Proceedings of the 32nd Conference
on Decision and Control
San Antonio, Texas • December 1993

NEURAL NET ROBOT CONTROLLER: STRUCTURE AND STABILITY PROOFS

F. L. Lewis, A. Yesildirek, and K. Liu
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 multilayer neural net (NN) controller for a general serial-link robot arm is developed. The structure of the NN controller is derived using a filtered error/passivity approach. No learning phase is needed. It is argued that standard backpropagation tuning, when used for real-time closed-loop control, can yield unbounded NN weights if: (1) the net cannot exactly reconstruct a certain required nonlinear control function, (2) there are bounded unknown disturbances in the robot dynamics, or (3) the robot arm has more than one link (i.e. nonlinear case). Novel on-line weight tuning algorithms given here include correction terms to backpropagation, plus an added robustifying signal, and guarantee tracking as well as bounded weights. Notions of NN passivity are given.

1. INTRODUCTION

Much has been written about NN for system identification (e.g. [8]) 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, as well as some notions on NN for robot control, are given in [3,6,9,11,13,14,17]. See also [5].

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'. Some of these problems have been addressed for the 2-layer NN, where linearity in the parameters holds [11,13,14].

In this paper we confront these deficiencies for the full nonlinear 3-layer NN. Some notions in robot control [4] are tied to some notions in NN theory. The NN weights are tuned on-line, with no 'learning phase' needed. Fast convergence of the tracking error is comparable to that of adaptive controllers. 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 a required nonlinear function.

It is shown here that the backpropagation tuning technique generally yields unbounded NN weights if the net cannot exactly reconstruct a certain nonlinear robot function, or if there are bounded unmodelled disturbances in the robot dynamics, or if the robot function is not linear (which it never is for arms with more than one link). Modified weight tuning rules introduced here guarantee tracking and bounded weights for the general case. The modifications consist of: (1) the e-modification in [7], and (2) a term corresponding to a second-order forward propagating wave in the backprop tuning network [8]. Also required is a robustifying extra control signal.

2. BACKGROUND

Let R denote the real numbers, R^n denote the real n-vectors, $R^{m\times n}$ the real m x n matrices. Let S be a compact simply connected set of R^n. With map f: $S \rightarrow R^m$, 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 $A=[a_{ij}]$, $B \in R^{m\times n}$ the Frobenius norm is defined by

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

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

2.1 Neural Networks

Given $x_k \in R$, a three-layer neural net (NN) (Fig. 2.1) has a net output given by

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

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 the number of neurons in layer ℓ is N_ℓ, with N_2 the number of hidden-layer neurons.

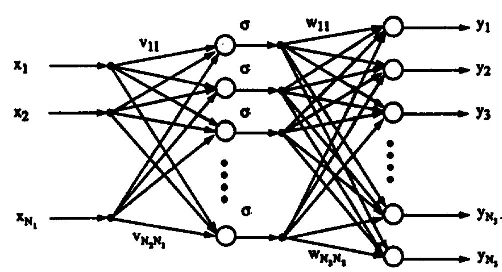

Figure 2.1 Three Layer Neural Network

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 thresholds θ_{vj} 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,

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

with the vector of activation functions defined by $\sigma(z) = [\sigma(z_1)\ldots\sigma(z_n)]^T$ for a vector $z \in R^n$ with components z_i (c.f. [13]). Including 1 as the first element of $\sigma(z)$ (i.e. above $\sigma(z_1)$) 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, vectors x and $\sigma()$ may be thus augmented, we loosely say that $x \in R^{N1}$ and σ: $R^{N2} \to R^{N2}$.

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

$$f(x) = W^T \sigma(V^T x) + \epsilon(x), \qquad (2.3)$$

with $N_1 = n$, $N_3 = m$, and $\epsilon(x)$ a NN functional reconstruction error vector. If there exist N_2 and constant 'ideal' weights W and V so that $\epsilon = 0$ for all $x \in S$, we say f(x) is in the functional range of the NN. In general, given a constant real number $\epsilon_N > 0$, 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 $\|\epsilon\| < \epsilon_N$.

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 [10,14]. In fact, the functional range of NN (2.2) can be shown to be dense in $C^m(S)$.

Typical selections for $\sigma()$ include the sigmoid, hyperbolic tangent, 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. [10]).

