Journal of Intelligent and Robotic Systems **23:** 105–128, 1998.

Robust Backstepping Control of Robotic Systems Using Neural Networks

S. JAGANNATHAN
Department of Electrical Engineering, University of Texas at San Antonio, San Antonio, TX 78249, USA; e-mail: saranj@gw1.cat.com

F. L. LEWIS
Automation and Robotics Research Institute, 7300 Jack Newell Blvd. South, Fort Worth, Texas 76118, USA

(Received: January 5 1998; accepted: 4 March 1998)

Abstract. Neural network (NN) controllers for the robust back stepping control of robotic systems in both continuous and discrete-time are presented. Control action is employed to achieve tracking performance for unknown nonlinear system. Tuning methods are derived for the NN based on delta rule. Novel weight tuning algorithms for the NN are obtained that are similar to ε-modification in the case of continuous-time adaptive control. Uniform ultimate boundedness of the tracking error and the weight estimates are presented without using the persistency of excitation (PE) condition. Certainty equivalence is not used and regression matrix is not computed. No learning phase is needed for the NN and initialization of the network weights is straightforward. Simulation results justify the theoretical conclusions.

Key words: neural networks, robust control, back stepping control, adaptive control.

1. Introduction

Adaptive control is an important area of research that has been widely pursued [1, 12, 15, 19]. Progress of adaptive control theory and the availability of microprocessors have led to a series of successful applications in the last several decades in the areas such as robotics, aircraft control, process control and estimation.

Controllers are usually implemented on actual systems using microprocessors. To implement a controller on a digital microprocessor, it is necessary to express it in terms of difference equations. This is accomplished using *digital* or *discrete-time* design. Unfortunately, most adaptive control results are available for continuous-time systems, where approaches such as the direct model-reference approach [1, 13] allow simultaneous proofs of tracking error and parameter error stability. Discrete-time adaptive control design is far more complex than continuous-time design, due primarily to the fact that discrete-time Lyapunov derivatives are *quadratic* in the state, not linear as in the continuous-time case. This has led to traditional techniques where the parameter identification problem is decoupled from the control problem using the so-called *certainty equivalence* assumption. In this approach, two separate proofs are essentially given, one for identification and one for control. Various elegant techniques for providing a posteriori proofs of overall convergence have been offered by Ljung and others [16]. Even recently, several authors (e.g., [9]) state that for discrete-time adaptive systems very few results exists, and one has to impose linear growth conditions on the nonlinearities to provide global stability.

Moreover, most adaptive control design algorithms are restricted to systems that are linear in the unknown parameters, and an often complex regression matrix must be computed for each system. Uncertainty in the regression matrix and unmodeled disturbances may cause the performance of the adaptive controllers to deteriorate considerably [1, 18]. In the recent adaptive and robust control literature [3, 20] there has been a tremendous amount of activity on a special control scheme known as “backstepping” [10]. When used under some mild assumptions, many existing robust and adaptive control techniques can be extended to a wide class of systems and applications [3, 20]. Several researchers have presented the backstepping control of nonlinear systems in continuous-time. Dawson et al. [3] have applied such techniques to induction motor control and in [20] the application of backstepping to power convertors is given. One of the problems with adaptive backstepping approaches is that certain functions must be expressed as linear in the unknown parameters and some tedious analysis is needed to determine a regression matrix.

Learning-based control using neural networks (NN) and has emerged as an alternative to adaptive control [6–8, 14, 19, 21]. These systems are nonlinear in the tunable parameters, and open-loop systems can be tuned using the essential backpropagation algorithm and its variants. For closed-loop feedback systems, however, it has not been fully understood until recently how to use the learning phenomenon for training. Research in NN for control applications is by now being pursued by several groups [6–8, 19, 21]. In [11], backstepping controller was designed for a class of nonlinear systems in continuous time using 2-layer feedforward neural networks. While some work in the design of discrete-time NN controllers has been performed [2, 6–8], there have been no results for backstepping control of nonlinear systems with multilayer NN in the discrete-time domain that employ *direct* techniques to estimate the controller parameters while guaranteeing boundedness of both the tracking and parameter estimation errors.

In [5–8, 20], adaptive control of a class of nonlinear systems in discrete time was presented with and without using neural networks. Although Lyapunov stability analysis and passivity properties were detailed, the analysis was limited to a class of nonlinear systems of the form $x(k+1) = f(x(k)) + g(x(k))u(k)$. In [2], CMAC NN are applied to control a robotic system by neglecting the actuator dynamics. Therefore, in this paper, the analysis is extended to a broader class of nonlinear systems where a backstepping controller is designed using CMAC NN in continuous time. On the other hand, multilayer feedforward NNs are used at each stage of the back stepping procedure to approximate certain nonlinear functions in discrete time.

A family of novel learning schemes is presented here for backstepping control of unknown nonlinear system that do not require preliminary off-line training. The traditional problems with adaptive control are overcome by using a *single Lyapunov function containing both the parameter identification error and the control errors*. This guarantees at once both stable identification and stable tracking. However, it leads to complex proofs for the discrete-time case where it is necessary to complete the square with respect to several different variables.

The use of a single Lyapunov function for tracking and estimation avoids the need for the certainty equivalence assumption. Along the way various other standard assumptions in adaptive control are also overcome, including persistence of excitation, linearity-in-the-parameters, and the need for tedious computation of a regression matrix.

2. Background

2.1. NEURAL NETWORK APPROXIMATION PROPERTY

A general function $f(x) \in C^{(S)}$ can be approximated using the n-layer neural network as

$$f\big(x(k)\big) = W_n^{\mathrm{T}}\varphi_n\big[W_n^{\mathrm{T}}\varphi_{n-1}\big[\dots\varphi_1\big(x(k)\big)\big]\big] + \varepsilon(k), \tag{1}$$

where $W_n^{\mathrm{T}}, W_{n-1}^{\mathrm{T}}, \dots, W_2^{\mathrm{T}}, W_1^{\mathrm{T}}$ are constant weights and $\varphi_i(k)$ denotes the vectors of activation functions at the instant k with the output layer activation functions being linear, and $\varepsilon(k)$ an NN functional reconstruction error vector. If there exist N_2 and constant ideal ('target') weights such that $\varepsilon = 0$ for all $x \in S$, then $f(x)$ is said to be *in the functional range of the NN*. In general, given a constant real number $\varepsilon_N \geqslant 0$, $f(x)$ is *within ε_N of the NN range* if there exist constant weights so that for all $x \in \mathbb{R}^n$, Equation (1) holds, for a sufficiently large number of hidden-layer neurons, with $\|\varepsilon\| \leqslant \varepsilon_N$.

In (1) for notational convenience, the vector of activation functions of the input layer at the instant k is denoted as $\varphi_1(k) = \varphi(x(k))$. Then, the vectors of hidden and output layer activation functions are denoted by

$$\varphi_{m+1}(k) = W_m^{\mathrm{T}}\varphi_m(k), \quad m = 1, \dots, n-1. \tag{2}$$

If the input layer weights are considered as identity matrix, then Equation (1) can be expressed as

$$f\big(x(k)\big) = W^{\mathrm{T}}(k)\varphi(k) + \varepsilon(k). \tag{3}$$

For Cerebellar Model Articulation Controller (CMAC) networks, the output is expressed as (3). Further, the activation functions are generally composed of splines [2]. The activation functions of a 2-D CMAC composed of second-order splines

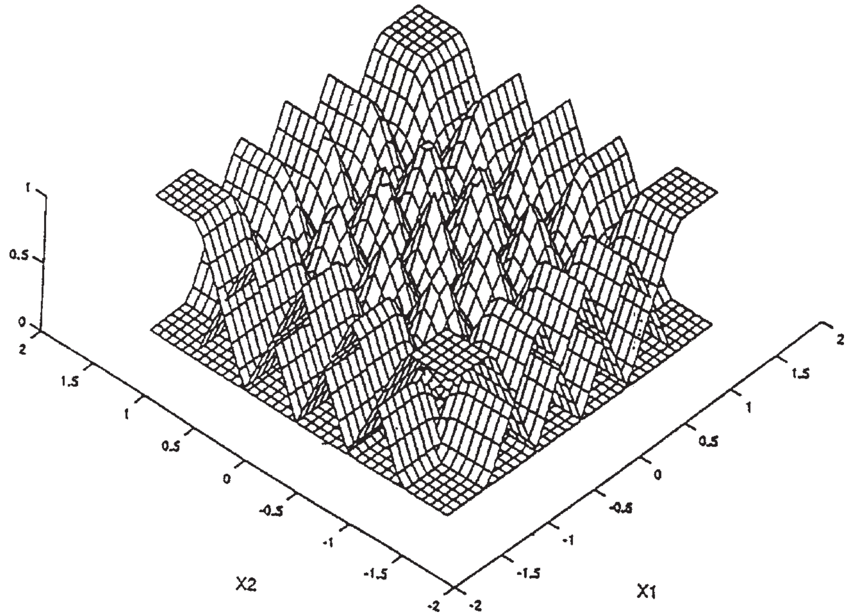

Figure 1. 2-D activation functions of a CMAC NN.

(e.g., triangle functions) are shown in Figure 1, with $L = 5 \times 5 = 25$ and L represent the number of activation functions. These activation functions of a CMAC NN are called receptive field functions in analogy with the optical receptor fields of the eye. An advantage of CMAC NN is that the receptive field functions based on splines have finite support so that they may be efficiently evaluated. An additional computational advantage is provided by the fact that higher-order splines may be computed recursively from lower-order splines. For suitable approximation using CMAC NN, the receptive field functions should form a basis.

