ROBUST ADAPTIVE CONTROL OF ROBOTS USING NEURAL NETWORK: GLOBAL STABILITY

C. Kwan, D.M. Dawson and F.L. Lewis

ABSTRACT

A desired compensation adaptive law-based neural network (DCAL-NN) controller is proposed for the robust position control of rigid-link robots. The NN is used to approximate a highly nonlinear function. The controller can guarantee the *global* asymptotic stability of tracking errors and boundedness of NN weights. In addition, the NN weights here are tuned on-line, with *no off-line learning phase required*. When compared with standard adaptive robot controllers, we do not require linearity in the parameters, or lengthy and tedious preliminary analysis to determine a regression matrix. The controller can be regarded as *a universal reusable controller* because the same controller can be applied to any type of rigid robots without any modifications. A comparative simulation study with different robust and adaptive controllers is included.

KeyWords: Robust adaptive control, neural networks, global stability, robot control.

I. INTRODUCTION

In the past decade, there has been much research on the applications of modern nonlinear control theory to the motion control of robots. Many robust and adaptive control schemes were developed with the aim of explicitly counteracting parametric uncertainties in the robot system and hence improving the accuracy of motion tracking. Most of these controllers can be found in a survey paper by Abdallah *et al*. [1].

Recently, many researchers have begun to focus on implementation issues such as the reduction of on-line computations [12,21]. For example, it is well known that one of the disadvantages of the adaptive robot controllers is that the regression matrix used as feedforward compensation must be determined by extensive preliminary analysis, and evaluated on-line to compute certain nonlinear functions. Moreover, some terms in the robot dynamics are not necessarily linear in the parameters (e.g. friction is actually described by complex nonlinear functions). Such concerns caused people to rethink the previous theoretical development of robot controllers.

To eliminate the need for on-line evaluation of the regression matrix, Sadegh and Horowitz [21] proposed the desired compensation adaptive law (DCAL). The idea is to make use of the desired joint trajectory and replace the actual joint angles, velocities with the desired joint angles, velocities in the evaluation of the regression matrix. Although this scheme is quite simple, some tedious and lengthy preliminary dynamical analysis is still needed to get the regression matrix. This preliminary effort can be quite cumbersome for the case of robots with multiple degrees of freedom. Moreover, this effort will start all over again for a new type of robot.

Neural network (NN) can be used for approximation of nonlinear systems, for classification of signals, and for associative memories. For control engineers, the approximation capability of NN is usually used for system identification [3,9], or indirect "identification-based" [2,14-16]. However there is very little about the use of NN in direct closed-loop controllers that yield guaranteed performance.

Problems that remain to be addressed in NN research include ad hoc controller structures and the inability to guarantee satisfactory performance of the system in terms of small tracking errors and bounded NN weights. Uncertainty on how to initialize the NN weights leads to the necessity for "preliminary off-line tuning" [5]. Some of these problems have by now been addressed [13,17-20].

In this paper, we propose a new robust robot control

Manuscript received September 14, 2000; revised February 5, 2001; accepted March 29, 2001.

C. Kwan is with Intelligent Automation Inc., 7519 Standish Place, Suite 200, Rockville, MD 20855, U.S.A.

D.M. Dawson is with Department of Electrical and Computer Engineering, Center for Advanced Manufacturing, Clemson University, Clemson, SC 29634, U.S.A.

F.L. Lewis is with Automation and Robotics Research Institute, The University of Texas at Arlington, 7300 Jack Newell Blvd. S, Fort Worth, Texas 76118, U.S.A.

scheme by combining the theory of Neural Network (NN) with DCAL. The main advantage is that the NN controller is the same irrespective of the kind of rigid-link robot under control. Hence it can be termed *a universal reusable controller* since no preliminary dynamical analysis, which can be quite cumbersome for robots with multiple degrees of freedom, is needed. Our NN weights here are tuned on-line, with *no off-line learning phase required*. Most importantly, we can guarantee the *global* asymptotic stability of joint position tracking error and the boundedness of NN weight updates. When compared with standard adaptive robot controllers, we do not require linearity in the parameters, and the evaluation of regression matrix.

The paper is organized as follows. In Section 2 we will review some basics of neural networks, the rigid robot model and its properties. Then in Section 3, we will introduce our DCAL-NN controller. A Lyapunov approach will be used to show that joint position error is *globally* asymptotically stable and the NN weights are all bounded. A detailed 2-link robot motion-tracking example will be presented in Section 4 to illustrate the performance of different controllers. Finally, conclusions will be given in Section 5.

II. MATHEMATICAL PRELIMINARIES

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

$$\bar{f} = \sup\|f(x)\|,\ f: S \to R^m,\ \forall x \in S. \tag{1}$$

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

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

with $tr(\bullet)$ the trace. The associated inner product is $\langle A, B\rangle_F = tr(A^T, B)$.

2.1 Neural networks

A three-layer neural network is shown in Fig. 1. Given an input vector x in R^{N_1}, a three-layer NN has an output given by

$$y_i = \sum_{j=1}^{N_2}\left[w_{ij}\sigma\left[\sum_{k=1}^{N_1} v_{jk}x_k + \theta_{vj}\right] + \theta_{wi}\right],\ i = 1, \ldots, N_3 \tag{3}$$

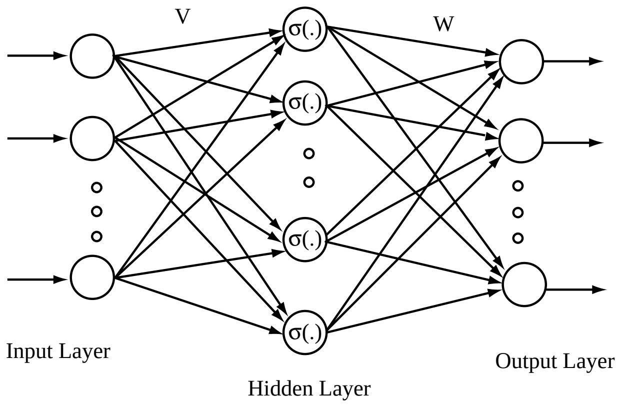

Fig. 1. Three-layer neural network.

with $\sigma(\bullet)$ the activation function such as a sigmoid function, v_{jk} the first-to-second layer interconnection weights, and w_{ij} the second-to-third layer interconnection weights. θ_{vm}, θ_{wm}, $m = 1, 2, \ldots$, are called the threshold offsets and the number of neurons in layer l is N_l, with N_2 the number of hidden-layer neurons.

