Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance

Frank L. Lewis, *Fellow, IEEE,* Aydin Yeşildirek, and Kai Liu, *Senior Member, IEEE*

Abstract—**A multilayer neural-net (NN) controller for a general serial-link rigid robot arm is developed. The structure of the NN controller is derived using a filtered error/passivity approach. No off-line learning phase is needed for the proposed NN controller and the weights are easily initialized. The nonlinear nature of the NN, plus NN functional reconstruction inaccuracies and robot disturbances, mean that the standard delta rule using backpropagation tuning does not suffice for closed-loop dynamic control. Novel on-line weight tuning algorithms, including correction terms to the delta rule plus an added robustifying signal, guarantee bounded tracking errors as well as bounded NN weights. Specific bounds are determined, and the tracking error bound can be made arbitrarily small by increasing a certain feedback gain. The correction terms involve a second-order forward-propagated wave in the backprop network. New NN properties including the notions of a passive NN, a dissipative NN, and a robust NN are introduced.**

I. Introduction

MUCH has been written about neural net (NN) for system identification (e.g., [5], [12], and [27]) or identification-based ("indirect") control, little about the use of NN in direct closed-loop controllers that yield guaranteed performance. Some results showing the relations between NN and direct adaptive control [10], [16], as well as some notions on NN for robot control, are given in [2], [14], [15], [25], [29], and [45], see also [24].

NN used in the dynamic control loop pose problems not present when they are used for classification or system identification. Persistent problems that remain to be adequately addressed include *ad hoc* controller structures and the inability to guarantee satisfactory performance of the system in terms of small tracking error and bounded NN weights (which ensures bounded control inputs). Uncertainty on how to initialize the NN weights leads to the necessity for "preliminary off-line tuning" [25], [7]. Some of these problems have been addressed for the two-layer NN, where linearity in the parameters holds (e.g., using radial basis function (RBF) nets) [38], [35], [37], [32], [33], [18]. Multilayer nets using a projection algorithm to tune the weights have been used for identification purposes (only) in [32] and [33]. A technique based on deadzone weight tuning is given in [4] and [22]. Generally, in all these papers initial estimates for the NN weights are needed that stabilize the closed-loop system; such stabilizing weights may be very hard to determine. In [21] a two-layer NN is given that does not need such initial estimates; in that work, the NN is linear in the tunable weights so that suitable basis functions must be first selected (e.g., RBF's).

In this paper we extend work in [19]–[21] to confront these deficiencies for the full nonlinear three-layer NN with arbitrary activation functions (as long as the function satisfies an approximation property and it and its derivatives are bounded). The approximation accuracy is generally better in nonlinear multilayer NN than in linear two-layer NN. Moreover, in comparison to the linear-in-the-parameter NN used in [38], [35], [37], [32], [33], [18], and [21], in the multilayer NN no basis functions are needed; it is only necessary that an approximation property be satisfied. This seems to be because the first layers effectively compute (i.e., "learn") the basis functions for the specific application while the last layer combines them appropriately.

In this paper, some notions in robot control [17] are tied to some notions in NN theory. The NN weights are tuned on-line, with no "off-line learning phase" needed. Fast convergence of the tracking error is comparable to that of adaptive controllers. A novel controller structure that includes an outer tracking loop ensures good performance during the initial period if the NN weights are initialized at zero. Tracking performance is guaranteed using a Lyapunov approach even though there do not exist "ideal" weights such that the NN perfectly reconstructs a required nonlinear robot function. By "guaranteed" we mean that both the tracking errors and the neural net weights are bounded.

The controller is composed of a neural net incorporated into a dynamic system, where the structure comes from some filtered error notions standard in robot control. Unlike adaptive robot control, where a "regression matrix of robot functions" must be tediously computed from the dynamics of each specific arm [6], [17], the NN approach allows the use of activation functions standard in NN theory for any serial-link robot arm. This means that the same NN controller works for any rigid robot with no preliminary analysis. The NN controller affords the design freedom of trading off the complexity (i.e., number of hidden-layer neurons) of the NN with the magnitude of a certain robust control term added for guaranteed stability.

It is shown that standard delta rule tuning technique using backpropagation [43], [44], [36] generally yields unbounded NN weights if: 1) the net cannot exactly reconstruct a certain required nonlinear robot function; 2) there are bounded unknown disturbances or modeling errors in the robot dynamics; or 3) the robot arm has more than one link (i.e., nonlinear

Manuscript received March 28, 1994; revised February 27, 1995. This work was supported by NSF Grant IRI-9216545.

The authors are with the Automation and Robotics Research Institute, The University of Texas at Arlington, Ft. Worth, TX 76118 USA.

Publisher Item Identifier S 1045-9227(96)00162-2.

case). Modified weight tuning rules based on the delta rule with backpropagation guarantee tracking and bounded weights for the general case. The modifications consist of: 1) the e-modification in [26]; 2) a novel term corresponding to a second-order forward propagating wave in the backprop tuning network [27]; and 3) a robustifying control signal.

New passivity properties of NN as introduced for linear nets in [21] are extended to the three-layer nonlinear net. It is shown that the backpropagation tuning algorithm yields a passive NN. This, coupled with the dissipativity of the robot dynamics, guarantees that all signals in the closed-loop system are bounded under additional observability or persistency of excitation (PE) conditions. The modified weight tuning algorithms given herein avoid the need for PE by making the NN robust, that is, strictly passive in a sense defined herein.

II. Background

Let $\boldsymbol{R}$ denote the real numbers, $\boldsymbol{R}^{\boldsymbol{n}}$ denote the real n-vectors, and $\boldsymbol{R}^{\boldsymbol{mxn}}$ the real $m \times n$ matrices. Let $\boldsymbol{S}$ be a compact simply connected set of $\boldsymbol{R}^{\boldsymbol{n}}$. With map f: $\boldsymbol{S} \rightarrow \boldsymbol{R}^{\boldsymbol{m}}$, define $C^m(\boldsymbol{S})$ as the space such that f is continuous. We denote by $\|\cdot\|$ any suitable vector norm. When it is required to be specific we denote the p-norm by $\|\cdot\|_p$. The supremum norm of $f(x)$ (over $\boldsymbol{S}$) is defined as [3]

$$\sup_{x\in\boldsymbol{S}}\|f(x)\|, \quad f: \boldsymbol{S} \rightarrow \boldsymbol{R}^{\boldsymbol{m}}.$$

Given $A = [a_{ij}], B \in \boldsymbol{R}^{\boldsymbol{mxn}}$ the Frobenius norm is defined by

$$\|A\|_F^2 = \operatorname{tr}(A^T A) = \sum a_{ij}^2$$

with tr( ) the trace. The associated inner product is $\langle A, B\rangle_F = \operatorname{tr}(A^T B)$. The Frobenius norm is nothing but the vector two-norm over the space defined by stacking the matrix columns into a vector. As such, it cannot be defined as the induced matrix norm for any vector norm, but is compatible with the two-norm so that $\|Ax\|_2 \leq \|A\|_F \|x\|_2$, with $A \in \boldsymbol{R}^{\boldsymbol{mxn}}$ and $x \in \boldsymbol{R}^{\boldsymbol{n}}$.

When $x(t) \in \boldsymbol{R}^{\boldsymbol{n}}$ is a function of time we may use the standard L_p norms [17]. We say $x(t)$ is bounded if its L_∞ norm is bounded. We say $A(t) \in \boldsymbol{R}^{\boldsymbol{mxn}}$ is bounded if its induced matrix ∞-norm is bounded.

A. Neural Networks

Given $x \in \boldsymbol{R}^{\boldsymbol{N_1}}$, a three-layer NN (Fig. 1) has a net output given by

$$y_i = \sum_{j=1}^{N_2}\left[w_{ij}\sigma\left[\sum_{k=1}^{N_1} v_{jk}x_k + \theta_{vj}\right] + \theta_{wi}\right]; \quad i = 1, \cdots, N_3 \tag{1}$$

with $\sigma(\cdot)$ the activation function, v_{jk} the first-to-second layer interconnection weights, and w_{ij} the second-to-third layer interconnection weights. The $\theta_{vm}, \theta_{wm}, m = 1, 2, \cdots,$ are threshold offsets and the number of neurons in layer ℓ is N_ℓ, with N_2 the number of hidden-layer neurons. In the NN we should like to adapt the weights and thresholds on-line in real time to provide suitable performance of the net. That is, the NN should exhibit "learning-while controlling" behavior.

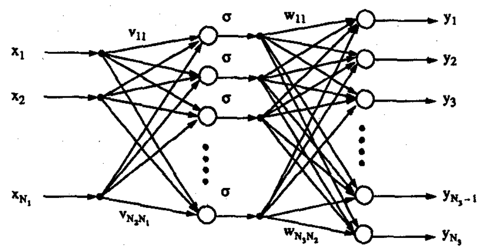


Fig. 1. Three-layer NN structure.