2.2 Stability and Passive Systems

Consider the nonlinear system

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

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

A system with input u(t) and output y(t) is said to be passive [4,16] if it verifies an equality of the so-called 'power form'

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

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

$$\int_0^T y^T(\tau) u(\tau)\, d\tau \geq \int_0^T g(\tau)\, d\tau \; - \gamma^2 \qquad (2.5)$$

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

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

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

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, and are not aware of its use previously in the literature. 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 assumption of persistence of excitation.

2.3 Robot Arm Dynamics

The dynamics of an n-link robot manipulator may be expressed in the Lagrange form [4]

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

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 (including e.g. unstructured unmodelled dynamics) are denoted by 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

$$e(t) = q_d(t) - q(t) \qquad (2.8)$$

and the filtered tracking error is

$$r = \dot{e} + \Lambda e \qquad (2.9)$$

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

$$M\dot{r} = -V_m r - \tau + f + \tau_d \qquad (2.10)$$

where the nonlinear robot function is

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

and, for instance,

$$x = [e^T \;\; \dot{e}^T \;\; q_d^T \;\; \dot{q}_d^T \;\; \ddot{q}_d^T]^T. \qquad (2.12)$$

Define now a control input torque as

$$\tau_o = \hat{f} + K_v r \qquad (2.13)$$

with $\hat{f}(x)$ an estimate of f(x) and a gain matrix $K_v = K_v^T > 0$. Note that τ_o incorporates a PD term $K_v r = K_v(\dot{e} + \Lambda e)$. The closed-loop system becomes

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

where the functional estimation error is given by

$$\tilde{f} = f - \hat{f} \qquad (2.15)$$

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, standard techniques [16] guarantee that e(t) exhibits stable behavior.

The following properties of the robot dynamics are required [4].

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 $\|\tau_d\| < b_d$, 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 nonnegative 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

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

3. NN CONTROLLER

In this section we derive a NN controller for the robot dynamics in Section 2. We propose various weight tuning algorithms that guarantee tracking, including standard backpropagation. It is shown that with backpropagation tuning the NN can only be guaranteed to perform suitably in closed-loop under ideal conditions (which require e.g. f(x) linear). A modified tuning algorithm is then proposed so that the NN controller performs under realistic conditions.

Thus, assume that the nonlinear robot function (2.11) is given by a neural net as in (2.3) for some constant 'ideal' NN weights W and V, where the net reconstruction error $\epsilon(x)$ is bounded by a known constant ϵ_N. We only need to know that such ideal weights exist; their actual values are not required. For notational convenience define the matrix of all the weights as

$$Z = \begin{bmatrix} W \\ V \end{bmatrix}, \tag{3.1}$$

with padding by zeros for dimensional consistency.

3.1 Some Bounding Assumptions and Facts

Some <u>bounding assumptions/facts</u> are now stated.

<u>Assumption 1</u>

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

$$\|Z\|_F \le Z_M \tag{3.2}$$

with Z_M known.

<u>Assumption 2</u>

The desired trajectory is bounded in the sense, for instance, that for some known constant $Q_d \in \mathbb{R}$,

$$\begin{Vmatrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{Vmatrix} \le Q_d. \tag{3.3}$$

■

<u>Fact 3</u>

For each time t, x(t) in (2.12) is bounded by

$$\|x\| \le c_1 Q_d + c_2 \|r\| \tag{3.4}$$

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

The next discussion is of major importance in this paper (c.f. [12]). With $\hat{V}$, $\hat{W}$ some estimates of the ideal weight values, define the weight deviations or <u>weight estimation errors</u> as

$$\tilde{V} = V - \hat{V}, \quad \tilde{W} = W - \hat{W}, \quad \tilde{Z} = Z - \hat{Z} \tag{3.5}$$

and the <u>hidden layer output error</u> for a given x as

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T x) - \sigma(\hat{V}^T x). \tag{3.6}$$

The Taylor series expansion for a given x is

$$\sigma(V^T x) = \sigma(\hat{V}^T x) + \sigma'(\hat{V}^T x)\,\tilde{V}^T x + O(\tilde{V}^T x)^2 \tag{3.7}$$

with $\sigma'(\hat{z}) = d\sigma(z)/dz|_{z=\hat{z}}$, and $O(z)^2$ denoting terms of order two. Denoting $\hat{\sigma}' = \sigma'(\hat{V}^T x)$, we have

$$\tilde{\sigma} = \sigma'(\hat{V}^T x)\tilde{V}^T x + O(\tilde{V}^T x)^2 = \hat{\sigma}'\tilde{V}^T x + O(\tilde{V}^T x)^2. \tag{3.8}$$

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

$$O(\tilde{V}^T x)^2 = [\sigma(V^T x) - \sigma(\hat{V}^T x)] - \sigma'(\hat{V}^T x)\,\tilde{V}^T x \tag{3.9}$$

we take the following.

