Proceedings of the 1993 International
Symposium on Intelligent Control
Chicago, Illinois, USA - August 1993

NEURAL NET ROBOT CONTROLLER WITH GUARANTEED TRACKING PERFORMANCE

F. L. Lewis, K. Liu, and A. Yesildirek
Automation and Robotics Research Institute
The University of Texas at Arlington
7300 Jack Newell Blvd. S
Ft. Worth, Texas 76118

Research supported by NSF Grants MSS-9114009, IRI-9216545

ABSTRACT

A neural net (NN) controller for a general serial-link robot arm is developed. The NN has 2 layers so that linearity in the parameters holds, but the 'net functional reconstruction error' is taken as nonzero. The structure of the NN controller is derived using a filtered error/passivity approach. 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. A novel on-line weight tuning algorithm including a correction term to backpropagation guarantees tracking as well as bounded weights. The notions of a passive NN and a robust NN are introduced.

1. INTRODUCTION

Much has been written about NN for system identification (e.g. [1,6,14]) 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 [4], as well as some notions on NN for robot control, are given in [8,11,12,15,18,19,20,24].

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 2-layer NN, where linearity in the parameters holds (c.f. [16,18,19,20]). This may be considered as a step in extending adaptive control theory to NN control theory. The general nonlinear (multilayer) NN is considered in [10]. Some notions in robot control [9] are tied here to some notions in NN theory. The NN structure comes from filtered error and passivity notions standard in robot control. 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.

Unlike adaptive robot control, where a 'matrix of robot functions' must be tediously computed from the dynamics of each specific arm [2,9], the basis functions for the proposed NN controller can be determined from the physics (Lagrangian dynamics) of general robot arms.

It is shown that the backpropagation tuning technique generally yields unbounded NN weights if the net cannot exactly reconstruct a certain nonlinear control function, or if there are bounded unmodelled disturbances in the robot dynamics. It is shown that the back-propagation tuning algorithm yields a passive neural net. This guarantees that all signals in the closed-loop system are bounded under an additional persistency of excitation (PE) condition. A modified weight tuning algorithm avoids the need for PE by making the NN robust, that is, strictly passive in a sense defined herein.

2. BACKGROUND

Let $\mathbf{R}$ denote the real numbers, $\mathbf{R}^n$ denote the real n-vectors, $\mathbf{R}^{m\times n}$ the real $m \times n$ matrices. Let S be a compact simply connected set of $\mathbf{R}^n$. With maps $f: S \to \mathbf{R}^m$, define $C^m(S)$ as the space such that f is continuous. We denote by $\|.\|$ any suitable vector norm. When it is required to be specific we denote the p-norm by $\|.\|_p$.

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

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

with tr() the trace. The associated inner product is $\langle A,B\rangle_F = \mathrm{tr}(A^TB)$. The Frobenius norm is nothing but the vector 2-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 2-norm so that $\|Ax\|_2 \le \|A\|_F \|x\|_2$, with $A \in \mathbf{R}^{m\times n}$ and $x \in \mathbf{R}^n$. When $x(t) \in \mathbf{R}^n$ is a function of time we use the standard L_p norms [9].

2.1 Neural Networks

Given $x \in \mathbf{R}^{N_1}$, a three layer neural net (NN) (Fig. 2.1) has a net output given by

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

with $\sigma(.)$ the activation functions, v_{jk} the first-to-second layer interconnection weights, and w_{ij} the second-to-third layer interconnection weights. The $\theta_{v\ell}$, $\theta_{w\ell}$, $\ell = 1,2,\ldots$, are threshold offsets and N_2 is the number of hidden-layer neurons.

