Implementation of a Neural Network Tracking Controller for a Single Flexible Link: Comparison with PD and PID Controllers

L. B. Gutiérrez, *Member, IEEE*, F. L. Lewis, *Fellow, IEEE*, and J. Andy Lowe

***Abstract*—The objective of this paper is to show the results of the practical implementation of a neural network (NN) tracking controller on a single flexible link and compare its performance to that of proportional derivative (PD) and proportional integral derivative (PID) standard controllers. The NN controller is composed of an outer PD tracking loop, a singular perturbation inner loop for stabilization of the fast flexible-mode dynamics, and an NN inner loop used to feedback linearize the slow pointing dynamics. No off-line training or learning is needed for the NN. It is shown that the tracking performance of the NN controller is far better than that of the PD or PID standard controllers. An extra friction term was added in the tests to demonstrate the ability of the NN to learn unmodeled nonlinear dynamics.**

***Index Terms*—Flexible structures, neurocontrollers.**

I. Introduction

IN RECENT literature, there have been many neural network (NN) controllers proposed for robot arms or other nonlinear systems [4], [20]–[26]. The performance of these NN controllers on actual systems has been open to question, despite the fact that several of these references provide stability proofs. In this paper, we implement the NN controller derived in [31] on an actual single-flexible-link robot arm which could emulate, for instance, a tank gun barrel in DOD applications. It is found that the NN controller far outperforms standard proportional derivative (PD) and proportional integral derivative (PID) controllers, even for the single-link arm, which is basically linear except for nonlinear friction effects.

The control of flexible-link robot arms belongs to a class of problems characterized by reduced control effectiveness and additional unstable zero dynamics. Some other problems in this category are large-scale space structures, overhead gantry cranes, and other industrial processes. The requirement of controllers with faster response and higher accuracy introduces a challenge that the researchers have faced in different ways.

Several researchers [19], [30] have observed that the approximate flexible-link robot arm dynamics is input–output feedback linearizable, but the zero dynamics is not asymptotically stable when the tip position is taken as the output. To control the arm, a modified output was defined to yield stable zero dynamics. However, this output does not correspond to practical tracking objectives, except in the set-point command case. In [27] and [29], input–output feedback linearization and a singular perturbation correction term [11] to stabilize the internal dynamics was used to control a multilink flexible arm. Finally, in [17], a Lyapunov approach is used to stabilize a component of the tracking error, but not the tracking error in its entirety.

There are different control techniques for rigid robot arms available in the literature. These techniques require an exact knowledge of the nonlinear terms (computed torque), knowledge of bounds on uncertainties (robust control), or knowledge of a nonlinear regression matrix of robot functions (adaptive control) [14]. In practice, it is very difficult to have such *a priori* knowledge of the arm dynamics, especially in the presence of frictional terms, which may not have a known dynamical form.

To overcome these limitations, an NN tracking controller for a rigid-link robot arm has been devised in [13] and [15]. In this scheme, there is an outer PD tracking loop, with the NN used in a feedback linearization inner loop. The weight-training rules include an e-modification term [22] and a term corresponding to a second-order term. Using a Lyapunov approach, it is shown that these training rules guarantee tracking performance and bounded weights, even though there do not exist ideal weights, such that the NN perfectly reconstructs the nonlinear robot function.

In [31], a tracking controller for a flexible arm is designed using singular perturbation plus an NN feedback linearization inner loop. There, a modified output for tracking is defined that does correspond to practical tracking requirements. The structure of that controller includes an outer PD tracking loop, a singular perturbation inner loop for stabilization of the fast dynamics, and an NN inner loop used to feedback linearize the rigid dynamics. Applying singular perturbation theory, it is shown that, after stabilizing the fast dynamics, the slow dynamics can be controlled using the same approach used in [13] and [15]. This approach avoids the requirement of the knowledge of friction, gravity, and Coriolis/centripetal terms, or any regression matrix. In contrast to other NN controllers in the literature, there is no off-line learning phase, the NN weights are easy to initialize without known "stabilizing initial weights" (the weights are initialized at zero), and the controller

Manuscript received August 6, 1996; revised August 11, 1997.