DEFINITION 2.1. Let S be a compact simply connected set of $\mathbb{R}^n$, and $\phi(x)\colon\ S \to \mathbb{R}^{N_2}$ be integrable and bounded. Then $\phi(x)$ is said to provide a basis for $C^m(S)$ if

1. A constant function on S can be expressed as (3) for finite N_2.
2. The functional range of NN (3) is dense in $C^m(S)$ for countable N_2.

2.2. STABILITY OF SYSTEMS

DEFINITION 2.2. To formulate the discrete-time controller, the following stability notions are needed. Consider the nonlinear system given by

$$x(k+1) = f\big(x(k), u(k)\big), \qquad y(k) = h\big(x(k)\big), \tag{4}$$

where $x(k)$ is a state vector, $u(k)$ is the input vector and $y(k)$ is the output vector. The solution is said to be *uniformly ultimately bounded* (*UUB*) if for all $x(k_0) =$

x_0, there exists an $\varepsilon \geqslant 0$ and a number $N(\varepsilon, x_0)$ such that $\|x(k)\| \leqslant \varepsilon$ for all $k \geqslant k_0 + N$.

3. Continuous-time Nonlinear System

Consider the following class of nonlinear system given by

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= F_1(x_1, x_2) + G_1(x_1, x_2)x_3 + d_1, && (5) \\ \dot{x}_3 &= F_2(x_1, x_2) + G_2(x_1, x_2)u + d_2 && (6) \end{aligned}$$

with $u, x_1, x_2, x_3 \in \mathbb{R}^n$, $G_i(\cdot) \in \mathbb{R}^{n \times n}$, and $F_i \in \mathbb{R}^n$, $i = 1, 2$. It was assumed that $F_i(\cdot)$ are smooth unknown functions where $G_i(\cdot)$ are considered to be known and invertible. For instance, rigid robot with actuator dynamics [3], power converters [21] can be expressed in the above form. The invertibility property assumed for $G_i(\cdot)$ is stringent but efforts are underway to relax this assumption in the future.

The control objective is to make x_1 follow a certain desired trajectory $x_{1\mathrm{d}}$. For the case of robot control, x_1 denote joint angles.

3.1. CONTROLLER STRUCTURE AND ERROR DYNAMICS

Since the nonlinear systems (5) and (6) fall into the general class of systems described by backstepping approach, a backstepping controller will be designed. x_3 is treated as a fictitious control of subsystem (5). A CMAC NN is used to design the fictitious controller $x_{3\mathrm{d}}$. An actual controller u is then designed to force the error between the actual x_3 and $x_{3\mathrm{d}}$ as small as possible. Finally, an overall closed-loop stability analysis is performed using CMAC weight tuning algorithms.

Step 1: Design a fictitious controller x_3.

Define a filtered tracking error as

$$r = \dot{e}_1 + \lambda e_1, \tag{7}$$

where

$$e_1 = x_1 - x_{1\mathrm{d}} \tag{8}$$

and λ, a positive definite matrix, is chosen to control the rate of convergence of e_1. Differentiating (7) yields

$$\dot{r} = \ddot{e}_1 + \lambda \dot{e}_1. \tag{9}$$

Substituting (5) in (9) results in the following tracking error dynamics as

$$\dot{r} = F_1(\cdot) + G_1 x_{3\mathrm{d}} + \lambda \dot{e}_1 - \ddot{x}_{1\mathrm{d}} + G_1(x_3 - x_{3\mathrm{d}}) + d_1. \tag{10}$$

Select the fictitious controller

$$x_{3d} = G_1^{-1}\big[-\widehat{F}_1(\cdot) - \lambda \dot{e}_1 + \ddot{x}_1 - K_1 r\big] \tag{11}$$

with $K_1 > 0$ a design parameter, $\widehat{F}_1(\cdot)$ an estimate of $F_1(\cdot)$. Substituting (11) into subsystem (6) and rewriting the error dynamics as

$$\dot{r} = F_1(\cdot) - \widehat{F}_1(\cdot) - K_1 r + G_1\eta + d_1 \tag{12}$$

with $\eta = (x_3 - x_{3d})$. The usual approach in backstepping is to assume that the unknown parameters for F_1 [10] is linearly parametrizable (LP). Then standard adaptive techniques can be used to design a controller. Here a CMAC NN will be used to approximate this unknown function so that the LP assumption can be relaxed.

Step 2: Design of the actual control input u.

Let us define the error between x_3 and x_{3d} as η, that is

$$\eta = x_3 - x_{3d}. \tag{13}$$

The subsystem (5) then becomes

$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = F_1(x_1, x_2) + G_1(x_1, x_2)(x_{3d} + \eta) + d_1. \tag{14}$$

Differentiating η in (13) and using (6), Equation (13) yields

$$\dot{\eta} = F_3 + G_2 u + d_2, \tag{15}$$

where

$$\begin{aligned} F_3(\cdot) \;=\; & F_2(\cdot) + G_1^{-1}(\lambda + K_1)F_1(\cdot) + \\ & + \dot{G}_1^{-1}\big(\widehat{F}_1 + \lambda\dot{e}_1 - \ddot{x}_{1d} + K_1 r\big) + F_3^k(\cdot) \end{aligned} \tag{16}$$

with

$$\begin{aligned} F_3^k(\cdot) \;=\; & G_1^{-1}\lambda(G_1 x_{3d} + G_1\eta - \ddot{x}_{1d}) + \\ & + G_1^{-1}K_1(\lambda\dot{e}_1 + G_1 x_{3d} + G_1\eta - \ddot{x}_{1d}) - G_1^{-1}x_{3d} + G_1^{-1}\dot{\widehat{F}}_1(\cdot). \end{aligned} \tag{17}$$

To make η as small as possible, the following control input u is selected as

$$u = G_2^{-1}\big[-\widehat{F}_3 - K_2\eta - G_1^{\mathrm{T}} r\big], \tag{18}$$

where $K_2 > 0$ is a design parameter and $\widehat{F}_3(\cdot)$ an estimate of the unknown function $F_3(\cdot)$. A CMAC NN will be employed to approximate the unknown function $F_3(\cdot)$ so that the linearity in the parameter assumption can be relaxed. Note further that the term $-G_1^{\mathrm{T}} r$ is added in Equation (18). This term is required to cancel the effect of $G_1 r$ in (12) so that the closed-loop stability can be proven. In the next subsection, two CMAC NN are considered one for approximating $F_1(\cdot)$ and the other to approximate $F_3(\cdot)$.

Step 3: Closed-loop stability analysis using CMAC NN.

The next step is to select appropriate weight tuning updates so that the Lyapunov's stability analysis can be performed for the closed-loop system. These updates are presented in the next section.

3.2. CMAC NN

Assume that the nonlinear functions $F_1(\cdot)$ and $F_3(\cdot)$ in (12) and (16), respectively, are approximated by CMAC NN for some constant known ideal weights $W_i,\ i = 1, 2$, which is,

$$F_1 = W_1^{\mathrm{T}}\phi_1 + \varepsilon_1, \qquad F_3 = W_2^{\mathrm{T}}\phi_2 + \varepsilon_2, \tag{19}$$

where

$$\|\varepsilon_1\| < \varepsilon_{N1}, \qquad \|\varepsilon_2\| < \varepsilon_{N2}, \tag{20}$$

and $\phi_1(\cdot)$ and $\phi_2(\cdot)$ provide suitable basis functions for the CMAC NNs. Define the NN functional estimate of $F_1(\cdot)$ in (5) as

$$\widehat{F}_1 = \widehat{W}_1^{\mathrm{T}}\phi_1 \tag{21}$$

with $\widehat{W}_1$ the current NN weight estimates provided by the tunning algorithms. Then the error dynamics (12) becomes

$$\dot{r} = \widetilde{W}_1^{\mathrm{T}}\phi_1 - K_1 r + G_1\eta + \varepsilon_1 + d_1. \tag{22}$$

Similarly, define the NN functional estimate $\widehat{F}_3(\cdot)$ in (15) by

$$\widehat{F}_3 = \widehat{W}_2^{\mathrm{T}}\phi_2. \tag{23}$$

The error dynamics for η in (15) is given by

$$\dot{\eta} = \widetilde{W}_2^{\mathrm{T}}\phi_2 - K_2\eta - G_1^{\mathrm{T}} r + \varepsilon_2 + d_2. \tag{24}$$

Note the presence of $G_1\eta$ in Equation (22) and $-G_1^{\mathrm{T}} r$ in Equation (24). This implies that the error dynamics (22) and (24) are coupled and therefore they can be combined. Then, using appropriate weight tuning laws the closed-loop stability analysis can be done. This analysis is discussed in the next section. The following assumptions are needed in order to proceed.

ASSUMPTION 3.1. The ideal weights are bounded by known positive values so that

$$\|W_1\|_F \leqslant W_{1M}, \tag{25}$$

$$\|W_2\|_F \leqslant W_{2M}, \tag{26}$$

or

$$\|Z_F\|_F \leqslant Z_M, \tag{27}$$

where $Z = \mathrm{diag}[W_1, W_2]$ with Z_M known.

Define

$$\begin{aligned}
&\omega = \left[r^{\mathrm{T}}\ \eta^{\mathrm{T}}\right]^{\mathrm{T}}, \qquad \widetilde{W}_i = W_i - \widehat{W}_i, \quad i = 1, 2,\\
&\widetilde{Z} = \mathrm{diag}[\widetilde{W}_1, \widetilde{W}_2], \qquad d = [d_1, d_2]^{\mathrm{T}},\\
&K = \mathrm{diag}[K_1, K_2], \qquad \phi = \left[\phi_1^{\mathrm{T}}, \phi_2^{\mathrm{T}}\right]^{\mathrm{T}}, \qquad \varepsilon = \left[\varepsilon_1^{\mathrm{T}}, \varepsilon_2^{\mathrm{T}}\right]^{\mathrm{T}}
\end{aligned} \tag{28}$$

and

$$G_1 = \begin{bmatrix} 0 & G_1 \\ -G_1^{-1} & 0 \end{bmatrix}. \tag{29}$$

The error dynamics (22) and (24) are combined and rewritten as

$$\dot{\omega} = -K\omega + \tilde{f}(\cdot) + G\omega + d + \varepsilon, \tag{30}$$

where $\hat{f} = [\widehat{F}_1(\cdot)\ \widehat{F}_5(\cdot)]^{\mathrm{T}}$ and $\|\varepsilon\| \leqslant \varepsilon_N$.

