Neural Net Robot Controller with Guaranteed Tracking Performance

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

***Abstract*—A neural net (NN) controller for a general serial-link robot arm is developed. The NN has two layers so that linearity in the parameters holds, but the "net functional reconstruction error" and robot disturbance input are taken as nonzero. The structure of the NN controller is derived using a filtered error/passivity approach, leading to new NN passivity properties. On-line weight tuning algorithms including a correction term to backpropagation, plus an added robustifying signal, guarantee tracking as well as bounded NN weights. The NN controller structure has an outer tracking loop so that the NN weights are conveniently initialized at zero, with learning occurring on-line in real-time. It is shown that standard backpropagation, when used for real-time closed-loop control, can yield unbounded NN weights if 1) the net cannot exactly reconstruct a certain required control function or 2) there are bounded unknown disturbances in the robot dynamics. The role of persistency of excitation is explored.**

I. Introduction

MUCH has been written about neural nets (NN) for system identification (e.g., [4], [12], and [22]) or identification-based ("indirect") control, yet 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], [23], as well as some notions on NN for robot control, are given in [2], [14], [15], [20], [25], [28], [32], [33], and [39]. See also [19].

Persistent problems that remain to be adequately addressed include ad hoc controller structures and the inability to guarantee satisfactory performance of the system. Uncertainty on how to initialize the NN weights leads to the necessity for "preliminary off-line tuning."

In this paper we take a step to confront these deficiencies by considering the two-layer NN, where linearity in the parameters holds (c.f. [7], [26], [28], [29], and [31]–[33]). This may be considered as a step in extending adaptive control theory to NN control theory. Some notions in robot control [17] are tied here to some notions in NN theory.

A NN controller structure derived using robot control techniques means that the NN weights are tuned on-line, with no "learning phase" needed. The controller structure 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 the required nonlinear function.

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 "matrix of robot functions" must be tediously computed from the dynamics of each specific arm [6], [17], the basis functions for the proposed NN controller can be determined from the physics (Lagrangian dynamics) of general robot arms. The proposed controller has more structure than the adaptive controllers standard in robotics and affords the possibility of trading off the complexity of the NN with the magnitude of a certain robust control term added for guaranteed stability. Simplified partitioned NN design makes for faster weight updates.

New NN passivity properties are explored that make the controller robust to unknown disturbances. It is shown that the backpropagation tuning technique generally yields unbounded NN weight if the net cannot exactly reconstruct a certain nonlinear control function or if there are bounded unmodeled disturbances in the robot dynamics. It is shown that the backpropagation tuning algorithm yields a passive neural net. This, coupled with the dissipativity of the robot dynamics, guarantees that all signals in the closed-loop system are bounded under an additional persistency of excitation (PE) condition. Modified weight tuning algorithms avoid the need for PE by making the NN robust, that is, strictly passive in a sense defined herein.

II. Background

Let $\mathbf{R}$ denote the real numbers, $\mathbf{R^n}$ denote the real n-vectors, and $\mathbf{R^{m\times n}}$ the real $m\times n$ matrices. Let $\mathbf{S}$ be a compact simply connected set of $\mathbf{R^n}$. With map $f:\mathbf{S}\rightarrow\mathbf{R^m}$, define $C^m(\mathbf{S})$ 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 $\mathbf{S}$) is defined as [3]

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

The vector (or matrix) absolute value is denoted $|x|$, which is the vector (or matrix) whose entries are the absolute values of the entries of x.

Given $A=[a_{ij}], B\in\mathbf{R^{m\times n}}$ the Frobenius norm is defined by

$$\|A\|_F^2=\mathrm{tr}(A^TA)=\sum_{i,j}a_{ij}^2$$

Manuscript received February 25, 1994; revised March 7, 1994. This research 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.

IEEE Log Number 9404862.

1045-9227/95$04.00 © 1995 IEEE

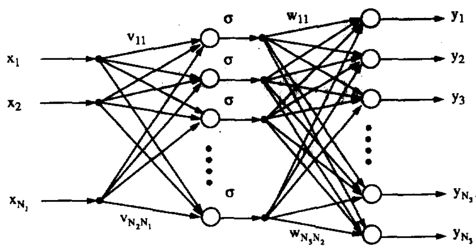


Fig. 1. Three layer neural net structure.

with tr () the trace. The associated inner product is $\langle A, B\rangle_F = \mathrm{tr}(A^T B)$. 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 $A \in \mathbf{R}^{\mathbf{m}\times\mathbf{n}}$ and $x \in \mathbf{R}^{\mathbf{n}}$.

When $x(t) \in \mathbf{R}^{\mathbf{n}}$ is a function of time we use the standard L_p norms [17]. We denote by L_p^n, L_∞^n the spaces such that $x(t)$ is, respectively, pth power integrable and essentially bounded. We say $x(t)$ is bounded if $x(t) \in L_\infty^n$. We say $A(t) \in \mathbf{R}^{\mathbf{m}\times\mathbf{n}}$ is bounded if its induced matrix ∞-norm is bounded (i.e., $A(t) \in L_\infty^{m\times n}$).

A. Neural Networks

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

$$y_i = \sum_{j=1}^{N_2}\left[w_{ij}\sigma\left[\sum_{k=1}^{N_1} v_{jk}x_k + \theta_{vj}\right] + \theta_{wi}\right]; \quad i = 1,\cdots,N_3 \tag{2.1}$$

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 θ_{vm}, θ_{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 behavior."

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

$$y = W^T\sigma(V^T x) \tag{2.2}$$

where, if $z = [z_1\ z_2\ \cdots\ ]^T$ a vector we define $\sigma(z) = [\sigma(z_1)\ \sigma(z_2)\ \cdots]^T$. Including "1" 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 $x \in \mathbf{R}^{\mathbf{N1}}$ and $\sigma: \mathbf{R}^{\mathbf{N2}} \to \mathbf{R}^{\mathbf{N2}}$.

In point of fact, (2.2) provides greater generality than (2.1), since each component of $\sigma(\cdot)$ can be a function of all components of $V^T x$. This has been termed a "functional link (perceptron) net" [26], [32] in the case of $V = I$ (e.g., two-layer net). Indeed, additional generality may be engendered by taking nonsquare σ, e.g., $\sigma: \mathbf{R}^{\mathbf{N2}'} \to \mathbf{R}^{\mathbf{N2}}$ with $N_2' \neq N_2$. Then, the number of hidden units is equal to N_2.

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

$$f(x) = W^T\sigma(V^T x) + \varepsilon(x) \tag{2.3}$$

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 $x \in \mathbf{S}$, we say $f(x)$ is in the functional range of the NN. In general, given a 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 $x \in \mathbf{R}^{\mathbf{n}}$, (2.3) holds with $\|\varepsilon\| < \varepsilon_N$. Unless the NN is in some sense "minimal," these various weights may not be unique [1], [37].

Various well-known results for various activation functions $\sigma()$, based, e.g., on the Stone–Weierstrass Theorem, say that any sufficiently smooth function can be approximated by a suitably large net [8], [13], [27], [33]. The functional range of NN (2.2) is said to be dense in $C^m(\mathbf{S})$ if for any $f \in C^m(\mathbf{S})$ and $\varepsilon_N > 0$ there exist finite N_2, and W and V such that (2.3) holds with $\|\varepsilon\| < \varepsilon_N$, $N_1 = n$, $N_3 = m$. Typical results are like the following, for the case of 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 (2.2) is dense in $C^m(\mathbf{S})$.

In this result, the metric defining denseness is the supremum norm. Moreover, the last layer thresholds θ_{wi} are not needed for this result.

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

$\sigma(z) = \frac{1}{1+e^{-\alpha z}}$	sigmoid
$\sigma(z) = \frac{1-e^{-\alpha z}}{1+e^{\alpha z}}$	hyperbolic tangent (tanh)
$\sigma(z) = e^{-(z-m_j)^2}/s_j$	radial basis functions (RBF).

The issues of selecting σ, and of choosing N_2 for a specified $\mathbf{S} \subset \mathbf{R}^{\mathbf{n}}$ and ε_N are current topics of research (see, e.g., [24], [27]).

B. Stability and Passive Systems

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

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

We say the solution is uniformly ultimately bounded (UUB) if there exists a compact set U such that, for all $x(t_0) = x_0 \in U$. there exists an $\varepsilon > 0$ and a number $T(\varepsilon, x_0)$ such that $\|x(t)\| < \varepsilon$ for all $t \geq t_0 + T$.