The NN equation can be conveniently expressed in matrix format by defining $x = [x_0\ \ x_1\ \ \ldots\ \ x_{N_1}]^T$, $y = [y_1\ \ \ldots\ \ y_{N_3}]^T$, and weight matrices $W^T = [w_{ij}]$, $V^T = [v_{jk}]$. Including $x_0 = 1$ in x allows one to include the threshold vector $[\theta_{v1}\ \ \theta_{v2}\ \ \ldots\ \ \theta_{vN_1}]^T$ as the first column of V^T. Hence, the NN outputs can be compactly written as

$$y = W^T\sigma(V^Tx) \tag{4}$$

where, if $z = [z_1\ \ z_2\ \ \ldots]^T$ is a vector we define $\sigma(z) = [\sigma(z_1)\ \ \sigma(z_2)\ \ \ldots]^T$. Including 1 as a first term in the vector $\sigma(V^Tx)$ 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.

A general nonlinear function $f(x)$, $x(t) \in S$, can be written as

$$f(x) = W^T\sigma(V^Tx) = \varepsilon(x) \tag{5}$$

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 constant real number $\varepsilon_N > 0$, we say $f(x)$ is within ε_N of the NN range if there exist N_2 constant weights so that for all $x \in S$, (5) holds with $\|\varepsilon\| < \varepsilon_N$.

For the case of the "squashing functions" (a bounded, measurable, nondecreasing function), it has been shown [6,9,13] that there exist finite N_2, and constant weights W and V such that (5) holds with $\|\varepsilon(x)\| < \varepsilon_N$, $N_1 = n$, and $N_3 = m$. Typical "squashing functions" include, with $z \in R$, $\alpha > 0$,

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

$$\sigma(z) = \frac{1-e^{-\alpha z}}{1+e^{-\alpha z}}, \text{ hyperbolic tangent function.} \tag{6}$$

2.2 Robot model and its properties

We will assume that the robot is an n-link, serially connected, rigid-link revolute robot with the following dynamic model [12]

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

with $q, \dot{q}, \ddot{q} \in R^n$ denoting the joint position (angle), velocity, and acceleration vectors, respectively, $M(q) \in R^{n\times n}$ the inertia matrix, $V_m(q, \dot{q}) \in R^{n\times n}$ the centripetal-Coriolis matrix, $G(q) \in R^n$ the gravity vector, $F(\dot{q}) \in R^n$ representing the friction terms, $T_d \in R^n$ the additive bounded disturbance, $\tau \in R^n$ the torque input vector. The rigid dynamics (7) has the following properties [4,10]. Note that linearity in the parameters assumption is not needed here.

Property 1. Boundedness of the inertia matrix
The inertia matrix $M(q)$ is symmetric and positive definite, and satisfies the following inequalities

$$\lambda_1 \| y \|^2 \le y^T M(q) y \le \lambda_2 \| y \|^2, \ \forall y \in R^n \tag{8}$$

where λ_1 and λ_2 are known positive constants, and $\| \bullet \|$ denotes the standard Euclidean norm.

Property 2. Skew symmetry
The inertia and centripetal-Coriolis matrices have the following property

$$y^T \left[\frac{1}{2}\dot{M}(q) - V_m(q,\dot{q}) \right] y = 0, \quad \forall y \in R^n, \ \forall q \in R^n, \ \forall \dot{q} \in R^n \tag{9}$$

where $\dot{M}(q)$ is the time derivative of the inertia matrix.

Property 3. Neural Network Approximator
We use a three layer NN to approximate a highly nonlinear robot function, i.e

$$f_d(q_d, \dot{q}_d, \ddot{q}_d) = W^T \sigma(V^T x) + \varepsilon(x), \tag{10}$$

where

$$f_d(q_d, \dot{q}_d, \ddot{q}_d) = M(q_d)\ddot{q}_d + V_m(q_d, \dot{q}_d)\dot{q}_d + G(q_d) + F(\dot{q}_d), \tag{11}$$

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

and W, V are constants. Throughout this paper, we will assume the desired trajectory, denoted by $q_d, \dot{q}_d, \ddot{q}_d \in S$, is bounded which implies that we are guaranteed that

$$\| \varepsilon(x) \| \le \varepsilon_b, \tag{13}$$

where ε_b is a positive constant.

Property 1 is very important in generating a positive definite function to prove stability of the closed-loop system. Property 2 will help in simplifying the controller. Many robust methods have incorporated Properties 1 and 2 in their controller design [4,10]. Property 3 is the key for our DCAL-NN controller design. It is a well-established result and follows directly from the universal approximation property of NN [6,9,13].

III. DCAL-NN CONTROLLER

The control objective is to develop a link position tracking controller for the robot dynamics given by (7) based on inexact knowledge of manipulator dynamics. To accomplish this purpose we first define the joint angle tracking error as

$$e = q_d - q \tag{14}$$

where $q_d(t)$ denotes the desired joint angle trajectory. In addition, we also define a filtered tracking error as

$$r = \dot{e} + \Lambda e \tag{15}$$

where $\Lambda \in R^{n\times n}$ is a diagonal, positive definite control gain matrix.. Using (15) and (7), we can derive the error equation

$$M\dot{r} = f(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d) - V_m r + T_d - \tau, \tag{16}$$

where the highly nonlinear robot function $f(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d)$ is defined by

$$f(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d) = M(q)(\ddot{q}_d + \Lambda\dot{e}) + V_m(q,\dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(\dot{q}). \tag{17}$$

3.1 Bounding assumptions and lemmas

For notational convenience we define the matrix of all weights as

$$Z = diag[W, V]. \tag{18}$$

Also define

$$\tilde{V} = V - \hat{V}, \tilde{W} = W - \hat{W}, \text{ and } \tilde{Z} = Z - \hat{Z} \tag{19}$$

with $\hat{V}$, $\hat{W}$ the estimates of the ideal weight values as provided by the weight tuning algorithms (to be discussed).

Two standard assumptions, which are quite common in the robotics literature [12] and neural network [8,11,13] are stated next. Two useful facts (Lemma 1 and Lemma 2) will be listed, which can be proven using these assumptions.

Assumption 1. The ideal weights are bounded by positive constants V_M, W_M, Z_M so that

$$\| V \|_F \le V_M, \| W \|_F \le W_M, \text{ or } \| Z \|_F \le Z_M.$$

Assumption 2. The desired trajectory is bounded, i.e.

$$\| x \| \le Q_d$$

where x is defined in (12) and Q_d is positive scalar constant.

