws of the neural networks and the dynamics of the plants are all described by differential equations, the simulation are implemented by ACSL (Advanced Continuous Simulation Language), which runs on a SUN SPARC station. The simulation is divided into three parts.

A. System Diverges if δ is Too Large

Equations (29) and (30) together imply that the initial parameter error δ, the modeling error ϵ, the dead-zone size d_0, and the gain α are related by

$$c_3\delta^2 + c_4\epsilon < \alpha d_0. \tag{38}$$

To demonstrate this relationship, the parameters of the pretrained neural network are perturbed by numbers randomly selected from $[-\delta, \delta]$. The perturbed network is then used to control the plant to track the reference command $y_m = 0.5(\sin((2\pi/30)t) + \sin((2\pi/20)t))$. With this periodic reference command, the error is said to have converged if it stays in the dead zone for two periods, i.e., 120 seconds, since it is observed that the tracking error never comes out of the dead zone after that. Table I shows the relationship between δ and d_0 for a fixed $\alpha(=1.0)$. The numbers in the table are the time required (in seconds) for convergence, ★ means that convergence has not been achieved up to 4800 seconds, and × means that the system diverges before 4800 seconds. It is observed from Table I that, when $\delta = 0$, a d_0 of size 0.01 is needed to tolerate the modeling error. As δ increases, d_0 has to increase for the tracking error to converge. Notice that the system diverges for $\delta \geq 2.0$. Table II shows the relationship between δ and α for a fixed $d_0(=0.01)$. It is observed that, as δ increases, α has to increase for the tracking error to converge. The system can diverge for $\delta > 1.5$.

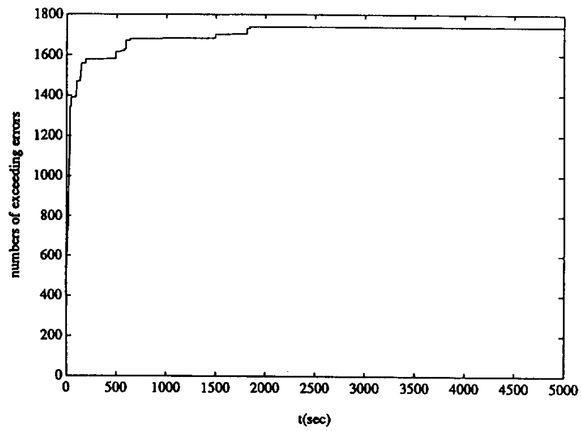

Fig. 1. The cumulated number of errors w.r.t. time when dead zone is used.

B. Tracking Error $e(t)$ Converges for A Range of $e(0)$

This simulation is designed to show that, for fixed α, d_0, ϵ and initial parameter errors, the tracking error would converge to the dead zone for a range of $e(0)$. For all the simulation runs, the parameters of the pretrained network are perturbed by a set of numbers randomly selected from $[-0.1, 0.1]$, d_0 is fixed at 0.02, α equals 4.0, and y_m is the same as in Part A. This simulation is also used to illustrate how μ is related to $e(0)$ (see (34)). For different $e(0)$, μ is gradually increased until convergence is not reached in 4800 seconds. The results are

$$\begin{aligned}
&\text{when} \quad e(0) = 0.5, \qquad \mu \leq 4.349\\
&\text{when} \quad e(0) = 1.0, \qquad \mu \leq 1.62\\
&\text{when} \quad e(0) = 1.5, \qquad \mu \leq 0.752\\
&\text{when} \quad e(0) = 2.0, \qquad \mu \leq 0.293\\
&\text{when} \quad e(0) = 2.5, \qquad \mu \leq 0.153.
\end{aligned}$$

These data show that our convergence result is not local in $e(0)$ and that the largest possible learning rate is inversely proportional to $e^2(0)$.

C. The Effect of Dead Zone Against Modeling Error

Here we want to demonstrate the advantage of using a dead zone in the presence of modeling errors. The plant, the networks and their initial weights are the same as those used in Part A, except that the high frequency term $0.1\sin((2\pi/0.1)y)$, which cannot be properly modeled by the neural network used, is added to f. The reference command is a smooth random trajectory. The parameters δ, d_0, α, and μ are chosen to be 0.1, 0.007, 17.0, and 2.0, respectively. We check for the output error exceeding d_0 every 0.005 sec. When an error is detected, we record the cumulated number of errors, its magnitude $(|e| - d_0)$, and the time when the error occurs. The results are shown in Figs. 1 and 2 for systems with dead zones and in Figs. 3 and 4 for systems without dead zone. We observe (from Figs. 3 and 4) that the frequency of error occurrence decreases toward zero and that $|e| - d_0$ tends to converge to zero for the system with dead zone. For the system without dead zone, both the error frequency and the error size show no tendency of decreaseness (Figs. 3 and 4).

V. CONCLUSION

The motivation of using multilayer networks in adaptive control problems is briefly discussed in Section I. The techniques employed for arriving at the convergence result in this note can also be applied to other approximation models which are nonlinearly parameterized,

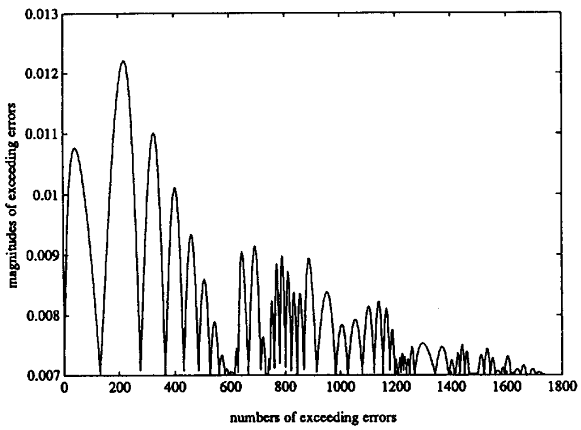

Fig. 2. The size of errors when dead zone is used.