L. B. Gutiérrez was with the Automation and Robotics Research Institute, The University of Texas at Arlington, Fort Worth, TX 76118-7115 USA. He is now with the School of Engineering, Universidad Pontificia Bolivariana, Medellín, Colombia.

F. L. Lewis and J. A. Lowe are with the Automation and Robotics Research Institute, The University of Texas at Arlington, Fort Worth, TX 76118-7115 USA.

Publisher Item Identifier S 0278-0046(98)01568-8.

guarantees boundedness of the tracking error and control signal.

In this paper, we present some practical implementation results for a single flexible link for the controller designed in [31]. Despite the fact that the dynamics of a single flexible link are linear, an extra friction term was added in the implementation to show the capability of the NN controller to compensate for nonlinearities in the model by learning. A comparison with the performance of standard PD and PID controllers is performed to show the superior tracking performance of the NN controller.

II. Dynamics of a Flexible-Link Robot Arm

In [3], [5]–[7], and [16], it is shown that the dynamics of any multilink flexible-link robot can be represented by

$$M(q)\ddot{q} + D(q,\dot{q})\dot{q} + Kq + F(q,\dot{q}) + G(q) = B(q)u \quad (1)$$

with

$$q = \begin{bmatrix} q_r \\ q_f \end{bmatrix}$$

where q_r is the vector of rigid modes (generalized joint coordinates) and q_f is the vector of flexible modes (the amplitudes of the flexible modes). $M(q)$ represents the inertia matrix, $D(q,\dot{q})$ is the Coriolis and centrifugal matrix, K is the stiffness matrix, $F(q,\dot{q})$ is the friction matrix, $G(q)$ is the gravity matrix, $B(q)$ is an input matrix dependent on the boundary conditions selected in the assumed mode shapes method, and u includes the control torques applied to each joint.

The model (1) follows the same properties of any standard rigid-link robot [16]. That is, $M(q)$ is positive definite and upper and lower bounded, $D(q,\dot{q})$ is bounded by $d_b(q)\|\dot{q}\|$, and $D(q,\dot{q})$ can be chosen such that $\dot{M}(q) - 2D(q,\dot{q})$ is skew symmetric [16].

III. NN Control of Flexible-Link Robot Arms

A. Singular Perturbation Approach

The singular perturbation approach basically consists of breaking the dynamics of the system into two parts, each of them in a separate time scale [10]–[12]. In this case, the slow dynamics correspond to the rigid modes q_r and the fast dynamics correspond to the flexible modes q_f. In order to apply singular perturbation, (1) can be split, as in [27] and [29]:

$$\begin{aligned}\ddot{q}_r &= -D^1_{rr}\dot{q}_r - D^1_{rf}\dot{q}_f - K^1_{rf}q_f - F^1_r - G^1_r + B^1_r u\\ \ddot{q}_f &= -D^1_{fr}\dot{q}_r - D^1_{ff}\dot{q}_f - K^1_{ff}q_f - F^1_f - G^1_f + B^1_f u. \end{aligned} \quad (2)$$

Now introduce the scale factor ϵ and define

$$\epsilon^2\xi = q_f \quad (3)$$

where $1/\epsilon^2$ is the smallest stiffness in K^1_{ff}. Define

$$\tilde{K}_{ff} \equiv \epsilon^2 K^1_{ff}. \quad (4)$$

Then, defining H_{rf}, such that $K^1_{rf} = H_{rf}(1/\epsilon^2)\tilde{K}_{ff}(q_r,\epsilon^2\xi)$ (H_{rf} is a normalized stiffness respect to K^1_{ff}), we get

$$\begin{aligned}\dot{q}_r &= -D^1_{rr}(\theta)\dot{q}_r - D^1_{rf}(\theta)\epsilon^2\dot{\xi} - H_{rf}(1/\epsilon^2)\tilde{K}_{ff}(q_r,\epsilon^2\xi)\epsilon^2\xi\\ &\quad - F^1_r(q_r,\epsilon^2\xi) - G^1_r(q_r,\epsilon^2\xi) + B^1_r(q_r,\epsilon^2\xi)u\\ \epsilon^2\ddot{\xi} &= -D^1_{fr}(\theta)\dot{q}_r - D^1_{ff}(\theta)\epsilon^2\dot{\xi} - (1/\epsilon^2)\tilde{K}_{ff}(q_r,\epsilon^2\xi)\epsilon^2\xi\\ &\quad - F^1_f(q_r,\epsilon^2\xi) - G^1_f(q_r,\epsilon^2\xi) + B^1_f(q_r,\epsilon^2\xi)u \end{aligned} \quad (5)$$

where $\theta = (q_r, \dot{q}_r, \epsilon^2\xi, \epsilon^2\dot{\xi})$.