Typical selections for $\sigma(\cdot)$ include, with $z \in \boldsymbol{R}$

$$\sigma(z) = \frac{1}{1+e^{\alpha z}}, \quad \text{sigmoid}$$

$$\sigma(z) = \frac{1-e^{-\alpha z}}{1+e^{-\alpha z}}, \quad \text{hyperbolic tangent (tanh)}$$

$$\sigma(z) = e^{-(z-m_j)^2}/s_j, \quad \text{radial basis functions (RBF).}$$

The NN equation may be conveniently expressed in matrix format by defining $x = [x_0 x_1 x_2 \cdots x_{N1}]^T, y = [y_1\, y_2 \cdots y_{N3}]^T$, and weight matrices $W^T = [w_{ij}], V^T = [v_{jk}]$. Including $x_0 \equiv 1$ in x allows one to include the threshold vector $[\theta_{v1}\theta_{v2}\cdots\theta_{vN2}]^T$ as the first column of V^T, so that V^T contains both the weights and thresholds of the first- to second-layer connections. Then

$$y = W^T\sigma(V^T x) \tag{2}$$

where, if $z = [z_1 z_2 \cdots]^T$ a vector we define $\sigma(z) = [\sigma(z_1)\, \sigma(z_2)\cdots]^T$. Including one as a first term in the vector $\sigma(V^T x)$ allows one to incorporate the thresholds θ_{wi} as the first column of W^T. Any tuning of W and V then includes tuning of the thresholds as well.

Although, to account for nonzero thresholds, x may be augmented by $x_0 = 1$ and σ by the constant first entry of one, we loosely say that $x \in \boldsymbol{R}^{\boldsymbol{N_1}}$ and σ: $\boldsymbol{R}^{\boldsymbol{N_2}} \rightarrow \boldsymbol{R}^{\boldsymbol{N_2}}$.

A general function $f(x) \in C^m(\boldsymbol{S})$ can be written as

$$f(x) = W^T\sigma(V^T x) + \varepsilon(x) \tag{3}$$

with $N_1 = n, N_3 = m$, and $\varepsilon(x)$ a NN functional reconstruction error vector. If there exist N_2 and constant "ideal" weights W and V so that $\varepsilon = 0$ for all $x \in \boldsymbol{S}$, we say $f(x)$ is in the functional range of the NN. In general, given a constant real number $\varepsilon_N > 0$, we say $f(x)$ is within ε_N of the NN range if there exist N_2 and constant weights so that for all $x \in \boldsymbol{R}^{\boldsymbol{n}}$, (3) holds with $\|\varepsilon\| < \varepsilon_N$.

Various well-known results for various activation functions $\sigma(\cdot)$, based, e.g., on the Stone–Weierstrass theorem, say that any sufficiently smooth function can be approximated by a suitably large net [8], [13], [31], [38]. The functional range of NN (2) is said to be dense in $C^m(\boldsymbol{S})$ if for any $f \in C^m(\boldsymbol{S})$ and $\varepsilon_N > 0$ there exist finite N_2, and W and V such that (3) holds with $\|\varepsilon\| < \varepsilon_N, N_1 = n, N_3 = m$. Typical results are like the following, for the case of σ the "squashing functions" (a bounded, measurable, nondecreasing function from the real numbers onto [0, 1]), which include for instance the step, the ramp, and the sigmoid.

Theorem 2.1: Set $N_1 = n, N_3 = m$ and let σ be any squashing function. Then the functional range of NN (2) is dense in $C^m(\boldsymbol{S})$. ■

In this result, the metric defining denseness is the supremum norm. Moreover, the last layer thresholds $\theta_{w\ell}$ w are not needed for this result. The issues of selecting σ, and of choosing N_2 for a specified $\boldsymbol{S} \subset \boldsymbol{R^n}$ and ε_N are current topics of research (see, e.g., [28] and [31]).

B. Stability and Passive Systems

Some stability notions are needed to proceed. Consider the nonlinear system

$$\dot{x} = f(x, u, t), \quad y = h(x, t)$$

with state $x(t) \in \boldsymbol{R^n}$. We say the solution is uniformly ultimately bounded (UUB) if there exists a compact set $\boldsymbol{U} \subset \boldsymbol{R^n}$ such that for all $x(t_0) = x_0 \in \boldsymbol{U}$, there exists an $\varepsilon > 0$ and a number $T(\varepsilon, x_0)$ such that $\|x(t)\| < \varepsilon$ for all $t \geq t_0 + T$. As we shall see in the proof of the theorems, the compact set $\boldsymbol{U}$ is related to the compact set on which NN approximation property (3) holds. Note that $\boldsymbol{U}$ can be made larger by selecting more hidden-layer neurons.

Some aspects of passivity will subsequently be important [11], [16], [17], [41]. A system with input $u(t)$ and output $y(t)$ is said to be passive if it verifies an equality of the so-called "power form"

$$\dot{L}(t) = y^T u - g(t) \tag{4}$$

with $L(t)$ lower bounded and $g(t) \geq 0$. That is

$$\int_0^T y^T(\tau) u(\tau)\, d\tau \geq \int_0^T g(\tau)\, d\tau - \gamma^2 \tag{5}$$

for all $T \geq 0$ and some $\gamma \geq 0$.

We say the system is dissipative if it is passive and in addition

$$\int_0^\infty y^T(\tau) u(\tau)\, d\tau \neq 0 \text{ implies } \int_0^\infty g(\tau)\, d\tau > 0. \tag{6}$$

A special sort of dissipativity occurs if $g(t)$ is a monic quadratic function of $\|x\|$ with bounded coefficients, where $x(t)$ is the internal state of the system. We call this state-strict passivity, and are not aware of its use previously in the literature (although cf. [11]). Then the L_2 norm of the state is overbounded in terms of the L_2 inner product of output and input (i.e., the power delivered to the system). This we use to advantage to conclude some internal boundedness properties of the system without the usual assumptions of observability (e.g., persistence of excitation), stability, etc.

C. Robot Arm Dynamics

The dynamics of an n-link robot manipulator may be expressed in the Lagrange form [17]

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \tag{7}$$

with $q(t) \in \boldsymbol{R^n}$ the joint variable vector, $M(q)$ the inertia matrix, $V_m(q, \dot{q})$ the coriolis/centripetal matrix, $G(q)$ the gravity vector, and $F(\dot{q})$ the friction. Bounded unknown disturbances (including, e.g., unstructured unmodeled dynamics) are denoted by τ_d, and the control input torque is $\tau(t)$.

Given a desired arm trajectory $q_d(t) \in \boldsymbol{R^n}$ the tracking error is

$$e(t) = q_d(t) - q(t). \tag{8}$$

In standard use in robotics is the filtered tracking error

$$r = \dot{e} + \Lambda e \tag{9}$$

where $\Lambda = \Lambda^T > 0$ is a design parameter matrix, usually selected diagonal. Differentiating $r(t)$ and using (7), the arm dynamics may be written in terms of the filtered tracking error as

$$M\dot{r} = -V_m r - \tau + f + \tau_d \tag{10}$$

where the nonlinear robot function is

$$f(x) = M(q)(\ddot{q}_d + \Lambda\dot{e}) + V_m(q, \dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(\dot{q}) \tag{11}$$

and, for instance, we may select

$$x = [e^T \dot{e}^T q_d^T \dot{q}_d^T \ddot{q}_d^T]^T. \tag{12}$$

Define now a control input torque as

$$\tau_o = \hat{f} + K_v r \tag{13}$$

gain matrix $K_v = K_v^T > 0$ and $\hat{f}(x)$ an estimate of $f(x)$ provided by some means not yet disclosed. The closed-loop system becomes

$$M\dot{r} = -(K_v + K_m)r + \tilde{f} + \tau_d \equiv -(K_v + V_m)r + \zeta_o \tag{14}$$

where the functional estimation error is given by

$$\tilde{f} = f - \hat{f}. \tag{15}$$

This is an error system wherein the filtered tracking error is driven by the functional estimation error.

The control τ_o incorporates a proportional-plus-derivative (PD) term in $K_v r = K_v(\dot{e} + \Lambda e)$.

In the remainder of the paper we shall use (14) to focus on selecting NN tuning algorithms that guarantee the stability of the filtered tracking error $r(t)$. Then, since (9), with the input considered as $r(t)$ and the output as $e(t)$ describes a stable system, standard techniques [23], [41] guarantee that $e(t)$ exhibits stable behavior. In fact, $\|e\|_2 \leq \|r\|_2/\sigma_{\min}(\Lambda)$, $\|\dot{e}\|_2 \leq \|r\|^2$, with $\sigma_{\min}(\Lambda)$ the minimum singular value of Λ. Generally Λ is diagonal, so that $\sigma_{\min}(\Lambda)$ is the smallest element of Λ.

The following standard properties of the robot dynamics are required [17] and hold for any revolute rigid serial robot arm.