ASSUMPTION 3.2. The desired trajectory $x_{1\mathrm{d}}$ and its derivatives up to the third order are bounded. In other words,

$$\left\|[x_{1\mathrm{d}}\ \dot{x}_{1\mathrm{d}}\ \ddot{x}_{1\mathrm{d}}]^{\mathrm{T}}\right\| \leqslant q_B. \tag{31}$$

LEMMA 3.1. *It is easy to show that for each time t, $x(t) = [x_1\ x_2\ x_3]^{\mathrm{T}}$ is bounded by*

$$\|x\| \leqslant c_1 + c_2\|\omega\| \leqslant q_B + c_0\left\|\omega(0)\right\| + c_2\|\omega\| \tag{32}$$

for computable positive constants c_0, c_1, c_2.

Figure 2 shows the continuous-time version of the backstepping controller structures for the systems (5) and (6). It is necessary to discuss the effect of initial conditions on the stability analysis.

Initial Tracking Errors and Initial CMAC NN weights

It is required to determine how to tune the CMAC NN weights to yield guaranteed closed-loop stability. Since the NN approximation property holds on a compact set, one must define an allowable initial condition set as follows. Let the NN approximating property hold for the function $f(x)$ given in (30) with a given accuracy of ε_N in (20) for all x inside the ball of radius $b_x > q_B$. Define the set of allowable initial tracking errors as

$$S_\omega = \left\{\omega \mid \|\omega\| < \frac{(b_x - q_B)}{(c_0 + c_2)}\right\}. \tag{33}$$

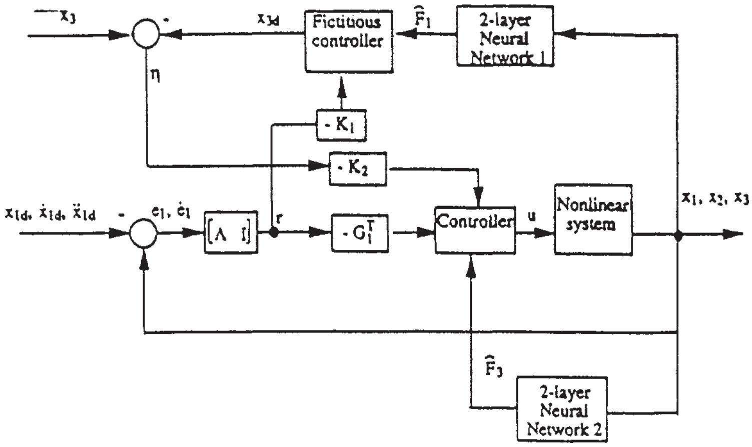

Figure 2. CMAC NN backstepping control structure.

Note that the approximation accuracy of the NN determines the size of S_ω. For a larger NN (i.e., more hidden layer units), ε_N is small for a larger radius b_x. Thus, the allowed initial condition set S_ω is larger. On the other hand, a more active desired trajectory (e.g., containing higher frequency components) results in a larger accelerations $\ddot{x}_1$, which yields a larger bound q_B, thereby decreasing S_ω. It is important to note the dependence of S_ω on the PD design ratio λ – both c_0 and c_2 depend upon λ.

A key feature of our initial condition requirement is its independence of the NN initial weights. This is in contrast to other techniques in the literature where the proofs of stability depend upon some initial stabilizing weights, which is very difficult to do.

Weight Initialization and On-Line Learning

For the NN control schemes derived in this paper, there is no preliminary off-line learning phase. The weights are simply initialized at zero, for then the Figure 2 shows that the controller is just a PD controller. Standard results indicate that for controlling systems of the form (5), a PD controller may result in bounded tracking errors if K is large enough and $\hat{f}(\cdot)$ is suitably approximated. Then, the closed-loop system remains stable until the NN begins to learn. Therefore the weights are tuned on-line as the system tracks the desired trajectory. The tracking performance improves as the NN learns the unknown functions, $F_1(\cdot)$ and $F_3(\cdot)$. Note that this controller structure results in a modular design for industrial applications wherein the underlying conventional controller is augmented with a CMAC NN to enhance the overall system performance. The receptive field functions $\phi(x)$ for the CMAC NN are selected in such a way that they form a set of basis for the unknown nonlinear functions $F_1(\cdot)$ and $F_3(\cdot)$.

3.3. WEIGHT UPDATES

In this section, the weight updates are presented for CMAC NNs. These novel weight tuning updates guarantee the boundedness of both weights and tracking errors. These updates contain additional terms to relax the persistency of excitation condition. Let the weight tuning updates are provided by

$$\dot{\widehat{W}}_1 = \delta_1 \phi_1 r^{\mathrm{T}} - k_v \delta_1 \|\omega\| \widehat{W}_1 \tag{34}$$

and

$$\dot{\widehat{W}}_2 = \delta_2 \phi_2 \eta^{\mathrm{T}} - k_v \delta_2 \|\omega\| \widehat{W}_2 \tag{35}$$

with any constant matrices $\delta_1 = \delta_1^{\mathrm{T}},\ \delta_2 = \delta_2^{\mathrm{T}}$ that are invertible and k_v being a positive scalar constant.

THEOREM 3.1. *Let the desired trajectory $x_{1\mathrm{d}}$ and its derivatives up to the third order be bounded and let the ideal weights are bounded above by Z_M. Select the control input as* (18) *with CMAC NN weight tuning provided by* (34) *and* (35) *and gain satisfying the condition*

$$K_{\min} > \frac{(k_v Z_M^2/4 + \varepsilon_N + d_M)(c_0 + c_2)}{(b_x - q_B)} \tag{36}$$

on the minimum singular value of K. Let the neural network functional reconstruction error ε be bounded by ε_N and disturbances are bounded by d_M. Then the errors $r(t)$, $\eta(t)$ and CMAC NN weight estimates are UUB with practical bounds given by the right-hand sides of (41) *and* (42). *Further, the tracking errors $r(t)$ and $\eta(t)$ can be kept small by increasing the gains K in* (36).

Proof. The error dynamics (30) is further expressed as

$$\dot{\omega} = -K\omega + \widetilde{Z}^{\mathrm{T}}\phi + G\omega + \varepsilon + d. \tag{37}$$

Let the NN approximation property (3) hold for the functions $F_1(\cdot)$ and $F_3(\cdot)$ given in (19) with a maximum bound of ε_N for all x in the compact set $S_x \equiv \{x \mid \|x\| < b_x\}$ with $b_x > q_B$. Let $\omega(0) \in S_\omega$. Then the approximation property holds at time $t = 0$.

Consider the following Lyapunov function candidate

$$V = \frac{1}{2}\omega^{\mathrm{T}}\omega + \frac{1}{2}\mathrm{tr}\big(\widetilde{Z}^{\mathrm{T}}\delta^{-1}\widetilde{Z}\big), \tag{38}$$

where $\delta = \mathrm{diag}[\delta_1, \delta_2]$. Differentiating (38) and using (37), the first derivative of the Lyapunov function can be written as

$$\dot{V} = -\omega^{\mathrm{T}}K\omega + \omega^{\mathrm{T}}(\varepsilon + d) + k_v\|\omega\|\mathrm{tr}\big(\widetilde{Z}^{\mathrm{T}}\big(Z - \widetilde{Z}\big)\big). \tag{39}$$

Since $\mathrm{tr}(\widetilde{Z}(Z-\widetilde{z})) = \langle \widetilde{Z}, Z\rangle_F - \|\widetilde{Z}\|_F^2\|Z\|_F - \|\widetilde{Z}\|_F^2$, Equation (39) can be expressed as

$$\begin{aligned}\dot{V} &\leqslant -K_{\min}\|\omega\|^2 + k_v\|\omega\|\left\|\widetilde{Z}\right\|\left(Z_M - \left\|\widetilde{Z}\right\|_F\right) + (\varepsilon_N + d_M)\|\omega\| \\ &= -\|\omega\|\left[K_{\min}\|\omega\| + k_v\left\|\widetilde{Z}\right\|_F\left(\|\widetilde{Z}\|_F - Z_M\right) - (\varepsilon_N + d_M)\right] \end{aligned} \tag{40}$$

which is negative as long as the term in square braces is positive. Therefore, $\dot{V}$ is guaranteed negative as long as

$$\|\omega\| > \frac{k_v Z_M^2/4 + (\varepsilon_N + d_M)}{K_{\min}} \equiv b_\omega \tag{41}$$

or

$$\left\|\widetilde{Z}\right\| > \frac{Z_M}{2} + \sqrt{\frac{Z_M^2}{4} + \frac{(\varepsilon_N + d_M)}{k_v}} \equiv b_z. \tag{42}$$

Thus, $\dot{V}$ is negative outside a compact set. Selecting the gains according to (36) ensures that the compact set defined by $\|\omega\| \leqslant b_\omega$ is contained in S_ω, so that the approximating property holds through out. This demonstrates the UUB of both $\|\omega\|$ and $\|\widetilde{Z}\|$. □

From Equations (41) and (42), the CMAC NN reconstruction error bound ε_N and the disturbance bound d_M increase the bounds of $\|\omega\|$ and $\|\widetilde{Z}\|_F$ in a very interesting way. For instance, the tracking error can be made smaller by increasing the control gain $K_{\min}$. The parameter k_v offers a design tradeoff between the relative magnitudes of $\|\omega\|$ and $\|\widetilde{Z}\|_F$, a smaller k_v yields a smaller $\|\omega\|$ and a larger $\|\widetilde{Z}\|_F$, and vice versa. Further note that the PE condition is not needed to establish these bounds for the tracking and weight estimation errors. Finally, note that two-layer CMAC NN were employed to design the backstepping controller in continuous time. If one choses to use a multilayer NN, then the derivative of $\hat{\phi}(\cdot)$ is required and it cannot be calculated. This problem can be avoided for the case of discrete-time control. However, it is very difficult to show the Lyapunov stability analysis for discrete-time systems. In fact, in the next section, a backstepping controller is designed in discrete time using multilayer NNs.

