SAGACIA turns out to be quite robust and is able to locate global optimum solutions for problems where gradient based algorithms fail. Furthermore, when SAGACIA failed to locate global optimal solutions, suboptimal solutions of good quality can be located quickly. SAGACIA can not only escape from local minima easily, but also converge to global minimum rapidly. It is an efficient and convenient optimization algorithm.

REFERENCES

[1] D. S. Johnson and C. R. Arago *et al.*, "Optimization by simulated annealing, an experimental evolution—Part I," *Graph Partitioning Oper. Res.*, vol. 37, no. 6, pp. 865–892, 1989.

[2] H. Das and P. T. Cumming *et al.*, "Scheduling multiproduct batch process via simulated annealing," *Comput. Chem. Eng.*, vol. 14, no. 12, pp. 1352–1362, 1990.

[3] I. P. Androulakis *et al.*, "A genetic algorithm framework for process design and optimization," *Comput. Chem. Eng.*, vol. 15, no. 4, pp. 217–228, 1991.

[4] L. Ding and X. F. Zhu, "New training method of artificial neural networks" (in Chinese), *Chemotaxis, Modeling and Control of Industrial Process*, vol. 5, pp. 332–337, 1992.

[5] B. Li and W. S. Jiang, "An improved genetic algorithm used to some combinatorial problems" (in Chinese), *Proc. Control and Decision*, pp. 360–362, 1995.

[6] J. C. Potts, T. D. Giddens, and S. B. Yadav, "The development and evaluation of an improved genetic algorithm based on migration and artificial selection," *IEEE Trans. Syst. Man, and Cybern.*, vol. 24, pp. 73–85.

[7] P. Tian, "The mix strategy of Darwin and Boltzman for combinatorial problems: Theory and applications," Ph.D. dissertation (in Chinese), 1993.

[8] F.-T. Lin *et al.*, "Applying the genetic approach to simulated annealing in solving some np-hard problems," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 1752–1767, Nov./Dec. 1993.

[9] M. Iosifescu, *Finite Markov Processes and Their Application*. New York: Wiley, 1980.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison Wesley, 1989.

[11] J. N. Aaral, K. Tumer, and J. Ghosh, "Designing genetic algorithms for the state assignment problem," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 687–694, 1995.

[12] B. Li and W. S. Jiang, "Quasisimulated annealing algorithm and its application" (in Chinese), *J. Inform. Contr.*, vol. 24, pp. 412–415, 1995.

[13] D. Dasgupta and D. R. Mcgregor, "A more biologically motivated genetic algorithm: The model and some results," *Cybernetics and Systems: An International Journal*, vol. 25, pp. 447–469, 1994.

[14] J. J. Grefenstette, "Optimization of control parameter for genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, pp. 122–128, 1986.

[15] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 761–680, 1983.

[16] V. Kreinovich, C. Quintana, and O. Fuentes, "Genetic algorithms: What fitness scaling is optimal?," *Cybernetics and Systems: An International Journal*, vol. 24, pp. 9–25, 1993.

[17] K. Krishna, K. Ganeshan, and D. J. Ram, "Distributed simulated annealing algorithms for job shop scheduling," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1102–1109, 1995.

[18] H. M. Cartwright and R. A. Long, "Simultaneous optimization of chemical flowshop sequencing and topology using genetic algorithms," *Ind. Eng. Chem. Res.*, vol. 32, pp. 2706–2713, 1993.

Neural Network-Based Model Reference Adaptive Control System

H. D. Patiño and Derong Liu

***Abstract*—In this paper, an approach to model reference adaptive control based on neural networks is proposed and analyzed for a class of first-order continuous-time nonlinear dynamical systems. The controller structure can employ either a radial basis function network or a feedforward neural network to compensate adaptively the nonlinearities in the plant. A stable controller-parameter adjustment mechanism, which is determined using the Lyapunov theory, is constructed using a σ-modification-type updating law. The evaluation of control error in terms of the neural network learning error is performed. That is, the control error converges asymptotically to a neighborhood of zero, whose size is evaluated and depends on the approximation error of the neural network. In the design and analysis of neural network-based control systems, it is important to take into account the neural network learning error and its influence on the control error of the plant. Simulation results showing the feasibility and performance of the proposed approach are given.**

***Index Terms*—Adaptive control, adaptive systems, neural network applications, neurocontrollers.**

I. INTRODUCTION

Let a plant be given by the following differential equation

$$\dot{y}(t) + f[y(t)] = u(t), \qquad t \geq 0 \tag{1}$$

where $y(t)$ is the output signal of the system, $u(t)$ is the input signal to the system, and $f\colon R \rightarrow R$ is the unknown static nonlinear function which is continuously differentiable and Lipschitz. Let a stable linear continuous-time reference model be specified by the following differential equation:

$$\dot{y}_m(t) + a_m y_m(t) = k_m r(t), \qquad t \geq 0 \tag{2}$$

where $y_m(t)$ is the output signal, $r(t)$ is the reference input signal, and $a_m > 0,\ k_m > 0$.

The objective of a model reference adaptive control (MRAC) system can be stated as follows. It is desired to obtain a *control law* $u(t)$, and an *updating law* of the controller parameters, such that one or more variables of the plant are kept within prescribed limits, and the closed-loop system maintains a performance specified by the reference model. In other words, it is desired to design a controller that computes a control action signal, such that the overall control system responds dynamically as the specified reference model. This may be expressed in mathematical terms as follows. A plant with an input–output pair $u(t)$, $y(t)$ is given as in (1), and a stable reference model specified by its input–output pair $r(t), y_m(t)$ is given as in (2) with the reference input signal of the system $r \in L_\infty$. Then, the objective is to determine a control action law, $u(t)$, for all $t > 0$, and an updating law of the controller parameters such that

$$\lim_{t\rightarrow\infty} |y(t) - y_m(t)| \leq \varepsilon$$

for some specified constant $\varepsilon > 0$.