Property 1: $M(q)$ is a positive definite symmetric matrix bounded by

$$m_1 I \leq M(q) \leq m_2 I$$

with m_1, m_2 known positive constants.

Property 2: $V_m(q, \dot{q})$ is bounded by $v_b(q)\|\dot{q}\|$, with $v_b(q) \in C^1(\boldsymbol{S})$.

Property 3: The matrix $\dot{M} - 2V_m$ is skew-symmetric.

Property 4: The unknown disturbance satisfies $\|\tau_d < b_d$, with b_d a known positive constant.

Property 5: The dynamics (14) from $\zeta_o(t)$ to $r(t)$ are a stat— strict passive system.

Proof of Property 5: See [21].

III. NN CONTROLLER

In this section we derive a NN controller for the robot dynamics in Section II. We propose various weight-tuning algorithms, including standard backpropagation. It is shown that with backpropagation tuning the NN can only be guaranteed to perform suitably in closed loop under unrealistic ideal conditions (which require, e.g., $f(x)$ linear). A modified tuning algorithm is subsequently proposed so that the NN controller performs under realistic conditions.

Thus, assume that the nonlinear robot function (11) is given by an NN as in (3) for some constant "ideal" NN weights W and V, where the net reconstruction error $\varepsilon(x)$ is bounded by a known constant ε_N. Unless the net is "minimal," suitable "ideal" weights may not be unique [1], [42]. The "best" weights may then be defined as those which minimize the supremum norm over $\boldsymbol{S}$ of $\varepsilon(x)$. This issue is not of major concern here, as we only need to know that such ideal weights exist; their actual values are not required.

According to Theorem 2.1, this mild approximation assumption always holds for continuous functions. This is in stark contrast to the case for adaptive control, where approximation assumptions such as the Erzberger or linear-in-the-parameters assumptions may not hold. The mildness of this assumption is the main advantage to using multilayer nonlinear nets over linear two-layer nets.

For notational convenience define the matrix of all the weights as

$$Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix}. \tag{16}$$

A. Some Bounding Assumptions and Facts

Some required mild bounding assumptions are now stated. The two assumptions will be true in every practical situation, and are standard in the existing literature. The facts are easy to prove given the assumptions.

Assumption 1: The ideal weights are bounded by known positive values so that $\|V\|_F \le V_M, \|W\|_F \le W_M$, or

$$\|Z\|_F \le Z_M \tag{17}$$

with Z_M known.

Assumption 2: The desired trajectory is bounded in the sense, for instance, that

$$\left\| \begin{matrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{matrix} \right\| \le Q_d \tag{18}$$

where $Q_d \in \boldsymbol{R}$ is a known constant. ■

Fact 3: For each time $t, x(t)$ in (12) is bounded by

$$\|x\| \le c_1 Q_d + c_2 \|r\| \tag{19}$$

for computable positive constants c_i (c_2 decreases as Λ increases). ■

The next discussion is of major importance in this paper; it is the key to extending linear NN results to nonlinear NN's. Proper use of these Taylor series-based results gives a requirement for new terms in the weight tuning algorithms for nonlinear NN's that do not occur in linear NN's.

Let $\hat{V}, \hat{W}$ be some estimates of the ideal weight values, as provided for instance by the weight tuning algorithms to be introduced. Define the weight deviations or weight estimation errors as

$$\tilde{V} = V - \hat{V}, \quad \tilde{W} = W - \hat{W}, \quad \tilde{Z} = Z - \hat{Z} \tag{20}$$

and the hidden-layer output error for a given x as

$$\tilde{\sigma} = \sigma - \hat{\sigma} \equiv \sigma(V^T x) - \sigma(\hat{V}^T x). \tag{21}$$

The Taylor series expansion for a given x may be written as

$$\sigma(V^T x) = \sigma(\hat{V}^T x) + \sigma'(\hat{V}^T x)\tilde{V}^T x + O(\tilde{V}^T x)^2 \tag{22}$$

with $\sigma'(\hat{z}) \equiv d\sigma(z)/dz|_{z=\hat{z}}$, and $O(z)^2$ denoting terms of order two. (Compare to [33] where a different Taylor series was used for identification purposes only.) Denoting $\hat{\sigma}' = \sigma'(\hat{V}^T x)$, we have

$$\tilde{\sigma} = \sigma'(\hat{V}^t x)\tilde{V}^T x + O(\tilde{V}^T x)^2 = \hat{\sigma}'\tilde{V}^T x + O(\tilde{V}^T x)^2. \tag{23}$$

Different bounds may be put on the Taylor series higher-order terms depending on the choice for $\sigma(\cdot)$. Noting that

$$O(\tilde{V}^T x)^2 = [\sigma(V^t x) - \sigma(\hat{V}^T x)] - \sigma'(\hat{V}^T x)\tilde{V}^T x \tag{24}$$

we take the following.

Fact 4: For sigmoid, RBF, and tanh activation functions, the higher-order terms in the Taylor series are bounded by

$$\|O(\tilde{V}^T x)^2\| \le c_3 + c_4 Q_d \|\tilde{V}\|_F + c_5 \|\tilde{V}\|_F \|r\|$$

where c_i are computable positive constants. ■

Fact 4 is direct to show using (19), some standard norm inequalities, and the fact that $\sigma(\cdot)$ and its derivative are bounded by constants for RBF, sigmoid, and tanh.

The extension of these ideas to nets with greater than three layers is not difficult, and leads to composite function terms in the Taylor series (giving rise to backpropagation filtered error terms for the multilayer net case—see Theorem 3.1).

B. Controller Structure and Error System Dynamics

Define the NN functional estimate of (11) by

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \tag{25}$$

with $\hat{V}, \hat{W}$ the current (estimated) values of the ideal NN weights V, W as provided by the tuning algorithms subsequently to be discussed. With τ_o defined in (13), select the control input

$$\tau = \tau_o - v = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v \tag{26}$$

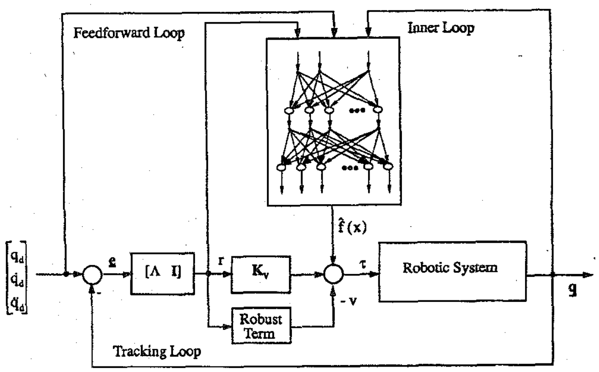


Fig. 2. NN control structure.

with $v(t)$ a function to be detailed subsequently that provides robustness in the face of higher-order terms in the Taylor series. The proposed NN control structure is shown in Fig. 2, where $\underline{q} \equiv [q^T \dot{q}^T]^T, \underline{e} \equiv [e^T \dot{e}^T]^T$.

Using this controller, the closed-loop filtered error dynamics become

$$M\dot{r} = -(K_v + V_m)r + W^T\sigma(V^Tx) - \hat{W}^T\sigma(\hat{V}^Tx) + (\varepsilon + \tau_d) + v.$$

Adding and subtracting $W^T\hat{\sigma}$ yields

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T\hat{\sigma} + W^T\tilde{\sigma} + (\varepsilon + \tau_d) + v$$

with $\hat{\sigma}$ and σ defined in (21). Adding and subtracting now $\hat{W}^T\tilde{\sigma}$ yields

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T\hat{\sigma} + \hat{W}^T\tilde{\sigma} + \tilde{W}^T\tilde{\sigma} + (\varepsilon + \tau_d) + v. \tag{27}$$

The key step is the use now of the Taylor series approximation (23) for $\tilde{\sigma}$, according to which the closed-loop error system is

$$M\dot{r} = -(K_v+V_m)r+\tilde{W}^T\hat{\sigma}+\hat{W}^T\hat{\sigma}'\tilde{V}^Tx+w_1+v \tag{28}$$

where the disturbance terms are

$$w_1(t) = \tilde{W}^T\hat{\sigma}'\tilde{V}^Tx + W^TO(\tilde{V}^Tx)^2 + (\varepsilon + \tau_d). \tag{29}$$

Unfortunately, using this error system does not yield a compact set outside which a certain Lyapunov function derivative is negative. Therefore, write finally the error system

$$\begin{aligned} M\dot{r} &= -(K_v + V_m)r + \tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^Tx) \\ &\quad + \hat{W}^T\hat{\sigma}'\tilde{V}^Tx + w + v \\ &\equiv -(K_v + V_m)r + \zeta_1 \end{aligned} \tag{30}$$

where the disturbance terms are

$$w(t) = \tilde{W}^T\hat{\sigma}'V^Tx + W^TO(\tilde{V}^Tx)^2 + (\varepsilon + \tau_d). \tag{31}$$

It is important to note that the NN reconstruction error $\varepsilon(x)$, the robot disturbances τ_d, and the higher-order terms in the Taylor series expansion of $f(x)$ all have exactly the same influence as disturbances in the error system. The next key bound is required. Its importance is in allowing one to overbound $w(t)$ at each time by a known computable function; it follows from Fact 4 and some standard norm inequalities.