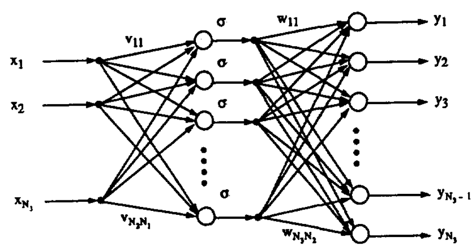

Figure 2.1 Three Layer Neural Network

The NN equation may be conveniently expressed in matrix format by defining $x = [x_0 \; x_1 \; x_2 \ldots x_{N1}]^T$, $y = [y_1 \; y_2 \ldots y_{N3}]^T$, and weight matrices $W^T = [w_{ij}]$, $V^T = [v_{jk}]$. Including $x_0 = 1$ in x allows one to include the threshold vector $[\theta_{v1} \; \theta_{v2} \ldots \theta_{vN1}]^T$ 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,

$$y = W^T \sigma(V^Tx). \qquad (2.2)$$

We define the action of the activation functions on a vector $v = [v_i]$ by $\sigma(v) = [\sigma(v_i)]$. Including a first entry of 1 in the vector $\sigma(V^T x)$ allows one to incorporate the thresholds θ_{wj} 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 $\sigma(V^T x)$ by $\sigma_0 = 1$, we loosely say that $x \in R^{N1}$ and $\sigma: R^{N2} \rightarrow R^{N2}$.

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 $x \in S$, 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 $x \in R^n$, (2.3) holds with $\|\varepsilon\|_\infty < \varepsilon_N$.

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 [3,7,17,20]. Typical selections for $\sigma()$ include the sigmoid, polynomial basis functions, radial basis functions, etc. The issues of selecting σ, and of choosing N_2 for a specified $S \subset R^n$ and ε_N are current topics of research (see e.g. [17]).

2.2 Stability and Passive Systems

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

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

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

A system with input u(t) and output y(t) is said to be passive [5,9,23] 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

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

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. 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 conclude some internal boundedness properties of the system without the usual assumptions of observability (e.g. persistence of excitation), stability, etc.

2.3 Robot Arm Dynamics

The dynamics of an n-link robot manipulator may be expressed in the Lagrange form [9]

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

with $q(t) \in R^n$ 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 τ_d and the control input torque is $\tau(t)$.

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

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

and the filtered tracking error 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

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

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

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

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.

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 [23] guarantee that e(t) exhibits stable behavior. The following properties of the robot dynamics are required [9].

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

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

Property 4: The unknown disturbance satisfies $\|\tau_d\|_\infty < b_d$, 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

$$L = \tfrac{1}{2} r^T M r$$

so that, using (2.14)

$$\dot{L} = r^T M\dot{r} + \tfrac{1}{2} r^T \dot{M} r = -r^T K_v r + \tfrac{1}{2} r^T(\dot{M} - 2V_m) r + r^T \zeta_o$$

whence skew-symmetry yields the power form

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

■

3. TWO-LAYER NN CONTROLLER

We consider in the remainder of the paper the NN for the case of fixed V. For the general case see [10]. We propose various weight tuning algorithms, showing their relation to standard backpropagation. 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 to guarantee stability.

Define $\phi(x)=\sigma(V^Tx)$ so that the net output is

$$y= W^T\phi(x). \tag{3.1}$$

Then, for suitable NN approximation properties, $\phi(x)$ must satisfy some conditions (e.g. [19]). Take $N_1= n$, $N_3= m$.

Definition 3.1

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

1. A constant function on S can be expressed as (3.1) for finite N_2.

2. The functional range of NN (3.1) is dense in $C^m(S)$ 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 $S\subset R^n$ and ϵ_N, are topics of current research.

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

$$f(x)= W^T\phi(x) + \epsilon(x), \tag{3.2}$$

where $\phi(x)$ provides a suitable basis and $\|\epsilon(x)\|_\infty< \epsilon_N$, with the bound ϵ_N 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 [2], and it is easy to determine the required robotic functions [9]. 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 5 how to select $\phi(x)$ 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, as required in standard adaptive approaches, is avoided.