<u>Fact 4</u>

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

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

where c_i are computable positive constants. ■

Fact 4 is direct to show using (3.4), some standard norm inequalities, and the fact that $\sigma()$ and its derivative are bounded by constants for RBF, sigmoid, and tanh.

3.2 Controller Structure and Error System Dynamics

Define the <u>NN functional estimate</u> of (2.11) by

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x), \tag{3.10}$$

with $\hat{V}$, $\hat{W}$ the current (estimated) values of the ideal NN weights V, W. With τ_0 defined in (2.13), select the control input

$$\tau = \tau_0 - v = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v, \tag{3.11}$$

with v(t) a function to be detailed subsequently that provides robustness in the face of higher-order terms in the Taylor series. The proposed NN control structure is shown in Fig. 3.1, where $\underline{q} = [q^T \ \dot{q}^T]^T$, $\underline{e} = [e^T \ \dot{e}^T]^T$.

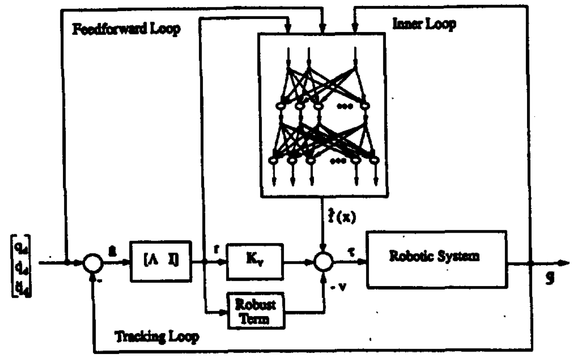

Figure 3.1 Neural Net Control Structure

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

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

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

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

with $\hat{\sigma}$ and $\tilde{\sigma}$ defined in (3.6). Adding and subtracting now $\hat{W}^T\tilde{\sigma}$ yields

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T\hat{\sigma} + \hat{W}^T\tilde{\sigma} + \tilde{W}^T\tilde{\sigma} + (\varepsilon + \tau_d) + v. \tag{3.12}$$

The key step is the use now of the Taylor series approximation (3.8) for $\tilde{\sigma}$, according to which the closed-loop error system is

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T\hat{\sigma} + \hat{W}^T\hat{\sigma}'\tilde{V}^T x + w_1 + v \tag{3.13}$$

where the disturbance terms are

$$w_1(t) = \tilde{W}^T\hat{\sigma}'\tilde{V}^T x + W^T O(\tilde{V}^T x)^2 + (\varepsilon + \tau_d). \tag{3.14}$$

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

$$\begin{aligned} M\dot{r} &= -(K_v + V_m)r + \tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x) + \hat{W}^T\hat{\sigma}'\tilde{V}^T x + w + v \\ &= -(K_v + V_m)r + \zeta_1. \end{aligned} \tag{3.15}$$

where the disturbance terms are

$$w(t) = \tilde{W}^T\hat{\sigma}' V^T x + W^T O(\tilde{V}^T x)^2 + (\varepsilon + \tau_d). \tag{3.16}$$

The next key bound is required. Its importance is in allowing one to overbound w(t) at each time by a <u>known computable function</u>; it follows from Fact 4 and some standard norm inequalities.

<u>Fact 5</u>

The disturbance term (3.16) is bounded according to

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

or

$$\|w(t)\| \le C_0 + C_1\|\tilde{Z}\|_F + C_2\|\tilde{Z}\|_F \|r\| \tag{3.17}$$

with C_i known positive constants. ■

3.3 Weight Updates for Guaranteed Tracking Performance

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

Note that the problem of net weight initialization occurring in other approaches in the literature does not arise, since if $\hat W(0)$, $\hat V(0)$ are taken as zero the PD term $K_v r$ in (3.11) 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 a certain idealized case that demands: (1) no net functional reconstruction error, (2) no unmodelled disturbances in the robot arm dynamics, and (3) no higher-order Taylor series terms. The last amounts to the assumption that $f(x)$ in (2.10) is linear. In this case the tuning rules are straightforward and familiar.