Some aspects of passivity will subsequently be important [11], [16, [17], [36]. 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"

$$\dot{L}(t) = y^T u - g(t) \tag{2.4}$$

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

$$\int_0^T y^T(\tau)u(\tau)d\tau \geq \int_0^T g(\tau)d\tau - \gamma^2 \tag{2.5}$$

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

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

$$\int_0^\infty y^T(\tau)u(\tau)d\tau \neq 0 \quad \text{implies} \quad \int_0^\infty g(\tau)d\tau > 0. \tag{2.6}$$

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 (SSP) and are unaware of its use elsewhere (though cf., [11]). For SSP systems, 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]

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \tag{2.7}$$

with $q(t) \in \mathbf{R}^n$ 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 are denoted by τ_d and the control input torque is $\tau(t)$.

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

$$e(t) = q_d(t) - q(t) \tag{2.8}$$

and the filtered tracking error (in standard use in robotics) is

$$r = \dot{e} + \Lambda e \tag{2.9}$$

where $\Lambda = \Lambda^T > 0$. Differentiating $r(t)$ and using (2.7), the arm dynamics may be written in terms of the filtered tracking error as

$$M\dot{r} = -V_m r - \tau + f + \tau_d \tag{2.10}$$

where the nonlinear robot function is

$$f(x) = M(q)(\ddot{q}_d + \Lambda\dot{e}) + V_m(q,\dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(\dot{q}) \tag{2.11}$$

and, for instance

$$x = [e^T \ \dot{e}^T \ q_d^T \ \dot{q}_d^T \ \ddot{q}_d^T]^T. \tag{2.12}$$

Define now a control input torque as

$$\tau_o = \hat{f} + K_v r \tag{2.13}$$

with $\hat{f}(x)$ an estimate of $f(x)$ and a gain matrix $K_v = K_v^T > 0$. The closed-loop system becomes

$$M\dot{r} = -(K_v + V_m)r + \tilde{f} + \tau_d \equiv -(K_v + V_m)r + \zeta_0 \tag{2.14}$$

where the functional estimation error is given by

$$\tilde{f} = f - \hat{f}. \tag{2.15}$$

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 (2.14) to focus on selecting NN tuning algorithms that guarantee the stability of the filtered tracking error $r(t)$. Then, since (2.9), with the input considered as $r(t)$ and the output as e(t) describes a stable system, standard techniques [18], [36] 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 properties of the robot dynamics are required [17]. They hold for all revolute rigid-link manipulators, and Properties 1–4 are standard in the literature. As far as we know, Property 5 is new.

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 $v_b(q) \in C^1(\mathbf{S})$.

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 (2.14) from $\zeta_o(t)$ to $r(t)$ are a state strict passive system.

Proof of Property 5: Take the Lyapunov function

$$L = 1/2 r^T M r$$

so that, using (2.14)

$$\begin{aligned}\dot{L} &= r^T M\dot{r} + 1/2 r^T \dot{M} r \\ &= -r^T K_v r + 1/2 r^T(\dot{M} - 2V_m)r + r^T\zeta_0\end{aligned}$$

whence skew-symmetry yields the power form

$$\dot{L} = r^T\zeta_0 - r^T K_v r.$$ □

III. Two-Layer NN Controller

We consider in the remainder of the paper the NN for the case of fixed V, which may be considered a first step to bridging the gap between adaptive control and NN control. This makes the NN linear in the parameters; this case has been treated for the radial basis functions in [28] and [33], using a projection algorithm for weight tuning in [28] and [31], and for discrete-time systems in [32]. Here, we consider general basis functions $\phi(x)$ and propose various weight tuning algorithms, showing their relation to standard backpropagation [38]. The tuning algorithms yield a passive NN, yet persistency of excitation (PE) is generally needed for suitable performance. A modified tuning algorithm is proposed to make the NN robust so that PE is not needed.

Define

$$\phi(x) = \sigma(V^T x) \tag{3.1}$$

so that the net output is

$$y = W^T \phi(x). \tag{3.2}$$

Then, for suitable NN approximation properties, $\phi(x)$ must satisfy some conditions (e.g., [32]). Take $N_1 = n, N_3 = m$.

Definition 3.1: Let $\mathbf{S}$ be a compact simply connected set of $\mathbf{R^n}$, and $\phi(x)\colon \mathbf{S} \to \mathbf{R^{N2}}$ be integrable and bounded. Then ϕ is said to provide a basis for $C^m(S)$ if

1) A constant function on $\mathbf{S}$ can be expressed as (3.2) for finite N_2.
2) The functional range of NN (3.2) is dense in $C^m(\mathbf{S})$ for countable N_2.

The issue of selecting σ and V so that ϕ provides a basis, as well as the further issue of selecting N_2 for a given $\mathrm{S} \subset \mathbf{R^n}$ and ε_N, are topics of current research. We shall assume that this problem has been addressed and that $\phi(x)$ is known. A popular set of basis function $\phi(x)$ in the literature is the radial basis functions, though we shall indicate in Section V how to choose a basis set specifically for rigid-link robot arms.

Assume that there exist constant ideal weights W so that the robot function in (2.11) can be written as

$$f(x) = W^T \phi(x) + \varepsilon(x) \tag{3.3}$$

where $\phi(x)$ provides a suitable basis and $\|\varepsilon(x)\|_{<}\varepsilon N(x)$, with the bounding function $\varepsilon_N(x) \in C^1(\mathrm{S})$ known. This is a very reasonable assumption for robotic systems. In fact, for a specific robot arm this corresponds to the standard linearity in the parameters assumption [6], and it is easy to determine the required robotic functions [17]. This leads to the standard adaptive control techniques for robot arms.

The difference between standard adaptive control and the NN approach proferred here is significant: we indicate in Section V how to select basis functions $\phi(x)$ for a general n-link rigid arm using the physics of the arm, (e.g., Lagrangian dynamics). Thus, the tedium of solving analytically for the robot basis functions needed for each given arm (e.g., regression matrix), as required in standard adaptive approaches, is avoided.

Note that the bounding function ε_N is in general a known function of x. This provides additional freedom over the case of constant ε_N, as it allows a design trade-off between NN complexity and robust control. In fact, it allows the possibility of selecting a simplified neural net structure based on a reduced basis $\phi(x)$ and compensating for the increased magnitude of $\varepsilon(x)$ using the robustifying control term soon to be introduced.

A. Controller Structure and Error System Dynamics

Define the NN functional estimate by

$$\hat{f}(x) = \hat{W}^T \phi(x) \tag{3.4}$$

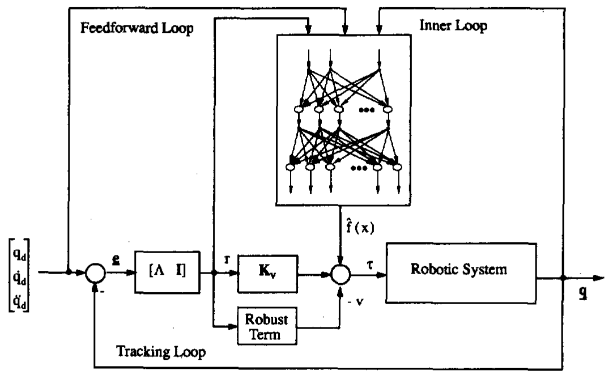


Fig. 2. Neural net control structure.

with $\hat{W}$ the current values of the NN weights as provided by the tuning algorithm. With W the ideal weights required in (3.3) define the weight deviations or weight estimation errors as

$$\tilde{W} = W - \hat{W}. \tag{3.5}$$

Assume the ideal weights are bounded by known values so that

$$\|W\|_F \le W_{\max}. \tag{3.6}$$

With τ_o defined in (2.13), select the control input

$$\tau = \tau_o - v = \hat{W}^T \phi(x) + K_v r - v \tag{3.7}$$

with $v(t)$ a function to be determined to provide robustness in the face of the net reconstruction error ε. Then, the closed-loop filtered error dynamics become

$$\begin{aligned} M\dot{r} &= -(K_v + V_m)r + \tilde{W}^T\phi(x) + (\varepsilon + \tau_d) + v \\ &\equiv -(K_v + V_m)r + \zeta_1. \end{aligned} \tag{3.8}$$

The proposed NN control structure is shown in Fig. 2, where $\underline{q} \equiv [q^T\ \dot{q}^T]^T$, $\underline{e} \equiv [e^T\ \dot{e}^T]^T$.

B. Weight Updates for Guaranteed Tracking Performance

We give here some NN weight tuning algorithms that guarantee the tracking stability of the closed-loop system. It is required to demonstrate that the tracking error $r(t)$ is suitably small and that the NN weights $\hat{W}$ remain bounded, for then the control $\tau(t)$ is bounded.