3.1 Controller Structure and Error System Dynamics

Define the NN functional estimate by

$$\hat f(x)= \hat W^T\phi(x), \tag{3.3}$$

with $\hat W$ the current values of the NN weights. With W the ideal weights required in (3.2) define the weight deviations or weight estimation errors as

$$\tilde W= W - \hat W. \tag{3.4}$$

Assume the ideal weights are bounded by known values so that

$$\|W\|_F \le W_{max}. \tag{3.5}$$

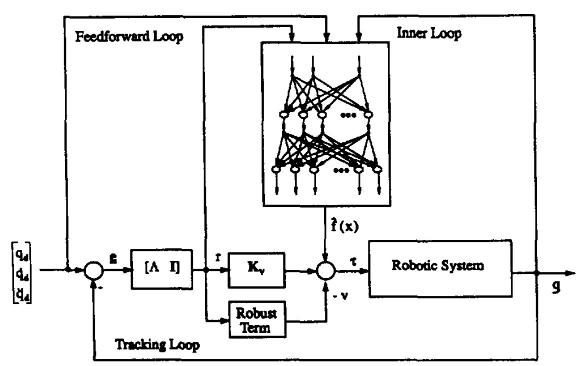

Figure 3.1 Neural Net Control Structure

Select the control input defined in (2.13), namely

$$\tau= \hat W^T\phi(x) + K_v r, \tag{3.6}$$

Then, the closed-loop filtered error dynamics become

$$M\dot r= -(K_v+V_m)r + \tilde W^T\phi(x) + (\epsilon+\tau_d) = -(K_v+V_m)r + \zeta_1. \tag{3.7}$$

The proposed NN control structure is shown in Fig. 3.1, where $\underline{q}= [q^T\ \dot q^T]^T$, $\underline{e}= [e^T\ \dot e^T]^T$.

3.2 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.

Observability and Persistency of Excitation

A technical result 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 $x\in R^n$, $u\in R^m$, $y\in R^p$. Since the state transition matrix is the identity matrix, the observability gramian is

$$N(t,t_0)= \int_{t_0}^{t} C^T(\tau)C(\tau)\ d\tau. \tag{3.8}$$

Then we say the system is uniformly completely observable [21] if there exist finite constants $\delta>0$, $\beta_1>0$, $\beta_2>0$ so that, for all $t\ge0$,

$$\beta_1 I \le N(t+\delta,t) \le \beta_2 I. \tag{3.9}$$

It is not difficult to show that if the system (0, B(t), C(t)) is uniformly completely observable with B(t) bounded, then boundedness of u(t) and y(t) guarantees the state x(t) is bounded. 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) [21] if there exist α_1, α_2, $\delta > 0$ so that

$$\alpha_1 I \le \int_{t_0}^{t_0+\delta} w(\tau)w^T(\tau)\ d\tau \le \alpha_2 I \quad , \text{ for all } t_0\ge0. \tag{3.10}$$

Case of Nonzero NN Reconstruction Error ϵ_N

There now follows a sequence of results designed to provide guaranteed tracking under increasingly mild assumptions.

Theorem 3.2

Let the desired trajectory q_d, $\dot q_d$, $\ddot q_d$ be bounded. Take the control input for (2.7) as (3.6) and weight tuning provided by

$$\dot{\hat W} = F\phi r^T, \tag{3.11}$$

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 GUUB and the NN weight estimates $\hat W$ are bounded.

Proof:

Define the Lyapunov function candidate

$$L= \tfrac12 r^TMr + \tfrac12 tr(\tilde W^TF^{-1}\tilde W) \tag{3.12}$$

Differentiating yields

$$\dot L= r^TM\dot r + \tfrac12 r^T\dot Mr + tr(\tilde W^TF^{-1}\dot{\tilde W}),$$

whence substitution from (3.7) yields

$$\dot L= -r^TK_vr + \tfrac12 r^T(\dot M-2V_m)r + tr\,\tilde W^T(F^{-1}\dot{\tilde W} + \phi r^T) + r^T(\epsilon + \tau_d).$$

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 (3.11).