Here is considered the case in which the stiffness of the links is sufficiently large, so that ϵ is sufficiently small. The control objective is that $q_r(t)$ should track $q_d(t)$, a prescribed trajectory. For that purpose, define the control

$$u = \overline{u} + u_F \quad (6)$$

where $\overline{u}$ is the slow component and u_F is the fast component. From this point, the bar over the variables is used to denote the slow part of them. To obtain the equations for the slow dynamics, set $\epsilon = 0$ in (5) to obtain

$$\ddot{\overline{q}}_r = -\overline{D}^1_{rr}\dot{\overline{q}}_r - \overline{H}_{rf}\tilde{K}_{ff}\overline{\xi} - \overline{F}^1_r - \overline{G}^1_r + \overline{B}^1_r\overline{u} \quad (7)$$

and the algebraic slow manifold equation

$$0 = -\overline{D}^1_{fr}\dot{\overline{q}}_r - \overline{H}_{ff}\tilde{K}_{ff}\overline{\xi} - \overline{F}^1_f - \overline{G}^1_f + \overline{B}^1_f\overline{u} \quad (8)$$

which is solved for the slow variables

$$\overline{\xi} = \tilde{K}^{-1}_{ff}\overline{H}^{-1}_{ff}(-\overline{D}^1_{fr}\dot{\overline{q}}_r - \overline{F}^1_f - \overline{G}^1_f + \overline{B}^1_f\overline{u}). \quad (9)$$

Substituting (9) in (7) and defining $\overline{M}^{-1}_{rr} \equiv \overline{B}^1_r - \overline{H}_{rf}\overline{H}^{-1}_{ff}\overline{B}^1_f$, $\overline{D}_{rr} \equiv \overline{M}_{rr}(\overline{D}^1_{rr} - \overline{H}_{rf}\overline{H}^{-1}_{ff}\overline{D}^1_{fr})$, $\overline{F}_r \equiv \overline{M}_{rr}(I - \overline{H}_{rf}\overline{H}^{-1}_{ff})\overline{F}^1_r$, and $\overline{G}_r \equiv \overline{M}_{rr}(I - \overline{H}_{rf}\overline{H}^{-1}_{ff})\overline{G}^1_r$ we get

$$\ddot{\overline{q}}_r = \overline{M}^{-1}_{rr}(-\overline{D}_{rr}\dot{\overline{q}}_r - \overline{F}_r - \overline{G}_r + \overline{u}). \quad (10)$$

For the fast subsystem, define the states

$$\begin{aligned}\zeta_1 &\equiv \xi - \overline{\xi}\\ \zeta_2 &\equiv \epsilon\dot{\xi}\end{aligned} \quad (11)$$

with a time scale $\tau = t/\epsilon$ resulting in

$$\begin{aligned}\frac{d\zeta_1}{d\tau} &= \zeta_2\\ \frac{d\zeta_2}{d\tau} &= -\overline{D}^1_{fr}\dot{\overline{q}}_r - D^1_{ff}\epsilon\zeta_2 - \overline{H}_{ff}\tilde{K}_{ff}(\zeta_1 + \overline{\xi})\\ &\quad - \overline{F}^1_f - \overline{G}^1_f + \overline{B}^1_f(\overline{u} + u_F)\end{aligned} \quad (12)$$

since $d\epsilon/d\tau \approx 0$. Setting $\epsilon = 0$ and substituting from (9) the fast dynamics are found to be

$$\frac{d}{d\tau}\begin{bmatrix}\zeta_1\\ \zeta_2\end{bmatrix} = \begin{bmatrix}0 & I\\ -\overline{H}_{ff}\tilde{K}_{ff} & 0\end{bmatrix}\begin{bmatrix}\zeta_1\\ \zeta_2\end{bmatrix} + \begin{bmatrix}0\\ \overline{B}^1_f\end{bmatrix}u_F \quad (13)$$

or

$$\frac{d\zeta}{d\tau} = A_F\zeta + B + Fu_F \quad (14)$$

with $\zeta = [\zeta_1^T \;\; \zeta_2^T]^T$.