Assumption 1 follows from Property 3 in Section 2 which states that W, V are constants and hence are bounded. The hidden-layer output error for a given x is given by

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T x) - \sigma(\hat{V}^T x). \tag{20}$$

The Taylor series expansion of σ in (20) for a given x may be written as

$$\sigma(V^T x) = \sigma(\hat{V}^T x) + \sigma'(\hat{V}^T x)\tilde{V}^T x + O(\tilde{V}^T x)^2 \tag{21}$$

with $\sigma' = \left. \frac{d\sigma(z)}{dz} \right|_{z=\hat{z}}$ and $O(z)^2$ denoting higher order terms.

Denoting $\hat{\sigma}' = \sigma'(\hat{V}^T x)$, we have

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

Noting that

$$O(\tilde{V}^T x)^2 = [\sigma(V^T x) - \sigma(\hat{V}^T x)] - \hat{\sigma}'\tilde{V}^T x, \tag{22}$$

we can then show the following Lemma for the higher-order term $Q(\bullet)^2$ in (22).

Lemma 1. For sigmoid and tanh activation functions, the higher-order term in the Taylor series (Eq. (22)) are bounded by

$$\left\| O(\tilde{V}^T x)^2 \right\| \le c_1 + c_2 \| \tilde{V} \|_F \tag{23}$$

where c_i's are computable positive constants. The proof is given in an Appendix.

The extension of these ideas to NNs with more than three layers is not difficult, and leads to composite function term in the Taylor series. Here we only consider one nonlinear hidden layer. The justification is that Cybenko [6] and Hornik *et al*. [9] have shown that a NN with one nonlinear hidden-layer can approximate any smooth nonlinear functions over a compact set. Thus it is not necessary to consider NNs with more nonlinear hidden layers which may be complicated to analyze and offer no significant advantage over the one nonlinear hidden layer case.

3.2 Controller structure and error dynamics

Let a NN be given such that (10) holds with $\| \varepsilon(x) \| \le \varepsilon_N$ where ε_N is a positive constant. According to the well-known NN approximation property [6,9], since x is bounded, (10) always holds globally in a sense that the approximation is state independent. We select the control input for (16) as

$$\tau = \hat{W}^T \sigma(\hat{V}^T x) + k_v r + k_p e + v_R \tag{24}$$

with $\hat{W}$, $\hat{V}$ the dynamic estimate of W, V. Vectors e and r are defined by (10), (14), (15), respectively. $k_v \in R^{n \times n}$, $k_p \in R^{n \times n}$, are diagonal, positive definite gain matrices. The nonlinear term $v_R(t)$ to be defined later is used to compensate for error functions that provides robustness in the face of higher order term in the Taylor series (Eq. (22)), functional reconstruction error $\varepsilon(x)$, and bounded disturbance T_d. The controller structure is shown in Fig. 2.

Substituting controller (24) into (16) gives the closed-loop dynamics

$$M\dot{r} = f(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d) - (k_v + V_m)r - \hat{W}^T \sigma(\hat{V}^T x)$$
$$- k_p e - v_R + T_d.$$

Adding and subtracting $f_d(q_d, \dot{q}_d, \ddot{q}_d)$ defined in (11) into the above expression yields

$$M\dot{r} = f(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d) - f_d(q_d, \dot{q}_d, \ddot{q}_d) + f_d(q_d, \dot{q}_d, \ddot{q}_d)$$
$$- (k_v + V_m)r - \hat{W}^T \sigma(\hat{V}^T x) - k_p e - v_R + T_d.$$

Now using the NN approximation property (10) gives

$$M\dot{r} = f(\bullet) - f_d(\bullet) + W^T \sigma(V^T x) + \varepsilon(x) - (k_v + V_m)r$$

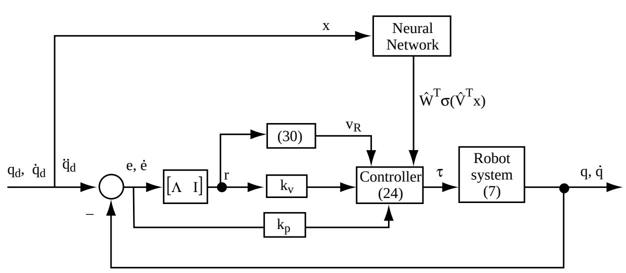

Fig. 2. DCAL-NN controller structure.

$$-\hat{W}^T\sigma(\hat{V}^Tx)-k_pe-v_R+T_d. \quad (25)$$

Expanding $\sigma(v^Tx)$ in (25) into Taylor series and manipulating the resulting expression yields

$$M\dot{r}=-(k_v+V_m)r-k_pe+\tilde{W}^T\hat{\sigma}+\hat{W}^T\hat{\sigma}'\tilde{V}^Tx+w-v_R, \quad (26)$$

where

$$w=f(\bullet)-f_d(\bullet)+\tilde{W}^T\hat{\sigma}'\tilde{V}^Tx+W^TO(\tilde{V}^Tx)^2+T_d+\varepsilon(x). \quad (27)$$

Lemma 2. The disturbance term w in (27) is bounded according to

$$\|w\|\le\rho(z,\hat{W},\hat{V}), \quad (28)$$

where the positive scalar function $p(\bullet)$ is defined by

$$\rho(z,\hat{W},\hat{V})=\zeta_0+\zeta_1\|z\|+\zeta_2\|z\|^2+\zeta_3\|\hat{W}\|_F+\zeta_4\|\hat{V}\|_F$$
$$+\zeta_5\|\hat{V}\|_F\|\hat{W}\|_F=S\varphi. \quad (29)$$

The ζ_i's are positive bounding constants that depend on the desired trajectory, the physical properties of the robot (i.e., link masses, link lengths, friction coefficients, etc.), the disturbance bound, the error bound, and on the control gain matrix Λ of (15). The vectors z, S, φ used in (28) and (29) are defined as follows

$$z=[e^T \;\; r^T]^T,$$

$$S=\left[1 \;\; \|z\| \;\; \|z\|^2 \;\; \|\hat{W}\|_F \;\; \|\hat{V}\|_F \;\; \|\hat{W}\|_F\|\hat{V}\|_F\right],$$

$$\varphi=[\zeta_0 \;\; \zeta_1 \;\; \zeta_2 \;\; \zeta_3 \;\; \zeta_4 \;\; \zeta_5]^T.$$

This Lemma is proven in an Appendix by using results in [12] and noting the fact that w in (28) depends on $\hat{W}$, $\hat{V}$, and their product.

The robustifying term $v_R(t)$ in the controller (24) is given by

$$v_R=\frac{r(S\hat{\varphi})^2}{(S\hat{\varphi})\|r\|+\delta}, \quad (30)$$

where

$\dot{\delta}=-\gamma\delta$, $\delta(0)$ = design constant > 0, $\gamma > 0$.