Fact 5: The disturbance term (31) is bounded according to

$$\|w(t)\| \le (\varepsilon_N + b_d + c_3 Z_M) + c_6 Z_M\|\tilde{Z}\|_F + c_7 Z_M\|\tilde{z}\|_f\|r\|$$

or

$$\|w(t)\| \le c_0 + c_1\|\tilde{Z}\|_F + C_2\|\tilde{Z}\|_F\|r\| \tag{32}$$

with C_i computable known positive constants. ■

C. Weight Updates for Guaranteed Tracking Performance

We give here some NN weight-tuning algorithms that guarantee the tracking stability of the closed-loop system under various assumptions. It is required to demonstrate that the tracking error $r(t)$ is suitably small and that the NN weights $\hat{V}, \hat{W}$ remain bounded, for then the control $\tau(t)$ is bounded. The key features of all our algorithms are that stability is guaranteed, there is no off-line learning phase so that NN control begins immediately, and the NN weights are very easy to initialize without the requirement for "initial stabilizing weights."

Ideal Case—Backpropagation Tuning of Weights: The next result details the closed-loop behavior in a certain idealized case that demands: 1) no net functional reconstruction error; 2) no unmodeled disturbances in the robot arm dynamics; and 3) no higher-order Taylor series terms. The last amounts to the assumption that $f(x)$ in (10) is linear. In this case the tuning rules are straightforward and familiar. Our contribution lies in the proof and the conditions thus determined showing when the algorithm works, and when it cannot be relied on.

Theorem 3.1: Let the desired trajectory be bounded and suppose the disturbance term $w_1(t)$ in (28) is equal to zero. Let the control input for (7) be given by (26) with $v(t) = 0$ and weight tuning provided by

$$\dot{\hat{W}} = F\hat{\sigma}r^T \tag{33}$$

$$\dot{\hat{V}} = Gx(\hat{\sigma}'^T\hat{W}r)^T \tag{34}$$

and any constant positive definite (design) matrices F, G. Then the tracking error $r(t)$ goes to zero with t and the weight estimates $\hat{V}, \hat{W}$ are bounded.

Proof: Define the Lyapunov function candidate

$$L = \tfrac{1}{2}r^TMr + \tfrac{1}{2}\mathrm{tr}\,(\tilde{W}^TF^{-1}\tilde{W}) + \tfrac{1}{2}\mathrm{tr}\,(\tilde{V}^TG^{-1}\tilde{V}). \tag{35}$$

Differentiating yields

$$\dot{L} = r^TM\dot{r} + \tfrac{1}{2}r^T\dot{M}r + \mathrm{tr}\,(\tilde{W}^TF^{-1}\dot{\tilde{W}}) + \mathrm{tr}\,(\tilde{V}^TG^{-1}\dot{\tilde{V}})$$

whence substitution from (28) (with $w_1 = 0, v = 0$) yields

$$\begin{aligned} \dot{L} &= -r^TK_vr + \tfrac{1}{2}r^T(\dot{M} - 2V_m)r + \mathrm{tr}\,\tilde{W}^T(F^{-1}\dot{\tilde{W}} + \hat{\sigma}r^T) \\ &\quad + \mathrm{tr}\,\tilde{V}^T(G^{-1}\dot{\tilde{V}} + xr^T\hat{W}^T\hat{\sigma}'). \end{aligned}$$

The skew symmetry property makes the second term zero, and since $\hat{W} = W - \tilde{W}$ with W constant, so that $d\tilde{W}/dt = -d\hat{W}/dt$ (and similarly for V), the tuning rules yield

$$\dot{L} = -r^TK_vr.$$

Since $L > 0$ and $\dot{L} \leq 0$ this shows stability in the sense of Lyapunov so that $r, \tilde{V}$, and $\tilde{W}$ (and hence $\hat{V}, \hat{W}$) are bounded. Moreover

$$\int_0^\infty -\dot{L}\, dt < \infty. \tag{36}$$

LaSalle's extension [17], [41] is now used to show that $r(t)$ in fact goes to zero. Boundedness of r guarantees the boundedness of e and $\dot{e}$, whence boundedness of the desired trajectory shows $q, \dot{q}, x$ are bounded. Property 2 then shows boundedness of $V_m(q, \dot{q})$. Now, $\ddot{L} = -2r^T K_v \dot{r}$, and the boundedness of $M^{-1}(q)$ and of all signals on the right-hand side of (28) verify the boundedness of $\ddot{L}$, and hence the uniform continuity of $\dot{L}$. This allows one to invoke Barbalat's Lemma [17], [41] in connection with (36) to conclude that $\dot{L}$ goes to zero with t, and hence that $r(t)$ vanishes. ■

Note that the problem of net weight initialization occurring in other approaches in the literature does not arise. In fact, selecting the initial weights $\hat{W}(0), \hat{V}(0)$ as zero takes the NN out of the circuit and leaves only the outer tracking loop in Fig. 2. It is well known that the PD term $K_v r$ in (26) can then stabilize the plant on an interim basis. A formal proof reveals that K_v should be large enough and the initial filtered error $r(0)$ small enough. The exact value of K_v needed for initial stabilization is given in [9], though for practical purposes it is only necessary to select K_v large.

Note next that (33) and (34) are nothing but the continuous-time version of the backpropagation algorithm. In the scalar sigmoid case, for instance

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)) \tag{37}$$

so that

$$\hat{\sigma}'^T \hat{W} r = \sigma(\hat{V}^T x)[1 - \sigma(\hat{V}^T x)]\hat{W} r \tag{38}$$

which is the filtered error weighted by the current estimate $\hat{W}$ and multiplied by the usual product involving the hidden-layer outputs.

Theorem 3.1 indicates when backprop alone should suffice; namely, when the disturbance $w_1(t)$ is equal to zero. Observing the first term in (29) reveals that this is a stronger assumption than simply linearity of the robot function $f(x)$ in (10). That is, even when $\varepsilon(x) = 0, \tau_d = 0$, and $f(x)$ is linear, backprop tuning is not guaranteed to afford successful tracking of the desired trajectory. Note that $f(x)$ is linear only in the one-link robot arm case. The assumption $w_1(t) = 0$ means, moreover, that the NN can exactly approximate the required function over all of $\boldsymbol{R}^n$. This is a strong assumption.

Theorem 3.2 further reveals the failure of simple backpropagation in the general case. In fact, in the two-layer NN case $V = I$ (i.e., linear in the parameters), it is easy to show that, using update rule (33), the weights $\hat{W}$ are not generally bounded unless the hidden-layer output $\sigma(x)$ obeys a stringent PE condition [18]. In the three-layer (nonlinear) case, PE conditions are not easy to derive as one is faced with the observability properties of a certain bilinear system. Thus, backpropagation used in a net that cannot exactly reconstruct $f(x)$, or on a robot arm with bounded unmodeled disturbances, or when $f(x)$ is nonlinear, cannot be guaranteed to yield bounded weights in the closed-loop system.

General Case: To confront the stability and tracking performance of a NN robot arm controller in the thorny general case, we require: 1) the modification of the weight tuning rules and 2) the addition of a robustifying term $v(t)$. The problem in this case is that, though it is not difficult to conclude that $r(t)$ is bounded, it is impossible without these modifications to show that the NN weights are bounded in general. Boundedness of the weights is needed to verify that the control input $\tau(t)$ remains bounded.

The next theorem relies on an extension to Lyapunov theory. The disturbance τ_d, the NN reconstruction error ε, and the nonlinearity of $f(x)$ make it impossible to show that the Lyapunov derivative $\dot{L}$ is nonpositive for all $r(t)$ and weight values. In fact, it is only possible to show that $\dot{L}$ is negative outside a compact set in the state space. This, however, allows one to conclude boundedness of the tracking error and the neural net weights. In fact, explicit bounds are discovered during the proof. The required Lyapunov extension is [17, Theorem 1.5–6], the last portion of our proof being similar to the proof used in [26].

Theorem 3.2: Let the desired trajectory be bounded by (18). Take the control input for (7) as (26) with robustifying term

$$v(t) = -K_Z(\|\hat{Z}\|_F + Z_M)r \tag{39}$$

and gain

$$K_z > C_2 \tag{40}$$

with C_2 the known constant in (32). Let NN weight tuning be provided by

$$\dot{\hat{W}} = F\hat{\sigma}r^T - F\hat{\sigma}'\hat{V}^T x r^T - \kappa F\|r\|\hat{W} \tag{41}$$

$$\dot{\hat{V}} = Gx(\hat{\sigma}'^T \hat{W} r)^T = \kappa G\|r\|\hat{V} \tag{42}$$

with any constant matrices $F = F^T > 0, G = G^T > 0$, and scalar design parameter $\kappa > 0$. Then, for large enough control gain K_v, the filtered tracking error $r(t)$ and NN weight estimates $\hat{V}, \hat{W}$ are UUB, with practical bounds given specifically by the right-hand sides of (43) and (44). Moreover, the tracking error may be kept as small as desired by increasing the gains K_v in (26).