Key features of all our tuning algorithms are that:

1) No off-line weight tuning is needed; on the contrary, the NN controller converges to small tracking errors and bounded weight errors on line in real time.
2) Initializing the NN weights is very easy. In fact, they are simply initialized at zero; the unique controller structure (Fig. 2) has an outer tracking loop that guarantees bounded errors until the NN begins to learn.

Observability and Persistency of Excitation: A technical lemma involving observability of a certain time-varying system is needed. Consider the linear time-varying system $(0, B(t), C(t))$ defined by $\dot{x} = B(t)u$, $y = C(t)x$ with the elements of $B(t)$ and $C(t)$ piecewise continuous functions and $x \in \mathbf{R^n}$, $u \in \mathbf{R^m}$, $y \in \mathbf{R^p}$. Since the state transition matrix is the identity matrix, the observability gramian is

$$N(t,t_0) = \int_{t_0}^{T} C^T(\tau)C(\tau)d\tau. \tag{3.9}$$

Then we say the system is uniformly completely observable [34] if there exist finite constants $\delta > 0$ $\beta_1 > 0$, $\beta_2 > 0$ so that, for all $t \geq 0$,

$$\beta_1 I \leq N(t+\delta, t) \leq \beta_2 I. \tag{3.10}$$

Technical Lemma 3.1: Let the system $(0, B(t), C(t))$ be uniformly completely observable with $B(t)$ bounded. Then if $u(t)$ and $y(t)$ are bounded, the state $x(t)$ is bounded.

Proof: See the Appendix. □

Note that this result holds despite the less-than-desirable stability properties of the system.

A vector $w(t)$ is said to be persistently exciting (PE) [34] if there exist $\alpha_1, \alpha_2, \delta > 0$ so that

$$\alpha_1 I \leq \int_{t_0}^{t_0+\delta} w(\tau)w^T(\tau)d\tau \leq \alpha_2 I, \quad \text{for all } t_0 \geq 0. \tag{3.11}$$

Case of Constant NN Reconstruction Error ε_N: There now follows a sequence of results designed to provide guaranteed tracking under increasingly mild assumptions. First we assume the NN reconstruction error bound ε_N is constant.

Theorem 3.2: Let the desired trajectory $q_d, \dot{q}_d, \ddot{q}_d$ be bounded and the NN functional reconstruction error bound ε_N be constant. Take the control input for (2.7) as (3.7) with $v(t) = 0$ and weight tuning provided by

$$\dot{\hat{W}} = F\phi r^T \tag{3.12}$$

where $F = F^T > 0$ is any constant matrix. Suppose the hidden layer output $\phi(x)$ is persistently exciting. Then the filtered tracking error $r(t)$ is UUB, with a practical bound given by the right-hand side of (3.14), and the NN weight estimates $\hat{W}$ are bounded. Moreover, the tracking error may be kept as small as desired by increasing the gain K_v.

Proof: Define the Lyapunov function candidate

$$L = 1/2r^T Mr + 1/2\text{tr}(\tilde{W}^T F^{-1}\tilde{W}). \tag{3.13}$$

Differentiating yields

$$\dot{L} = r^T M\dot{r} + 1/2r^T\dot{M}r + \text{tr}(\tilde{W}^T F^{-1}\dot{\tilde{W}})$$

whence substitution from (3.8) yields

$$\dot{L} = -r^T K_v r + 1/2r^T(\dot{M} - 2V_m)r + \text{tr}\tilde{W}^T(F^{-1}\dot{\tilde{W}} + \phi r^T) + r^T(\varepsilon + \tau_d).$$

The skew symmetry property makes the second term zero and the third term is zero if we select

$$\dot{\tilde{W}} = -F\phi r^T.$$

Since $\tilde{W} = W - \hat{W}$ and W is constant, this yields the weight tuning law (3.12).

Now

$$\dot{L} = -r^T K_v r + r^T(\varepsilon + \tau_d)$$
$$\leq -K_{v\min}\|r\|^2 + (\varepsilon_N + b_d)\|r\|.$$

with $K_{v\min}$ the minimum singular value of K_v. Since $\varepsilon_N + b_d$ is constant, $\dot{L} \leq 0$ as long as

$$\|r\| > (\varepsilon_N + b_d)/K_{v\min}. \tag{3.14}$$

This demonstrates that the tracking error $r(t)$ is bounded and continuity of all functions shows as well the boundedness of $\dot{r}(t)$. It remains to show that $\hat{W}$, or equivalently $\tilde{W}$, is bounded.

Boundedness of r guarantees the continuity of e and $\dot{e}$, whence boundedness of the desired trajectory shows q and $\dot{q}$ are bounded. Property 2 then shows boundedness of $V_m(q, \dot{q})$. These facts guarantee boundedness of the function

$$y \equiv M\dot{r} + (K_v + V_m)r - (\varepsilon + \tau_d)$$

since $M(q)$ is bounded. Therefore, according to (3.12), (3.8) the dynamics relative to $\tilde{W}$ are given by

$$\dot{\tilde{W}} = -F\phi r^T$$
$$y^T = \phi^T\tilde{W} \tag{3.15}$$

with $y(t)$ and $r(t)$ bounded.

Using the Kronecker product $\otimes$ [5] allows one to write

$$\frac{d}{dt}\text{vec}(\dot{\tilde{W}}) = -(I \otimes F\phi)r$$
$$y = (I \otimes \phi^T)\text{vec}(\tilde{W})$$

where the $\text{vec}(A)$ operator stacks the columns of a matrix A to form a vector, and one notes that $\text{vec}(z^T) = z$ for a vector z. Now, the PE condition on ϕ is equivalent to PE of $(I \otimes \phi)$, and so to the uniform complete observability of this system, so that by Lemma 3.1 boundedness of $y(t)$ and $r(t)$ assures the boundedness of $\tilde{W}$, and hence of $\hat{W}$. [Note that boundedness of $x(t)$ verifies boundedness of $F\phi(x(t))$.] □

The right-hand side of (3.14) can be taken as a practical bound on the tracking error in the sense that $r(t)$ will never stray far above it. It is important to note from this that the tracking error increases with the NN reconstruction error bound ε_N and robot disturbance bound b_d, yet arbitrarily small tracking errors may be achieved by selecting large gains K_v. (If K_v is taken as a diagonal matrix, $K_{v\min}$ is simply the smallest gain element.)

Note further that the problem of net weight initialization occurring in other approaches in the literature does not arise, since if $\hat{W}(0)$ is taken as zero the PD term $K_v r$ stabilizes the plant on an interim basis. A formal proof reveals that the gain K_v should be large enough and the initial filtered error $r(0)$ small enough. The minimum value of the gain needed can be computed and is given in [9].

The primary impact of this is that the proposed NN controller requires no off-line learning phase. On the contrary, as seen in the simulation examples, the NN learns in real time within fractions of a second and suitable control performance is guaranteed throughout.

Note finally that the assumption of constant ε_N allows one to bound $r(t)$ in the first phase of the proof without the use of $v(t)$ in (3.7). This assumption is relaxed in a subsequent theorem.

Ideal Case—Backpropagation Tuning of Weights: The next result details the closed-loop behavior in the idealized case of no net functional reconstruction error and no unmodeled disturbances in the robot arm dynamics. In this case the PE assumption on $\phi(x)$ is not needed. Interestingly enough, in advanced adaptive controllers PE is not needed for good control performance. For parameter estimation, however, it is generally required. A similar situation should also eventually be discovered in NN, where PE is closely connected to the generalization capability of the NN.

Corollary 3.3: Suppose $q_d, \dot{q}_d, \ddot{q}_d$ are bounded, and the NN functional reconstruction error ε_N and unmodeled disturbances τ_d are equal to zero. Let the control input for (2.7) be given by

$$\tau = \hat{W}^T \phi(x) + K_v r \tag{3.16}$$

with weight tuning provided by

$$\dot{\hat{W}} = F\phi r^T \tag{3.17}$$

with any constant matrix $F = F^T > 0$. Then the tracking error $r(t)$ goes to zero with t and the weight estimates $\hat{W}$ are bounded.