4. Discrete-time Nonlinear System Description

Consider the following class of discrete-time nonlinear system, to be controlled, given in multivariable form as

$$\begin{aligned} x_1(k+1) &= x_2(k), \\ x_2(k+1) &= F_1\big(x_1(k), x_2(k)\big) + G_1\big(x_1(k), x_2(k)\big)x_3(k) + d_1(k), \qquad (43) \\ x_3(k+1) &= F_2\big(x_1(k), x_2(k)\big) + G_2\big(x_1(k), x_2(k)\big)u(k) + d_2(k) \qquad (44) \end{aligned}$$

with state $x_i(k) \in \mathbb{R}^n$, $i = 1, \ldots, 3$, $F_1(\cdot), F_2(\cdot) \in \mathbb{R}^n$, $G_1(\cdot), G_2(\cdot) \in \mathbb{R}^{n\times n}$, and control $u(k) \in \mathbb{R}^n$. The nonlinear functions $F_i(\cdot)$ are assumed unknown whereas

$G_i(\cdot)$ are assumed known and invertible. Several physical nonlinear systems can be expressed in the above form including rigid robot with actuator dynamics [3] and power convertors [21].

4.1. CONTROLLER STRUCTURE AND ERROR DYNAMICS

The control objective is to make $x_1(k)$ follow a desired trajectory $x_{1\mathrm{d}}(k)$. As explained in the previous section, in the backstepping controller design, the state $x_3(k)$ is treated as a fictitious control to subsystem (43). A multilayer NN is employed to design the fictitious controller $x_{3\mathrm{d}}$. Then, an actual controller $u(k)$ is selected to force the error between the actual $x_3(k)$ and $x_{3\mathrm{d}}(k)$ as small as possible, using the second NN. Finally, an overall closed-loop stability analysis is carried out using the weight tuning algorithms.

Given a desired trajectory $x_{1\mathrm{d}}(k)$ and its delayed values, define the tracking error as

$$e_1(k) = x_1(k) - x_{1\mathrm{d}}(k), \tag{45}$$

and the filtered tracking error, $r(k) \in \mathbb{R}^n$,

$$r(k) = e_1(k+1) + \lambda e_1(k), \tag{46}$$

where $e_1(k+1)$ is the future value for the error $e_1(k)$ and λ is a constant positive definite matrix selected to control the rate of convergence of $e_1(k)$. Equation (46) can be expressed as

$$r(k+1) = e_2(k+1) + \lambda e_2(k). \tag{47}$$

Assume that $G_i(\cdot) \in C^\infty[U]$, i.e., a smooth function $U \to R$, so that the Taylor series expansion of $G_i(\cdot)$ exists and $x(k) = [x_1(k), x_2(k)]^{\mathrm{T}}$. One can derive that $\|x(k)\| \leqslant d_{01} + d_{11}\|r(k)\|$. Using the bound on $x(k)$ and expressing $G_i(\cdot)$ using Taylor series, similar to $\|x(k)\|$, results in

$$\big\|G_1(\cdot)\big\| = C_{01} + C_{11}\big\|x(k)\big\| \tag{48}$$

and

$$\big\|G_2(\cdot)\big\| = C_{02} + C_{12}\big\|x(k)\big\|. \tag{49}$$

By applying the assumption that $G_i = C_{0i}$, Equations (43) and (44) are rewritten as

$$\begin{aligned} x_1(k+1) &= x_2(k), \\ x_2(k+1) &= F_1\big(x_1(k), x_2(k)\big) + C_{01}x_3(k) + d_1(k), \end{aligned} \tag{50}$$

$$x_3(k+1) = F_2\big(x_1(k), x_2(k)\big) + C_{02}u(k) + d_2(k). \tag{51}$$

Using Equation (50) in Equation (47), the dynamics of the MIMO system (47) can be written in terms of the filtered tracking error as

$$\begin{aligned} r(k+1) &= F_1\big(x_1(k), x_2(k)\big) + C_{01}x_{3d} + d_1(k) - \\ &\quad - x_{2d}(k+1) + C_{01}\big(x_3(k) - x_{3d}(k)\big) + \lambda e_2(k). \end{aligned} \tag{52}$$

Select the following fictitious controller

$$x_{3d}(k) = C_{01}^{-1}\big[-\widehat{F}_3\big(x_1(k), x_2(k)\big) + x_{2d}(k+1) - \lambda e_2(k) + k_{v1}r(k)\big] \tag{53}$$

with $k_{v1} > 0$ a design parameter, $\widehat{F}_3(\cdot)$ the estimate of $F_3(\cdot)$. Substituting (53) into subsystem (50) yields the error dynamics

$$r(k+1) = k_{v1}r(k) + \widetilde{F}_3(\cdot) + C_{01}\eta(k) + d_1(k) \tag{54}$$

with $\eta(k)$ defined in (55). The usual approach in the adaptive backstepping is to assume that $F_3(\cdot)$ can be expressed as linear in the unknown parameters. Here, a multilayer NN will be employed to approximate the nonlinear function $F_3(\cdot)$.

Step 2: Design of actual control input $u(k)$.

After the fictitious controller $x_{3d}(k)$ is designed, a way need to be found to realize it. Define the error

$$\eta = x_3(k) - x_{3d}(k). \tag{55}$$

The error system (55) can be rewritten as

$$\begin{aligned} \eta(k+1) &= F_2\big(x_1(k), x_2(k)\big) + C_{02}u(k) - C_{01}^{-1}(k)\big[-\widehat{F}_1(k) + \\ &\quad + x_{2d}(k+2) - \lambda e_2(k+1) + k_{v1}r(k+1)\big] + d_2(k). \end{aligned} \tag{56}$$

Equation (56) can be rewritten as

$$\eta(k+1) = F_3(\cdot) + C_{02}^{-1}u(k) + d_2(k). \tag{57}$$

Define the control input $u(k)$ as

$$u(k) = C_{02}^{-1}\big[-\widehat{F}_3(\cdot) + k_{v2}\eta(k) + C_{01}r(k)\big] \tag{58}$$

with a gain matrix k_{v2}, and $\widehat{F}_3(\cdot)$ an estimate of the unknown nonlinear function $F_3(\cdot)$. Then (56) becomes

$$\eta(k+1) = k_{v2}\eta(k) + \widetilde{F}_3(\cdot) + C_{01}r(k) + d_2(k). \tag{59}$$

Equations (54) and (59) can be combined as

$$\omega(k+1) = k_v\omega(k) + \hat{f}\big(x_1(k), x_2(k)\big) + d(k), \tag{60}$$

where $\omega = [r(k)\ \eta(k)]^{\mathrm{T}}$, $d(k) = [d_1\ d_2]^{\mathrm{T}}$, $\tilde{f}(\cdot) = [\widetilde{F}_1(\cdot)\ \widetilde{F}_3(\cdot)]^{\mathrm{T}}$, and

$$k_v = \begin{bmatrix} k_{v1} & C_{01} \\ C_{01} & k_{v2} \end{bmatrix}. \tag{61}$$

Note that the error system is driven by the functional estimation error. Here, multilayer NNs are employed to estimate the nonlinear functions $f(\cdot)$. In this paper, multilayer NNs are used in discrete-time to provide the estimate $\hat{f}(\cdot)$ for the error systems (60). This error system is used to focus on selecting discrete-time NN tuning algorithms that guarantee the stability of the tracking error $\omega(k)$ and weight estimates.

5. Multilayer NN Controller Design

Two three-layer NN are considered for the error system (60) and stability analysis is carried out for the closed-loop system. In this section, stability analysis by Lyapunov's direct method is performed for a family of multilayer NN weight tuning algorithms using a delta rule in each layer. These weight tuning paradigms guarantee boundedness of all closed-loop signals without the requirement of the PE condition.

Assume that there exists some constant ideal weights W_{11}, W_{12}, W_{21}, W_{22} and W_{31}, W_{32} for three-layer NN so that the nonlinear function in (52) and (58) can be written as

$$F_1(\cdot) = W_{31}^{\mathrm{T}}\varphi_{31}\big[W_{21}^{\mathrm{T}}\varphi_{21}\big[W_{11}^{\mathrm{T}}\varphi_{11}\big(x(k)\big)\big]\big] + \varepsilon_1(k) \tag{62}$$

and

$$F_3(\cdot) = W_{32}^{\mathrm{T}}\varphi_{32}\big[W_{22}^{\mathrm{T}}\varphi_{22}\big[W_{12}^{\mathrm{T}}\varphi_{12}\big(x(k)\big)\big]\big] + \varepsilon_2(k) \tag{63}$$

with $\|\varepsilon_1(k)\| < \varepsilon_{1N}$, and $\|\varepsilon_2(k)\| < \varepsilon_{2N}$, with the bounding constants known.