The parameter tuning law for φ is chosen as

$$\dot{\hat{\varphi}}=\Gamma S^T\|r\|=\dot{\tilde{\varphi}}, \quad (31)$$

where Γ is a symmetric and positive definite matrix and $\tilde{\varphi} = \varphi - \hat{\varphi}$. The overall stability analysis will be given next.

3.3 Weight updates for NN theorem

Let the desired trajectory be bounded (Assumption 2). Let the control input be given by (24), (30), (31) and NN weight tuning provided by

$$\dot{\hat{W}}=F\hat{\sigma}r^T, \quad (32a)$$

$$\dot{\hat{V}}=Gx(\hat{\sigma}'^T\hat{W}r)^T \quad (32b)$$

with F, G some constant symmetric and positive definite matrices. Then the tracking errors e, $\dot{e}$ go to zero and the weight estimates $\hat{W}$, $\hat{V}$ are bounded.

Proof. Consider the following Lyapunov function candidate

$$L=\frac{1}{2}r^TMr+\frac{1}{2}e^Tk_pe+\frac{1}{2}tr(\tilde{W}^TF^{-1}\tilde{W})$$
$$+\frac{1}{2}tr(\tilde{V}^TG^{-1}\tilde{V})+\frac{1}{2}L_R, \quad (33)$$

where

$$L_R=\frac{1}{2}\tilde{\varphi}^T\Gamma^{-1}\tilde{\varphi}+\frac{1}{2}\frac{\delta}{\gamma}.$$

Differentiating (33) and using (26) yields

$$\dot{L}\le-z^TQz+\dot{L}_1+\dot{L}_2,$$

where

$$z=[e^T \;\; r^T]^T,$$

$$Q=\begin{bmatrix} k_p\Lambda & -\frac{k_p}{2} \\ -\frac{k_p}{2} & k_v \end{bmatrix} > 0 \text{ for suitable choices of } k_p, k_v,$$

$$\dot{L}_1=\|r\|S\tilde{\varphi}-r^Tv_R-\delta, \quad (34a)$$

$$\dot{L}_2=tr[\tilde{W}^TF^{-1}\dot{\tilde{W}}+W^T\hat{\sigma}r^T]+tr[\tilde{V}^TG^{-1}\dot{\tilde{V}}+\tilde{V}^Tx(\hat{\sigma}'^T\hat{W}r)^T]. \quad (34b)$$

Using (30) in (34a) yields

$$\dot{L}_1=\|r\|S\hat{\varphi}-\frac{r^Tr(S\hat{\varphi})^2}{(S\hat{\varphi})\|r\|+\delta}-\delta\le\frac{\delta\|r\|S\hat{\varphi}}{(S\hat{\varphi})\|r\|+\delta}-\delta\le 0.$$

Substituting the weight update rules (32) into (34b) gives

$$\dot{L}_2 = 0. \tag{35}$$

Combining the above results, we have

$$\dot{L} \le -z^T QZ,\ Q > 0 \tag{36}$$

(36) implies L, z, e, r, $\hat{W}$, $\hat{V}$ and $\hat{\varphi}$ are all bounded. From the form of (36), it is easy to show that $z(t)$ is square integrable; hence e and r are square integrable. From the closed-loop error system (16), it is also easy to show that $\dot{r}$ is bounded; therefore, r is uniformly continuous. From Barbalat's Lemma [12], we know that e and $\dot{e}$ both go to zero asymptotically. ■

Remarks.

1. It should be emphasized that the controller (24) can be applied to any revolute rigid-ink robots without any prior knowledge about the system such as masses and lengths of every link. This is in sharp contrast to adaptive control approach which requires very tedious preliminary dynamical analysis of a given robot system to get the regression matrix. If a new robot comes in, the whole procedure has to start all over again. This can be quite a formidable task for robots with multiple degrees of freedom.
2. Note that the problem of *net weight initialization* does not arise. In fact, selecting the initial weights $\hat{W}(0)$, $\hat{V}(0)$ as zero leaves only the outer tracking loop in Fig. 2 at the beginning of tracking. The PD terms $k_v r$ and $k_p e$ in (24) can stabilize the robot arm on an interim basis until the NN begins to learn. This also means that there is *no off-line learning phase* for this NN controller. A more practical approach will be to run a few simulations and store the steady-state NN weights. Then the initial NN weights can be set to values near those steady-state weights. This will avoid some transient problems.
3. There is, however, no assurance for $\hat{W}$, $\hat{V}$ to converge to the true W, V. This phenomenon is reminiscent of similar things in adaptive control where no parameter convergence is assured unless certain signals in the system are persistently exciting. This does not concern us as the objective of tracking with bounded controls has been achieved.
4. Since the parameter δ in (30) decays to zero exponentially, this may cause a chattering phenomenon when δ gets very small. In practice, it is suggested that δ should be reset periodically to avoid this problem.
5. A comparison with standard DCAL controller will be carried out in Section 4. The advantage of DCAL-NN is that it is robust to unmodeled dynamics whereas the performance of DCAL degrades if certain unknown parameters were not accounted for in the design process. In terms of computational burden, the two approaches are comparable since both require on-line computations to compensate for the effects of nonlinearity.
6. A comparison with robust control will also be carried out in Section 4. Although the robust controller is computationally efficient, the price is the achievable performance. As will be seen shortly, the bounds in the robust controller are usually conservative that may cause high frequency chattering. It is well known that chattering may cause wear in system component and excite unmodeled dynamics.

IV. EXAMPLE

Consider a simple 2-link manipulator shown in Fig. 3. The model for this robot system can be described in the form of (1) with

$$M(q) = \begin{bmatrix} a + b\cos q_2 & c + \frac{b}{2}\cos q_2 \\ c + \frac{b}{2}\cos q_2 & c \end{bmatrix},$$

$$V_m(q, \dot{q}) = \begin{bmatrix} -b\dot{q}_2 \sin q_2 & -\frac{b}{2}\dot{q}_2 \sin q_2 \\ \frac{b}{2}\dot{q}_1 \sin q_2 & 0 \end{bmatrix},$$

$$G(q) = \begin{bmatrix} d\cos q_1 + e\cos(q_1 + q_2) \\ e\cos(q_1 + q_2) \end{bmatrix},$$

$$a = l_2^2 m_2 + l_1^2(m_1 + m_2),\ b = 2l_1 l_2 m_2,$$

$$c = l_2^2 m_2,\ d = (m_1 + m_2)l_1 g_0,\ e = m_2 l_2 g_0. \tag{37}$$

Parameters a, b, c, d, and e are unknown constants. The parameter values, which correspond to the first two links of a PUMA-560 robot [7], are $l_1 = 0.432\ m$, $l_2 = 0.432\ m$, $m_1 = 15.61\ kg$, $m_2 = 11.36\ kg$, $g_0 = 10\ m/s^2$. The desired joint