Proof: Let the approximation property (3) hold with a given accuracy ε_N for all x in the compact set $U_x \equiv \{x \| x\| \leq b_x\}$ with $b_x > c_1 Q_d$ in (19). Define $U_r = \{r \| r\| \leq (b_x - c_1 Q_d)/c_2\}$. Let $r(0) \in U_r$. Then the approximation property holds.

Selecting now the Lyapunov function (35), differentiating, and substituting now from the error system (30) yields

$$\begin{aligned}\dot{L} = &-r^T K_v r + \tfrac{1}{2} r^T(\dot{M} - 2V_m)r \\ &+ \operatorname{tr} \tilde{W}^T(F^{-1}\dot{\tilde{W}} + \hat{\sigma}r^T - \hat{\sigma}'\hat{V}^T x r^T) \\ &+ \operatorname{tr} \tilde{V}^T(G^{-1}\dot{\tilde{V}} + x r^T \hat{W}^T \hat{\sigma}') + r^T(w + v).\end{aligned}$$

The tuning rules give

$$\begin{aligned}\dot{L} &= -r^T K_v r + \kappa\|r\| \mathrm{tr}\ \tilde{W}^T(W-\tilde{W}) \\ &\quad + \kappa\|r\|\mathrm{tr}\ \tilde{V}^T(V-\tilde{V}) + r^t(w+v) \\ &= -r^T K_v r + \kappa\|r\|\mathrm{tr}\ \tilde{Z}^T(Z-\tilde{Z}) + r^T(w+v).\end{aligned}$$

Since $tr\ \tilde{Z}^T(Z-\tilde{Z}) = \langle \tilde{Z}, Z\rangle_F - \|\tilde{Z}\|_F^2 \le \|\tilde{Z}\|_F\|Z\|_F - \|\tilde{Z}\|_F^2$, there results

$$\begin{aligned}\dot{L} &\le -K_{v\min}\|r\|^2 + \kappa\|r\|\|\tilde{Z}\|_F(Z_M - \|\tilde{Z}\|_F) \\ &\quad - K_Z(\|\hat{Z}\|_F + Z_M)\|r\|^2 + \|r\|\|w\| \\ &\le -K_{v\min}\|r\|^2 + \kappa\|r\|\|\tilde{z}\|_F(Z_M - \|\tilde{Z}\|_F) \\ &\quad - K_Z(\|\hat{Z}\|_F + Z_M)\|r\|^2 \\ &\quad + \|r\|[C_0 + C_1\|\tilde{Z}\|_F + C_2\|\tilde{Z}\|_F\|r\|] \\ &\le -\|r\|[K_{v\min}\|r\| + \kappa\|\tilde{Z}\|_F(\|\tilde{Z}\|_F - Z_M) \\ &\quad - C_0 - C_1\|\tilde{Z}\|_F]\end{aligned}$$

where $K_{v\min}$ is the minimum singular value of K_v and the last inequality holds due to (40). Thus, $\dot{L}$ is negative as long as the term in braces is positive.

Defining $C_3 = Z_M + C_1/\kappa$ and completing the square yields

$$\begin{aligned}K_{v\min}&\|r\| + \kappa\|\tilde{Z}\|_F(\|\tilde{Z}\|_F - C_2) - C_0 \\ &= \kappa(\|\tilde{Z}\|_F - C_3/2)^2 - \kappa C_3^2/4 + K_{v\min}\|r\| - C_0\end{aligned}$$

which is guaranteed positive as long as either

$$\|r\| > \frac{\kappa C_3^2/4 + C_0}{K_{v\min}} \equiv b_r \tag{43}$$

or

$$\|\tilde{Z}\|_F > C_3/2 + \sqrt{C_3^2/4 + C_0/\kappa} \equiv b_Z \tag{44}$$

where

$$C_3 = Z_M + C_1/\kappa. \tag{45}$$

Thus, $\dot{L}$ is negative outside a compact set. The form of the right-hand side of (43) shows that the control gain K_v can be selected large enough so that $b_r < (b_x - c_1 Q_d)/c_2$. Then, any trajectory $r(t)$ beginning in U_r evolves completely within U_r. According to a standard Lyapunov theorem extension [17], [26], this demonstrates the UUB of both $\|r\|$ and $\|\tilde{Z}\|_F$. ■

Some remarks are in order. First, since any excursions of $\|r\|$ or $\|\tilde{Z}\|_F$ beyond the bounds given in (43) and (44), respectively, lead to a decrease in the Lyapunov function L, it follows that the right-hand sides of (43) and (44) can be taken as practical bounds on $\|r\|$ and $\|\tilde{Z}\|_F$, respectively, in the sense that the norms will be restricted to these values plus ε for any arbitrarily small $\varepsilon > 0$.

A comparison with the results of [26] for adaptive control shows that the NN reconstruction error ε_N, the bounded disturbances b_d, and the higher-order Taylor series terms, all embodied in the constants C_0, C_3, increase the bounds on $\|r\|$ and $\|\tilde{Z}\|_F$ in a very interesting way. Note from (43), however, that arbitrarily small tracking error bounds may be achieved by selecting large control gains K_v. (If K_v is taken as a diagonal matrix, $K_{v\min}$ is simply the smallest gain element.) On the other hand, (44) reveals that the NN weight errors are fundamentally bounded by Z_M (through C_3). The parameter κ offers a design trade-off between the relative eventual magnitudes of $\|r\|$ and $\|\tilde{Z}\|_F$.

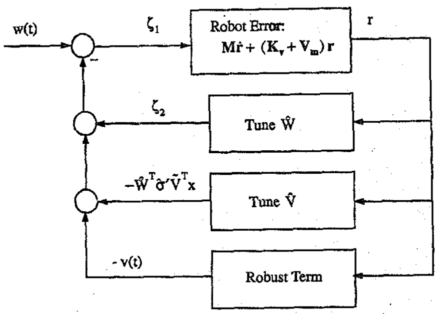


Fig. 3. Neural net closed-loop error system.

The first terms of (41) and (42) are nothing but the standard backpropagation algorithm. The last terms correspond to the e-modification [26] in standard use in adaptive control to guarantee bounded parameter estimates; they form a special sort of forgetting term in the weight updates. The second term in (41) is novel and bears discussion. The standard backprop terms can be thought of as backward propagating signals in a nonlinear "backprop" network [27] that contains multipliers. The second term in (41) seems to correspond to a forward travelling wave in the backprop net that provides a second-order correction to the weight tuning for $\hat{W}$.

Note that there is design freedom in the degree of complexity (e.g., size) of the NN. For a more complex NN (e.g., more hidden units), the bounding constants will decrease, resulting in smaller tracking errors. On the other hand, a simplified NN with fewer hidden units will result in larger error bounds; this degradation can be compensated for, as long as bound ε_N is known, by selecting a larger value for K_z in the robustifying signal $v(t)$, or for Λ in (9).

An alternative to guaranteeing the boundedness of the NN weights for the two-layer case (i.e., linear in the parameters) is presented in [32], [33], and [35], where projection algorithms are used for tuning $\hat{W}$. In the nonlinear case a deadzone tuning algorithm is given in [22].

IV. Passivity Properties of the NN

The closed-loop error system appears in Fig. 3, with the signal ζ_2 defined as

$$\begin{aligned}&\zeta_2(t) = -\tilde{W}^T\hat{\sigma}, \quad \text{for error system (28)} \\ &\zeta_2(t) = -\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x), \quad \text{for error system (30).}\end{aligned} \tag{46}$$

(In the former case, signal $w(t)$ should be replaced by $w_1(t)\cdot$) Note the role of the NN, which is decomposed into two effective blocks appearing in a typical feedback configuration, in contrast to the role of the NN in the controller in Fig. 2.

Passivity is important in a closed-loop system as it guarantees the boundedness of signals, and hence suitable performance, even in the presence of additional unforseen disturbances as long as they are bounded. In general, an NN cannot be guaranteed to be passive. The next results show, however, that the weight tuning algorithms given here do in fact guarantee desirable passivity properties of the NN, and hence of the closed-loop system.

The first result is with regard to error system (28).

Theorem 4.1: The backprop weight tuning algorithms (33), (34) make the map from $r(t)$ to $-\tilde{W}^T\hat{\sigma}$, and the map from $r(t)$ to $-\hat{W}^T\hat{\sigma}'\tilde{V}^T x$, both passive maps.