According to Tikhonov's theorem [10], [11], the original system (2) can be described to order ϵ using (10) and (14) with

$$\begin{aligned} q_r &= \overline{q}_r + O(\epsilon) \\ q_f &= \epsilon^2(\overline{\xi} + \zeta_1) + O(\epsilon) \end{aligned} \tag{15}$$

with $O(\epsilon)$ denoting terms of order ϵ.

Now, define the tracking output

$$y = \begin{bmatrix} \overline{q}_r \\ \dot{\overline{q}}_r \end{bmatrix} \tag{16}$$

which corresponds to the *slow part* of the rigid-mode variables (e.g., of the link-tip motion). Assume that (A_F, B_F) is stabilizable, the fast system parameters have bounded uncertainties and perturbations (slow subsystem variables), and the slow system variables vary smoothly with time. The stabilizing assumption on (A_F, B_F) is satisfied in practical systems and is far milder that the requirement for stable zero dynamics. Moreover, the definition (16) corresponds to practical tracking objectives in contrast to the "reflected" outputs defined in [19] and [30]. Under these assumptions, a stabilizing control $u_F(t)$ can easily be designed using linear techniques (e.g., H_∞ design), so that

$$u_F = -[K_{pF} \quad K_{dF}] \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = -\frac{K_{pF}}{\epsilon^2} q_f - \frac{K_{dF}}{\epsilon} \dot{q}_f + K_{pF}\overline{\xi} \tag{17}$$

stabilizes (14), with $\overline{\xi}$ given by (9).

B. NN Control of the Rigid Dynamics

The slow dynamics given by (10) can be rewritten as

$$\overline{M}_{rr}\ddot{\overline{q}}_r + \overline{D}_{rr}\dot{\overline{q}}_r + \overline{F}_r + \overline{G}_r = \overline{u} \tag{18}$$

which is exactly the Lagrange form of an n-link rigid robot arm, satisfying the standard robot properties. For this part, an NN controller can be designed [13], [15]. Note that $\dot{\overline{M}}_{rr} - 2\overline{D}_{rr}$ is skew symmetric.

Given a desired trajectory $q_d(t)$ for $\overline{q}_r$, the tracking error is

$$e = q_d - \overline{q}_r. \tag{19}$$

Define the filtered tracking error as

$$r = \dot{e} + \Lambda e \tag{20}$$

where $\Lambda = \Lambda^T > 0$. Using (20), the arm dynamics can be rewritten in terms of the filtered tracking error as

$$\overline{M}_{rr}\dot{r} = -\overline{D}_{rr}r - \overline{u} + h(x) \tag{21}$$

where the nonlinear robot function is

$$\begin{aligned} h(x) =& \overline{M}_{rr}(\overline{q})\,(\ddot{\overline{q}}_d + \Lambda\dot{e}) + \overline{D}_{rr}(\overline{q}, \dot{\overline{q}})\,(\dot{\overline{q}}_d + \Lambda e) \\ &+ \overline{F}_r(\dot{\overline{q}}) + \overline{G}_r(\overline{q}) \end{aligned} \tag{22}$$

with $x = [e^T \; \dot{e}^T \; \overline{q}_d^T \; \ddot{\overline{q}}_d^T]^T$. It is assumed that $h(x)$ is unknown.

An NN can be used to estimate $h(x)$ based on the *universal approximation property* of NN's, which is stated in the following theorem.

Theorem—Universal Approximation Property of the NN: Let $f(x)$: $\Re^n \to \Re^m$ be a smooth function. Then, given a compact set $S \in \Re^n$ and a positive number ϵ_N, there exists a two-layer NN, such that

$$f(x) = W^T\sigma(V^Tx) + \epsilon \tag{23}$$

with $\|\epsilon\| < \epsilon_N$ for all $x \in S$, for some (sufficiently large) number L of hidden-layer neurons. ϵ is generally a function of x and is called the NN function approximation error. ϵ decreases as L increases. □

The estimate of $h(x)$ is given by

$$\hat{h}(x) = \hat{W}^T\sigma(\hat{V}^Tx). \tag{24}$$

Let

$$Z \equiv \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \tag{25}$$

be the ideal weight matrix, which is unknown.