For notational convenience define the matrix of all the ideal weights as

$$Z_1 = \mathrm{diag}\{W_{11}, W_{21}, W_{31}\} \quad \text{and} \quad Z_2 = \mathrm{diag}\{W_{12}, W_{22}, W_{32}\}.$$

Similarly, for notational convenience define the matrix of ideal weights as $W_1 = \mathrm{diag}\{W_{11}, W_{12}\}$, $W_2 = \mathrm{diag}\{W_{21}, W_{22}\}$ and $W_3 = \mathrm{diag}\{W_{31}, W_{32}\}$, and $Z = \mathrm{diag}\{Z_1, Z_2\}$. Define $\varphi_1(k) = \mathrm{diag}\{\varphi_{11}(k), \varphi_{12}(k)\}$, $\varphi_2(k) = \mathrm{diag}\{\varphi_{21}(k), \varphi_{22}(k)\}$ and $\varphi_3(k) = \mathrm{diag}\{\varphi_{31}(k), \varphi_{32}(k)\}$. Then some bounding assumption can be stated.

ASSUMPTION 5.1. The ideal weights are bounded by known positive values so that $\|W_{11}\| \leqslant W_{11\max}$, $\|W_{21}\| \leqslant W_{21\max}$, and $\|W_{31}\| \leqslant W_{31\max}$, or $\|Z_1\| \leqslant Z_{1\max}$.

ASSUMPTION 5.2. The ideal weights are bounded by known positive values so that $\|W_{12}\| \leqslant W_{12\max}$, $\|W_{22}\| \leqslant W_{22\max}$, and $\|W_{32}\| \leqslant W_{32\max}$, or $\|Z_2\| \leqslant Z_{2\max}$.

ASSUMPTION 5.3. The ideal weights are bounded by known positive values so that $\|W_1\| \leqslant W_{1\max}$, $\|W_2\| \leqslant W_{2\max}$, and $\|W_3\| \leqslant W_{3\max}$, or $\|Z\| \leqslant Z_{\max}$.

5.1. STRUCTURE OF THE NN CONTROLLER

Assume that the nonlinear functions $F_1(\cdot)$ and $F_3(\cdot)$ can be represented by two three-layer NN for some constant ideal weights as presented above. Define the NN functional estimate by

$$\widehat{F}_1(\cdot) = \widehat{W}_{31}^{\mathrm{T}}(k)\varphi_{31}\big(\widehat{W}_{21}^{\mathrm{T}}(k)\varphi_{21}\big(\widehat{W}_{11}^{\mathrm{T}}(k)\varphi_{11}\big(x_1(k), x_2(k)\big)\big)\big), \tag{64}$$

$$\widehat{F}_3(\cdot) = \widehat{W}_{32}^{\mathrm{T}}(k)\varphi_{32}\big(\widehat{W}_{22}^{\mathrm{T}}(k)\varphi_{22}\big(\widehat{W}_{12}^{\mathrm{T}}(k)\varphi_{12}\big(x(k), u(k)\big)\big)\big) \tag{65}$$

with $\widehat{W}_{31}(k)$, $\widehat{W}_{21}(k)$, $\widehat{W}_{32}(k)$, $\widehat{W}_{22}(k)$, and $\widehat{W}_{11}(k)$, $\widehat{W}_{12}(k)$ be the current value of the weights. The vector of input layer activation functions is given by $\widehat{\varphi}_{11}(k) = \varphi_{11}(k) = \varphi_{11}(x(k))$ and $\widehat{\varphi}_{12}(k) = \varphi_{12}(k) = \varphi_{12}(x(k))$. Then the vector of activation functions of the hidden and output layer with the actual weights at the instant k is denoted by

$$\widehat{\varphi}_{(m+1)1}(k) = \varphi\big(\widehat{W}_{m1}^{\mathrm{T}}\widehat{\varphi}_{m1}(k)\big), \quad m = 1, \ldots, n-1, \tag{66}$$

$$\widehat{\varphi}_{(m+1)2}(k) = \varphi\big(\widehat{W}_{m2}^{\mathrm{T}}\widehat{\varphi}_{m2}(k)\big), \quad m = 1, \ldots, n-1. \tag{67}$$

Fact 2. For a given trajectory, the activation functions are bounded by known positive values so that

$$\begin{aligned}
&\big\|\widehat{\varphi}_{11}(k)\big\| \leqslant \varphi_{11\max}, \qquad \big\|\widehat{\varphi}_{21}(k)\big\| \leqslant \varphi_{21\max}, \quad \text{and} \quad \big\|\widehat{\varphi}_{31}(k)\big\| \leqslant \varphi_{31\max},\\
&\big\|\widehat{\varphi}_{12}(k)\big\| \leqslant \varphi_{12\max}, \qquad \big\|\widehat{\varphi}_{22}(k)\big\| \leqslant \varphi_{22\max}, \quad \text{and} \quad \big\|\widehat{\varphi}_{32}(k)\big\| \leqslant \varphi_{32\max}.
\end{aligned}$$

For convenience define

$$\widehat{\varphi}_1(k) = \mathrm{diag}\{\widehat{\varphi}_{11}(k), \widehat{\varphi}_{12}(k)\}, \qquad \widehat{\varphi}_2(k) = \mathrm{diag}\{\widehat{\varphi}_{21}(k), \widehat{\varphi}_{22}(k)\}$$

and the output layer activation functions as $\widehat{\varphi}_3(k) = \mathrm{diag}\{\widehat{\varphi}_{31}(k), \widehat{\varphi}_{32}(k)\}$. The error in the weights or weight estimation errors are given by

$$\begin{aligned}
&\widetilde{W}_{31}(k) = W_{31} - \widehat{W}_{31}(k), \qquad \widetilde{W}_{21}(k) = W_{21} - \widehat{W}_{21}(k),\\
&\widetilde{W}_{11}(k) = W_{11} - \widehat{W}_{11}(k), \qquad \widetilde{Z}_1(k) = Z_1 - \widehat{Z}_1(k),
\end{aligned} \tag{68}$$

$$\begin{aligned}
&\widetilde{W}_{32}(k) = W_{32} - \widehat{W}_{32}(k), \qquad \widetilde{W}_{22}(k) = W_{22} - \widehat{W}_{22}(k),\\
&\widetilde{W}_{12}(k) = W_{12} - \widehat{W}_{12}(k), \qquad \widetilde{Z}_2(k) = Z_2 - \widehat{Z}_2(k),
\end{aligned} \tag{69}$$

$$\begin{aligned}
&\widetilde{W}_1(k) = W_1 - \widehat{W}_1(k), \qquad \widetilde{W}_2(k) = W_2 - \widehat{W}_2(k),\\
&\widetilde{W}_3(k) = W_3 - \widehat{W}_3(k), \qquad \widetilde{Z}(k) = Z - \widehat{Z}(k),
\end{aligned} \tag{70}$$

where

$$\begin{aligned}\widehat{Z}_1(k) &= \operatorname{diag}\{\widehat{W}_{11}(k), \widehat{W}_{21}(k), \widehat{W}_{31}(k)\},\\ \widehat{Z}_2(k) &= \operatorname{diag}\{\widehat{W}_{12}(k), \widehat{W}_{22}(k), \widehat{W}_{32}(k)\},\end{aligned}$$

and the hidden-layer output errors are defined as

$$\widetilde{\varphi}_{21}(k) = \varphi_{21}(k) - \widehat{\varphi}_{21}(k), \qquad \widetilde{\varphi}_{31}(k) = \varphi_{31}(k) - \widehat{\varphi}_{31}(k), \tag{71}$$

$$\widetilde{\varphi}_{22}(k) = \varphi_{22}(k) - \widehat{\varphi}_{22}(k), \qquad \widetilde{\varphi}_{32}(k) = \varphi_{32}(k) - \widehat{\varphi}_{32}(k) \tag{72}$$

and

$$\widetilde{\varphi}_{2}(k) = \varphi_{2}(k) - \widehat{\varphi}_{2}(k), \qquad \widetilde{\varphi}_{3}(k) = \varphi_{3}(k) - \widehat{\varphi}_{3}(k). \tag{73}$$

Select the following fictitious controller

$$x_{3\mathrm{d}}(k) = C_{01}^{-1}\big[-\widehat{W}_{31}^{\mathrm{T}}(k)\widehat{\varphi}_{31}(k) + x_{2\mathrm{d}}(k+1) - \lambda e_2(k) + k_{v1}r(k)\big]. \tag{74}$$

Define the control input $u(k)$ as

$$u(k) = C_{02}^{-1}\big[-\widehat{W}_{32}^{\mathrm{T}}(k)\widehat{\varphi}_{32}(k) + k_{v2}r(k) + C_{01}r(k)\big], \tag{75}$$

where the functional estimates (74) and (75) are provided by two three-layer NN and denoted in Equations (74) and (75) by $\widehat{W}_{31}^{\mathrm{T}}(k)\widehat{\varphi}_{31}(k)$ and $\widehat{W}_{32}^{\mathrm{T}}(k)\widehat{\varphi}_{32}(k)$, respectively. Then, the closed-loop filtered dynamics (60) become

$$\omega(k+1) = k_v\omega(k) + \bar{e}_i(k) + W_3^{\mathrm{T}}\widetilde{\varphi}_3(k) + \varepsilon(k) + d(k), \tag{76}$$

where the identification error is defined by

$$\bar{e}_i(k) = \widetilde{W}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k). \tag{77}$$

The outputs and inputs of the plant are processed through a series of delays to obtain the past values of the outputs and inputs, and fed as current inputs to the multilayer NN so that the nonlinear function $f(\cdot)$ in Equation (60) can be suitably approximated. The next step is to determine the weight updates so that the tracking performance of the closed-loop error dynamics is guaranteed.