Proof: The new assumptions yield the error system

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T \phi(x). \tag{3.18}$$

Using the Lyapunov function candidate (3.13) with the new assumptions results in

$$\dot{L} = -r^T K_v r.$$

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

$$\int_0^\infty -\dot{L} dt < \infty. \tag{3.19}$$

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 (3.18) verify the boundedness of $\ddot{L}$ and hence the uniform continuity of $\dot{L}$. This allows us to invoke Barbalat's Lemma [17], [36] in connection with (3.19) to conclude that $\dot{L}$ goes to zero with t and hence that $r(t)$ vanishes. □

Note that (3.17) is nothing but the continuous-time backpropagation algorithm [38] for the one-layer case; Corollary 3.3 indicates when backprop alone should suffice. Theorem 3.2, however, reveals the failure of simple backpropagation if there are net functional reconstruction errors or bounded disturbances. Thus, backpropagation used in a net that cannot exactly reconstruct $f(x)$, or on a robot arm with bounded unmodeled disturbances, cannot be guaranteed to yield bounded weights. Then, the PE condition is required to guarantee boundedness of the weight estimates. Unfortunately, it may be difficult to verify the PE of the hidden layer output functions $\phi(x)$.

Relaxation of Persistence of Excitation Condition: In adaptive control the possible unboundedness of the weight (e.g., "parameter") estimates when PE fails to hold is known as "parameter drift." An alternative to correcting this problem that does not require the PE condition is to use σ-modification [28], or e-modification [21] of the weight tuning equation as follows.

The next theorem relies on an extension to Lyapunov theory. The disturbance τ_d and the NN reconstruction error ε 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 Theorem 1.5–6 in [17], the last portion of our proof being similar to the proof used in [21].

Theorem 3.4 Given the hypotheses of Theorem 3.2, let the control input for the robot arm (2.7) be given by

$$\tau = \hat{W}^T \phi(x) + K_v r \tag{3.20}$$

and let the weight tuning be modified as

$$\dot{\hat{W}} = F\phi r^T - \kappa F\|r\|\hat{W} \tag{3.21}$$

with $\kappa > 0$ a design parameter. Make no assumptions of any sort of PE condition on $\phi(x)$. Then the filtered tracking error $r(t)$ and the NN weight estimates $\hat{W}(t)$ are UUB with practical bounds given respectively by the right-hand sides of (3.22), (3.23). Moreover, according to the former, the tracking error may be made as small as desired by increasing the gain K_v.

Proof: Using the Lyapunov function candidate (3.13) with tuning rule (3.21) yields (c.f. proof of Theorem 3.2)

$$\dot{L} = -r^T K_v r + \kappa\|r\| \operatorname{tr} \tilde{W}^T (W - \tilde{W}) + r^T(\varepsilon + \tau_d).$$

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

$$\begin{aligned}\dot{L} \le & -K_{v\min}\|r\|^2 + \kappa\|r\| \|\tilde{W}\|_F (W_{\max} - \|\tilde{W}\|_F) \\ & + (\varepsilon_N + b_d)\|r\| = -\|r\| [K_{v\min}\|r\| \\ & + \kappa\|\tilde{W}\|_F (\|\tilde{W}\|_F - W_{\max}) - (\varepsilon_N + b_d)]\end{aligned}$$

which is negative as long as the term in braces is positive. Completing the square yields

$$\begin{aligned}& K_{v\min}\|r\| + \kappa\|\tilde{W}\|_F(\|\tilde{W}\|_F - W_{\max}) - (\varepsilon_N + b_d) \\ & = \kappa(\|\tilde{W}\|_F - W_{\max}/2)^2 - \kappa W_{\max}^2/4 + K_{v\min}\|r\| \\ & - (\varepsilon_N + b_d)\end{aligned}$$

which is guaranteed positive as long as either

$$\|r\| > \frac{\kappa W_{\max}^2/4 + (\varepsilon_N + b_d)}{K_{v\min}} \tag{3.22}$$

or

$$\|\tilde{W}\|_F > W_{\max}/2 + \sqrt{W_{\max}^2/4 + (\varepsilon_N + b_d)/\kappa}. \tag{3.23}$$

Thus, $\dot{L}$ is negative outside a compact set. According to a standard Lyapunov theorem extension [17], [21], this demonstrates the UUB of both $\|r\|$ and $\|\tilde{W}\|_F$. □

The right-hand sides of (3.22), (3.23), respectively, may be taken as practical bounds on the tracking error and NN weight errors in the sense that excursions beyond these bounds will be very small. Note, moreover, from the former that arbitrarily small tracking error bounds may be achieved by selecting large control gains K_v. On the other hand, the NN weight error is fundamentally bounded by $W_{\max}$, the known bound on the ideal weights W. The parameter κ offers a design trade-off between the relative eventual magnitudes of $\|r\|$ and $\|\tilde{W}\|_F$; a smaller κ yields a smaller $\|r\|$ and a larger $\|\tilde{W}\|_F$, and vice versa.

A comparison with the results of [21] shows that the NN reconstruction error ε_N and the bounded disturbances b_d increase the bounds on $\|r\|$ and $\|\tilde{W}\|_F$ in a very interesting way.

Note that PE is not needed to establish the bounds on $\tilde{W}$ with the modified weight tuning algorithm.

Finally, it is emphasized that the NN weights may be initialized at zero, and stability will be provided by the outer tracking loop in (3.1) until the NN learns. This means that there is no off-line learning phase, but NN learning occurs in real-time.

Nonconstant Reconstruction Error Bound ε_n: The final result is meant to address the case where the NN reconstruction error bound $\varepsilon_N(x)$ is not a constant. Indeed, though the case of constant ε_N is treated in the literature to date, it is known that reasonable choices for $\phi(x)$ in robot control virtually doom ε_N to be quadratic in $\|\dot{e}\|$, so that it depends on both q_d and q (i.e., on x). In fact, a very reasonable assumption (c.f. [6] and [17]) is that

$$\varepsilon_N(x) = \alpha_0(q_d, \dot{q}_d, \ddot{q}_d) + \rho(r) \tag{3.24}$$

with α_0 and ρ continuous polynomial functions. The coefficients of $\rho(r)$ depend on $q_d, \dot{q}_d, \ddot{q}_d$.

To attack this problem we require the addition of a robustifying term $v(t)$ based on the function defined for $y \in \mathbf{R}$ as

$$\text{sat}(y) = \begin{cases} 1 - e^{-y/\gamma}, & y \geq 0 \\ -(1 - e^{y/\gamma}), & y < 0 \end{cases}$$

with γ a small positive (design) parameter. As $\gamma \to 0$, sat(y) approaches a step transition from -1 at $y = 0^-$ to 1 at $y = 0$, however, sat(y) is a continuous function. When $y \in \mathbf{R}^n$, sat(y) is the vector with entries sat(y_i), with $y_i \in \mathbf{R}$ the components of y.

Theorem 3.5: Let the desired trajectory $q_d, \dot{q}_d, \ddot{q}_d$ be bounded and the NN functional reconstruction error bound be of the form (3.24). Take the control input for (2.7) as

$$\tau = \hat{W}^T \phi(x) + K_v r - v \tag{3.25}$$

with auxiliary signal

$$v = -(\varepsilon_N + b_d)\ \text{sat}\,(r) \tag{3.26}$$

and weight tuning provided by

$$\dot{\hat{W}} = F\phi r^T \tag{3.27}$$

with $F = F^T > 0$ a constant matrix. Suppose the hidden layer output $\phi(x)$ is persistently exciting. Then the filtered tracking error $r(t)$ is UUB and the NN weight estimates $\hat{W}$ are bounded. A practical bound on the tracking error is given by the right-hand side of (3.28), which can be made as small as desired by choosing large control gains K_v.

Proof: Proceeding as in the proof of Theorem 3.2, there results finally

$$\dot{L} = -r^T K_v r + r^T(\varepsilon + \tau_d + v).$$

Selecting (3.26) guarantees that the last term

$$r^T(\varepsilon + \tau_d + v) = \sum_i r_i(\varepsilon_i + \tau_{di} + v_i)$$

satisfies

$$r_i(\varepsilon_i + \tau_{di} + v_i) \leq \begin{cases} r_i(|\varepsilon_i| + |\tau_{di}| + v_i) \leq r_i[(\varepsilon_N + b_d)e^{-r_i/\gamma} - \delta_i], & r_i \geq 0 \\ -r_i(|\varepsilon_i| + |\tau_{di}| - v_i) \leq -r_i[(\varepsilon_N + b_d)e^{r_i/\gamma} - \delta_i], & r_i < 0 \end{cases}$$

where $\delta_i = [(\varepsilon_N - |\varepsilon_i|) + (b_d - |\tau_{di}|)] \geq 0$, so that

$$r_i(\varepsilon_i + \tau_{di} + v_i) \leq (\varepsilon_N + b_d)|r_i|e^{-|r_i|/\gamma}.$$

Thus

$$r^T(\varepsilon + \tau_d + v) \leq (\varepsilon_N + b_d)|r|^T \text{vec}\ (e^{-|r|/\gamma})$$

with vec(y_i) the vector with components y_i.