The functional approximation error of the NN is

$$\tilde{h}(x) = h(x) - \hat{h}(x) \tag{26}$$

which can be written using a Taylor expansion, assuming smooth activation functions, as

$$\tilde{h}(x) = \tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}_h^Tx) + \hat{W}^T\hat{\sigma}'\tilde{V}^Tx + w \tag{27}$$

where

$$\hat{\sigma} \equiv \sigma(\hat{V}^Tx), \quad \hat{\sigma}' \equiv \left.\frac{\partial\sigma(z)}{\partial z}\right|_{z=\hat{z}} \tag{28}$$

and the additional error term

$$w(t) = \tilde{W}^T\hat{\sigma}'V^Tx + W^TO(\tilde{V}^Tx)^2 + \epsilon_{l_h}(x) \tag{29}$$

bounded according to

$$w(t) \le C_0 + C_1\|\tilde{Z}\| + C_2\|x\|\,\|Z\|. \tag{30}$$

The Jacobian $\hat{\sigma}'$ is an easily computed function of $\hat{V}^Tx$.

It is assumed that the ideal weights of the NN are bounded, so that

$$\|Z\| \le Z_m \tag{31}$$

with Z_m a known bound, and the desired trajectory is bounded according to

$$\left\| \begin{matrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{matrix} \right\| \le Q \tag{32}$$

with Q a known bound.

Definition: The solution to

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

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

Under all the assumptions stated above, an NN controller is defined by the following theorem [31].

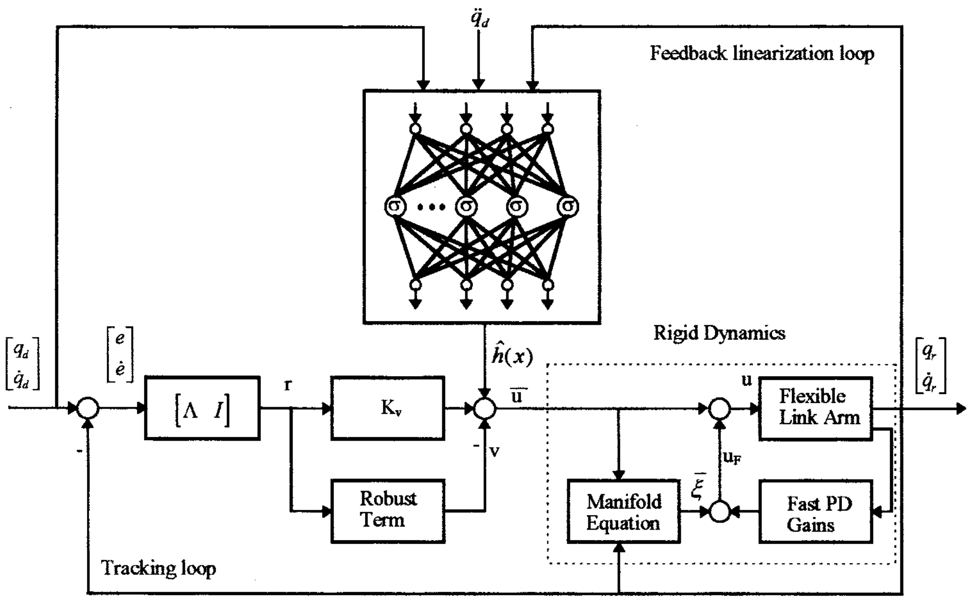


Fig. 1. Overall control structure of the NN controller for a flexible link robot.

Theorem: Let the desired trajectory and the ideal unknown weights be bounded according to the assumptions. Let the control input for (18) be defined by

$$\overline{u} = \hat{h} + K_v r - v, \quad \text{for } K_v = K_v^T > 0 \tag{33}$$

with robustifying term

$$v(t) = -K_z(\|\hat{Z}\| + Z_m)r \tag{34}$$

and gain $K_z > C_2$.