5.2. MULTILAYER NEURAL NETWORK WEIGHT UPDATES

A family of NN weight tuning paradigms that guarantee the stability of the closed-loop system (76) is presented in this section. It is required to demonstrate that the tracking error ω is suitably small and that the NN weights $\widehat{W}_3^{\mathrm{T}}(k)$, $\widehat{W}_2^{\mathrm{T}}(k)$, $\widehat{W}_1^{\mathrm{T}}(k)$, remain bounded, for then the control $u(k)$ is bounded. The results of this section present tuning algorithms that overcome the need for PE in the case of multilayered

NN. The next theorem proves the stability of the NN controllers, which have tuning algorithms augmented to avoid PE.

THEOREM 5.1. *Let the desired trajectory $x_{nd}(k)$ be bounded and the NN functional reconstruction error and the disturbance bounds, ε_N, d_M, respectively, be known constant. Take the control input as in Equation* (75) *for the system Equation* (43). *Let the weight tuning provided for the input and hidden layers as*

$$\begin{aligned}\widehat{W}_1(k+1) &= \widehat{W}_1(k) - \alpha_1\widehat{\varphi}_1(k)\big[\hat{y}_1(k) + B_1 k_v \omega(k)\big]^{\mathrm{T}} - \\ &\quad - \Gamma\big\|I - \alpha_1\widehat{\varphi}_1(k)\widehat{\varphi}_1^{\mathrm{T}}(k)\big\|\widehat{W}_1(k),\end{aligned} \tag{78}$$

$$\begin{aligned}\widehat{W}_2(k+1) &= \widehat{W}_2(k) - \alpha_2\widehat{\varphi}_2(k)\big[\hat{y}_2(k) + B_2 k_v \omega(k)\big]^{\mathrm{T}} - \\ &\quad - \Gamma\big\|I - \alpha_2\widehat{\varphi}_2(k)\widehat{\varphi}_2^{\mathrm{T}}(k)\big\|\widehat{W}_2(k).\end{aligned} \tag{79}$$

Let the weight update for the output layer given by either

$$\text{(a)}\quad \begin{aligned}\widehat{W}_3(k+1) &= \widehat{W}_3(k) + \alpha_3\widehat{\varphi}_3(k)\bar{f}^{\mathrm{T}}(k) - \\ &\quad - \Gamma\big\|I - \alpha_3\widehat{\varphi}_3(k)\widehat{\varphi}_3^{\mathrm{T}}(k)\big\|\widehat{W}_3(k),\end{aligned} \tag{80}$$

$$\text{(b)}\quad \begin{aligned}\widehat{W}_3(k+1) &= \widehat{W}_3(k) + \alpha_3\widehat{\varphi}_3(k)\omega^{\mathrm{T}}(k+1) - \\ &\quad - \Gamma\big\|I - \alpha_3\widehat{\varphi}_3(k)\widehat{\varphi}_3^{\mathrm{T}}(k)\big\|\widehat{W}_3(k)\end{aligned} \tag{81}$$

with $\Gamma > 0$ a design parameter. Then the filtered tracking error $\omega(k)$ and the NN weight estimates $\widehat{W}_1(k)$, $\widehat{W}_2(k)$, and $\widehat{W}_3(k)$ are UUB provided the following conditions hold:

$$\text{(1)}\quad \alpha_i\big\|\widehat{\varphi}_i(k)\big\|^2 < \begin{cases} 2, & i = 1, 2, \\ 1, & i = 3, \end{cases} \tag{82}$$

$$\text{(2)}\quad 0 < \Gamma < 1, \tag{83}$$

$$\text{(3)}\quad k_{v\max} < \frac{1}{\sqrt{\overline{\sigma}}}, \tag{84}$$

where $\overline{\sigma}$ is given by

$$\overline{\sigma} = \beta_3 + \sum_{i=1}^{2}\beta_i\kappa_i^2 \tag{85}$$

with

$$\begin{aligned}\beta_i &= \alpha_i\big\|\widehat{\varphi}_i(k)\big\|^2 + \\ &\quad + \frac{[(1-\alpha_i\|\widehat{\varphi}_i(k)\|^2) - \Gamma\|I - \alpha_i\widehat{\varphi}_i(k)\widehat{\varphi}_i^{\mathrm{T}}(k)\|]^2}{(2-\alpha_i\|\widehat{\varphi}_i(k)\|^2)}, \quad i = 1, 2.\end{aligned} \tag{86}$$

For Algorithm (a) β_3 *is given by*

$$\beta_3 = 1 + \frac{1}{(1 - \alpha_3 \|\widehat{\varphi}_3(k)\|^2)}, \tag{87}$$

whereas for Algorithm (b) *it is given by*

$$\begin{aligned}\beta_3 \;=\; & 1 + \alpha_3 \big\|\widehat{\varphi}_3(k)\|^2 + \\ & + \frac{1}{(1-\alpha_3)\|\widehat{\varphi}_3(k)\|^2}\Big[\alpha_3\big\|\widehat{\varphi}_3(k)\big\|^2 + \Gamma\big(1 - \alpha_3\big\|\widehat{\varphi}_3(k)\big\|^2\big)\Big]^2.\end{aligned} \tag{88}$$

Note. The parameters $\beta_i,\ \alpha_i,\ i = 1, 2, 3$, and $\overline{\sigma}$ are dependent upon the trajectory.

Proof. Let the NN approximation property (3) hold for the functions $F_1(\cdot)$ and $F_3(\cdot)$ given in (19) with a given accuracy of ε_N for all x in the compact set $S_x \equiv \{x \mid \|x\| < b_x\}$ with $b_x > q_B$. Let $\omega(0) \in S_\omega$. Then the approximation property holds at time $k = 0$.

Algorithm (a): Define the Lyapunov function candidate

$$\begin{aligned}J = \omega^{\mathrm{T}}(k)\omega(k) \;+\; & \frac{1}{\alpha_1}\mathrm{tr}\big(\widetilde{W}_1^{\mathrm{T}}(k)\widetilde{W}_1(k)\big) + \frac{1}{\alpha_2}\mathrm{tr}\big(\widetilde{W}_2^{\mathrm{T}}(k)\widetilde{W}_2(k)\big) + \\ & + \frac{1}{\alpha_3}\mathrm{tr}\big(\widetilde{W}_3^{\mathrm{T}}(k)\widetilde{W}_3(k)\big).\end{aligned} \tag{89}$$

The first difference is given by

$$\begin{aligned}\Delta J \;=\; & \omega^{\mathrm{T}}(k+1)\omega(k+1) - \omega^{\mathrm{T}}(k)\omega(k) + \\ & + \frac{1}{\alpha_1}\mathrm{tr}\big(\widetilde{W}_1^{\mathrm{T}}(k+1)\widetilde{W}_1(k+1) - \widetilde{W}_1^{\mathrm{T}}(k)\widetilde{W}_1(k)\big) + \\ & + \frac{1}{\alpha_2}\mathrm{tr}\big(\widetilde{W}_2^{\mathrm{T}}(k+1)\widetilde{W}_2(k+1) - \widetilde{W}_2^{\mathrm{T}}(k)\widetilde{W}_2(k)\big) + \\ & + \frac{1}{\alpha_3}\mathrm{tr}\big(\widetilde{W}_3^{\mathrm{T}}(k+1)\widetilde{W}_3(k+1) - \widetilde{W}_3^{\mathrm{T}}(k)\widetilde{W}_3(k)\big).\end{aligned} \tag{90}$$

Use the tracking error dynamics (76) and tuning mechanism (78)–(80) in (90) to obtain (91)

$$\begin{aligned}\Delta J \;\leqslant\; & -\big[1 - \overline{\sigma}k_{v\max}^2\big]\big\|\omega(k)\big\|^2 + 2k_{v\max}\gamma\,\big\|\omega(k)\big\| + \rho - \\ & - \sum_{i=1}^{2}\big[2 - \alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)\big]\Big\|\widehat{W}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k) - \\ & - \frac{(1 - \alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)) - \Gamma\|I - \alpha_i\widehat{\varphi}_i(k)\widehat{\varphi}_i^{\mathrm{T}}(k)\|}{[2 - \alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)]}\big(W_i^{\mathrm{T}}\widehat{\varphi}_i(k) + B_i k_v \omega(k)\big)\Big\|^2 \\ & - \big[1 - \alpha_3\widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k)\big]\Big\|\overline{e}_i(k) -\end{aligned}$$

$$
\begin{aligned}
&- \frac{k_v\omega(k) + (\alpha_3\widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k) + \Gamma\|I - \alpha_3\widehat{\varphi}_3(k)\widehat{\varphi}_3^{\mathrm{T}}(k)\|)}{(1 - \alpha_3\widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k))} \times \\
&\times \left(W_3^{\mathrm{T}}\widetilde{\varphi}_3(k) + \varepsilon(k) + d(k)\right)\Big\|^2 + \sum_{i=1}^{3}\frac{1}{\alpha_i}\left\|I - \alpha_i\widehat{\varphi}_i(k)\widehat{\varphi}_i^{\mathrm{T}}(k)\right\|^{\mathrm{T}} \times \\
&\times \mathrm{tr}\left[\Gamma^2\widehat{W}_i^{\mathrm{T}}(k)\widehat{W}_i(k) + 2\Gamma\widehat{W}_i^{\mathrm{T}}(k)\widetilde{W}_i(k)\right], \qquad (91)
\end{aligned}
$$

where

$$
\begin{aligned}
\gamma \;=\; &\left[1 + \frac{1}{(1-\alpha_3\varphi_{3\max}^2)}\left(\alpha_3\varphi_{3\max}^2 + \Gamma\left(1-\alpha_3\varphi_{3\max}^2\right)\right)\right] \times \\
&\times \left(W_{3\max}\widetilde{\varphi}_{3\max} + \varepsilon_N + d_M\right) + \sum_{i=1}^{2}\left[\beta_i + \Gamma\left(1-\alpha_i\varphi_{i\max}^2\right)W_{i\max}\varphi_{i\max}\right]\kappa_i, \qquad (92)
\end{aligned}
$$

and

$$
\begin{aligned}
\rho \;=\; &\left[1 + \alpha_3\varphi_{3\max}^2 + \frac{1}{(1-\alpha_3\varphi_{3\max}^2)}\left(\alpha_3\varphi_{3\max}^2 + \Gamma\left(1-\alpha_3\varphi_{3\max}^2\right)\right)\right] \times \\
&\times \left(W_{3\max}\widetilde{\varphi}_{3\max} + \varepsilon_N\right)^2 + 2\Gamma\left(1-\alpha_3\varphi_{3\max}^2\right) \times \\
&\times W_{3\max}\varphi_{3\max}\left(W_{3\max}\widetilde{\varphi}_{3\max} + \varepsilon_N + d_M\right) + \\
&+ \sum_{i=1}^{2}\left[\beta_i + 2\Gamma\left(1-\alpha_i\varphi_{i\max}^2\right)\right]\varphi_{i\max}^2 W_{i\max}^2. \qquad (93)
\end{aligned}
$$