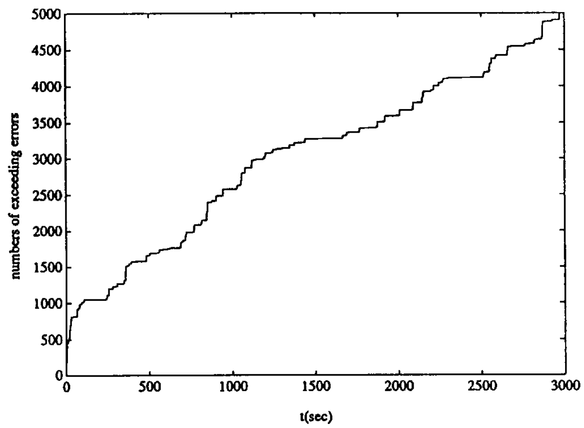

Fig. 3. The cumulated number of errors w.r.t. time when dead zone is not used.

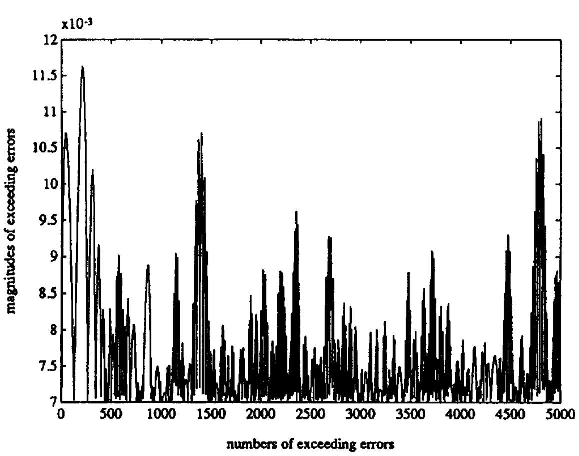

Fig. 4. The size of errors when dead zone is not used.

as long as a related algorithm is available to modify the nonlinear parameters. Although the control problem considered is a simpler one (for SISO, relative-degree-one systems only), this allows us to focus on the new aspects introduced by multilayer neural networks when they are used in adaptive control problems. Extensions to more general control problems (e.g., multi-input/multi-output systems with relative-degree higher than one) will be reported in a forthcoming paper.

References

[1] S. S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Trans. Automat. Contr.*, vol. 34, no. 11, pp. 1123–1131, Nov. 1989.

[2] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, "Systematic design of adaptive controllers for feedback linearizable systems," *IEEE Trans. Automat. Contr.*, vol. 36, no. 11, pp. 1241–1253, Nov. 1991.

[3] D. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, Rumelhart and McClelland (Eds). Cambridge, MA: vol. 1, MIT Press, 1986.

[4] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, pp. 4–27, Mar. 1990.

[5] F.-C. Chen, "Back-propagation neural networks for nonlinear self-tuning adaptive control," *IEEE Contr. Syst. Mag.*, Apr. 1990. ⟨page numbers⟩

[6] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, pp. 183–192, 1989.

[7] F.-C. Chen and H. K. Khalil, "Adaptive control of nonlinear systems using neural networks," *Inter. J. Contr.*, vol. 55, no. 6, pp. 1299–1317, Jun. 1992.

[8] ——, "Adaptive control of nonlinear systems using neural networks—A dead zone approach," in *Proc. 1991 Amer. Contr. Conf.*, 1991, pp. 667–672.

[9] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 837–863, Nov. 1992.

[10] W. Hahn, *Stability of Motion.* New York: Springer-Verlag, 1967.

[11] B. B. Peterson and K. S. Narendra, "Bounded error adaptive control," *IEEE Trans. Automat. Contr.*, vol. AC-27, no. 6, pp. 1161–1168, Dec. 1982.

[12] S. R. Chu and R. Shoureshi, "A neural network approach for identification of continuous-time nonlinear dynamic systems," in *Proc. 1991 Amer. Contr. Conf.*, 1991, pp. 1–5.

Robust Kalman Filtering for Uncertain Discrete-Time Systems

Lihua Xie, Yeng Chai Soh, and Carlos E. de Souza

***Abstract*—This note is concerned with the problem of a Kalman filter design for uncertain discrete-time systems. The system under consideration is subjected to time-varying norm-bounded parameter uncertainty in both the state and output matrices. The problem addressed is the design of a linear filter such that the variance of the filtering error is guaranteed to be within a certain bound for all admissible uncertainties. Furthermore, the guaranteed cost can be optimized by appropriately searching a scaling design parameter.**

Manuscript received February 22, 1993; revised July 30, 1993. This work was supported by the Nanyang Technological University Applied Research Fund and the Australian Research Council.

L. Xie and Y. C. Soh are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 2263.

C. E. de Souza is with the Laboratoire d'Automatique de Grenoble, ENSIEG, Saint-Martin-d'Hères, France and is on leave from the Department of Electrical and Computer Engineering, The University of Newcastle, NSW 2308, Australia.

IEEE Log Number 9401654.

0018–9286/94$04.00 © 1994 IEEE