Proof: The dynamics with respect to $\tilde{W}, \tilde{V}$ are

$$\dot{\tilde{W}} = -F\hat{\sigma}r^T \quad (47)$$

$$\dot{\tilde{V}} = -Gx(\hat{\sigma}'^T\hat{W}r)^T. \quad (48)$$

1) Selecting the nonnegative function

$$L = \tfrac{1}{2}\text{tr } \tilde{W}^T F^{-1}\tilde{W}$$

and evaluating $\dot{L}$ along the trajectories of (47) yields

$$\dot{L} = \text{tr } \tilde{W}^T F^{-1}\dot{\tilde{W}} = -\text{tr } \tilde{W}^T\hat{\sigma}r^T = r^T(-\tilde{W}^T\hat{\sigma})$$

which is in power form (4).

2) Selecting the nonnegative function

$$L = \tfrac{1}{2}\text{tr } \tilde{V}^T G^{-1}\tilde{V}$$

and evaluating $\dot{L}$ along the trajectories of (48) yields

$$\dot{L} = \text{tr } \tilde{V}^T G^{-1}\dot{\tilde{V}} = -\text{tr } \tilde{V}^T x(\hat{\sigma}'^T\hat{W}r)^T = r^T(-\hat{W}^T\hat{\sigma}'\tilde{V}^T x).$$

which is in power form.

■

Thus, the robot error system in Fig. 3 is state strict passive (SSP) and the weight error blocks are passive; this guarantees the dissipativity of the closed-loop system [41]. Using the passivity theorem one may now conclude that the input–output signals of each block are bounded as long as the external inputs are bounded.

Unfortunately, though dissipative, the closed-loop system is not SSP so, when disturbance $w_1(t)$ is nonzero, this does not yield boundedness of the internal states of the weight blocks (i.e., $\tilde{W}, \tilde{V}$) unless those blocks are observable, that is persistently exciting (PE). Unfortunately, this does not yield a convenient method for defining PE in a three-layer NN, as the two weight-tuning blocks are coupled, forming in fact a bilinear system. By contrast, PE conditions for the two-layer case $V = I$ (i.e., linear NN) are easy to deduce [18].

The next result shows why a PE condition is not needed with the modified weight update algorithm of Theorem 3.2; it is in the context of error system (30).

Theorem 4.2: The modified weight tuning algorithms (41), (42) make the map from $r(t)$ to $-\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)$, and the map from $r(t)$ to $-\hat{W}^T\hat{\sigma}'\tilde{V}^T x$, both SSP maps.

Proof: The revised dynamics relative to $\tilde{W}, \tilde{V}$ are given by

$$\dot{\tilde{W}} = -F\hat{\sigma}r^T + F\hat{\sigma}'\hat{V}^T x r^T + \kappa F\|r\|\hat{W} \quad (49)$$

$$\dot{\tilde{V}} = -Gx(\hat{\sigma}'^T\hat{W}r)^T + \kappa G\|r\|\hat{V}. \quad (50)$$

1) Selecting the nonnegative function

$$L = \tfrac{1}{2}\text{tr } \tilde{W}^T F^{-1}\tilde{W}$$

and evaluating $\dot{L}$ yields

$$\begin{aligned}\dot{L} &= \text{tr } \tilde{W}^T F^{-1}\dot{\tilde{W}} \\ &= \text{tr } \{[-\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)]r^T + \kappa\|r\|\tilde{W}^T\hat{W}\}.\end{aligned}$$

Since $\text{tr } \tilde{W}^T(W - \tilde{W}) = \langle\tilde{W}, W\rangle_F - \|\tilde{W}\|_F^2 \le \|\tilde{W}\|_F\|W\|_F - \|\tilde{W}\|_F^2$ there results

$$\begin{aligned}\dot{L} &\le r^T[-\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)] \\ &\quad -\kappa\|r\|(\|\tilde{W}\|_F^2 - \|\tilde{W}\|_F\|W\|_F) \\ &\le r^T[-\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)] - \kappa\|r\|(\|\tilde{W}\|_F^2 - W_M\|\tilde{W}\|_F)\end{aligned}$$

which is in power form with the last function quadratic in $\|\tilde{W}\|_F$.

2) Selecting the nonnegative function

$$L = \tfrac{1}{2}\text{tr } \tilde{V}^T G^{-1}\tilde{V}$$

and evaluating $\dot{L}$ yields

$$\begin{aligned}\dot{L} &= \text{tr } \tilde{V}^T G^{-1}\dot{\tilde{V}} = r^T(-\hat{W}^T\hat{\sigma}'\tilde{V}^T x) \\ &\quad - \kappa\|r\|(\|\tilde{V}\|_F^2 - \langle\tilde{V}, V\rangle_F) \\ &\le r^T(-\hat{W}^T\hat{\sigma}'\tilde{V}^T x) - \kappa\|r\|(\|\tilde{V}\|_F^2 - V_M\|\tilde{V}\|_F)\end{aligned}$$

which is in power form with the last function quadratic in $\|\tilde{V}\|_F$.

■

It is exactly the special forms of $\dot{L}$ that allow one to show the boundedness of $\tilde{W}$ and $\tilde{V}$ when the first terms (power input) are bounded, as in the proof of Theorem 3.2.

It should be noted that SSP of both the robot dynamics and the weight-tuning blocks does guarantee SSP of the closed-loop system, so that the norms of the internal states are bounded in terms of the power delivered to each block. Then, boundedness of input–output signals assures state boundedness without any sort of observability requirement.

We define an NN as passive if, in the error formulation, it guarantees the passivity of the weight-tuning subsystems. Then, an extra PE condition is needed to guarantee boundedness of the weights [18]. We define an NN as robust if, in the error formulation, it guarantees the SSP of the weight-tuning subsystem. Then, no extra PE condition is needed for boundedness of the weights. Note that 1) SSP of the open-loop plant error system is needed in addition for tracking stability and 2) the NN passivity properties are dependent on the weight tuning algorithm used.

V. Illustrative Design and Simulation

A planar two-link arm used extensively in literature for illustration purposes appears in Fig. 4. The dynamics are given, for instance in [17]; no friction term was used in this example. This arm is simple enough to simulate conveniently, yet contains all the nonlinear terms arising in general n-link manipulators. The joint variable is $q = [q_1 q_2]^T$. We should like to illustrate the NN control scheme derived herein, which will require no knowledge of the dynamics, not even their structure which is needed for adaptive control.

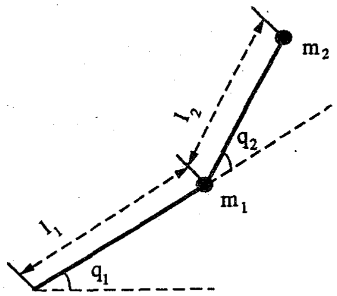
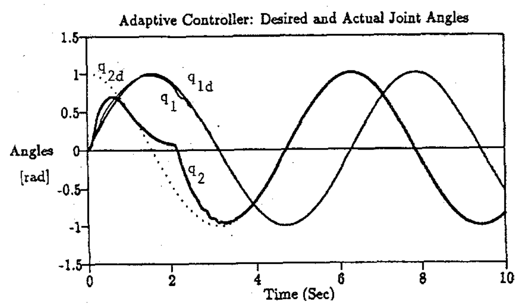
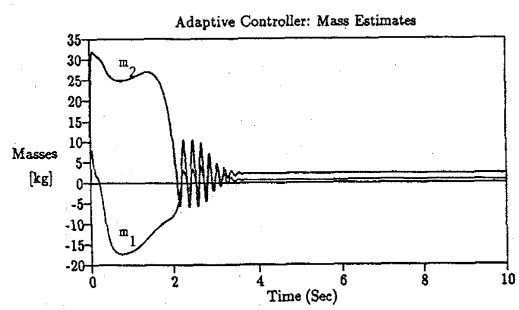


Fig. 4. Two-link planar elbow arm.



(a)



(b)

Fig. 5. Response of adaptive controller. (a) Actual and desired joint angles. (b) Parameter estimates.

Adaptive Controller—Baseline Design: For comparison, a standard adaptive controller is given by [40]

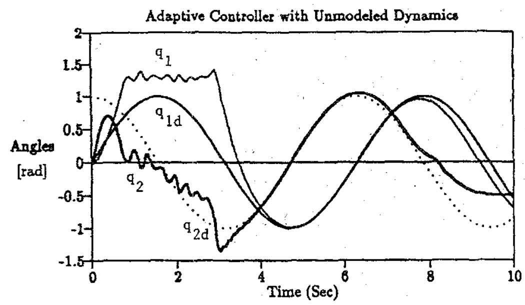
$$\tau = Y\hat{\Psi} + K_v r \tag{51}$$

$$\dot{\hat{\Psi}} = FY^T r \tag{52}$$

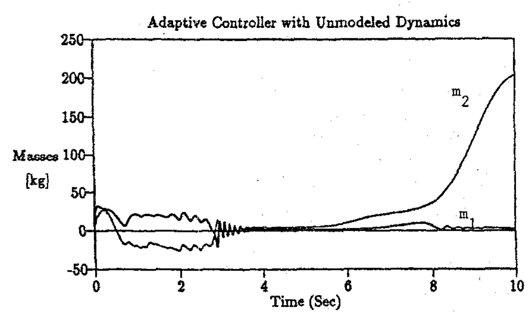
with $F = F^T > 0$ a design parameter matrix, $Y(e, \dot{e}, q_d, \dot{q}_d, \ddot{q}_d)$ a fairly complicated matrix of robot functions that must be explicitly derived from the dynamics for each arm, and Ψ the vector of unknown parameters, in this case simply the link masses m_1, m_2.