Manuscript received November 19, 1997; revised April 17, 1999. This work was supported in part by the National Science Foundation under Grant ECS-9732785. This paper was recommended by Associate Editor R. A. Hess.

H. D. Patiño is with the Instituto de Automatica, Facultad de Ingenieria, Universidad Nacional de San Juan, 5400 San Juan, Argentina.

D. Liu is with the Department of Electrical Engineering and Computer Science, University of Illinois, Chicago, IL 60607-7053 USA.

Publisher Item Identifier S 1083-4419(00)01213-9.

It is shown in [5], [6], [13], and [14], and others that any well-behaved nonlinear function can be approximated to any desired accuracy by a two-layer feedforward neural network or a radial basis function (RBF) network, with a sufficiently large number of neurons, over a compact domain of a finite dimensional normed vector space. In [1], [2], [7], [12], [15]–[17], and [20], it is shown that neural networks can be used for both identification and control of dynamical systems. In these works, the nonlinear mapping capability of neural networks is exploited for forward and inverse plant models in order to develop different adaptive control schemes.

In this paper, a neural network controller based on MRAC is developed, in which the error between the outputs of the plant and the reference model is used to adapt the controller parameters. The present results are established for MRAC with a first-order reference model. The nonlinear part of the controller, which compensates the plant nonlinearity $f(\cdot)$, can be implemented by either an RBF network or a feedforward neural network. The learning law used to train on-line the RBF network or the feedforward neural network is a σ-modification-type updating law [8]. The adjustment mechanism is determined by the Lyapunov stability analysis of the overall adaptive control system. This kind of neural network-based adaptive controller is applicable to a wide variety of practical problems. Another interesting contribution of the present paper is the evaluation of control error (the error between the outputs of the plant and the reference model) in terms of the neural network learning error. In the design and analysis of neural network-based control systems, it is important to take into account the neural network learning error and its influence on the control error of the plant.

II. Main Results

The nonlinear adaptive control system considered in the present paper is generalized from the well-known linear model reference adaptive control systems [9], [11]. Consider the plant to be controlled given by (1) and a reference model given by (2). Assume that a_m, k_m, and $r(t)$ have been chosen such that a desired trajectory $y_m(t)$ is obtained for the plant output $y(t)$ to follow. The proposed control law has the following form:

$$u(t) = -a_m y(t) + k_m r(t) + N_f[y(t),\, w(t)] = \theta^T \phi(t) \quad (3)$$

where $N_f(\cdot,\, \cdot)$ is implemented using an RBF network or a feedforward-type neural network that approximates the function $f(\cdot)$, w is the parameter vector of the neural network ($w \in R^p$), and $\theta = [a_m,\, k_m,\, 1]^T \in R^3$ is a vector of constant parameters, and $\phi(t) = [-y,\, r,\, N_f]^T \in R^3$ is a vector of functions. The vector w represents the neurocontroller parameters to be tuned (Fig. 1).

Define the error signal as

$$e(t) \triangleq y(t) - y_m(t).$$

When the neural network exactly represents the function $f(\cdot)$, i.e., when $N_f[y(t),\, w(t)] = f[y(t)]$ for all t, the closed-loop system equation, in terms of the error signal, is obtained by substituting (2) and (3) into (1) as

$$\dot{e}(t) + a_m e(t) = 0. \quad (4)$$

Note that (4) represents an unforced linear system with a unique equilibrium point at the origin, and it is asymptotically stable since $a_m > 0$. Thus, the control objective of $y(t)$ tracking $y_m(t)$ is achieved, i.e., $e(t) = y(t) - y_m(t) \to 0$ as $t \to \infty$.

Consider the neural network learning error, i.e., the approximation error in the representation of the function $f(\cdot)$ by the neural network, given by

$$\Delta(y,\, w) \triangleq N_f[y(t),\, w(t)] - f[y(t)]. \quad (5)$$

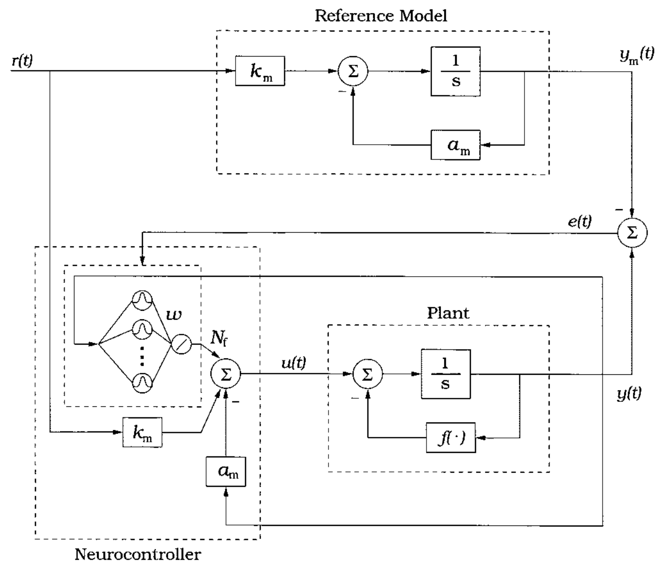

Fig. 1. Neural network-based model reference adaptive control system structure.