Therefore

$$\dot{L} \leq -K_{v\min}\|r\|^2 + (\varepsilon_N + b_d)|r|^T \text{vec}(e^{-|r|/\gamma})$$

whence (3.24) yields

$$\dot{L} \leq -K_{v\min}\|r\|^2 + (\alpha_0 + b_d)|r|^T \text{vec}(e^{-|r|/\gamma}) + \rho(r)|r|^T \text{vec}(e^{-|r|/\gamma}).$$

Since, for $y \in \mathbf{R}$, $ye^{-y/\gamma}$ has a maximum value of γ/e at $y = \gamma$, and the exponential dominates any polynomial, it may be argued [18] that for suitably small γ

$$\dot{L} \leq -K_{v\min}\|r\|^2 + (\alpha_0 + b_d)n\gamma/e + \alpha_1 n\gamma/e$$

with α_1 a continuous function of $q_d, \dot{q}_d, \ddot{q}_d$ depending on the coefficients of $\rho(r)$.

Since the last two terms are independent of r, $\dot{L}$ is negative as long as

$$\|r\| \geq \left[\frac{(b_d + \alpha_0 + \alpha_1)n\gamma}{eK_{v\min}}\right]^{1/2} \tag{3.28}$$

where e is the exponential function.

Since the desired trajectory is bounded, α_0 and α_1 are bounded, therefore, $\|r\|$ is bounded by the right-hand side of (3.28) plus an arbitrarily small positive constant. Having demonstrated the boundedness of $r(t)$, the remainder of the proof mimics that of Theorem 3.2. □

The right-hand side of (3.28) provides a practical bound on $\|r\|$; note that the tracking error can be made as small as desired by increasing the gain K_v, and that the deleterious effects of $\varepsilon(x)$ can be additionally mitigated by using a small γ. The use of a small γ has little detrimental effect as the function sat() remains bounded between ± 1. This may be contrasted to other robust control techniques which often: 1) assume constant disturbance bounds and 2) use functions like $(\varepsilon_N + b_d)r/\gamma$, that increase with the design parameter γ.

According to (3.28), in the case of nonconstant NN reconstruction error bounds, increasing the desired velocity $\dot{q}_d$ or acceleration $\ddot{q}_d$ can be counted on to increase the eventual bounds on the tracking error $r(t)$.

It is straightforward though tedious to demonstrate that the term $v(t)$ added to the control in Theorem 3.4 allows guaranteed tracking in case of $\varepsilon_N(x)$ nonconstant and in the absence of persistent excitation. The hurdle is in demonstrating a compact set outside of which $\dot{L} < 0$.

Note that the assumption that ε_N can be a function on x of this control structure allows for a design trade-off between NN complexity and magnitude of the robustifying term. Specifically, if the set of basis functions $\phi(x)$ is reduced to simplify the NN, then in (3.3) the bounding function $\varepsilon_N(x)$ increases, resulting in a concomitant increase in the magnitude of the robustifying signal $v(t)$ [e.g., (3.26)]. Tracking and stability are still guaranteed.

As in the other tuning schemes, we may select $\hat{W}(0) = 0$, and learning occurs on-line in real time. There is no off-line learning phase.

IV. Passivity Properties of the NN

The closed-loop error system (3.8) appears in Fig. 3. Note the role of the NN, which appears in a typical feedback configuration, as opposed 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.

Theorem 4.1: The weight tuning algorithm (3.12) makes the map from $r(t)$ to $-\tilde{W}^T\phi$ a passive map.

Proof: Selecting the Lyapunov function

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

and evaluating $\dot{L}$ along the trajectories of (3.15) yields

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

which is in power form. □

Thus the robot error system in Fig. 3 is (SSP) and the weight error block is passive; this guarantees the dissipativity of the closed-loop system [36]. 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.

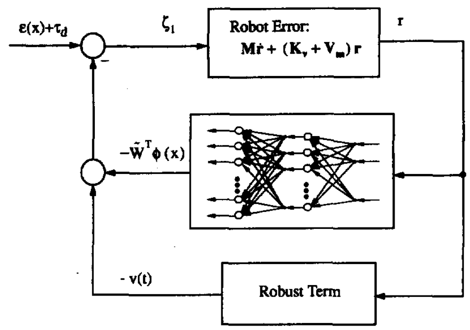


Fig. 3 Neural net closed-loop error system.

Unfortunately, though dissipative, the closed-loop system is not SSP so this does not yield boundedness of the internal states of the lower block (i.e., $\tilde{W}$) unless that block is observable, that is unless PE holds.

The next result shows why PE is not needed with the modified weight update algorithm of Theorem 3.4.

Corollary 4.2: The weight tuning algorithm (3.21) makes the map from $r(t)$ to $-\tilde{W}^T\phi$ a state strict passive map.

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

$$\begin{aligned} \dot{\tilde{W}} &= -\kappa F\|r\|\tilde{W} - F\phi r^T + \kappa F\|r\|W \\ y &= \phi^T\tilde{W} \end{aligned} \tag{4.1}$$

with $y(t)$, $r(t)$, and W bounded. Selecting the Lyapunov function candidate

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

and evaluating $\dot{L}$ yields

$$\begin{aligned} \dot{L} &= \text{tr } \tilde{W}^{TF^{-1}} \dot{\tilde{W}} \\ &= r^T(-\tilde{W}^T\phi) - \kappa\|r\|(\|\tilde{W}\|_F^2 - \langle\tilde{W}, W\rangle_F) \\ &\le r^T(-\tilde{W}^T\phi) - \kappa\|r\|(\|\tilde{W}\|_F^2 - W_{\max}\|\tilde{W}\|_F) \end{aligned}$$

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

It is not difficult to demonstrate that the form of $\dot{L}$ allows one to show the boundedness of $\tilde{W}$ when the first term is bounded (see, e.g., the proof of Theorem 3.4).

It should be noted that SSP of both the robot dynamics and the weight tuning block 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 even without PE.

The function of the robustifying term $v(t)$ is to guarantee boundedness of $\varepsilon_N + b_d + v$ when $\varepsilon_N(x)$ is not constant.

We define an NN plus its tuning algorithm as passive if, in the error formulation, it guarantees the passivity of the subsystem $r(t)$ to $-\tilde{W}^T\phi$. Then, an extra PE condition is needed to guarantee boundedness of the weights. We define an NN plus its tuning algorithm as robust if, in the error formulation, it guarantees the state strict passivity of the subsystem $r(t)$ to $-\tilde{W}^T\phi$. Then, no extra PE condition is needed for boundedness of the weights. Note that 1) dissipativity of the

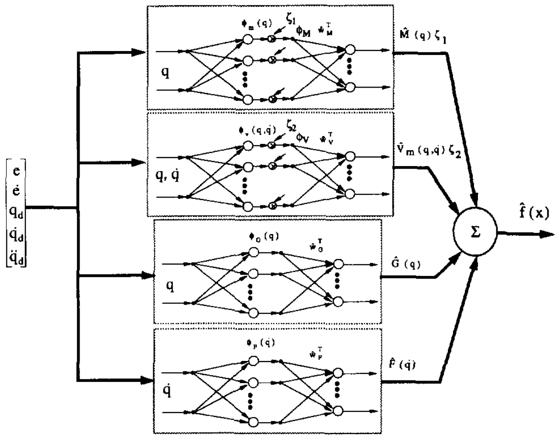


Fig. 4. Structured neural net.

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. Partitioned Neural Nets, Basis Functions, and Preprocessing of NN Input Signals

Given a robot arm, it is desired to select a set of basis functions and determine the NN reconstruction error bound so that (3.3) holds; that is, determine a basis function vector $\phi(x)$ so that the nonlinear robot function

$$f(x) = M(q)\zeta_1(t) + V_m(q,\dot{q})\zeta_2(t) + G(q) + F(\dot{q}) \quad (5.1)$$

can be expressed, for some W, on a compact set as

$$f(x) = W^T\phi(x) + \varepsilon(x) \quad (5.2)$$

where $\|\varepsilon(x)\| < \varepsilon_N(x)$, with the bounding function $\varepsilon_N(x)$ known. For controls purposes, $\zeta_1(t) = \ddot{q}d + \Lambda\dot{e}$, $\zeta_2(t) = \dot{q}_d + \Lambda e$.

A major advantage of the NN controller is that it allows one to design in terms of partitioned NN or neural subnets. This: 1) simplifies the design, 2) gives added controller structure, and 3) makes for faster weight tuning algorithms.