Now,

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

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

$$\|r\| > (\varepsilon_N + b_d)/K_{vmin}. \tag{3.13}$$

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

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

since M(q) is bounded. Therefore, according to (3.11), (3.7) the dynamics relative to $\tilde{W}$ are given by

$$\begin{aligned} \dot{\tilde{W}} &= -F\phi r^T \\ y^T &= \phi^T\tilde{W} \end{aligned} \tag{3.14}$$

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

Using the Kronecker product $\otimes$ allows one to write

$$\frac{d}{dt}\mathrm{vec}(\tilde{W}) = -(I \otimes F\phi)r$$

$$y = (I \otimes \phi^T)\,\mathrm{vec}(\tilde{W})$$

where the vec(A) operator stacks the columns of a matrix A to form a vector, and one notes that $\mathrm{vec}(z^T) = z$ 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 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 $F\phi(x(t))$.] ∎

Note from (3.13) 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_{vmin} 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 K_v should be large enough and the initial filtered error r(0) small enough.

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 unmodelled disturbances in the robot arm dynamics. In this case the PE assumption on $\phi(x)$ is not needed.

Corollary 3.3

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

$$\tau = \hat{W}^T\phi(x) + K_v r \tag{3.15}$$

with weight tuning provided by

$$\dot{\hat{W}} = F\phi r^T, \tag{3.16}$$

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

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T\phi(x). \tag{3.17}$$

Using the Lyapunov function candidate (3.12) with the new assumptions results in

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

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

$$\int_0^\infty -\dot{L}\, dt < \infty . \tag{3.18}$$

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 (3.17) verify the boundedness of $\ddot{L}$, and hence the uniform continuity of $\dot{L}$. This allows us to invoke Barbalat's Lemma [9,23] in connection with (3.18) to conclude that $\dot{L}$ goes to zero with t, and hence that r(t) vanishes. ∎

Note that (3.16) is nothing but the backpropagation algorithm for the 1-layer case; Corollary 3.3 indicates when backprop alone should suffice. However, Theorem 3.2 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 unmodelled 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 [18], or e-modification [13] of the weight tuning equation as follows.

Theorem 3.4

Given the hypotheses of Theorem 3.2, let the NN weight tuning be modified as

$$\dot{\hat{W}} = F\phi r^T - \kappa F\|r\|\hat{W}, \tag{3.19}$$

with $\kappa > 0$ a design parameter, and assume no PE condition on $\phi(x)$. Then the filtered tracking error r(t) and the NN weight estimates $\hat{W}(t)$ are GUUB.

Proof:

Using the Lyapunov function candidate (3.12) with tuning rule (3.19) yields (c.f. proof of Theorem 3.2)

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

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

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

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

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

which is guaranteed positive as long as

$$\|r\| > \frac{\kappa W_{max}^2/4 + (\varepsilon_N + b_d)}{K_{vmin}} \tag{3.20}$$

or

$$\|\tilde{W}\|_F > W_{max}/2 + \sqrt{W_{max}^2/4 + (\varepsilon_N + b_d)/\kappa} . \tag{3.21}$$

This demonstrates the GUUB of both $\|r\|$ and $\|\tilde{W}\|_F$.

∎

A comparison with the results of [13] 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, however, that arbitrarily small tracking error bounds may be achieved by selecting large control gains K_v. PE is not needed to establish the bounds on $\hat{W}$ with the modified weight tuning algorithm.

4. PASSIVITY PROPERTIES OF THE NN

The closed-loop error system (3.7) appears in Fig. 4.1. 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. 3.1.

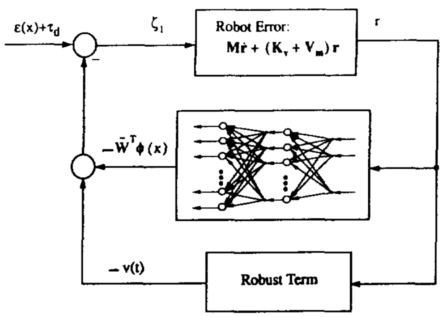

Figure 4.1 Neural Net Closed-loop Error System

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 a 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 backprop weight tuning algorithm (3.11) makes the map from r(t) to $-\tilde{W}^T\phi$ a passive map.