We took the arm parameters as $\ell_1 = \ell_2 = 1m, m_1 = 0.8$ kg, $m_2 = 2.3$ kg, and selected $q_{1d}(t) = \sin\, t, q_{2d}(t) = \cos\, t$, $K_v = \text{diag}\ \{20, 20\}, F = \text{diag}\ \{10, 10\}, = \Lambda = \text{diag}\ \{5, 5\}$. The response with this controller when $q(0) = 0, \dot{q}(0) = 0, \hat{m}_1(0) = 0, \hat{m}_2(0) = 0$ is shown in Fig. 5.

Note the good behavior, which obtains since there are only two unknown parameters, so that the single mode (e.g., two poles) of $q_d(t)$ guarantees PE [11].



(a)



(b)

Fig. 6. Response of adaptive controller with unmodeled dynamics. (a) Actual and desired joint angles. (b) Parameter estimates.

The (1, 1) entry of the robot function matrix $Y$is $\ell_1^2(\ddot{q}_{d1} + \lambda_1 \dot{e}_1) + \ell_1 g\, \cos\, q_1$ (with $\Lambda = \text{diag}\{\lambda_1, \lambda 2\}$). To demonstrate the deleterious effects of unmodeled dynamics in adaptive control, the term $\ell_1 g \cos q_1$ was now dropped in the controller. The result appears in Fig. 6 and is unsatisfactory. It is emphasized that in the NN controller all the dynamics are unmodeled.

NN Controller with Backprop Weight Tuning: Some preprocessing of signals yields a more advantageous choice for $x(t)$ than (12) that already contains some of the nonlinearities inherent to robot arm dynamics. Since the only occurrences of the revolute joint variables are as sines and cosines, the vector x can be taken as

$$x = [\zeta_1^T \zeta_2^T\ \cos\,(q)^T\ \sin\,(q)^T \dot{q}^T\ \text{sgn}\,(\dot{q})^T]^T \tag{53}$$

where $\zeta_1 = \ddot{q}_d + \Lambda\dot{e}, \zeta_2 = \dot{q}_d + \Lambda e$ and the signum function is needed in the friction terms. The NN controller appears in Fig. 2.

The response of the controller (26) (with $v(t) = 0$) with backprop weight tuning (e.g., Theorem 3.1) appears in Fig. 7. The sigmoid activation functions were used, and 10 hidden-layer neurons. The values for $q_d(t), \Lambda, F, K_v$ were the same as before, and we selected $G = \text{diag}\{10, 10\}$. In this case the NN weights appear to remain bounded, though this cannot in general be guaranteed.

The choice of 10 hidden-layer neurons was made as follows. Three simulations were performed, using five, 10, then 15 hidden-layer neurons. It was observed that going from five–10 neurons significantly improved the performance, but going from 10–15 neurons made no perceptible improvement. It is

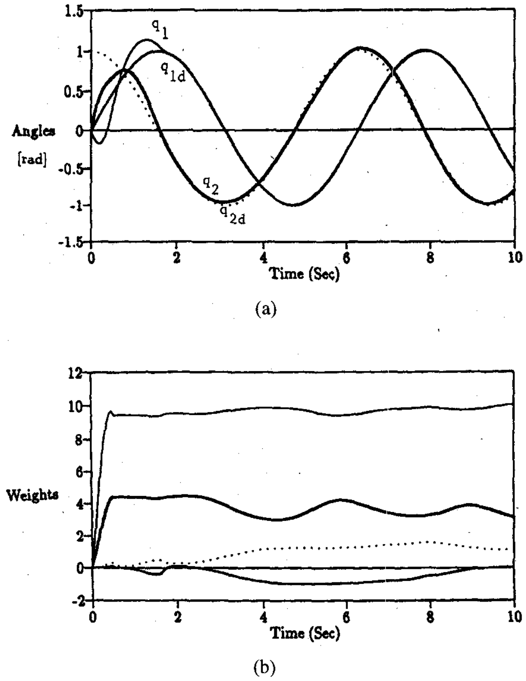


Fig. 7. Response of NN controller with backprop weight tuning. (a) Actual and desired joint angles. (b) Representative weight estimates.

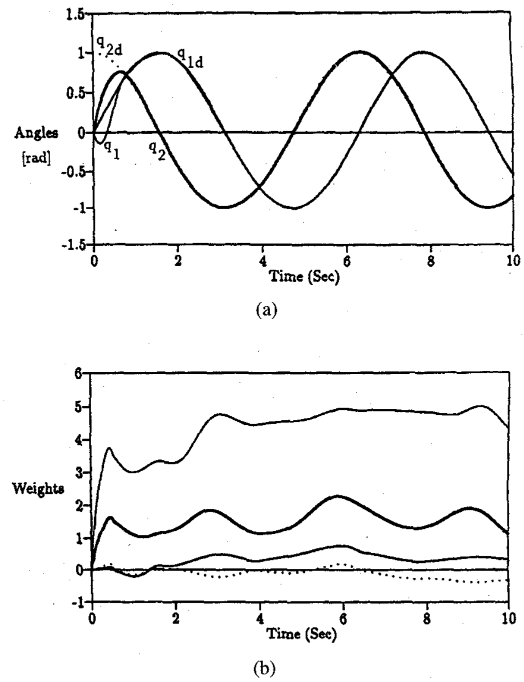


Fig. 8. Response of NN controller with improved weight tuning. (a) Actual and desired joint angles. (b) Representative weight estimates.

easy in our software to specify a new number of hidden-layer neurons, and each simulation run took about one minute of computer time. Similar simulation iterations may be made to determine effective numbers for the various design gains.

NN Controller with Improved Weight Tuning: The response of the controller (26) with the improved weight tuning in Theorem 3.2 appears in Fig. 8, where we took $\kappa = 0.1$. The tracking response is better than that using straight backprop, and the weights are guaranteed to remain bounded even though PE may not hold. The comparison with the performance of the standard adaptive controller in Fig. 5 is impressive, even though the dynamics of the arm were not required to implement the NN controller.

No initial NN training or learning phase was needed. The NN weights were simply initialized at zero in this figure.

To study the contribution of the NN, Fig. 9 shows the response with the controller $\tau = K_v r$, that is, with no neural net. Standard results in the robotics literature indicate that such a PD controller should give bounded errors if K_v is large enough. This is observed in the figure. It is very clear, however, that the addition of the NN makes a very significant improvement in the tracking performance.

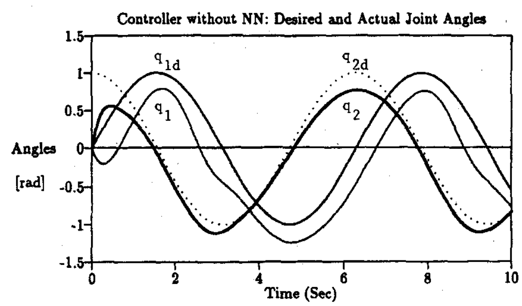


Fig. 9. Response of controller without NN. Actual and desired joint angles.

VI. Conclusion

A multilayer (nonlinear) NN controller for a serial-link robot arm was developed. The NN controller has a structure derived from robot control theory passivity notions and offers guaranteed tracking behavior. Backpropagation tuning was shown to yield a passive NN, so that it only performs well under ideal conditions that require linear robot dynamics, no NN reconstruction error, and no robot unmodeled disturbances.

An improved weight tuning algorithm was derived to correct these deficiencies. The improved algorithm consists of a backprop term, plus the e-modification term from adaptive control, plus a novel second-order forward propagating wave from the backprop network. A robustifying control term is also needed to overcome higher-order modeling error terms. The improved tuning algorithm makes the NN strictly state passive, so that bounded weights are guaranteed in practical nonideal situations.

No NN off-line learning or training phase was needed; simply initializing the NN weights at zero made for fast convergence and bounded errors. Structured or partitioned NN's can be used to simplify the controller design as well as make for faster weight updates [21].

References

[1] F. Albertini and E. D. Sontag, "For neural nets, function determines form," in *Proc. IEEE Conf. Decision Contr.*, Dec. 1992, pp. 26–31.

[2] E. Alcorta and E. N. Sánchez, "On-line function approximation via neural networks," preprint, 1992.