Theorem 3.1

Let the desired trajectory be bounded and suppose the disturbance term $w_1(t)$ in (3.13) is equal to zero. Let the control input for (2.7) be given by (3.11) with $v(t)=0$ and weight tuning provided by

$$\dot{\hat W} = F\hat\sigma r^T \tag{3.18}$$

$$\dot{\hat V} = Gx(\hat\sigma'^T\hat W r)^T \tag{3.19}$$

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

Proof:

Define the Lyapunov function candidate

$$L = \tfrac12 r^TMr + \tfrac12 tr(\tilde W^T F^{-1}\tilde W) + \tfrac12 tr(\tilde V^T G^{-1}\tilde V) \tag{3.20}$$

Differentiating yields

$$\dot L = r^TM\dot r + \tfrac12 r^T\dot M r + tr(\tilde W^T F^{-1}\dot{\tilde W}) + tr(\tilde V^T G^{-1}\dot{\tilde V}),$$

whence substitution from (3.13) (with $w_1=0$, $v=0$) yields

$$\dot L = -r^TK_vr + \tfrac12 r^T(\dot M-2V_m)r + tr\,\tilde W^T(F^{-1}\dot{\tilde W} + \hat\sigma r^T) + tr\,\tilde V^T(G^{-1}\dot{\tilde V} + xr^T\hat W^T\hat\sigma')$$

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

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

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

$$\int_0^\infty -\dot L\, dt < \infty . \tag{3.21}$$

Boundedness of r guarantees the boundedness of e and $\dot e$, whence boundedness of the desired trajectory shows q, $\dot q$, x are bounded. Property 2 then shows boundedness of $V_m(q,\dot q)$. Now, $\ddot L = -2r^TK_v\dot r$, and the boundedness of $M^{-1}(q)$ and of all signals on the right-hand side of (3.13) verify the boundedness of $\ddot L$, and hence the uniform continuity of $\dot L$. This allows one to invoke Barbalat's Lemma [4,16] in connection with (3.21) to conclude that $\dot L$ goes to zero with t, and hence that $r(t)$ vanishes. ■

Note that (3.18), (3.19) is nothing but the continuous version of the backpropagation algorithm. In the scalar sigmoid case, for instance,

$$\sigma'(z) = \sigma(z)(1-\sigma(z)), \quad \hat\sigma'^T\hat W r = \sigma(\hat V^T x)[1-\sigma(\hat V^T x)]\,\hat W r,$$

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

Theorem 3.1 reveals the failure of standard backpropagation in the general case. In fact, in the 2-layer NN case $V = I$ (i.e. linear in the parameters), it is easy to show that, using update rule (3.18), the weights $\hat W$ are not generally bounded unless the hidden layer output $\sigma(x)$ obeys a stringent persistency of excitation (PE) condition. In the 3-layer (nonlinear) case, PE conditions are not easy to derive or guarantee as one is faced with the observability properties of a certain bilinear system. Thus, backpropagation used in a net that cannot exactly reconstruct $f(x)$, or on a robot arm with bounded unmodelled disturbances, or when $f(x)$ is nonlinear, cannot be guaranteed to yield bounded weights in the closed-loop system.

General Case

To attack the stability and tracking performance of a NN robot arm controller in the thorny general case we require the modification of the weight tuning rules as well as the addition of a robustifying term $v(t)$.

Theorem 3.2

Let the desired trajectory be bounded by (3.3). Take the control input for (2.7) as (3.11) with

$$v(t) = -K_Z(\|\hat Z\|_F + Z_M)\, r \tag{3.22}$$

and gain

$$K_z > C_2 \tag{3.23}$$

with C_2 the known constant in (3.17). Let weight tuning be provided by

$$\dot{\hat W} = F\hat\sigma r^T - F\hat\sigma'\hat V^T x r^T - \kappa F\|r\|\hat W \tag{3.24}$$

$$\dot{\hat V} = Gx(\hat\sigma'^T\hat W r)^T - \kappa G\|r\|\hat V \tag{3.25}$$

with any constant matrices $F = F^T > 0$, $G = G^T > 0$, and scalar design parameter $\kappa > 0$. Then the filtered tracking error $r(t)$ and NN weight estimates $\hat V$, $\hat W$ are GUUB.