To see this, let $q \in \mathbf{R^n}$. Taking the four terms in $f(x)$ one at a time, assume the columns $m_i(q)$ of $M(q)$ can be expressed as

$$m_i = W_{mi}^T\phi_{mi}(q)$$

with vector $\phi_{mi}(q)$ containing a basis set for the ith column. Then, with ζ_{1i} the components of ζ_1, there follows the decomposition

$$M(q)\zeta_1(t) = \sum_{i=1}^{n} W_{mi}^T\phi_{mi}(q)\zeta_{1i}(t)$$

which one may write as

$$\begin{aligned} M(q)\zeta_1(t) &= \left[W_{m1}^T\ W_{m2}^T\ \cdots\ W_{mn}^T\right][\zeta_1(t)\otimes\phi_m(q)] \\ &\equiv W_M^T\phi_M(q) \end{aligned} \quad (5.3)$$

where $\otimes$ denotes Kronecker product [17], and it has been assumed that the same basis $\phi_m(q)$ serves for each column.

Similarly

$$\begin{aligned} V_m(q,\dot{q})\zeta_2(t) &= W_V^T[\zeta_2(t)\otimes\phi_v(q,\dot{q})] \\ &\equiv W_V^T\phi_V(q,\dot{q}) \end{aligned} \quad (5.4)$$

with ϕ_v a basis for each column of V_m. It is direct to write

$$\begin{aligned} G(q) &= W_G^T\phi_G(q) \\ F(\dot{q}) &= W_F^T\phi_F(\dot{q}). \end{aligned} \quad (5.5)$$

This procedure involves separately determining basis functions for the four terms in the robot dynamics, a much simplified problem.

Now, write $f(x)$ as

$$f(x) = [W_M^T\ W_V^T\ W_G^T\ W_F^T]\begin{bmatrix}\phi_M\\ \phi_V\\ \phi_G\\ \phi_F\end{bmatrix}$$

whence it is clear that the required basis functions in (5.2) are given in terms of the basis functions for the individual terms as

$$\phi(x) = \begin{bmatrix}\zeta_1(t)\otimes\phi_m(q)\\ \zeta_2(t)\otimes\phi_v(q,\dot{q})\\ \phi_G\\ \phi_F\end{bmatrix} \equiv \begin{bmatrix}\phi_M\\ \phi_V\\ \phi_G\\ \phi_F\end{bmatrix}. \quad (5.6)$$

This procedure results in four neural subnets, which we term a structured NN, as shown in Fig. 4. It is direct to show that the individual partitioned NN's can be separately tuned, making for a faster weight update procedure. In fact, if F and κ are selected as block diagonal matrices, then the weight update law (3.21) is

$$\begin{aligned} \dot{\hat{W}}_M &= F_M\phi_M r^T - \kappa_M F_M\|r\|\hat{W}_M \\ \dot{\hat{W}}_V &= F_V\phi_V r^T - \kappa_V F_V\|r\|\hat{W}_V \\ \dot{\hat{W}}_G &= F_G\phi_G r^T - \kappa_G F_G\|r\|\hat{W}_G \\ \dot{\hat{W}}_F &= F_F\phi_F r^T - \kappa_F F_F\|r\|\hat{W}_F. \end{aligned} \quad (5.7)$$

Using this structure in the theorems in Section III allows one to prove rigorously that individual tuning of the partitioned NN is equivalent to tuning the complete NN.

The selection of the basis functions $\phi_M, \phi_V, \phi_G, \phi_F$ remains to be addressed. It is first advantageous to note that some preprocessing of the NN input signals considerably aids in NN design and function. Thus, let an n-link robot have n_r revolute joints with joint variables q_r, and n_p prismatic joints with joint variables q_p. Define $n = n_r + n_p$. Since the only occurrences of the revolute joint variables are as sines and cosines, transform $q = [q_r^T q_p^T]^T$ by preprocessing to $[\cos(q_r)^T\ \sin(q_r)^T\ q_p^T]^T$ to be used as arguments for the basis functions. Then the vector x can be taken as

$$x = \left[\zeta_1^T\ \zeta_2^T\ \cos(q_r)^T\ \sin(q_r)^T\ q_p^T\ \dot{q}^T\ \mathrm{sgn}(\dot{q})^T\right]^T \quad (5.8)$$

(where the signum function is needed in the friction terms).

The proposed procedure allows one to determine the basis functions from the physical (Lagrange dynamics) properties of a general serial-link robot arm. According to [17], the gravity $G(q)$ and inertia matrix $M(q)$ are given explicitly in terms of partial derivatives of the arm T_i matrices with respect

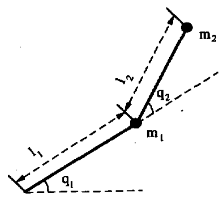


Fig. 5. Two-link planar elbow arm.

to $\dot{q}$. Then, $V_m(q,\dot{q})$ may be explicitly computed in terms of $\partial M(q)/\partial q$ That is, it is only necessary to find a set of basis functions for the arm T_i matrix partial derivatives then a complete basis set for $f(x)$ follows in terms of Kronecker products with the joint variables and velocities. Thus, it is not necessary to determine the dynamics for any specific arm. The requisite basis set will consist of polynomials up to a known maximum order in the variables of x. The basis set for the friction term $F(\dot{q})$ is easily found in terms of $\dot{q}$ and $\mathrm{sgn}(\dot{q})$.

VI. Illustrative Design and Simulation

A planar two-link arm used extensively in the literature for illustration purposes appears in Fig. 5; the dynamics are given, for instance in [17]. The joint variable is $q = [q_1\ q_2]^T$. We should like to illustrate the NN control schemes derived herein, which will require no knowledge of the dynamics, not even their structure which is needed for adaptive control. We shall use the results of Section V to derive a simplified structured neural net controller that allows easier NN design, based on the physics of the arm, and admits faster tuning than a single large net.

Adaptive Controller Baseline Design: For comparison, a standard adaptive controller is given by [35]

$$\tau = Y\hat{\psi} + K_v r \tag{6.1}$$

$$\dot{\hat{\psi}} = FY^T r \tag{6.2}$$

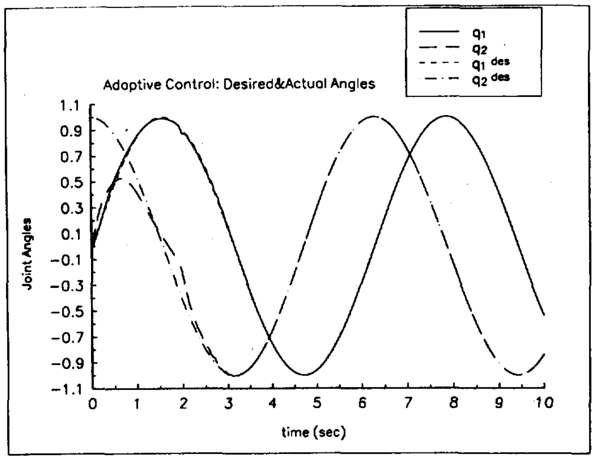
with $F = F^T > 0$ a design parameter matrix, $Y(e, \dot{e}, q_d, \dot{q}_d, \ddot{q}_d)$ a fairly complicated matrix of robot functions that must be explicitly derived from the known arm dynamics, and Ψ the vector of unknown parameters, in this case simply the link masses m_1, m_2.

We took the arm parameters as $\ell_1 = \ell_2 = 1$ m, $m_1 = 0.8$ kg, $m_2 = 2.3$ kg, and selected $q_{1d}(t) = \sin t$, $q_{2d}(t) = \cos t$, $K_v = \mathrm{diag}\{20, 20\}$, $F = \mathrm{diag}\{50, 50\}$, $\Lambda = \mathrm{diag}\{5, 5\}$. 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. 6.

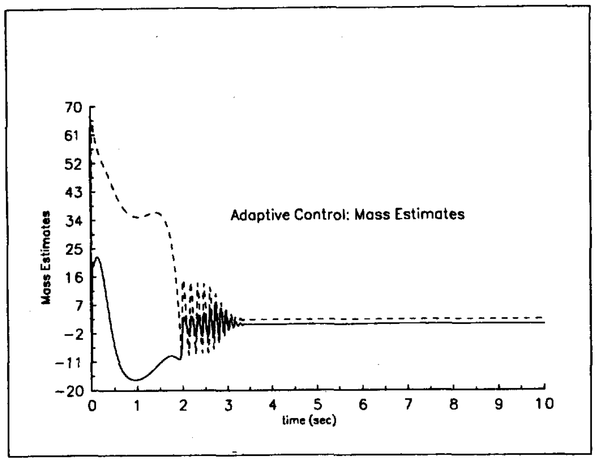
Note the good behavior, which obtains since there are only two unknown parameters, so that the single mode (e.g., two poles) of $q_d(t)$ guarantees persistence of excitation [11].