[3] R. G. Bartle, *The Elements of Real Analysis.* New York: Wiley, 1964.

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

[5] S. R. Chu and R. Shoureshi, "Neural-based adaptive nonlinear system identification," *Intelligent Control Systems,* DSC-vol. 45, ASME Winter Annual Meeting, 1992.

[6] J. J. Craig, *Adaptive Control of Robot Manipulators.* Reading, MA: Addison-Wesley, 1988.

[7] X. Cui and K. G. Shin, "Direct control and coordination using neural networks," *IEEE Trans. Syst., Man, Cybern.,* vol. 23, no. 3, May/June 1993.

[8] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Contr. Signals Syst.,* vol. 2, no. 4, pp. 303–314, 1989.

[9] D. M. Dawson, Z. Qu, F. L. Lewis, and J. F. Dorsey, "Robust control for the tracking of robot motion," *Int. J. Contr.,* vol. 52, no. 3, pp. 581–595, 1990.

[10] G. C. Goodwin, "Can we identify adaptive control?" in *Proc. European Contr. Conf.,* July 1991, pp. 1714–1725.

[11] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction, and Control.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[12] B. Horn, D. Hush, and C. Abdallah, "The state space recurrent neural network for robot identification," *Advanced Control Issues for Robot Manipulators,* DSC-vol. 39, ASME Winter Annual Meeting, 1992.

[13] K. Hornik, M. Stinchombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks,* vol. 2, pp. 359–366, 1989.

[14] Y. Iiguni, H. Sakai, and H. Tokumaru, "A nonlinear regulator design in the presence of system uncertainties using multilayer neural networks," *IEEE Trans. Neural Networks,* vol. 2, no. 4, pp. 410–417, July 1991.

[15] L. G. Kraft and D. P. Campagna, "A summary comparison of CMAC neural network and traditional adaptive control systems," in *Neural Networks for Control,* W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1991, pp. 143–169.

[16] Y. D. Landau, *Adaptive Control: The Model Reference Approach.* New York: Marcel Dekker, 1979.

[17] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators.* New York: Macmillan, 1993.

[18] F. L. Lewis, K. Liu, and A. Yeşildirek, "Neural net robot controller with guaranteed tracking performance," in *Proc. IEEE Int. Symp. Intell. Contr.,* Chicago, IL, Aug. 1993, pp. 225–231.

[19] F. L. Lewis, A. Yeşildirek, and K. Liu, "Neural net robot controller: Structure and stability proofs," in *Proc. IEEE Conf. Decision Contr.,* San Antonio, TX, Dec. 1993, pp. 2785–2791.

[20] ——, "Neural net robot controller: Structure and stability proofs," *J. Intell. Robotic Syst.,* to appear.

[21] F. L. Lewis, K. Liu, and A. Yeşildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks,* to appear.

[22] C.-C. Liu and F.-C. Chen, "Adaptive control of nonlinear continuous-time systems using neural networks general relative degree and MIMO cases," *Int. J. Contr.,* vol. 58, no. 2, pp. 317–335, 1993.

[23] K. Liu and F. L. Lewis, "Robust control techniques for general dynamic systems," *J. Intell. Robotic Syst.,* vol. 6, pp. 33–49, 1992.

[24] W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds., *Neural Networks for Control.* Cambridge: MIT Press, 1991.

[25] K. S. Narendra, "Adaptive control using neural networks," *Neural Networks for Control,* W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge: MIT Press, 1991, pp. 115–142.

[26] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Automat. Contr.,* vol. AC-32, no. 2, pp. 134–145, Feb. 1987.

[27] 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.

[28] J. Olvera, X. Guan, and M. T. Manry, "Theory of monomial networks," in *Proc. Symp. Implicit Nonlinear Syst.,* Dec. 1992, pp. 96–101.

[29] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma, and Y. Uchikawa, "Trajectory control of robotic manipulators," *IEEE Trans. Ind. Electron.,* vol. 38, pp. 195–202, June 1991.

[30] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks.* Reading, MA: Addison-Wesley, 1989.

[31] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computa.,* vol. 3, pp. 246–257, 1991.

[32] M. M. Polycarpou and P. A. Ioannu, "Identification and control using neural network models: design and stability analysis," Dep. Elect. Eng. Syst., Univ. S. Cal., Tech. Rep. 91-09-01, Sept. 1991.

[33] ——, "Neural networks as on-line approximators of nonlinear systems," in *Proc. IEEE Conf. Decision Contr.,* Tucson, AZ, Dec. 1992, pp. 7–12.

[34] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," Artificial Intell. Lab, Massachusettes Inst. Technol., Tech. Rep. 1140, 1989.

[35] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Syst., Man, Cybern.,* to appear, 1994.

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

[37] N. Sadegh, "Nonlinear identification and control via neural networks," *Control Systems with Inexact Dynamics Models,* DSC-vol. 33, ASME Winter Annual Meeting, 1991.

[38] R. M. Sanner and J.-J. E. Slotine, "Stable adaptive control and recursive identification using radial Gaussian networks," in *Proc. IEEE Conf. Decision Contr.,* Brighton, U.K., 1991.

[39] S. S. Sastry and M. Bodson, *Adaptive Control.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[40] J.-J. E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Automat. Contr.,* vol. AC-33, no. 11, pp. 995–1003, Nov. 1988.

[41] ——, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[42] H. J. Sussman, "Uniqueness of the weights for minimal feedforward nets with a given input–output map," *Neural Networks,* vol. 5, pp. 589–593, 1992.

[43] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavior sciences," Ph.D. dissertation, Committee Appl. Math., Harvard Univ., 1974.

[44] ——, "Backpropagation: Past and future," in *Proc. 1988 Int. Conf. Neural Nets,* vol. 1, pp. I343–I353, 1989.

[45] T. Yabuta and T. Yamada, "Neural network controller characteristics with regard to adaptive control," *IEEE Trans. Syst., Man, Cybern.,* vol. 22, no. 1, pp. 170–176, Jan. 1992.

Frank L. Lewis (S'78–M'81–SM'86–F'94) was born in Würzburg, Germany, subsequently studying in Chile and Scotland. He received the bachelor's degree in physics/electrical engineering and the master's degree in electrical engineering at Rice University, Houston, TX, in 1971. In 1977 he received the M.S. degree in aeronautical engineering from the University of West Florida, Pensacola. In 1981 he received the Ph.D. degree at The Georgia Institute of Technology in Atlanta, where he was employed from 1981 to 1990.

He spent six years in the U.S. Navy, serving as Navigator aboard the frigate USS Trippe (FF-1075), and Executive Officer and Acting Commanding Officer aboard USS Salinan (ATF-161). He has studied the geometric properties of the Riccati equation and implicit systems; his current interests include robotics, intelligent control, nonlinear systems, and manufacturing process control.

Dr. Lewis was awarded the Moncrief-O'Donnell Endowed Chair in 1990 at the Automation and Robotics Research Institute of The University of Texas at Arlington. He is the author/co-author of 80 journal papers and the books *Optimal Control, Optimal Esitmation, Applied Optimal Control and Estimation, Aircraft Control and Simulation, Control of Robot Manipulators,* and the IEEE reprint volume *Robot Control.* He is a registered Professional Engineer in the State of Texas, Associate Editor of *Circuits, Systems, and Signal Processing*, and the recipient of an NSF Research Initiation Grant, a Fulbright Research Award, the American Society of Engineering Education F. E. Terman Award, three Sigma Xi Research Awards, the UTA Halliburton Engineering Research Award, and the UTA University-Wide Distinguished Research Award.

Aydin Yeşildirek was born in Istanbul, Turkey, in 1965. He received the B.S. degree in electronics and telecommunication engineering from Istanbul Technical University, Turkey, in 1986, the M.S. degree in electrical engineering and applied physics from Case Western Reserve University, Cleveland, OH in 1992, and the Ph.D. degree from The University of Texas at Arlington, in 1994.

He is currently a Research Associate at the Automation and Robotics Research Institute, Fort Worth, TX. His research interests include neural networks, robotics, nonlinear systems, adaptive control, and image processing.

Dr. Yeşildirek is a recipient of the Turkish Educational Ministry Scholarship.

Kai Liu (S'89–M'90–SM'95) received the B.S. degree in physics in 1980 and the M.S. degree in electrical engineering in 1982, both from the Beijing Polytechnic University, Beijing, P.R. China, and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in 1990.

From 1982 to 1986 he was a faculty member in the Department of Electrial Engineering, Beijing Polytechnic University, where his research interests focused on the design techniques of microcomputer-based control systems. He is the Head Research Engineer in the Advanced Controls Group, the Automation and Robotics Research Institute of The University of Texas at Arlington. His current research interests include fuzzy logic and neural nets controls, automation, modeling and control of dynamic systems, robust control theory, and robotics.

Dr. Liu is a member of Sigma Xi, the IEEE Control Society, and the IEEE Robotics and Automation Society.