Proof:

Selecting the Lyapunov function (3.20), differentiating, and substituting now from the error system (3.15) yields

$$\dot L = -r^TK_vr + \tfrac12 r^T(\dot M-2V_m)r + tr\,\tilde W^T(F^{-1}\dot{\tilde W} + \hat\sigma r^T - \hat\sigma'\hat V^T x r^T) + tr\,\tilde V^T(G^{-1}\dot{\tilde V} + xr^T\hat W^T\hat\sigma') + r^T(w+v).$$

The tuning rules give

$$\dot L = -r^TK_vr + \kappa\|r\|\, tr\,\tilde W^T(W-\tilde W) + \kappa\|r\|\, tr\,\tilde V^T(V-\tilde V) + r^T(w+v)$$
$$= -r^TK_vr + \kappa\|r\|\, tr\,\tilde Z^T(Z-\tilde Z) + r^T(w+v).$$

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

$$\dot L \le -K_{vmin}\|r\|^2 + \kappa\|r\|\,\|\tilde Z\|_F(Z_M - \|\tilde Z\|_F) - K_Z(\|\hat Z\|_F + Z_M)\|r\|^2 + \|r\|\,\|w\|$$
$$\le -K_{vmin}\|r\|^2 + \kappa\|r\|\,\|\tilde Z\|_F(Z_M - \|\tilde Z\|_F) - K_Z(\|\hat Z\|_F + Z_M)\|r\|^2 + \|r\|\,[C_0 + C_1\|\tilde Z\|_F + C_2\|\tilde Z\|_F\|r\|]$$
$$\le -\|r\|\,[K_{vmin}\|r\| + \kappa\|\tilde Z\|_F(\|\tilde Z\|_F - Z_M) - C_0 - C_1\|\tilde Z\|_F],$$

where K_{vmin} is the minimum singular value of K_v and the last inequality holds due to (3.23). Thus, $\dot L$ is negative as long as the term in braces is positive.

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

$$K_{vmin}\|r\| + \kappa\|\tilde Z\|_F(\|\tilde Z\|_F - C_3) - C_0$$
$$= \kappa(\|\tilde Z\|_F - C_3/2)^2 - \kappa C_3^2/4 + K_{vmin}\|r\| - C_0,$$

which is guaranteed positive as long as either

$$\|r\| > \frac{\kappa C_3^2/4 + C_0}{K_{vmin}} \tag{3.26}$$

or

$$\|\tilde{Z}\|_F > C_3/2 + \sqrt{C_3^2/4 + C_0/\kappa}\ . \tag{3.27}$$

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

The first terms of (3.24), (3.25) are nothing but the standard backpropagation algorithm. The last terms correspond to the e-modification [7] in standard use in adaptive control to guarantee bounded parameter estimates. The second term in (3.24) is very interesting and bears discussion. The standard backprop terms can be thought of as backward propagating signals in a nonlinear 'backprop' network [8] that contains multipliers. The second term in (3.24) corresponds to a forward travelling wave in the backprop net that provides a second-order correction to the weight tuning for W.

Note from (3.26), that arbitrarily small tracking error bounds may be achieved by selecting large control gains K_v. On the other hand, (3.27) reveals that the NN weight errors are fundamentally bounded by Z_M (through C_3). The parameter κ offers a design tradeoff between the relative eventual magnitudes of $\|r\|$ and $\|\tilde{Z}\|_F$.

Note that there is design freedom in the degree of complexity (e.g. size) of the NN. For a more complex NN (e.g. more hidden units), the bounding constants will decrease, resulting in smaller tracking errors. On the other hand, a simplified NN with fewer hidden units will result in larger error bounds; this degradation can be compensated for, as long as bound ϵ_N is known, by selecting a larger value for K_z in the robustifying signal v(t), or for Λ in (2.9).

4. PASSIVITY PROPERTIES OF THE NN

The closed-loop error system appears in Fig. 4.1, with the signal ζ_2 defined as

$$\begin{aligned} \zeta_2(t) &= -\tilde{W}^T\hat{\sigma} &&, \text{ for error system (3.13)} \\ \zeta_2(t) &= -\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x), && \text{ for error system (3.15).} \end{aligned} \tag{4.1}$$

(In the former case, signal w(t) should be replaced by $w_1(t)$.) The NN appears here in standard feedback configuration.

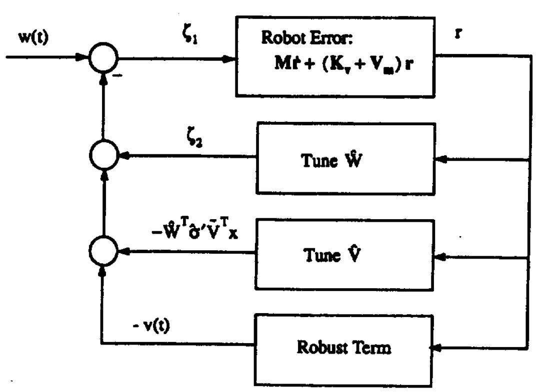

Figure 4.1 Neural Net Closed-loop Error System

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.