Substituting (2), (3), and (5) into (1), the closed-loop system equation becomes

$$\dot{e}(t) + a_m e(t) = \Delta(y,\, w). \quad (6)$$

Note that when in (6) the learning error tends to zero, i.e., when $N_f \to f$, the control error $e(t)$ tends to zero too. Define the neural network weight parameter error as $\tilde{w} = w - w^*$, where w^* is the optimal parameter vector corresponding to the global minimum error of the network which minimizes $|\Delta(y,\, w)|$; i.e., the minimum value of $|\Delta(y,\, w)|$ that could be reached is $|\Delta(y,\, w^*)|$. In the sequel, a stable parameter adjustment law is determined using the Lyapunov stability theory, and the practical stability for the neural network-based MRAC system is demonstrated. In addition, an explicit evaluation of the control error in function of the neural network learning error and the design parameter will be given.

A. Stability Analysis

In this subsection, analysis will be given for the case in which an RBF network is employed to approximate the nonlinear function of the plant. It is assumed that both RBF centers and widths have been chosen and fixed adequately, and the weight values of the linear combiner will be adjusted by a learning law such that the stability of the whole adaptive control system can be guaranteed. Background material on practical stability will be first introduced.

Definition 1—[10]: Let a system be given by

$$\dot{x} = X(x,\, t), \qquad t \geq 0 \quad (7)$$

which has the equilibrium state at the origin, i.e., $X(0,\, t) = 0$ for all $t \geq 0$. Let the perturbed system be given by

$$\dot{x} = X(x,\, t) + p(x,\, t), \qquad t \geq 0. \quad (8)$$

Let Q be a set which is closed and bounded containing the origin and let Q_0 be a subset of Q. Let $x(t,\, x_0,\, t_0)$ be the solution of (8) satisfying $x(t_0,\, x_0,\, t_0) = x_0$. Let P be the set of perturbations satisfying $|p(x,\, t)| \leq \delta$ for all $t \geq 0$ and for all x, where $\delta > 0$. If for each p in P, each x_0 in Q_0, and each $t_0 \geq 0$, $x(t,\, x_0,\, t_0)$ is in Q for all $t \geq 0$, then, the equilibrium of (7) at the origin is said to be *practically*

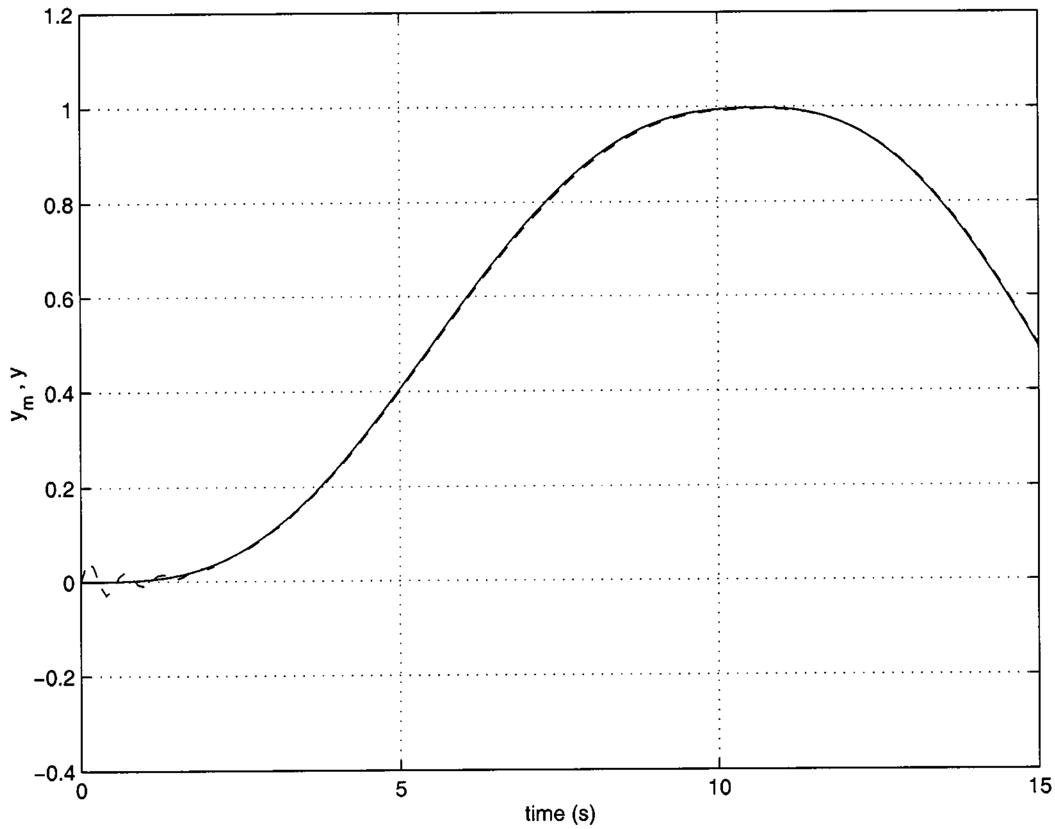

Fig. 2. Evolution of desired output and measured output (dashed line) of the adaptive control system in Example 1.

stable. In other words, practical stability of (7) implies that the solution of (8) which starts initially in Q_0 remains thereafter in Q. ■

Practical stability implies that the system may oscillate sufficiently close to its operating state with acceptable performance even though the operating state of the system may be unstable in the sense of Lyapunov [10]. The concept of practical stability is relative to the sets Q and Q_0. Q is the set of acceptable states, i.e., if the state $x(t)$ of the system at time t is in Q, then the system at that time is operating satisfactorily. The subset Q_0 is a set of initial states. It is necessary to take into account the concept of practical stability in the analysis and design of certain control systems. This will specify how close the state of the system to operating state is acceptable (the set Q), the magnitude of perturbations to be expected (the number δ), and how well the initial conditions should be controlled (the set Q_0).