Consider (91), rewrite the last term in terms of $\|\widetilde{Z}(k)\|$, denote the positive constants $c_{\max}$ and $c_{\min}$ as the maximum and minimum singular value of the diagonal matrix and is given as

$$
\begin{bmatrix}
\frac{\|I-\alpha_1\widehat{\varphi}_1(k)\widehat{\varphi}_1^{\mathrm{T}}(k)\|^2}{\alpha_1} & 0 & 0 \\
0 & \frac{\|I-\alpha_2\widehat{\varphi}_2(k)\widehat{\varphi}_2^{\mathrm{T}}(k)\|^2}{\alpha_2} & 0 \\
0 & 0 & \frac{\|I-\alpha_3\widehat{\varphi}_3(k)\widehat{\varphi}_3^{\mathrm{T}}(k)\|^2}{\alpha_3}
\end{bmatrix}. \qquad (94)
$$

Completing the squares for $\|\widetilde{Z}(k)\|$ to obtain

$$
\begin{aligned}
\Delta J \;\leqslant\; &-\left[1-\overline{\sigma}k_{v\max}^2\right]\left[\left\|\omega(k)\right\|^2 - \frac{2\gamma k_{v\max}}{1-\overline{\sigma}k_{v\max}^2}\left\|\omega(k)\right\| - \frac{\rho + c_0 Z_{\max}^2}{1-\overline{\sigma}k_{v\max}^2}\right] - \\
&- \sum_{i=1}^{2}\left[2-\alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)\right]\Big\|\widehat{W}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k) - \\
&- \frac{(1-\alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)) - \Gamma\|I-\alpha_i\widehat{\varphi}_i(k)\widehat{\varphi}_i^{\mathrm{T}}(k)\|}{2-\alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)} \times
\end{aligned}
$$

$$\times \left(W_i^{\mathrm{T}}\widehat{\varphi}_i(k) + B_i k_v \omega(k)\right)\Big\|^2 - \left[1 - \alpha_3 \widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k)\right] \times$$
$$\times \left\| \overline{e}_i(k) - \frac{k_v \omega(k) + (\alpha_3 \widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k) + \Gamma \| I - \alpha_3 \widehat{\varphi}_3(k)\widehat{\varphi}_3^{\mathrm{T}}(k)\|)}{1 - \alpha_3 \widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k)} \times \right.$$
$$\times \left(W_3^{\mathrm{T}}\widetilde{\varphi}_3(k) + \varepsilon(k) + d(k)\right)\Big\|^2 -$$
$$- \Gamma(2-\Gamma)c_{\min}\left[\left\|\widetilde{Z}(k)\right\| - \frac{(1-\Gamma)c_{\max}Z_{\max}}{(2-\Gamma)c_{\min}}\right]^2 \tag{95}$$

with

$$c_0 = \frac{c_{\max}}{c_{\min}}\frac{1}{(2-\Gamma)}\left[(1-\Gamma)^2 c_{\max} + \Gamma^2(2-\Gamma)c_{\min}\right]. \tag{96}$$

Then $\Delta J \leqslant 0$ as long as (82) through (84) hold and the quadratic term of $\omega(k)$ in (95) is positive, which is guaranteed when

$$\left\|\omega(k)\right\| > \frac{1}{(1-\overline{\sigma}k_{v\max}^2)}\left[\gamma k_{v\max} + \right.$$
$$\left. + \sqrt{\gamma^2 k_{v\max}^2 + [\rho + c_0 Z_{\max}^2](1-\overline{\sigma}k_{v\max}^2)}\right] \equiv b_\omega. \tag{97}$$

Similarly, completing the squares $\|\omega(k)\|$ using (95) yields

$$\Delta J \leqslant -\left[1-\overline{\sigma}k_{v\max}^2\right]\left[\left\|\omega(k)\right\| - \frac{\gamma k_{v\max}}{1-\overline{\sigma}k_{v\max}^2}\right]^2 - \Gamma(2-\Gamma)c_{\min}\left\|\widetilde{Z}(k)\right\|^2 -$$
$$- 2(1-\Gamma)c_{\max}\left\|\widetilde{Z}(k)\right\|Z_{\max} - \left(\Gamma c_{\max}Z_{\max}^2 + \frac{\gamma^2 k_{v\max}^2}{(1-\overline{\sigma}k_{v_{\max}}^2)} + \rho\right) -$$
$$- \sum_{i=1}^{2}\left[2 - \alpha_i \widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)\right] \times$$
$$\times \left\| \widehat{W}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k) - \frac{(1-\alpha_i \widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)) - \Gamma\|I - \alpha_i\widehat{\varphi}_i(k)\widehat{\varphi}_i^{\mathrm{T}}(k)\|}{2 - \alpha_i \widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)} \times \right.$$
$$\times \left(W_i^{\mathrm{T}}\widehat{\varphi}_i(k) + B_i k_v \omega(k)\right)\Big\|^2 - \left[1 - \alpha_3 \widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k)\right] \times$$
$$\times \left\| \overline{e}_i(k) - \frac{k_v \omega(k) + \left(\alpha_3 \widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k) + \Gamma \| I - \alpha_3 \widehat{\varphi}_3(k)\widehat{\varphi}_3^{\mathrm{T}}\|\right)}{1 - \alpha_3 \widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k)} \times \right.$$
$$\times \left(W_3^{\mathrm{T}}\widetilde{\varphi}_3(k) + \varepsilon(k) + d(k)\right)\Big\|^2, \tag{98}$$

where γ and ρ are given in (92) and (93), respectively. Then $\Delta J \leqslant 0$ as long as (82)–(84) hold and the quadratic term for $\|\widetilde{Z}(k)\|$ is positive, which is guaranteed when

$$\|\widetilde{Z}(k)\| > \frac{\Gamma(1-\Gamma)c_{\max}Z_{\max} + \sqrt{\Gamma^2(1-\Gamma)^2 c_{\max} Z_{\max}^2 + \Gamma(2-\Gamma)c_{\min}\theta}}{\Gamma(2-\Gamma)} \equiv b_Z, \tag{99}$$

where

$$\theta = \left[\Gamma^2 c_{\max} Z_{\max}^2 + \frac{\gamma^2 k_{v\max}^2}{(1-\overline{\sigma}k_{v\max}^2)} + \rho\right]. \tag{100}$$

From (97) and (99), it can be concluded that the tracking error $\omega(k)$ and the error in weight estimates $\widetilde{Z}(k)$ are UUB.

Algorithm (b): Select the Lyapunov function candidate (89). Use the tracking error dynamics (76) and tuning mechanism (78), (79), and (81) to obtain

$$\begin{aligned}\Delta J \;\leqslant\; & -\big[1-\overline{\sigma}k_{v\max}^2\big]\big\|\omega(k)\big\|^2 + 2k_{v\max}\gamma\,\big\|\omega(k)\big\| + \rho - \\ & -\sum_{i=1}^{2}\big[2-\alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)\big]\bigg\|\widehat{W}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k) - \\ & -\frac{(1-\alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)) - \Gamma\|I-\alpha_i\widehat{\varphi}_i(k)\widehat{\varphi}_i^{\mathrm{T}}(k)\|}{2-\alpha_i\widehat{\varphi}_i^{\mathrm{T}}(k)\widehat{\varphi}_i(k)} \times \\ & \times\big(W_i^{\mathrm{T}}\widehat{\varphi}_i(k) + B_i k_v\omega(k)\big)\bigg\|^2 - \big[1-\alpha_3\widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k)\big] \times \\ & \times\bigg\|\overline{e}_i(k) - \frac{(\alpha_3\widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k) + \Gamma\|I-\alpha_3\widehat{\varphi}_3(k)\widehat{\varphi}_3^{\mathrm{T}}(k)\|)}{1-\alpha_3\widehat{\varphi}_3^{\mathrm{T}}(k)\widehat{\varphi}_3(k)} \times \\ & \times\big(k_v\omega(k) + W_3^{\mathrm{T}}\widetilde{\varphi}_3(k) + \varepsilon(k) + d(k)\big)\bigg\|^2 + \\ & + \sum_{i=1}^{3}\frac{1}{\alpha_i}\big\|I-\alpha_i\widehat{\varphi}_i(k)\widehat{\varphi}_i^{\mathrm{T}}(k)\big\|^2 \mathrm{tr}\big[\Gamma^2\widehat{W}_i^{\mathrm{T}}(k)\widehat{W}_i(k) + 2\Gamma\widehat{W}_i^{\mathrm{T}}(k)\widetilde{W}_i(k)\big],\end{aligned} \tag{101}$$

where

$$\begin{aligned}\gamma \;=\; & \beta_3\big(W_{3\max}\widetilde{\varphi}_{3\max} + \varepsilon_N + d_M\big) + \big(1-\alpha_3\varphi_{3\max}^2\big)W_{3\max}\varphi_{3\max} + \\ & + \sum_{i=1}^{2}\big[\big(\beta_i + \Gamma\big(1-\alpha_i\varphi_{i\max}^2\big)\big)W_{i\max}\varphi_{i\max}\big]\kappa_i,\end{aligned} \tag{102}$$

and

$$\begin{aligned}\rho \;=\; & \big[\beta_3\big(W_{3\max}\widetilde{\varphi}_{3\max} + \varepsilon_N + d_M\big) + 2\Gamma\big(1-\alpha_3\varphi_{3\max}^2\big)W_{3\max}\varphi_{3\max}\big] \times \\ & \times\big[W_{3\max}\widetilde{\varphi}_{3\max} + \varepsilon_N + d_M\big] + \sum_{i=1}^{2}\big[\big(\beta_i + 2\Gamma\big(1-\alpha_i\varphi_{i\max}^2\big)\big)\big]\varphi_{i\max}^2 W_{i\max}^2.\end{aligned} \tag{103}$$