The first result is with regard to error system (3.13).

Theorem 4.1

The backprop weight tuning algorithms (3.18), (3.19) make the map from r(t) to $-\tilde{W}^T\hat{\sigma}$, and the map from r(t) to $-\hat{W}^T\hat{\sigma}'\tilde{V}^T x$, both passive maps.

Proof:

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

$$\dot{\tilde{W}} = -F\hat{\sigma}r^T \tag{4.2}$$

$$\dot{\tilde{V}} = -Gx(\hat{\sigma}'^T\hat{W}r)^T \tag{4.3}$$

1. Selecting the nonnegative function

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

and evaluating $\dot{L}$ along the trajectories of (4.2) yields

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

which is in power form (2.4).

2. Selecting the nonnegative function

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

and evaluating $\dot{L}$ along the trajectories of (4.3) yields

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

which is in power form. ∎

Thus, the robot error system in Fig. 4.1 is state strict passive (SSP) and the weight error blocks are passive; this guarantees the dissipativity of the closed-loop system [16]. Using the passivity theorem one may now conclude that the input/output signals of each block are bounded as long as the external inputs are bounded.

Unfortunately, though dissipative, the closed-loop system is not SSP so, when disturbance $w_1(t)$ is nonzero, this does not yield boundedness of the internal states of the weight blocks (i.e. $\tilde{W}$, $\tilde{V}$) unless those blocks are observable, that is persistently exciting (PE). PE is very difficult to check or guarantee for a NN.

The next result shows why a PE condition is not needed with the modified weight update algorithm of Theorem 3.2; it is in the context of error system (3.15).

Theorem 4.2

The modified weight tuning algorithms (3.24), (3.25) make the map from r(t) to $-\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)$, and the map from r(t) to $-\hat{W}^T\hat{\sigma}'\tilde{V}^T x$, both state strict passive (SSP) maps.

Proof:

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

$$\dot{\tilde{W}} = -F\hat{\sigma}r^T + F\hat{\sigma}'\hat{V}^T x r^T + \kappa F\|r\|\hat{W} \tag{4.4}$$

$$\dot{\tilde{V}} = -Gx(\hat{\sigma}'^T\hat{W}r)^T + \kappa G\|r\|\hat{V}. \tag{4.5}$$

1. Selecting the nonnegative function

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

and evaluating $\dot{L}$ yields

$$\begin{aligned} \dot{L} = \mathrm{tr}\ \tilde{W}^T F^{-1}\dot{\tilde{W}} &= r^T[-\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)] - \kappa\|r\|(\|\tilde{W}\|_F^2 - \langle\tilde{W},W\rangle_F) \\ &\le r^T[-\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)] - \kappa\|r\|(\|\tilde{W}\|_F^2 - W_M\|\tilde{W}\|_F) \end{aligned}$$

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

2. Selecting the nonnegative function

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

and evaluating $\dot{L}$ yields

$$\dot{L} = \mathrm{tr}\, \tilde{V}^T F^{-1} \dot{\tilde{V}} = r^T(-\hat{W}^T \hat{\sigma}' \tilde{V}^T x) - \kappa \|r\| (\|\tilde{V}\|_F^2 - \langle \tilde{V}, V \rangle_F)$$

$$\leq r^T(-\hat{W}^T \hat{\sigma}' \tilde{V}^T x) - \kappa \|r\| (\|\tilde{V}\|_F^2 - V_M \|\tilde{V}\|_F)$$

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

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

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

5. 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 [4]; no friction term was used. The joint variable is $q = [q_1 \;\; q_2]^T$. We should like to illustrate the NN control scheme derived herein, which will require no knowledge of the dynamics, not even their structure which is needed for adaptive control.

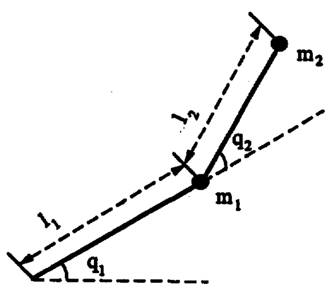

Fig 6.1 2-Link Planar Elbow Arm

Adaptive Controller: Baseline Design

For comparison, a standard adaptive controller is given by [15]

$$\tau = Y\hat{\psi} + K_v r \tag{5.1}$$

$$\dot{\hat{\psi}} = FY^T r \tag{5.2}$$

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 dynamics for each arm, and ψ the vector of unknown parameters, in this case simply the link masses m_1, m_2.