Let the NN weights be tuned by

$$\begin{aligned}\dot{\hat{W}} &= M(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)r^T - \kappa\|r\|M\hat{W} \\ \dot{\hat{V}} &= Nxr^T\hat{W}^T\hat{\sigma}' - \kappa\|r\|N\hat{V}\end{aligned} \tag{35}$$

with any constant matrices $M = M^T > 0, N = N^T > 0$, and a scalar design parameter $\kappa > 0$.

Then, the filtered tracking error $r(t)$ and the NN weight errors $\tilde{V}$, $\tilde{W}$ are GUUB. Moreover, the tracking error may be kept as small as desired by increasing the gains K_v. □

The proof of this theorem uses Lyapunov theory and is given in [31], where explicit bounds on $\|r\|$ and $\|\tilde{Z}\|$ are given. Notice that the training rules in (35) include the standard backpropagation terms plus an e-modification [22] and a second-order correction term. Furthermore, the NN weights can be easily initialized at zero, since the PD control stabilizes the system while the NN is learning. The NN controller is designed to control the robot arm while it is learning to improve the performance, hence, no off-line training is required.

The overall structure of the controller defined in Sections III-A and B is shown in Fig. 1.

C. Simulation

The simulation of the NN controller was performed in Matlab for a single flexible link. The model of the flexible link included three flexible modes, even though the controller only compensated for the first two modes (this was to corroborate that the controller works well even compensating for only a finite number of modes). The model was obtained as described in [7], using the parameters of the flexible-link test bed at the Automation and Robotics Research Institute (ARRI), The University of Texas at Arlington. The modal frequencies for the first three modes for this flexible link are 1.6, 10.0, and 28.1 Hz.

The controller used the following parameters:

$$\begin{aligned}K_v &= 36 \\ \Lambda &= \frac{200}{36} \\ K_z &= 0.2 \\ Z_m &= 50 \\ \frac{K_{pf}}{\epsilon^2} &= [-8 \quad 10] \\ \frac{K_{df}}{\epsilon} &= [0 \quad 0].\end{aligned}$$

The NN in the controller included ten neurons in the hidden layer and used the following parameters:

$$\begin{aligned}F &= 20 \\ G &= 20 \\ \kappa &= 0.000\,001.\end{aligned}$$

The activation functions for the neurons in the hidden layer were selected as the sigmoid functions

$$\sigma_k(z) = \frac{1}{1 + e^{-k\alpha z}}, \qquad \text{for } k = 1, 2, \cdots, 10; \alpha = 1.$$

It was observed in practice that, with the sigmoid functions defined this way, the NN learned faster and was able to reduce the tracking error more.

The results of the simulation are plotted in Fig. 2. Notice that, after some time, the NN learns the model of the link, reducing the tracking error to almost zero. Using a bigger