In analogy to the classic asymptotic stability in the large, we can consider a *strong practical stability*. For this case, given δ, Q, and Q_0, if the origin is practically stable and if, in addition, we require that every solution of the system (8) for each $p \in P$ be ultimately in Q, then we say that the system (7) has a strong practical stability. The following result is from [10].

Lemma 1: Let $V(x)$ be a scalar function which has continuous first partial derivative for all x and with the property that $V(x) \to \infty$ as $|x| \to \infty$. Let $\dot{V}_{(8)}$ denote the time derivative of V along the solutions of system (8). If $\dot{V}_{(8)} \leq -\varepsilon$ for all x outside Q_0, for all p in P, and for all $t \geq 0$ and if $V(x) \leq V(y)$ for all x in Q_0 and all y outside Q, then system (7) possesses strong practical stability. ■

The following is an application of Lemma 1 to our case.

Theorem 1: Suppose that the control law is given by (3) and the parameter updating law is given by

$$\dot{w}(t) = -\Psi B(t)e(t) - Kw(t) \tag{9}$$

where

$w \in R^p$	weight vector of the linear combiner of the RBF units;
Ψ and K	diagonal positive definite matrices, i.e., $\Psi = \mathrm{diag}(\Psi_i)$ and $K = \mathrm{diag}(K_i)$ with $\Psi_i > 0$ and $K_i > 0$;
$B(t) \in R^p$	output weight vector of the RBF units, i.e., $N_f[y(t), w(t)] = B^T(t)w(t)$.

Then, the whole system given by

$$\begin{cases} \dot{e}(t) + a_m e(t) = 0 \\ \dot{w}(t) = -\Psi B(t)e(t) - Kw(t) \end{cases} \tag{10}$$

possesses strong practical stability under perturbation given by $[\Delta(y, w), 0]^T$, where $\tilde{w}(t) = w(t) - w^*$ and w^* is the optimal parameter vector as defined before.

Proof: System (10) under the given perturbation is described by

$$\begin{cases} \dot{e}(t) + a_m e(t) = \Delta(y, w) \\ \dot{w}(t) = -\Psi B(t)e(t) - Kw(t). \end{cases} \tag{11}$$

Consider the positive definite function

$$V(e, \tilde{w}) = \tfrac{1}{2}e^2 + \tfrac{1}{2}\tilde{\omega}^T \Psi^{-1} \tilde{\omega}$$

in which Ψ is a diagonal positive definite matrix defined above. We need to show that the derivative of V along the solutions of system (11) satisfies the conditions in Lemma 1.

Clearly, V can be upper bounded by

$$V(e, \tilde{w}) \leq \tfrac{1}{2}e^2 + \tfrac{1}{2}\left\|\Psi^{-1}\right\| \cdot \|\tilde{w}\|^2.$$

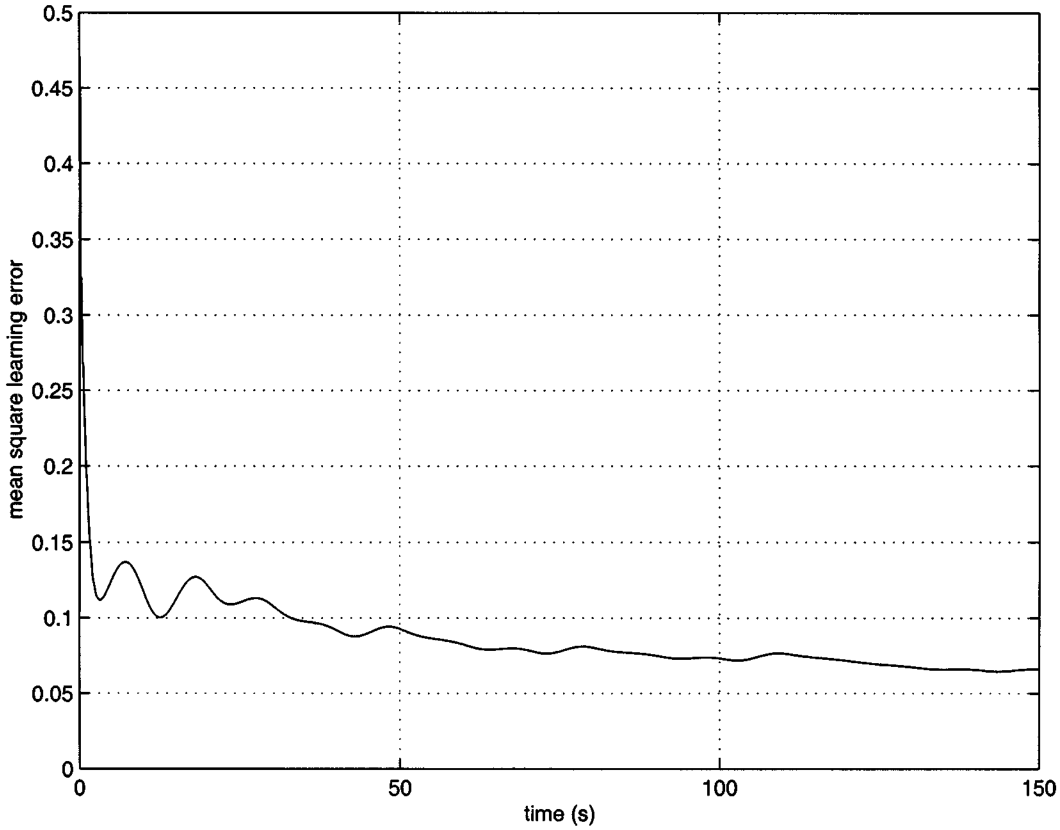

Fig. 3. Learning curve of the system in Example 1.