Proof:
Selecting the Lyapunov function

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

and evaluating $\dot{L}$ along the trajectories of (3.14) yields

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

which is in power form. ■

Thus, the robot error system in Fig. 4.1 is state strict passive (SSP) and the weight error block is passive; this guarantees the dissipativity of the closed-loop system [23]. 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 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 (3.19) 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

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

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

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

and evaluating $\dot{L}$ yields

$$\dot{L} = \mathrm{tr}\, \tilde{W}^T F^{-1} \dot{\tilde{W}} = r^T(-\tilde{W}^T\phi) - \kappa\|r\| \left(\|\tilde{W}\|_F^2 - \langle\tilde{W}, W\rangle_F\right)$$
$$\leq r^T(-\tilde{W}^T\phi) - \kappa\|r\| \left(\|\tilde{W}\|_F^2 - W_{max}\|\tilde{W}\|_F\right)$$

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

■

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.

5. PARTITIONED NEURAL NETS AND BASIS FUNCTIONS

A major advantage of the 2-layer NN approach over adaptive control, where all the unknown dynamics are lumped together, 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.

In fact, one can use four NN, one to reconstruct each term in f(x) of (2.11). We call this a structured NN, and it is direct to show that the individual partitioned NNs can be separately tuned, making for a faster weight update procedure. Indeed, if F and κ are selected as block diagonal matrices, then the weight update law (3.19) is

$$\begin{aligned}
\dot{\hat{W}}_M &= F_M\phi_M r^T - \kappa_M F_M\|r\|\hat{W}_M \\
\dot{\hat{W}}_V &= F_V\phi_V r^T - \kappa_V F_V\|r\|\hat{W}_V \\
\dot{\hat{W}}_G &= F_G\phi_G r^T - \kappa_G F_G\|r\|\hat{W}_G \\
\dot{\hat{W}}_F &= F_F\phi_F r^T - \kappa_F F_F\|r\|\hat{W}_F .
\end{aligned} \tag{5.1}$$

It is not difficult to show that the basis functions ϕ_M, ϕ_V, ϕ_G, ϕ_F can be chosen from the physics of the robot dynamics. It is only necessary to find a set of basis functions for the arm T_i matrix partial derivatives $\partial T_i/\partial q$, then a complete basis set for f(x) follows in terms of Kronecker products with the joint variables and velocities. This basis set works for a general robot arm. The requisite basis set will consist of polynomials up to a known maximum order in the variables of x.

6. ILLUSTRATIVE DESIGN AND SIMULATION

A planar 2-link arm used extensively in the literature for illustration purposes appears in Fig. 6.1; the dynamics are given, for instance in [9]. 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.

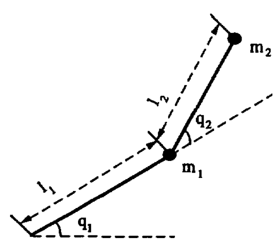

Fig 6.1 2-Link Planar Elbow Arm

Adaptive Controller: Baseline Design

For comparison, a standard adaptive controller is given by [22]

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

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

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. 6.2. 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 [5].

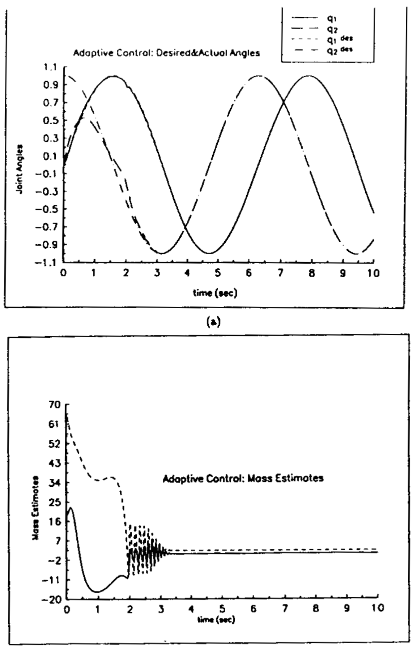

Figure 6.2. Response of Adaptive Controller. (a) Actual and Desired Joint Angles. (b) Representative Weight Estimates.