Neural Net Controller Structure: The NN controller appears in Fig. 2, where the NN has the structure appearing in Fig. 4, with $\zeta_1 = \ddot{q}_d + \Lambda\dot{e}$, $\zeta_2 = \dot{q}_d + \Lambda e$.

From well-known properties of the dynamics of any two-link revolute robot, the robot inertia matrix requires terms like $\sin(q)$, $\cos(q)$, and constant terms. Therefore, the neural subnet required to construct $M(q)\zeta_1$ appears in Fig. 7, where $\otimes$ denotes Kronecker product. The coriolis/centripetal matrix needs terms like $\sin(q)$, $\cos(q)$, multiplied generally in all possible combinations by $\dot{q}(t)$. Therefore, the neural subnet required to estimate $V_m(q,\dot{q})\zeta_2$ appears in Fig. 8. Likewise, the sub NN required for the gravity and frcition terms appear respectively in Figs. 9 and 10.



(a)



(b)

Fig. 6. Response of adaptive controller. (a) Actual and desired joint angles. (b) Representative weight estimates.

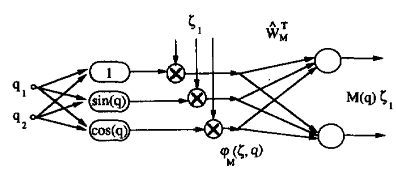


Fig. 7. Neural subnet for estimating $M(q)\zeta_1(t)$.

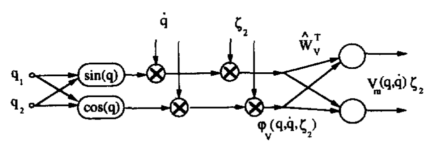


Fig. 8. Neural subnet for estimating $V_m(q,\dot{q})\zeta_2(t)$.

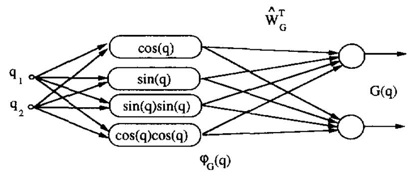


Fig. 9. Neural subnet for estimating $G(q)$.

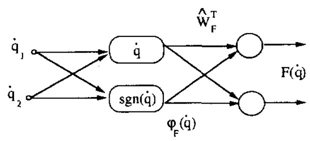
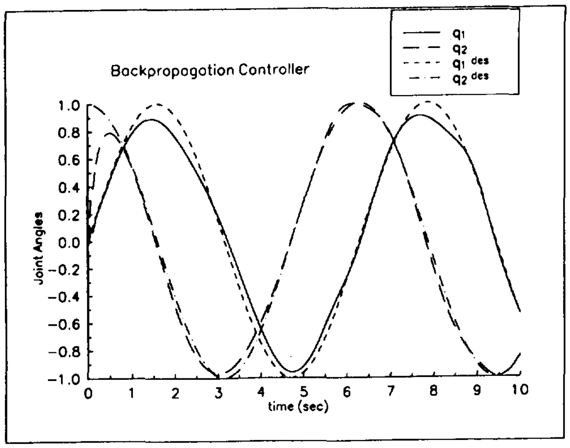
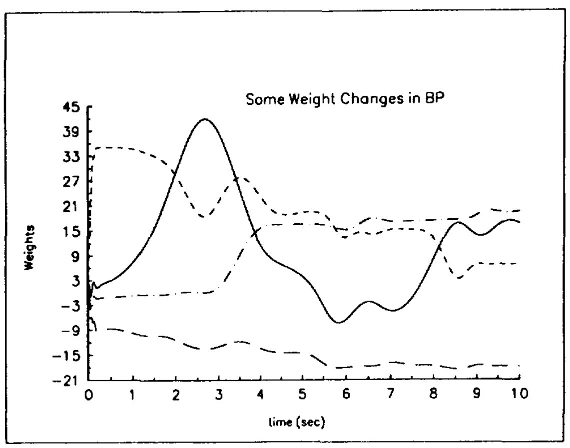


Fig. 10. Neural subnet for estimating $F(\dot{q})$.



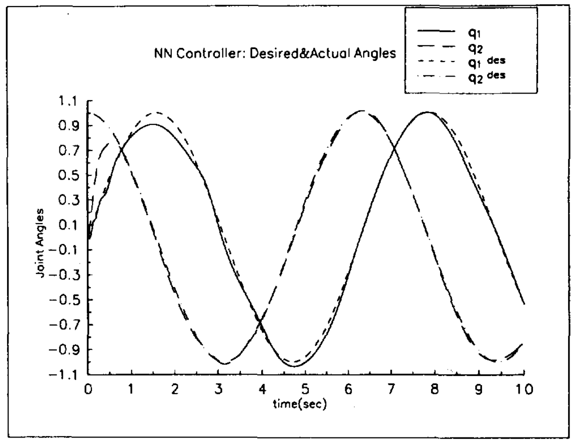
(a)



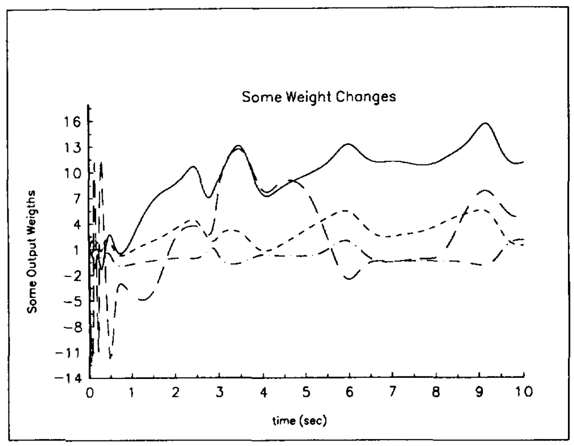
(b)

Fig. 11. Response of NN controller with backprop weight tuning. (a) Actual and desired joint angles. (b) Representative weight estimates.

NN Controller with Backprop Weight Tuning: The response of the controller (3.7) (with $v(t) = 0$) with backprop weight tuning (3.12) (e.g., Theorem 3.2) appears in Fig. 11. Note



(a)



(b)

Fig. 12. Response of NN controller with improved weight tuning. (a) Actual and desired joint angles. (b) Representative weight estimates.

the large values of weights required. In this case they appear to remain bounded, though this cannot in general be guaranteed.

NN Controller with Improved Weight Tuning: The response of the controller (3.7) (with $v(t) = 0$) with the improved weight tuning (3.21) (e.g., Theorem 3.4) appears in Fig. 12. We used $\kappa = 0.1$. The tracking response is better than that using straight backprop, and the weights are guaranteed to remain bounded even though PE does not hold. The comparison with the performance of the standard adaptive controller in Fig. 6 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. 13 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

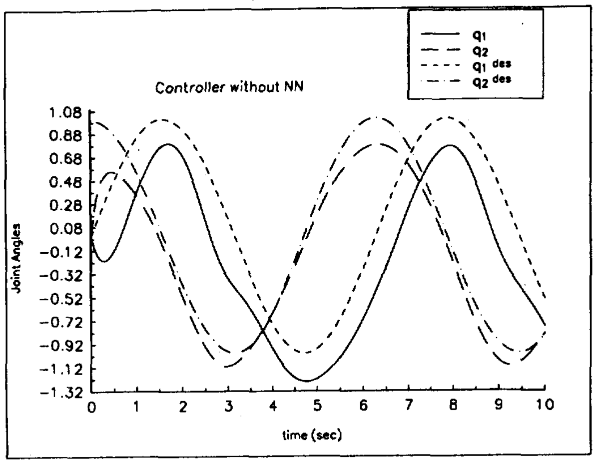


Fig. 13. Response of controller without NN. Actual and desired joint angles.

large enough. This is observed in the figure. It is now clear, however, that the addition of the NN makes a very significant improvement in the tracking performance.