The time derivative of V evaluated along the trajectories of system (11) is

$$\begin{aligned}\dot{V}_{(11)} &= e\dot{e} + \tilde{w}^T \Psi^{-1} \dot{\tilde{w}} \\ &= -a_m e^2 + \Delta(y, w)e + \tilde{w}^T \Psi^{-1} \dot{\tilde{w}}. \qquad (12)\end{aligned}$$

Since both $f(y)$ and $N_f(y, w)$ are continuously differentiable with respect to their arguments, so is $\Delta(y, w)$. One can apply the mean value theorem and obtain

$$\Delta(y, w) = \Delta(y, w^*) + \tilde{w}^T \left(\frac{\partial \Delta(y, w_\xi)}{\partial w_\xi}\right) \qquad (13)$$

for some w_ξ, where $\Delta(y, w^*)$ is the learning error evaluated at the global minimum $w = w^*$.

Considering (5) and

$$\frac{\partial \Delta(y, w_\xi)}{\partial w_\xi} = \frac{\partial N_f(y, w_\xi)}{\partial w_\xi} = B(t)$$

and substituting (13) into (12), one gets

$$\dot{V}_{(11)} = -a_m e^2 + \Delta(y, w^*)e + \tilde{w}^T \left(Be + \Psi^{-1}\dot{\tilde{w}}\right). \qquad (14)$$

The second term of (14) is partially cancelled out by the following parameter updating law

$$\dot{\tilde{w}}(t) = -\Psi B(t)e(t) - Kw(t). \qquad (15)$$

As w^* is a constant vector, the adjusting law of w can be determined as in (9), i.e.,

$$\dot{w}(t) = -\Psi B(t)e(t) - Kw(t).$$

Then, considering (15) [or (11)] and $\tilde{w} = w - w^*$, (14) becomes

$$\dot{V}_{(11)} = -a_m e^2 + \Delta(y, w^*)e - \tilde{w}^T \Psi^{-1} K\tilde{w} - \tilde{w}^T \Psi^{-1} K w^*$$

and $\dot{V}_{(11)}$ can be upper bounded by

$$\begin{aligned}\dot{V}_{(11)} \le& -a_m e^2 + |\Delta(y, w^*)| \cdot |e| - \mu_1 \|\tilde{w}\|^2 \\ &+ \mu_2 \|\tilde{w}\| \cdot \|w^*\| \qquad (16)\end{aligned}$$

where $\mu_1 = \min_i\{K_i/\Psi_i\}$ and $\mu_2 = |\Psi^{-1}K|$. Taking into account the fact that the bilinear terms can be expressed as

$$\begin{aligned}|\Delta(y, w^*)| \cdot |e| =& -\frac{1}{2}\left(\frac{|e|}{\eta} - |\Delta(y, w^*)|\eta\right)^2 + \frac{1}{2}\,\frac{e^2}{\eta^2} \\ &+ \frac{1}{2}\,\eta^2 |\Delta(y, w^*)|^2\end{aligned}$$

and

$$\begin{aligned}\|\tilde{w}\| \cdot \|w^*\| =& -\frac{1}{2}\left(\frac{\|\tilde{w}\|}{\xi} - \|w^*\|\xi\right)^2 + \frac{1}{2}\,\frac{\|\tilde{w}\|^2}{\xi^2} \\ &+ \frac{1}{2}\xi^2 \|w^*\|^2\end{aligned}$$

for some $\eta \in R$ and $\xi \in R$, (16) can be rewritten as

$$\begin{aligned}\dot{V}_{(11)} \le& -\left(a_m - \frac{1}{2\eta^2}\right)e^2 - \left(\mu_1 - \frac{\mu_2}{2\xi^2}\right)\|\tilde{w}\|^2 \\ &+ \frac{1}{2}\left(\eta^2 |\Delta(y, w^*)|^2 + \mu_2 \xi^2 \|w^*\|^2\right). \qquad (17)\end{aligned}$$

Equivalently, (17) can be written as

$$\dot{V}_{(11)} \le -\sigma V + \rho \qquad (18)$$

with

$$\sigma = \min\left(2a_m - \frac{1}{\eta^2},\ \frac{2\mu_1 \xi^2 - \mu_2}{\|\Psi^{-1}\| \cdot \xi^2}\right)$$