Neural Net Controller Structure

The NN controller appears in Fig. 3.1. From well-known properties of the dynamics of any 2-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. 6.3, where $\otimes$ denotes Kronecker product and $\zeta_1 = \ddot{q}_d + \Lambda\dot{e}$. The coriolis/centripetal matrix needs terms like sin(q),

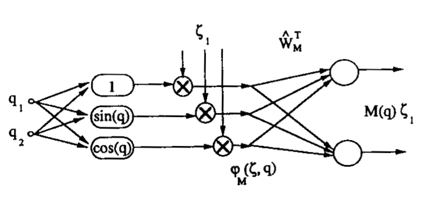

Figure 6.3 Neural Subnet for Estimating $M(q)\,\zeta_1(t)$

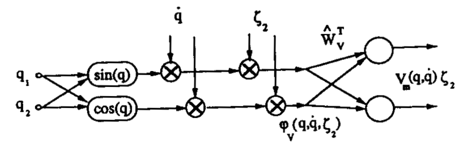

Figure 6.4 Neural Subnet for Estimating $V_m(q,\dot{q})\zeta_2$

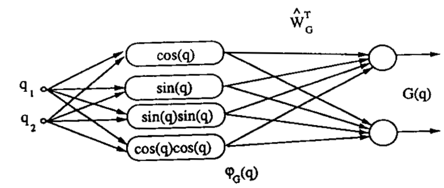

Figure 6.5 Neural Subnet for Estimating G(q)

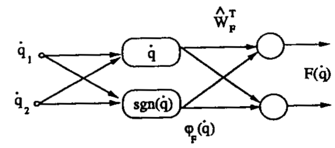

Figure 6.6 Neural Subnet for Estimating $F(\dot{q})$

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. 6.4, where $\zeta_2 = \dot{q}_d + \Lambda e$. Likewise, the sub NN required for the gravity and friction terms appear respectively in Fig. 6.5, Fig. 6.6.

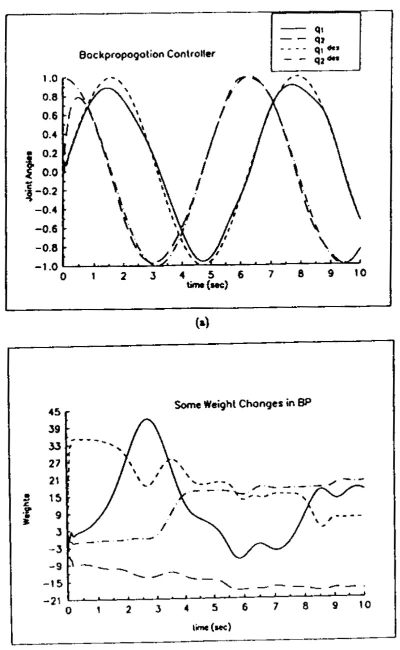

Figure 6.7 Response of NN Controller with Backprop Weight Tuning. (a) Actual and Desired Joint Angles. (b) Representative Weight Estimates.

NN Controller With Backprop Weight Tuning

The response of the controller (3.6) with backprop weight tuning (3.11) (e.g. Theorem 3.2) appears in Fig. 6.7. Note 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 (3.6) with the improved weight tuning (3.19) (e.g. Theorem 3.4) appears in Fig. 6.8. 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. 6.2 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. 6.9 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. However, it is now clear that the addition of the NN makes a very significant improvement in the tracking performance.

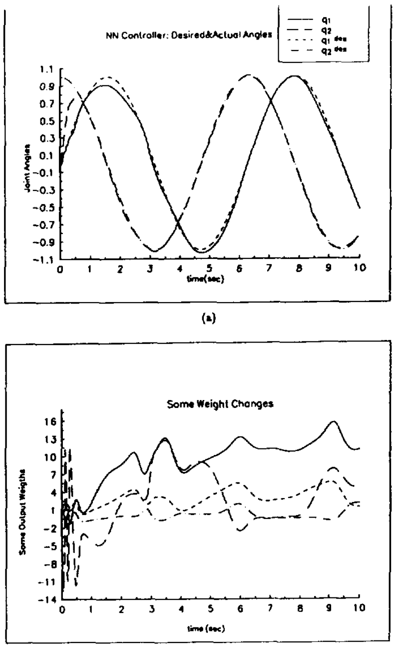

Figure 6.8 Response of NN Controller with Improved Weight Tuning. (a) Actual and Desired Joint Angles. (b) Representative Weight Estimates.