We took the arm parameters as $\ell_1 = \ell_2 = 1$ m, $m_1 = 0.8$ kg, $m_2 = 2.3$ kg, and selected $q_{1d}(t) = \sin t$, $q_{2d}(t) = \cos t$, $K_v = \mathrm{diag}(20,20)$, $F = \mathrm{diag}(10,10)$, $\Lambda = \mathrm{diag}(5,5)$. 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.

The (1,1) entry of the robot function matrix Y is $\ell_1^2(\ddot{q}_{d1} + \lambda_1 \dot{e}_1) + \ell_1 g \cos q_1$ (with $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2)$). To demonstrate the deleterious effects of unmodeled dynamics in adaptive control, the term $\ell_1 g \cos q_1$ was now dropped in the controller. The result appears in Fig. 6.3 and is very bad. It is emphasized that in the NN controller all the dynamics are unmodeled.

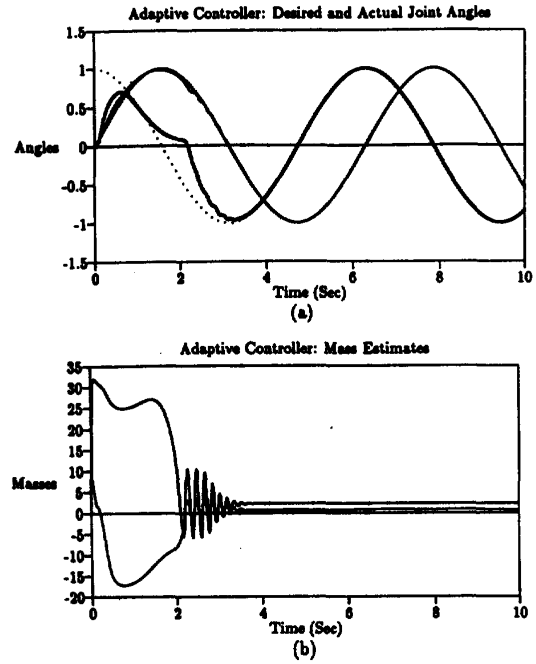

Figure 6.2 Response of Adaptive Controller. (a) Actual and Desired Joint Angles. (b) Parameter Estimates.

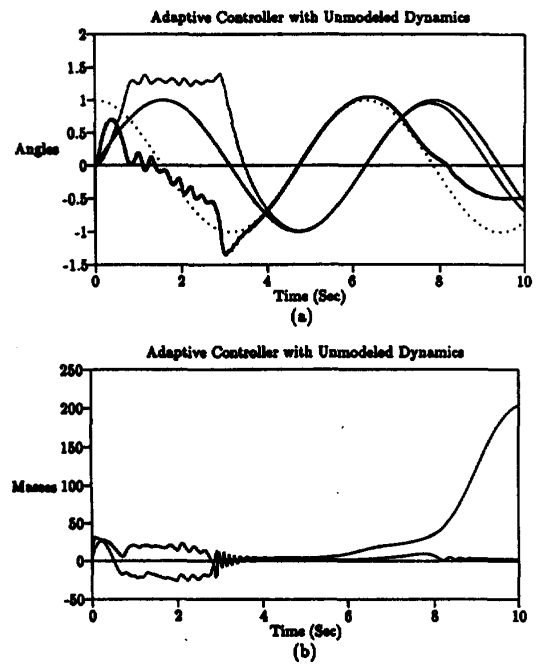

Figure 6.3 Response of Adaptive Controller with Unmodeled Dynamics. (a) Actual and Desired Joint Angles. (b) Parameter Estimates.

NN Controller With Backprop Weight Tuning

The NN controller appears in Fig. 3.1, with the NN input x(t) given by (2.12) and $\zeta_1 = \ddot{q}_d + \Lambda e$, $\zeta_2 = \dot{q}_d + \Lambda e$. We selected 10 hidden-layer neurons. The sigmoid activation functions were used.

The response of the controller (3.11) (with $v(t) = 0$) with backprop weight tuning (e.g. Theorem 3.2) appears in Fig. 6.4. In this case the NN weights appear to remain bounded, though this cannot in general be guaranteed.