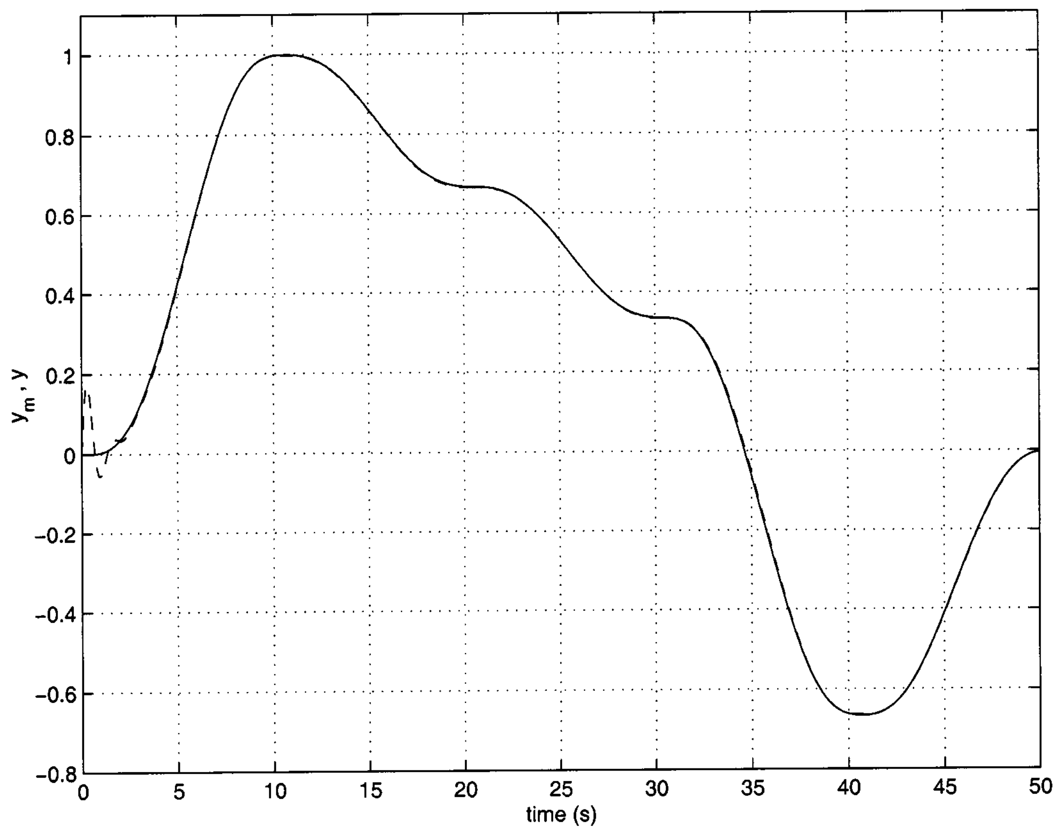

Fig. 4. Evolution of desired output and measured output (dashed line) of the adaptive control system in Example 2.

and $\rho = 1/2(\eta^2|\Delta(y, w^*)|^2 + \mu_2\xi^2\|w^*\|^2)$. It is always possible to choose $\eta^2 > (1/2a_m)$ and $\xi^2 > (\mu_2/2\mu_1)$, i.e., $\sigma > 0$. Now, consider

$$Q = Q_0 = \left\{(e, \tilde{w}): V(e, \tilde{w}) \leq \frac{\rho + \varepsilon}{\sigma}\right\}$$

for some $\varepsilon > 0$. It results from (18) that $\dot{V}_{(11)} < -\varepsilon$ for all e and $\tilde{w}$ outside Q since

$$V(e, \tilde{w}) > \frac{\rho + \varepsilon}{\sigma}$$

and that $V(x) \leq V(y)$ for all $x \in Q$ and all y outside Q. Then, applying Lemma 1, (18) implies that (10) possesses strong practical stability. ■

Remark 1: Theorem 1 implies that all trajectories of system (11) will be ultimately in Q, which implies that $e(t) \in L_\infty$ and $\tilde{w}(t) \in L_\infty^p$ for all $t \geq 0$. ■

Remark 2: The nonlinear part of the controller that compensates the nonlinearities $f[y(t)]$ of the plant can also be approximated by feedforward neural networks. In this case, for demonstrating the stability of the control system it is necessary to make modifications to the learning law of the network. ■

B. Control Error Evaluation as a Function of the Neural Network Learning Error

In this subsection an evaluation of the control error as an explicit function of the neural network learning error and design parameter will be conducted.

Define the Laplace transfer function of system (6) as $G(s)$, i.e.,

$$G(s) = \mathcal{L}\{g(t)\} = \frac{E(s)}{\Delta(s)} = \frac{1}{s + a_m}. \tag{19}$$

The temporal response of system (6) to the input $\Delta[y(t), w(t)]$ can be obtained by the following convolution

$$e(t) = g(t) * \Delta[y(t), w(t)]. \tag{20}$$

Considering the truncation $\Delta_T[y(t), w(t)]$ of $\Delta[y(t), w(t)]$ to the interval $[0, T]$, (20) becomes

$$e_T(t) = (g(t) * \Delta_T[y(t), w(t)])_T$$

where $e_T(t)$ is the truncation of $e(t)$ to the same interval. Applying the property that [20, p. 251]

$$\|p * q\|_\infty \leq \|p\|_A \cdot \|q\|_\infty$$

where the A-norm for function $p(t)$ is defined as [20, p. 246]

$$\|p(t)\|_A = \int_0^\infty |p(t)|\, dt$$

and the infinity norm for function $q(t)$ is defined as

$$\|q(t)\|_\infty = \text{ess.sup}_{t\geq 0}|q(t)|.$$

In our case, we have for $g(t)$

$$\|g(t)\|_A = \int_0^\infty |g(t)|\, dt = \frac{1}{a_m}.$$

Then, one has

$$\begin{aligned}\|e_T(t)\|_\infty &\leq \|g(t)\|_A \cdot \|\Delta_T[y(t), w(t)]\|_\infty \\ &= \frac{1}{a_m}\, |\Delta_T[y(t), w(t)]|_\infty.\end{aligned} \tag{21}$$

Notice that this expression can be used to evaluate the performance of the system through the evaluation of the learning error and design parameter a_m.