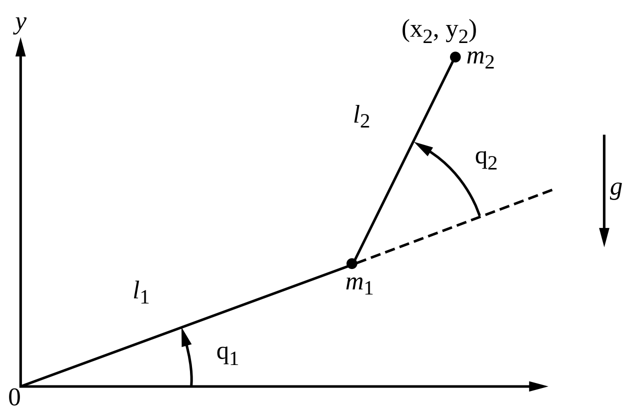

Fig. 3. A two-link manipulator.

trajectories are defined as $g_{d1}(t) = \sin(t)$, $g_{d2}(t) = \cos(t)$. In all the subsequent simulations, the initial joint positions for joint 1 and 2 start from 0 and 1 radians, respectively.

We divide our studies into four parts:

(i) PD controller

The controller is given by

$$\tau = k_v r + k_p e$$

with $k_v = \text{diag}\{10, 10\}$, $k_p = \text{diag}\{40, 40\}$. The value of Λ in (15) is given by diag{10, 10}. The simulation results are shown in Fig. 4. The results are not very encouraging since there exist large steady-state errors in both links. Of course one can increase the gains to reduce the errors. However high-gain feedback is undesirable as it may excite some high frequency unmodeled dynamics.

(ii) Robust controller

The robust controller that we used here is similar to the one described in [3,12]. The controller is of the form

$$\tau = k_v r + k_p e + v_R$$

with $v_R = \dfrac{r(S\bar{\varphi})^2}{(S\bar{\varphi})\|r\| + \delta}$, $S = \left[1 \;\; \|z\| \;\; \|z\|^2\right]$, $z = [e^T \;\; r^T]^T$, $\bar{\varphi} = [\bar{\zeta}_0 \;\; \bar{\zeta}_1 \;\; \bar{\zeta}_2]^T$, $\dot{\delta} = -\gamma\delta$, $\delta(0)$ = design constant > 0, $\gamma > 0$. A preliminary analysis of the nonlinear system (37) gave the following bounds for $\bar{\varphi} = [\bar{\zeta}_0 \;\; \bar{\zeta}_1 \;\; \bar{\zeta}_2]^T = [343 \;\; 220 \;\; 120]^T$. Note that these bounds are very conservative which is a typical situation in robust controllers since large bounds are necessary to suppress the effect of unknown system parameters. The initial condition $\delta(0)$ is chosen to be 30 and $\gamma = 0.1$. Simulation result is shown in Fig. 5. It can be seen that good performance comes at a cost of high-frequency chattering effect. If the bounds are chosen less conservatively, then there exist steady-state errors in the system as shown in Fig. 6. These two simulations clearly demonstrate that there is a trade-off between performance and robustness in robust control

(iii) Standard DCAL controller

Following the design steps of DCAL [14], we arrive

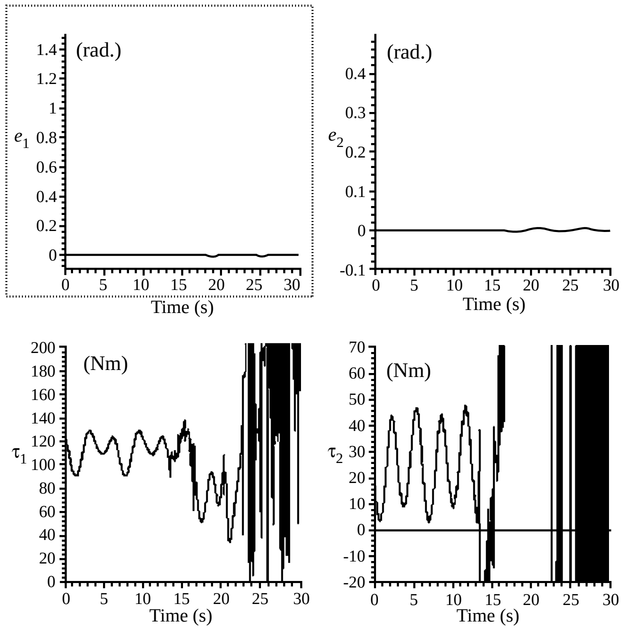

Fig. 5. Performance of robust controller.

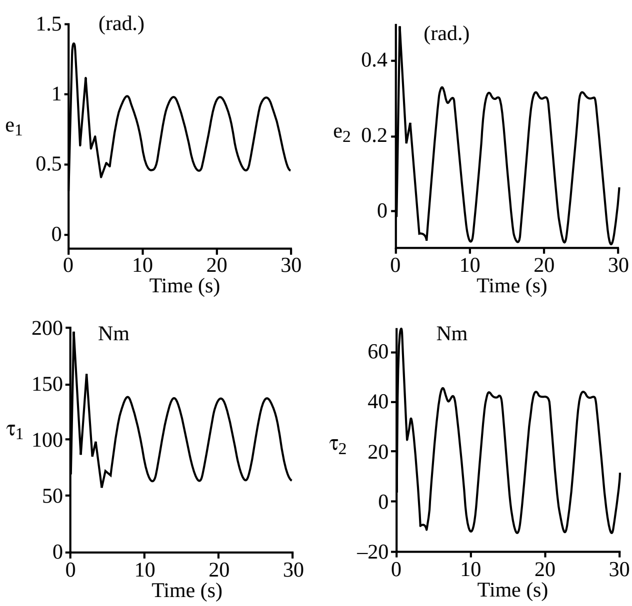

Fig. 4. Performance of PD controller.