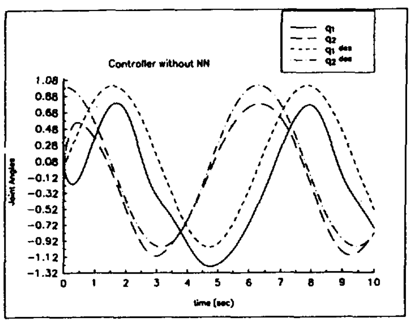

Figure 6.9 Control Response without NN. Actual and Desired Joint Angles.

REFERENCES

[1] S.R. Chu and R. Shoureshi, "Neural-based adaptive nonlinear system identification," Inteligent Control Systems, DSC-vol. 45, ASME Winter Annual Meeting, 1992.
[2] J.J. Craig, Adaptive Control of Robot Manipulators, Reading, VA: Addison-Wesley, 1988.
[3] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Mathematics of Control, Signals and Systems, vol. 247, pp. 303-314, 1989.
[4] G.C. Goodwin, "Can We Identify Adaptive Control?" Proc. European Control Conf., pp. 1714-1725, July 1991.
[5] G.C. Goodwin and K.S. Sin, Adaptive Filtering, Prediction, and Control, New Jersey: Prentice-Hall, 1984.
[6] 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.
[7] K. Hornik, M. Stinchombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks, vol. 2, pp. 359-366, 1989.
[8] Y. Iiguni, H. Sakai, and H. Tokumaru, "A nonlinear regulator design in the presence of system uncertaities using multilayer neural networks," IEEE Trans. Neural Networks, vol. 2, no. 4, pp. 410-417, July 1991.
[9] F.L. Lewis, C.T. Abdallah, and D.M. Dawson, Control of Robot Manipulators, New York: Macmillan, 1993.
[10] F.L. Lewis, K. Liu, and A. Yesildirek, "Multilayer neural net robot controller with guaranteed tracking performance," submitted, 1993.
[11] W.T. Miller, R.S. Sutton, P.J. Werbos, ed., Neural Networks for Control, Cambridge: MIT Press, 1991.
[12] K.S. Narendra, "Adaptive Control Using Neural Networks," Neural Networks for Control, pp 115-142. ed. W.T. Miller, R.S. Sutton, P.J. Werbos, Cambridge: MIT Press, 1991.
[13] K.S. Narendra and A.M. Annaswamy, "A new adaptive law for robust adaptation without persistent adaptation," IEEE Trans. Automat. Control, vol. AC-32, no. 2, pp. 134-145, Feb. 1987.
[14] 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.
[15] 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.
[16] Y.H. Pao, Adaptive Pattern Recognition and Neural Networks, Reading, VA: Addison-Wesley, 1989.
[17] J. Park and I.W. Sandberg, "Universal approximation using radial-basis-function networks," Neural Comp., vol. 3, pp. 246-257, 1991.
[18] M.M. Polycarpou and P.A. Ioannu, "Identification and control using neural network models: design and stability analysis," Tech. Report 91-09-01, Dept. Elect. Eng. Sys., Univ. S. Cal., Sept. 1991.
[19] N. Sadegh, "Nonlinear identification and control via neural networks," Control Systems with Inexact Dynamics Models, DSC-vol. 33, ASME Winter Annual Meeting, 1991.
[20] R.M. Sanner and J.-J.E. Slotine, "Stable adaptive control and recursive identification using radial gaussian networks," Proc. IEEE Conf. Decision and Control, Brighton, 1991.
[21] S.S. Sastry and M. Bodson, Adaptive Control, New Jersey: Prentice-Hall, 1989.
[22] 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.
[23] J.-J.E. Slotine and W. Li, Applied Nonlinear Control, New Jersey: Prentice-Hall, 1991.
[24] 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.