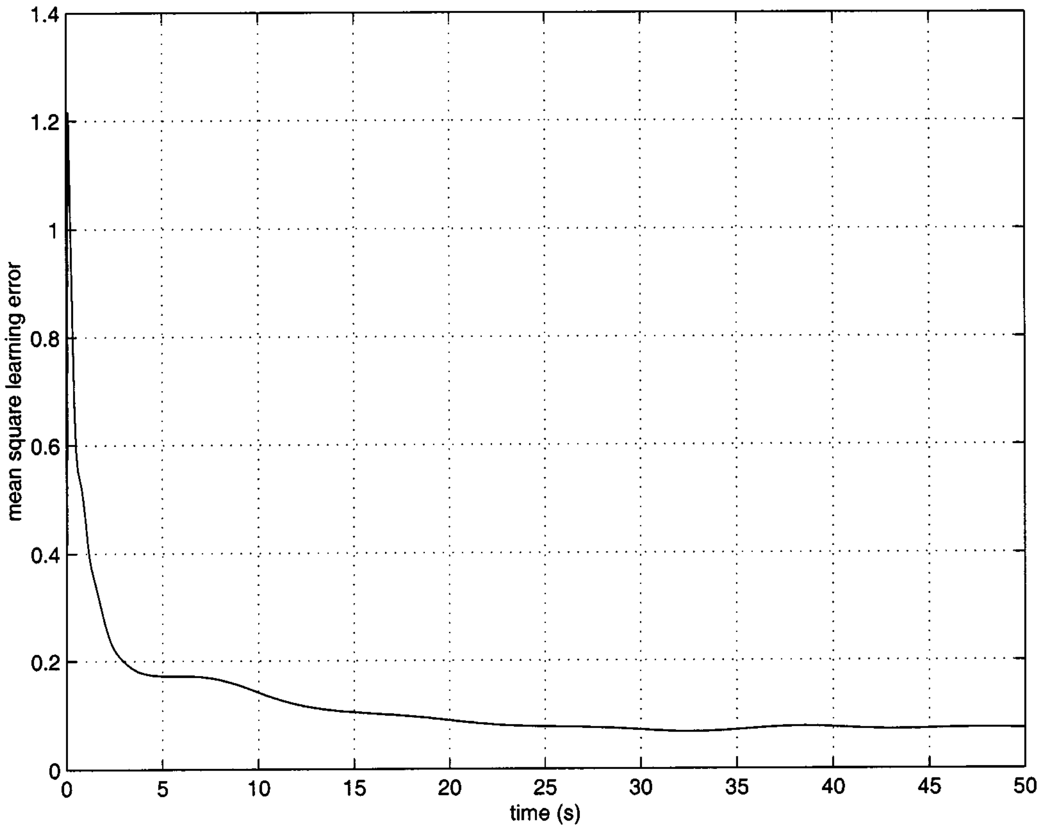

Fig. 5. Learning curve of the system in Example 2.

We note that the bound obtained in (21) is established by an analysis on the bounds for input–output relation and it is useful to evaluate the performance of control system both in the transient and steady state.

III. Simulation Results

In order to show the feasibility and performance of the proposed neural network-based adaptive control algorithm, as well as the stability properties obtained in the preceding theoretical development, a simulation study has been carried out for two examples of nonlinear plants.

Example 1: The plant to be controlled is governed by the nonlinear differential equation given by (1), in which the unknown function has the form

$$f[y(t)] = 2y(t) + 0.8y^3(t).$$

Notice that this mathematical model has similar structure to ship-steering model [4], [18]. The reference model is described by a first-order differential equation

$$\dot{y}_m(t) + 2y_m(t) = 2r(t)$$

where $r(t)$ is a smooth bounded reference input signal in the interval [−1, 1].

The function $f(\cdot)$ is estimated on-line using a Gaussian-type radial basis function network with input $y(t)$ and output $N_f(\cdot, \cdot)$. The number of nodes of the hidden layer used is 100 with a spread of 0.03. The centers are distributed uniformly along [−1, 1]. Other design procedures for placing the nodes in the input domain, such as the ones described in [3], are not possible to use due to the fact that the error between the actual and desired network output is not explicitly available.

The weight vector of the RBF network is adjusted according to (9) with

$$\Psi = \text{diag}[0.0125, \cdots, 0.0125]$$

and

$$K = \text{diag}[0.999, \cdots, 0.999].$$

After the training process is completed, $\|\Delta(y, w)\|_\infty = 0.204$ is obtained for the learning error. In this case, (21) gives a bound

$$\|e_T(t)\|_\infty \le \frac{1}{a_m} \|\Delta_T[y(t), w(t)]\|_\infty = \frac{0.204}{2} = 0.102$$

where $T = 30$ s.

The evolution of the desired and measured output signals of the system is presented in Fig. 2 (the dashed line is for y). The learning process can be seen in Fig. 3, which represents the mean square neural network learning error during the first 150 s. ■

Example 2: In this case, the plant to be controlled is described by the nonlinear differential equation (1), where the unknown function $f(\cdot, \cdot)$ has the form

$$f[y(t), u(t)] = \frac{y(t)}{1 + y^2(t)} + 0.1 \tanh[u(t)]. \tag{22}$$

The last term of (22) represents a small-bounded perturbation.

The reference model employed is described by

$$\dot{y}_m(t) + 2.5y_m(t) = 2.5r(t). \tag{23}$$

Here, the reference input signal $r(t)$ is also in the interval [−1, 1]. The function $f(\cdot, \cdot)$ is estimated on-line using a Gaussian-type RBF network, which has two inputs $y(t)$ and $u(t)$, and one output $N_f(\cdot, \cdot)$.