APPENDIX

Proof of Technical Lemma 3.1: In this proof we use the two-norm. First note that (3.10) implies that $\|N^{-1}(t,t-\delta)\| \leq 1/\beta_1$. Moreover, it also implies that for some positive constant β_3

$$\int_{t-\delta}^{t} \|C(\tau)\|^2 d\tau < \beta_3.$$

Note as well

$$\int_{t-\delta}^{t} \|C(\tau)\|\, d\tau = \langle \|C\|, 1\rangle \leq \|\, \|C\|\, \| \cdot \|1\| = \left[\int_{t-\delta}^{t} \|C(\tau)\|^2 d\tau\right]^{1/2} \left[\int_{t-\delta}^{t} 1 d\tau\right]^{1/2} \leq \beta_3^{1/2}\delta^{1/2}.$$

Now, the state trajectory is given by

$$x(t) = x(t_0) + \int_{t_0}^{t} B(\tau)u(\tau)\, d\tau.$$

Using standard techniques involving the adjoint operator to $C(t)$, the initial condition can be determined in terms of $u(t)$ and $y(t)$ over a finite interval, whence it is found that, for all t and with δ the finite constant in (3.10)

$$\begin{aligned} x(t) &= N^{-1}(t,t-\delta) \int_{t-\delta}^{t} C^T(\tau)Y(\tau)\, d\tau \\ &\quad + N^{-1}(t,t-\delta) \int_{t-\delta}^{t} C^T(\lambda)C(\lambda) \int_{\lambda}^{t} B(\tau)u(\tau)\, d\tau\, d\lambda \\ &\equiv x_1(t) + x_2(t). \end{aligned}$$

Now

$$\begin{aligned} \|x_1(t)\| &\leq \|N^{-1}(t,t-\delta)\| \quad \| \int_{t-\delta}^{t} C^T(\tau)y(\tau)d\tau\| \\ &\leq \frac{1}{\beta_1}\int_{t-\delta}^{t} \|C(\tau)\|\|y(\tau)\|d\tau \\ &\leq \frac{Y}{\beta_1}\int_{t-\delta}^{t} \|C(\tau)\|d\tau \leq \frac{Y(\delta\beta_3)^{1/2}}{\beta_1} \end{aligned}$$

with Y an upper bound on $\|y(t)\|$, which is finite since $y(t) \in L_\infty p$.

As for the second term

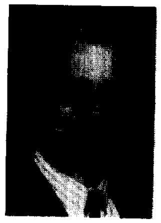
$$\begin{aligned} \|x_2(t)\| &\leq \frac{1}{\beta_1}\int_{t-\delta}^{T} \|C^T(\lambda)C(\lambda)\| \quad \int_{\lambda}^{t} \|B(\tau)u(\tau)\|d\tau d\lambda \\ &\leq \frac{1}{\beta_1}\int_{t-\delta}^{t} \|C^T(\lambda)C(\lambda)\| \quad \int_{t-\delta}^{t} \|B(\tau)u(\tau)\|d\tau d\lambda \\ &\leq \frac{1}{\beta_1}\int_{t-\delta}^{t} \|C(\lambda)\|^2 d\lambda \quad \int_{t-\delta}^{t} \|B(\tau)\|\; \|u(\tau)\|d\tau \\ &\leq \frac{\beta_3}{\beta_1}\beta_4\delta U \end{aligned}$$

with U an upper bound on $\|u(t)\|$ and β_4 an upper bound on $\|B(t)\|$, both guaranteed finite as $u(t) \in L_\infty^m$, $B(t) \in L_\infty^{n\times m}$. □

REFERENCES

[1] F. Albertini and E. D. Sontag, "For neural nets, function determines form," in *Proc. IEEE Conf. Decis. and Control,* 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] S. R. Chu and R. Shoureshi, "Neural-based adaptive nonlinear system identification," in *Proc. Intelligent Contr. Syst.,* vol. 45, ASME Winter Annual Meeting, 1992.
[5] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Syst.,* vol. CAS-25, no. 9, pp. 772–781, Sep. 1978.
[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," in *Proc. Advanced Contr. Issues Robot Manipulators,* 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,.

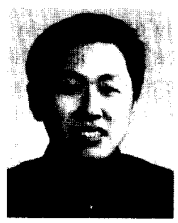
[16] Y. D. Landau, *Adaptive Control: The Model Reference Approach.* New York: Marcel Dekker, 1979.
[17] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators.* New York: Macmillan, 1993.
[18] K. Liu and F. L. Lewis, "Robust control techniques for general dynamic systems," *J. Intell. Robotic Syst.*, vol. 6, pp. 33–49, 1992.
[19] W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds., *Neural Networks for Control.* Cambridge, MA: MIT Press, 1991.
[20] K. S. Narendra, "Adaptive control using neural networks," *Neural Networks for Control.* W. T. Miller, R. S. Sutton, and P. J. Werbos, Eds. Cambridge, MA: MIT Press, 1991, pp. 115–142.
[21] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Automat. Control,* vol. 32, no. 2, pp. 134–145, Feb. 1987.
[22] 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.
[23] K. S. Narendra, "Adaptive control of dynamical systems using neural networks," *Handbook of Intelligent Control,* D. A. White and D. A. Sofge, Eds. New York: van Nostrand Reinhold, 1992, pp. 141–183.
[24] J. Olvera, X. Guan, and M. T. Manry, "Theory of monomial networks," in *Proc. Symp. Implicit and Nonlinear Systems,* Dec. 1992, pp. 96–101.
[25] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma, and Y. Uchikawa, "Trajectory control of robotic manipulators," *IEEE Trans. Ind. Elec.,* vol. 38, pp. 195–202, June 1991.
[26] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks.* Reading, MA: Addison-Wesley, 1989.
[27] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Comp.*, vol. 3, pp. 246–257, 1991.
[28] M. M. Polycarpou and P. A. Ioannu, "Identification and control using neural network models: design and stability analysis," Dept. Elect. Eng. Syst., Univ. S. Cal., Tech. Report 91-09-01, Sep. 1991.
[29] M. M. Polycarpou and P. A. Ioannu, "Neural networks as on-line approximators of nonlinear systems," in *Proc. IEEE Conf. Decision Contr.,* Dec. 1992, pp. 7–12.
[30] T. Poggio and F. Girosi, "A theory of networks with for approximation and learning," Artif. Intel. Lab, MIT, Tech. Report Memo no. 1140, 1989.
[31] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Syst., Man, Cybern.,* to appear, 1994.
[32] N. Sadegh, "Nonlinear identification and control via neural networks," in *Proc. Contr. Syst. Inexact Dynamics Models,* vol. 33, ASME Winter Annual Meeting, 1991.
[33] R. M. Sanner and J.-J. E. Slotine, "Stable adaptive control and recursive identification using radial gaussian networks," in *Proc. IEEE Conf. Decis. Contr.,* 1991.
[34] S. S. Sastry and M. Bodson, *Adaptive Control.* Englewood Cliffs, NJ: Prentice-Hall, 1989.
[35] J.-J. E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Automat. Control,* vol. 33, no. 11, pp. 995–1003, Nov. 1988.
[36] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice-Hall, 1991.
[37] 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.
[38] P. J. Werbos, "Backpropagation: Past and future," in *Proc. 1988 Int. Conf. Neural Nets,* vol. 1, 1989, pp. I343–I353.
[39] 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, and studied 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 in 1971. In 1977 he received the Master's of Science in Aeronautical Engineering from the University of West Florida. 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.

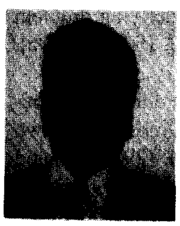
Dr. Lewis was awarded the Moncrief-O'Donnell Endowed Chair in 1990 at the Automation and Robotics Research Institute of The University of Texas at Arlington. He is the author/co-author of 80 journal papers and the following books *Optimal Control Optimal Estimation, Applied Optimal Control and Estimation, Aircraft Control and Simulation, Control or 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 Eduction F. E. Terman Award, three Sigma Xi Research Awards, and the UTA Halliburton Research Award.



Kai Liu (S'89–M'90) 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, GA, in 1990.

From 1982 to 1986 he was a faculty member in the Department of Electrical Engineering, Beijing Polytechnic University, where his research interests focused on the design techniques of microcomputer-based control systems. He is currently working at the Automation & Robotics Research Institute, University of Texas at Arlington, as a research scientist. 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, IEEE Control Society, and IEEE Robotics & Automation Society.



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 and the M.S. degree in electrical engineering and applied physics from Case Western Reserve University, Cleveland, OH, in 1992. Since 1992, he has been a doctoral student at the University of Texas at Arlington and research assistant at the Automation and Robotics Research Institute, Fort Worth, TX.

His research interest include neural networks, robotics, nonlinear systems, adaptive control and image processing.

Mr. Yeşildirekis a recipient of the Turkish Educational Ministry Scholarship.