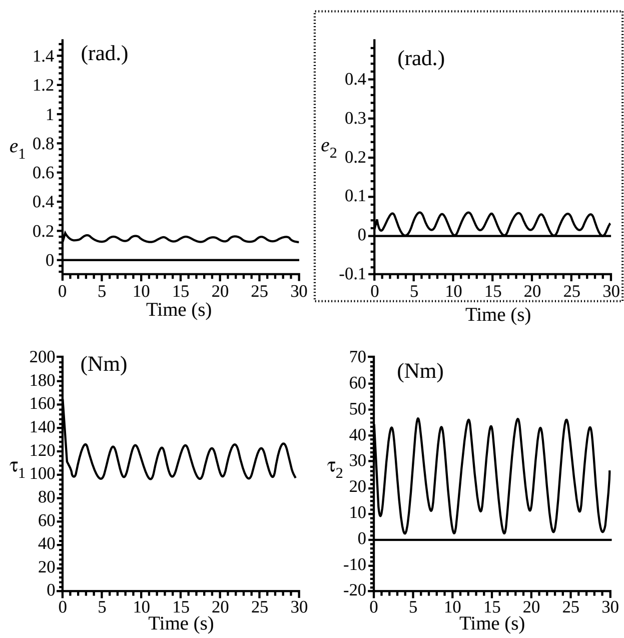

Fig. 6. Performance of robust controller with less conservative gains.

at the following controller

$$\tau = Y_d \hat{\theta} + k_v r + k_p e + v_R$$

with $Y_d\theta = f_d(\bullet)$ defined by (11) and $\hat{\theta} \in R^5$ the vector of unknown parameters a, b, c, d, e defined in (37). Gains k_p, k_v are the same as the PD controller. The robustifying term v_R is defined by

$$v_R = \frac{r(S\hat{\varphi})^2}{(S\hat{\varphi})\| r \| + \delta},$$

where

$$S = \left[1 \quad \| z \| \quad \| z \|^2\right],$$

$$z = [e^T \quad r^T]^T,$$

$$\varphi = [\zeta_0 \quad \zeta_1 \quad \zeta_2]^T,$$

$$\dot{\delta} = -\gamma\delta,\ \delta(0)\ ,= \text{design constant} > 0,\ \gamma > 0.$$

The parameter tuning law for φ is chosen as

$$\dot{\hat{\varphi}} = \Gamma S^T \| r \| = -\dot{\tilde{\varphi}},$$

where Γ is a symmetric and positive definite matrix and $\tilde{\varphi} = \varphi - \hat{\varphi}$. The estimates of the unknown parameters $\hat{\theta}$ is tuned by the following law Lewis *et al*. [12]

$$\dot{\hat{\theta}} = \Gamma_1 Y_d^T r$$

where Γ_1 is a symmetric and positive definite matrix. For the simulations, we choose $\Gamma = \Gamma_1 = 5$ I, $\gamma = 0.1$, $\delta(0) = 30$. All initial conditions of $\hat{\theta}$, $\hat{\varphi}$ are set to zero. The results are shown in Figs. 7 and 8. Figure 7 shows that DCAL works reasonably well. Figure 8 shows results where a term in $\hat{\theta}$, i.e. $\hat{e}$, is missing. This simulates the case of unmodeled dynamics. It can be seen that the performance in Fig. 8 is not very good since there exist steady-state errors in both links. This demonstrates that the DCAL controller is not robust to unmodeled dynamics since the performance in Fig. 8 is worse than that of Fig. 7.

(iv) DCAL-NN

The controller is given by (24), (30), (31), and (32). The parameters are $\Gamma = 5$ I, $\gamma = 0.1$, $\delta(0) = 30$, $F = G = 0.5$ I. k_p, k_v are the same as the PD controller. All initial conditions of $\hat{W}$, $\hat{V}$, $\hat{\varphi}$ are set to zero. There are ten units in the hidden layer. Although more units could be used, our results show that 10 units are quite enough in this example. The results are shown in Fig. 9. It can be seen that the convergence is not only slightly faster than DCAL

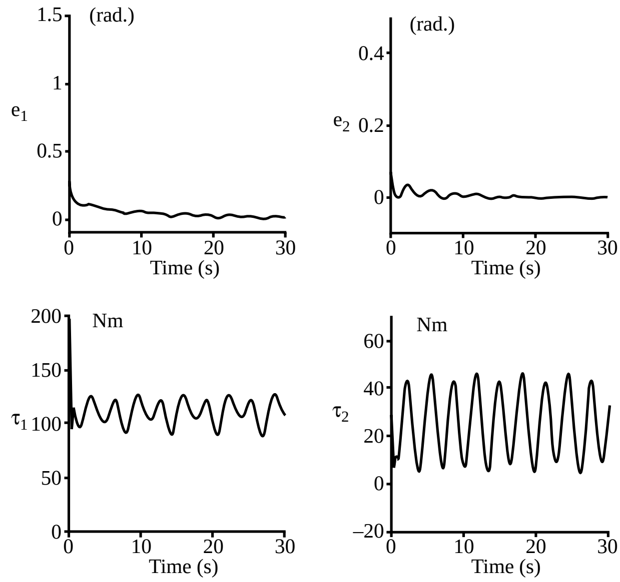

Fig. 7. Performance of DCAL controller.

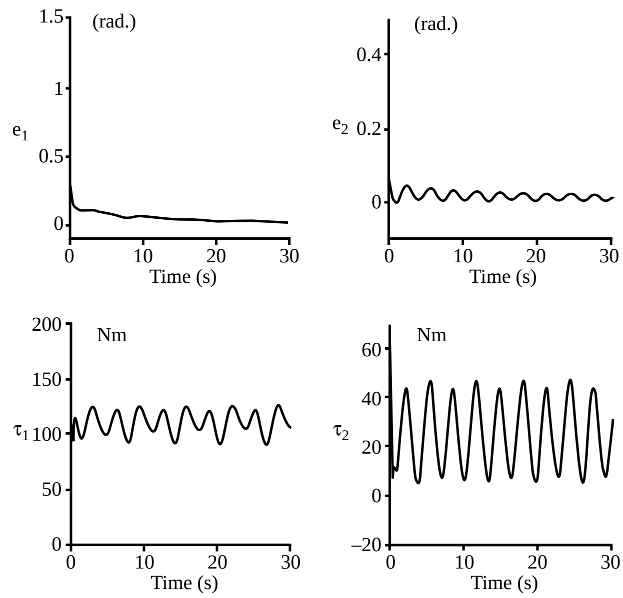

Fig. 8. Performance of DCAL controller with unmodeled dynamics.

but also the controller is very robust since we do not know any prior dynamics of the robot.