The number of nodes of the hidden layer utilized is 250, with a spread of 0.03 along y and 0.06 along u. The centers are distributed uniformly along [−1, 1] in the y axis and along [−3, 3] in the u axis. The RBF network weights are adjusted according to (9) with a similar Ψ and K as in Example 1. During the operation of the system, $\|\Delta(y, w)\|_\infty = 0.48$ is obtained for the learning error. The bound in (21) in this case is

$$\|e_T(t)\|_\infty \le \frac{0.48}{2.5} = 0.192$$

where $T = 50$ s. The desired and measured output signals of the controlled system can be observed in Fig. 4. The learning process, in terms of the evolution of mean square learning error during the first 50 s, is shown in Fig. 5. ■

IV. Conclusions

A neural network-based model reference adaptive controller for a class of nonlinear dynamical plants has been presented. The results obtained can be extended to systems of higher order and multivariable. The design of the present adaptive controller is based on the Lyapunov stability theory. The controller structure is a direct type and can employ either radial basis function networks or feedforward neural networks to compensate adaptively the nonlinearities in the plant. In comparison to the traditional learning systems, here the error between the actual nonlinear function output and desired neural network output is not available. Instead, it is shown that the discrepancy between the reference model output and the plant output can be used as the activation signal of the parameter adjusting law. The controller parameter adjustment law is determined using the Lyapunov stability theory. A practical stability result for the proposed control system is given, which takes into account the neural network learning error. In other words, the convergence of the control error to a neighborhood of zero can be assured, and the radius of the neighborhood is evaluated and depends on the approximation error of the network. In addition, a result is established that relates explicitly the control error to the neural network learning error and the design parameter. We believe that the procedure carried out in the proof of stability of the adaptive control system will be useful to the proof of the stability of other kinds of neural network-based adaptive control structures. To show the practical feasibility and performance of the proposed neural network-based adaptive control algorithm as well as the stability properties obtained in the present paper, a simulation study was carried out for two examples of nonlinear plants. The directions for future investigation will be oriented at the robustness issues of the neural network-based adaptive control structures, and applications of the control algorithm to other real plants.

References

[1] R. Carelli, E. F. Camacho, and D. Patiño, "A neural network based feedforward adaptive controller for robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1281–1288, Sept./Oct. 1995.
[2] F. Chen, "Back-propagation neural networks for nonlinear self-tuning adaptive control," *IEEE Contr. Syst. Mag.*, vol. 10, pp. 44–48, Apr. 1990.
[3] H. Demuth and M. Beale, *User's Guide of the Neural Network Toolbox*: The Math Works Inc., 1994.
[4] T. I. Fossen, "High performance ship autopilot with wave filter," in *Proc. Ship Control System Symp.*, Ottawa, Canada, Oct. 1993, pp. 25–29.
[5] K. Funahashi, "On the approximate realization of continuous mapping by neural networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.
[6] K. Hornik, M. Stinchcombe, and H. White, "Multilayered feedforward network are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
[7] K. J. Hunt, D. Sbarbaro, R. Żbikowski, and P. J. Gawthrop, "Neural networks for control systems—A survey," *Automatica*, vol. 28, pp. 1083–1112, 1992.
[8] P. A. Ioannou and P. A. Kokotovic, *Adaptive Systems with Reduced Models*. Berlin, Germany: Springer-Verlag, 1983.
[9] I. D. Landau, *Adaptive Control: The Model Reference Approach*. New York: Marcel Dekker, 1979.
[10] J. La Salle and S. Lefschetz, *Stability by Lyapunov: Direct Method with Applications*. New York: Academic, 1961.
[11] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[12] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, Mar. 1990.
[13] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, pp. 246–257, 1991.
[14] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, Sept. 1990.
[15] M. M. Polycarpou and P. A. Ioannou, "Learning and convergence analysis of neural-type structured networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 39–50, Jan. 1992.
[16] M. Saerens and A. Soquet, "Neural-controller based on back-propagation algorithm," *Proc. Inst. Elect. Eng.*, pt. F, vol. 138, pp. 55–62, 1991.
[17] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, pp. 837–863, Nov. 1992.
[18] J. Van Amerongen, "Adaptive steering of ships—A model reference approach," *Automatica*, vol. 20, pp. 3–14, 1984.
[19] M. Vidyasagar, *Nonlinear System Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1978.
[20] K. Warwick, G. W. Irwin, and K. J. Hunt, *Neural Networks for Control and Systems*. Stevenage, U.K.: Peregrinus, 1992.

Dynamic Output Feedback Controller Design forFuzzy Systems

Z. X. Han, G. Feng, B. L. Walcott, and J. Ma

***Abstract*—This paper presents dynamic output feedback controller design for fuzzy dynamic systems. Three kinds of controller design methods are proposed based on a smooth Lyapunov function or a piecewise smooth Lyapunov function. The controller design involves solving a set of linear matrix inequalities (LMI's) and the control laws are numerically tractable via LMI techniques. The global stability of the closed-loop fuzzy control system is also established.**

***Index Terms*—Dynamic output feedback controller, fuzzy systems, linear matrix inequality.**

I. Introduction

Fuzzy logical control (FLC) techniques represent a means of both collecting human knowledge and expertise and dealing with uncertainties in the process of control. In many cases, it has been suggested as an alternative approach to conventional control techniques, (see , for example, [2]–[4]). The first attempt to design a fuzzy control system was made in [2]. Many researchers have since followed that method which was based on the compositional rule of inference [1] and approximate reasoning [6]. Though the method has been practically successful, it

Manuscript received October 7, 1998; revised October 31, 1999. This paper was recommended by Associate Editor C. P. Neuman.
Z. X. Han and B. L. Walcott are with the Department of Electrical Engineering, University of Kentucky, Lexington, KY 40506 USA.
G. Feng is with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia.
J. Ma is with the Department of Information Systems, City University of Hong Kong, Kowloon Tong, Hong Kong.
Publisher Item Identifier S 1083-4419(00)01398-4.