Completing the squares for $\|\widetilde{Z}(k)\|$ in (101) similar to Algorithm (a) results in $\Delta J \leqslant 0$ as long as the conditions in (82) through (84) are satisfied and with the upper bound on the tracking error given by (97).

On the other hand, completing the squares for $\|\omega(k)\|$ in (101) results in $\Delta J \leqslant 0$ as long as the conditions (82)–(84) are satisfied and we have (99) where θ is given in (100).

In general $\Delta J \leqslant 0$ as long as (82)–(84) are satisfied and either (97) or (99) holds. In other words, if the right-hand sides of (97) and (99) are denoted as two constants δ_1 and δ_2, respectively, then $\Delta J \leqslant 0$ whenever $\|\omega(k)\| > \delta_1$ or $\|\widetilde{Z}(k)\| > \delta_2$. Let us denote $(\|\omega(k)\|, \|\widetilde{Z}(k)\|)$ by a new coordinate system $(\vartheta_1, \vartheta_2)$. Define the region D: $(\vartheta \mid \vartheta_1 < \delta_1, \vartheta_2 < \delta_2)$ then there exists an open set Ω: $(\vartheta \mid \vartheta_1 < \bar{\delta}_1, \vartheta_2 < \bar{\delta}_2)$, where $\bar{\delta}_i > \delta_i$ implies that $D \subset \Omega$. This further implies that the Lyapunov function J will stay in the region Ω which is an invariant set. This demonstrates that the tracking error and the error in weight estimates are UUB. □

Remark. For practical purposes, (97) and (99) for the case of Algorithms (a) and (b) can be considered as bounds for $\|\omega(k)\|$ and $\|\widetilde{Z}(k)\|$.

Note that the NN reconstruction error bound ε_N increase the bounds on $\|\omega(k)\|$ and $\|\widetilde{Z}(k)\|$ in a very interesting way. Note that small tracking error bounds, but not arbitrarily small, may be achieved by placing the closed-loop poles inside the unit circle and near the origin through the selection of the largest eigenvalue $k_{v\max}$. On the other hand, the NN weight error estimates are fundamentally bounded by $Z_{\max}$, the known bound on ideal weights W. The parameter Γ offers a design tradeoff between the relative eventual magnitudes of $\|\omega(k)\|$ and $\|\widetilde{Z}(k)\|$; a smaller Γ yields a smaller $\|\omega(k)\|$ and a larger $\|\widetilde{Z}(k)\|$, and vice versa.

The effect of adaptation gains α_1, α_2 and α_3 at each layer on the weight estimation error $\widetilde{Z}(k)$ and tracking error $\omega(k)$ can be easily observed by using the bounds presented in (97) and (99) through $c_{\min}$ and $c_{\max}$. Large values of α_1, and α_2 forces smaller weight estimation error whereas the tracking error is unaffected. In contrast, a large value of α_3 forces smaller tracking and weight estimation errors.

6. Design Example

Consider a single-link manipulator with the inclusion of actuator dynamics. This system model is expressed as

$$M\ddot{q} + Bq + N\sin(q) = \tau, \qquad D\dot{\tau} + H\tau = u - K_m\ddot{q}. \tag{104}$$

System (104) can be expressed in the form (5) and (6) by noting that $x_1 = q$, $x_2 = \dot{q}$, $x_3 = \tau$, and

$$F_1 = \frac{[-N\sin(x_1) - Bx_2]}{M}, \qquad G_1 = \frac{1}{M},$$

$$F_2 = \frac{-(-K_m x_2 - H x_3)}{D}, \qquad G_2 = \frac{1}{D}.$$

The parameter values are given by $M = 1$, $D = 0.05$, $B = 1$, $K_m = 10$, $H = 0.5$, and $N = 10$. The desired trajectory $x_{1\mathrm{d}} = \sin(2\pi t)$. Using the design procedure described in the paper, a fictitious controller was designed using a CMAC NN for τ followed by a second controller u using another CMAC NN. The controller parameters were selected as $K_1 = 100$, $K_2 = 150$, $k_v = 0.5$ and $\lambda = 10$. The number of receptive field functions selected in each dimension was 4. The initial conditions for x_1, x_2 and x_3 were selected as 0.1, 6.28 and 0, respectively. The weight updates developed in this paper in continuous time were employed for the simulation. Simulation results (not given in this paper) show impressive performance of the closed-loop system when there is no disturbances present.

7. Conclusions

Neural Network controllers for robust backstepping control of a class of nonlinear system were presented. A backstepping controller was designed in continuous-time using CMAC NN whereas multilayer NNs were employed in discrete time. This approach (both in continuous and discrete time) does not require the information about the system dynamics. Compared with the conventional adaptive control, no linearity in the unknown parameters is needed and persistency of excitation is not required. Further certainty equivalence is not employed and no tedious computation of the regression matrix is needed. No initial off-line learning phase is required in this approach. It was shown using the Lyapunov analysis that the tracking error and weight estimates are guaranteed to be bounded. In addition, the tracking error can be reduced to arbitrarily small values by choosing the gains large enough. A design example is given in continuous-time to illustrate the approach.

References

1. Åström, K. J. and Wittenmark, B.: *Adaptive Control*, Addison-Wesley, Reading, MA, 1989.
2. Commuri, S., Jagannathan, S., and Lewis, F. L.: CMAC NN control of robot manipulators, *J. Robotic Systems* **14**(6) (1997), 465–482.
3. Dawson, D. W., Qu, Z., and Hu, J.: Robust trasking control of an induction motor, in: *Proc. of American Control Conf.*, 1993, pp. 648–652.
4. Goodwin, G. C. and Sin, K. S.: *Adaptive Filtering, Prediction, and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1984.
5. Jagannathan, S. and Lewis, F. L.: Robust implicit self tuning regulator, *Automatica* **12**(12) (1996).
6. Jagannathan, S. and Lewis, F. L.: Multilayer discrete-time neural net controller with guaranteed performance, *IEEE Trans. Neural Networks* **7**(1) (1996), 107–130.
7. Jagannathan, S. and Lewis, F. L.: Discrete-time neural net controller with guaranteed performance, *IEEE Trans Automat. Control* **41**(11) (1996), 1693–1699.
8. Jagannathan, S.: Adaptive control of a class of feedback linearizable nonlinear systems using neural networks, in: *Proc. of the IEEE Conf. on Robotics and Automation*, Vol. 1, April 1996, pp. 258–263.

9. Kanellakopoulos, I.: A discrete-time adaptive nonlinear system, *IEEE Trans. Automat. Control* **39**(11) (1994), 2362–2365.
10. Kokotovic, P. V.: Bode lecture: The joy of feedback, *IEEE Control Systems Magazine* **3** (June 1992), 7–17.
11. Kwan, C. M. and Lewis, F. L.: Robust backstepping control of nonlinear systems using neural networks, in: *European Control Conf.*, 1994.
12. Landau, I. D.: *Adaptive Control: The Model Reference Approach*, Marcel Dekker, New York, 1979.
13. Landau, I. D.: Evolution of adaptive control, *ASME J. Dynamic Syst. Measurements Control* **115** (June 1993), 381–391.
14. Lewis, F. L., Liu, K., and Yesilderik, A.: Multilayer neural robot controller with guaranteed performance, *IEEE Trans. Neural Networks* **6**(3) (1995), 703–715.
15. Lewis, F. L., Abdallah, C. T., and Dawson, D. M.: *Control of Robot Manipulators*, MacMillan, New York, 1993.
16. Ljung, L. and Söderström, T.: *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA, 1993.
17. Narendra, K. S. and Annaswamy, A. M.: A new adaptive law for robust adaptation without persistent excitation, *IEEE Trans. Automat. Control* **AC-32**(2) (1987), 134–145.
18. Narendra, K. S. and Annaswamy, A. M.: *Stable Adaptive Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
19. Narendra, K. S. and Parthasarathy, K. S.: Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Networks* **1**(1) (1990), 4–27.
20. Sabanovic, A., Sabanovic, N., and Ohnishi, K.: Sliding modes in power converters and motion control systems, *Internat. J. Control* **57** (1993), 1237–1259.
21. Sanner, R. M. and Slotine, J.-J.: Gaussian networks for direct adaptive control, *IEEE Trans. Neural Networks* **3**(6) (1992), 837–863.
22. Slotine, J.-J. E. and Li, W.: *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.