V. CONCLUSIONS

A new type of adaptive controller is proposed in this paper. Our method has the following advantages: a) global stability comparing to conventional NN methods (all the methods in [13]) yield local stability); b) zero steady-state tracking error comparing with PD control; c) no control chattering as compared to robust and sliding mode control methods [12]; d) on-line tuning of NN weights as compared to many off-line NN methods; e) can deal with

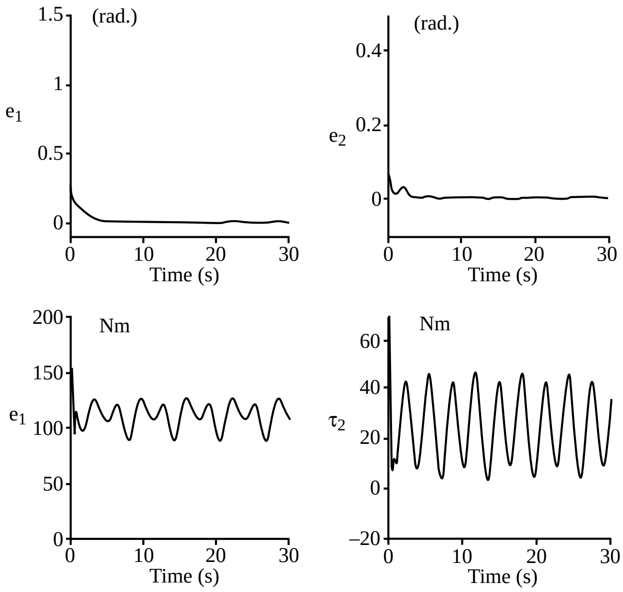

Fig. 9. Performance of DCAL-NN controller in this paper.

unmodelled dynamics (see simulations in Section 4) as compared to adaptive DCAL controller.

It is worth to mention some key differences between our earlier work in 1998 [11] and the current one. First, the NN method in this paper provides global stability of the closed-loop system whereas the one in [11] guarantees only local stability. The reason that the current paper achieves global stability is because NN is used to approximate a regression matrix that depends only on the desired trajectory, i.e. independent of system states. Second, [11] used only one layer of NN. Here we used two layers of NN. The weight updating is more complicated here and yet the approximation accuracy should be better in the current paper than that of [11]. Third, the application is different. The paper in 1998 [11] emphasized the application of NN to rigid link robot with slow actuator dynamics. Here we applied NN to robot systems where actuator dynamics can be ignored

APPENDIX

Proof of Lemma 1.

Since $\sigma(\bullet)$ and $\hat{\sigma}'(\bullet)$ in (22) are both bounded, then we have

$$\left\| Q(\tilde{V}^T x)^2 \right\| \le c_1 + \alpha \left\| \tilde{V}^T x \right\|, \; c_1 > 0, \; \alpha > 0.$$

$$\le c_1 + \alpha \left\| \tilde{V} \right\|_F Q_d$$

where Q_d is defined in Assumption 2. Defining $c_2 = \alpha Q_d$ yields the Lemma.

Proof of Lemma 2.

The first two terms in (27) can be shown to be bounded by [14]

$$\left\| f(\bullet) - f_d(\bullet) \right\| \le \alpha_0 + \alpha_1 \left\| z \right\| + \alpha_2 \left\| z \right\|^2$$

where α_i's are positive scalar constants. Noting the fact that $\hat{\sigma}'(\bullet)$ is bounded, the third term in (27) can be bounded by

$$\alpha_3 \left\| \tilde{W} \right\|_F \left\| \tilde{V} \right\|_F Q_d, \; \alpha_3 > 0 \qquad \text{(using Assumption 2)}$$

$$\le \alpha_3 (\left\| W \right\|_F + \left\| \hat{W} \right\|_F)(\left\| V \right\|_F + \left\| \hat{V} \right\|_F) Q_d$$

$$\le \alpha_3 (W_M + \left\| \hat{W} \right\|_F)(V_M + \left\| \hat{V} \right\|_F) Q_d. \qquad \text{(using Assumption 1)}$$

Using Lemma 1, the fourth term in (27) can be similarly bounded by

$$\alpha_4 + \alpha_5 \left\| \hat{V} \right\|_F$$

with $\alpha_4 > 0$, $\alpha_5 > 0$. The fifth and sixth terms are bounded by constants. Thus, combining the above results and manipulating some of the terms yields

$$\rho(z, \hat{W}, \hat{V}) = \zeta_0 + \zeta_1 \left\| z \right\| + \zeta_2 \left\| z \right\|^2 + \zeta_3 \left\| \hat{W} \right\|_F + \zeta_4 \left\| \hat{V} \right\|_F$$

$$+ \zeta_5 \left\| \hat{V} \right\|_F \left\| \hat{W} \right\|_F = S\varphi$$

where ζ_i*'s* are positive scalar constants.

REFERENCES

1. Abdallah, C., D. Dawson, P. Dorato and M. Jamishidi, “Survey of the Robust Control Robots,” *IEEE Contr. Syst. Mag.*, Vol. 11, No. 2, pp. 24-30 (1991).
2. Chen, F.C. and H.K. Khalil, “Adaptive Control of Nonlinear Systems Using Neural Networks,” *Int. J. Contr.*, Vol. 55, No. 6, pp. 1299-1317 (1992).
3. Corless, M., “Control of Uncertain Nonlinear Systems,” J. Dyn. Syst., Meas. Contr., Vol. 115 (1993).
4. Craig, J.J., *Adaptive Control of Mechanical Manipulators*, Wiley, New York (1986).
5. Cui, X. and K.G. Shin, “Direct Control and Coordination Using Neural Networks,” *IEEE Trans. Syst., Man. Cybern.*, Vol. 23, No. 3 (1993).
6. Cybenko, G., “Approximation by Superpositions of a Sigmoidal Function,” *Math. Contr. Signals Syst.*, Vol. 2, No. 4, pp. 303-314 (1989).
7. Flores, J., “Position and Force Control of Flexible and Rigid Manipulators,” Ph.D. Dissertation, The University of Texas at Arlington (1991).
8. Ge, S.S. and C.C. Hang, “Direct Adaptive Neural Network Control of Robots,'' *Int. J. Syst. Sci.*, Vol. 27,