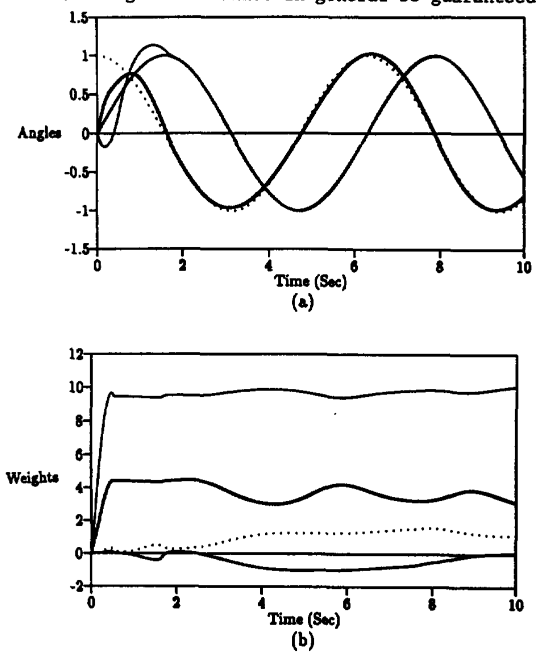

Figure 6.4 Response of NN Controller with BP Weight Tuning. (a) Actual and Desired Joint Angles. (b) Representative Weight Estimates.

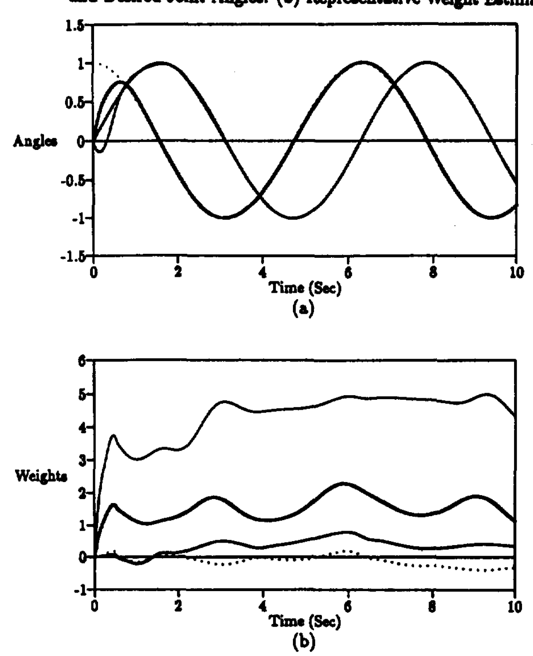

Figure 6.5 Response of Improved NN Controller. (a) Actual and Desired Joint Angles. (b) Representative Weight Estimates.

NN Controller With Improved Weight Tuning

The response of the controller (3.11) with the improved weight tuning in Theorem 3.2 appears in Fig. 6.5, where we took $\kappa = 0.1$. The tracking response is better than that using straight backprop, and the weights are guaranteed to remain bounded even though PE may not hold. The comparison with the performance of the standard adaptive controller in Fig. 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.

REFERENCES

[1] R.G. Bartle, The Elements of Real Analysis, New York: Wiley, 1964.
[2] J.J. Craig, Adaptive Control of Robot Manipulators, Reading, VA: Addison-Wesley, 1988.
[3] 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.
[4] F.L. Lewis, C.T. Abdallah, and D.M. Dawson, Control of Robot Manipulators, New York: Macmillan, 1993.
[5] W.T. Miller, R.S. Sutton, P.J. Werbos, ed., Neural Networks for Control, Cambridge: MIT Press, 1991.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] J. Park and I.W. Sandberg, "Universal approximation using radial-basis-function networks," Neural Comp., vol. 3, pp. 246-257, 1991.
[11] 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.
[12] M.M. Polycarpou and P.A. Ioannu, "Neural networks as on-line approximators of nonlinear systems," Proc. IEEE Conf. Decision and Control, pp. 7-12, Tucson, Dec. 1992.
[13] N. Sadegh, "Nonlinear identification and control via neural networks," Control Systems with Inexact Dynamics Models, DSC-vol. 33, ASME Winter Annual Meeting, 1991.
[14] 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.
[15] 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.
[16] J.-J.E. Slotine and W. Li, Applied Nonlinear Control, New Jersey: Prentice-Hall, 1991.
[17] 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.