No. 6, pp. 533-542 (1996).
9. Hornik, K., M. Stinchombe and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, pp. 359-366 (1989).
10. Slotine, J.J. and W. Li, "Adaptive Manipulator Control: A Case Study," *IEEE Trans. Automat. Contr*., Vol. 33, No.11, pp. 995-1003 (1988).
11. Kwan, C.M., F.L. Lewis and D. Dawson, "Robust Neural-Network Control of Rigid-Link Electrically Driven Robots," *IEEE Trans. Neural Networks*, Vol. 9, No. 4 (1998).
12. F.L. Lewis, C. Abdallah and D. Dawson, *Control of Robot Manipulators*, MacMillan Publishing Company, New York (1993).
13. Lewis, F.L., S. Jaganathan and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor and Francis (1998).
14. Narendra, K.S., "Adaptive Control Using Neural Networks," *Neural Networks for Control*, pp. 115-142, MIT Press, Boston (1991).
15. Narendra, K.S. and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks*, Vol.1, pp. 4-27 (1990).
16. Ozaki, T., T. Suzuki, T. Furuhashi, S. Okuma and Y. Ushikawa, "Trajectory Control of Robotic Manipulators," *IEEE Trans. Ind. Electron*., Vol. 38, pp. 195-202 (1991).
17. Polycarpou, M.M., and P.A. Ioannou, "Identification and Control Using Neural Network Models: Design and Stability Analysis," Technical Report 91-09-01, Department of Electrical Engineering System, University of Southern California (1991).
18. Rovithakis, G.A. and M.A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks," *IEEE Trans. Syst., Man Cybern.* (1994).
19. Sanner, R.M. and J.J. Slotine, "Stable Adaptive Control and Recursive Identification Using Radial Gaussian Networks," *Proc. IEEE Conf. Decis. Contr*., Brighton (1991).
20. Sadegh, N., "A Perceptron Network for Functional Identification and Control of Nonlinear Systems," *IEEE Trans. Neural Networks*, Vol 4, No. 6, pp. 982-988 (1993).
21. Sadegh, N. and R. Horowitz, "Stability and Robustness Analysis of a Class of Adaptive Controllers for Robotic Manipulators," *Int. J. Rob. Res*., Vol.9, No.3, pp. 74-92 (1990).

Chiman Kwan (S'85-M'93-SM'98) was born on 19 February 1966. He received his B.S. degree in electronics with honors from the Chinese University of Hong Kong in 1988 and M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Arlington in 1989 and 1993, respectively.

From April 1991 to February 1994, he worked in the Beam Instrumentation Department of the SSC (Superconducting Super Collider Laboratory) in Dallas, Texas, where he was heavily involved in the modeling, simulation and design of modern digital controllers and signal processing algorithms for the beam control and synchronization system. He received an invention award for his work at SSC. He later joined the Automation and Robotics Research Institute in Fort Worth, where he applied intelligent control methods such as neural networks and fuzzy logic to the control of power systems, robots, and motors. Since July 1995, he has been with Intelligent Automation, Inc. in Rockville, Maryland. He has served as Principal Investigator/Program Manager for more than twenty different projects such as modeling and control of advanced machine tools, digital control of high precision electron microscope, enhancement of microscope images, and adaptive antenna arrays for beam forming, automatic target recognition of FLIR and SAR images, fast flow control in communication networks, vibration management of gun pointing system, health monitoring of flight critical systems, high speed piezoelectric actuator control, fault tolerant missile control, active speech enhancement, fault detection and isolation of various electromechanical systems, and underwater vehicle control. Currently, he is the Director of Research and Development, leading research efforts in signal/image processing, and controls. His primary research areas include fault detection and isolation, robust and adaptive control methods, signal and image processing, communications, neural networks, and fuzzy logic applications.

Dr. Kwan is listed in the New Millennium edition of Who's Who in Science and Engineering and is a member of Tau Beta Pi.

Darren M. Dawson received a B.S. Degree in Electrical Engineering from the Georgia Institute of Technology in 1984. He then worked for Westinghouse as a control engineer from 1985 to 1987. In 1987, he returned to the Georgia Institute of Technology where he received the Ph.D. Degree in Electrical Engineering in March 1990. In

July 1990, he joined the Electrical and Computer Engineering Department at Clemson University where he currently holds the position of Centennial Professor. Professor Dawson research interests Nonlinear Control Techniques for Mechatronic Applications such as Electric Machinery, Robotic Systems, Aerospace Systems, Acoustic Noise, Underactuated Systems, Magnetic Bearings, Mechanical Friction, Paper Handling/Textile Machines, Flexible Beams/Robots/Rotors, Cable Structures, and Vision-Based Systems. He also focuses on the development of Realtime Hardware and Software Systems for Control Implementation.

Dr. Lewis was born in Würzburg, Germany, subsequently studying in Chile and Gordonstoun School in Scotland. He obtained the Bachelor's Degree in Physics/Electrical Engineering and the Master's of Electrical Engineering Degree at Rice University in 1971. 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). In 1977 he received the Master's of Science in Aeronautical Engineering from the University of West Florida. In 1981 he obtained the Ph.D. degree at The Georgia Institute of Technology in Atlanta, where he was employed as a professor from 1981 to 1990 and is currently an Adjunct Professor. He is a Professor of Electrical Engineering at The University of Texas at Arlington, where he was awarded the Moncrief-O'Donnell Endowed Chair in 1990 at the Automation and Robotics Research Institute.

Dr. Lewis has studied the geometric, analytic, and structural properties of dynamical systems and feedback control automation. His current interests include robotics, intelligent control, neural and fuzzy systems, nonlinear systems, and manufacturing process control. He is the author/co-author of 2 U.S. patents, 124 journal papers, 20 chapters and encyclopedia articles, 210 refereed conference papers, seven books: *Optimal Control, Optimal Estimation, Applied Optimal Control and Estimation, Aircraft Control and Simulation, Control of Robot Manipulators, Neural Network Control, High-Level Feedback Control with Neural Networks* and the IEEE reprint volume *Robot Control*. Dr. Lewis is a registered Professional Engineer in the State of Texas and was selected to the Editorial Boards of *International Journal of Control, Neural Computing and Applications, and Int. J. Intelligent Control Systems*. He is currently an Editor for the flagship journal *Automatica*. He is the recipient of an NSF Research Initiation Grant and has been continuously funded by NSF since 1982. Since 1991 he has received \$1.8 million in funding from NSF and upwards of \$1 million in SBIR/industry/state funding. He has received a Fulbright Research Award, the American Society of Engineering Education *F.E. Terman* Award, three Sigma Xi Research Awards, the UTA Halliburton Engineering Research Award, the UTA University-Wide Distinguished Research Award, the ARRI Patent Award, various Best Paper Awards, the IEEE Control Systems Society Best Chapter Award (as Founding Chairman), and the National Sigma Xi Award for Outstanding Chapter (as President). He was selected as Engineer of the year in 1994 by the Ft. Worth IEEE Section and is a Fellow of the IEEE. He was appointed to the NAE Committee on Space Station in 1995 and to the IEEE Control Systems Society Board of Governors in 1996. In 1998 he was selected as an IEEE Control Systems Society *Distinguished Lecturer*. He is a Founding Member of the Board of Governors of the Mediterranean Control Association.