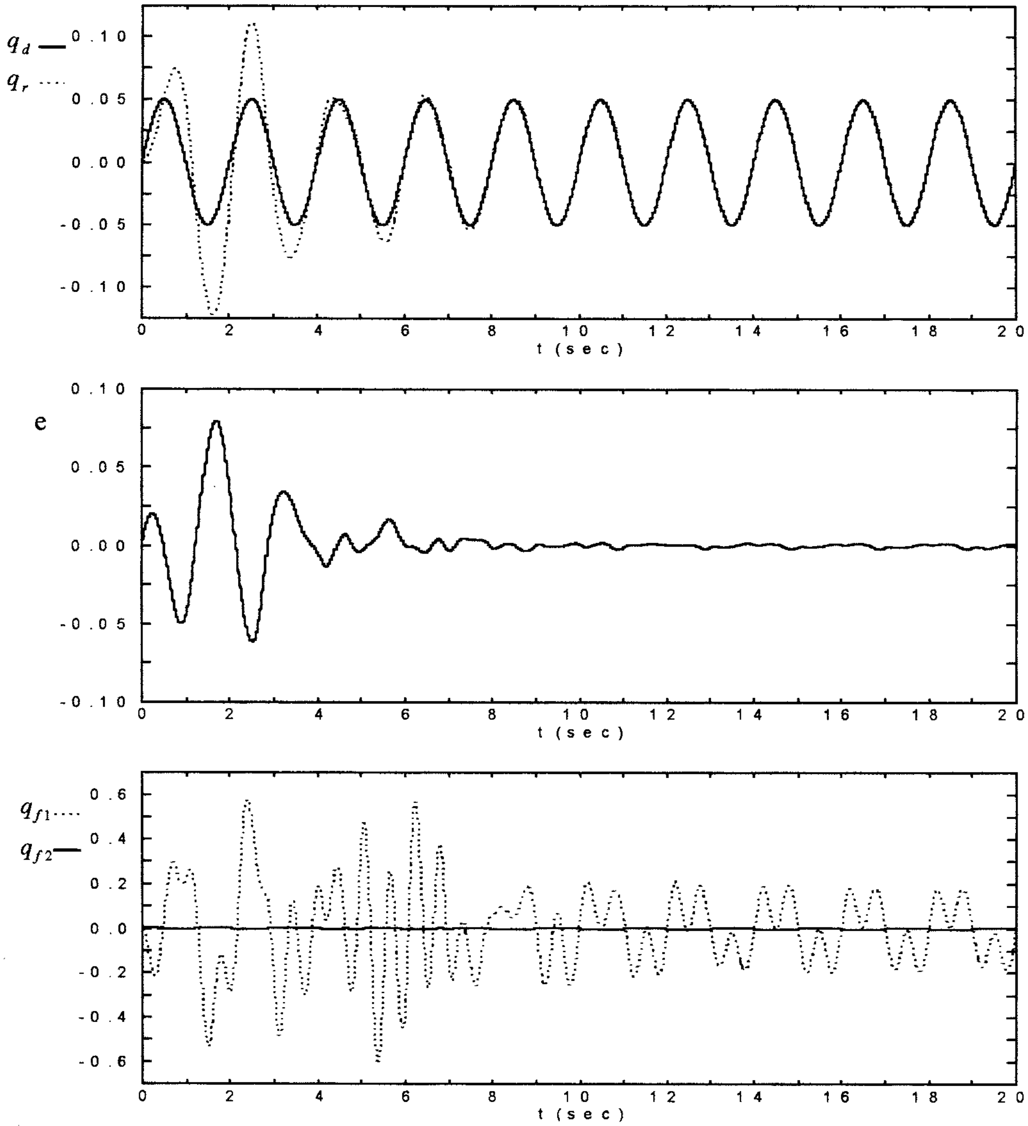


Fig. 2. Simulation of the NN controller with a single flexible link.

value for the learning rates F and G improved the tracking performance (faster learning and less tracking error), but produced a worse transient response (more oscillatory) and more excitation of the flexible modes (increased the magnitude of q_{f1} and q_{f2}, the flexible modes).

IV. Implementation of the Controller in the Flexible-Link Test Bed

The NN controller discussed in Section III was implemented on a single-flexible-link test bed at the ARRI, and some of the results obtained are presented here.

A. Description of the Implementation

The actual test bed at the ARRI is shown in Fig. 3. A list of the main characteristics of the practical implementation is given below.

- The flexible link is an aluminum beam with dimensions: 48 in × 2 in × 1/8 in.

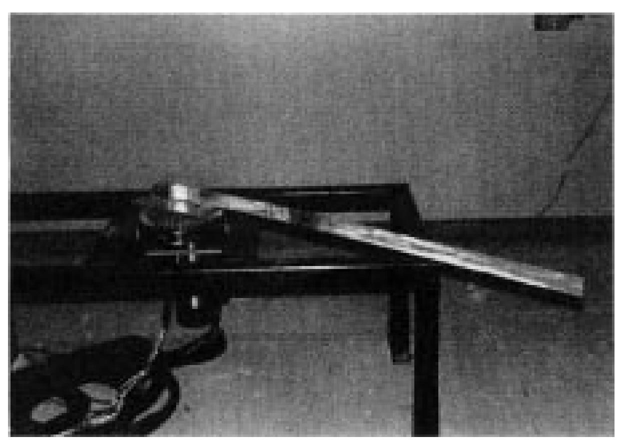


Fig. 3. Actual flexible-link test bed at the ARRI.

- Only the first two flexible modes were considered.
- The robust term v was not included. K_v was selected big enough to avoid the necessity of v. The manifold term was not included, since the actual model of the flexible

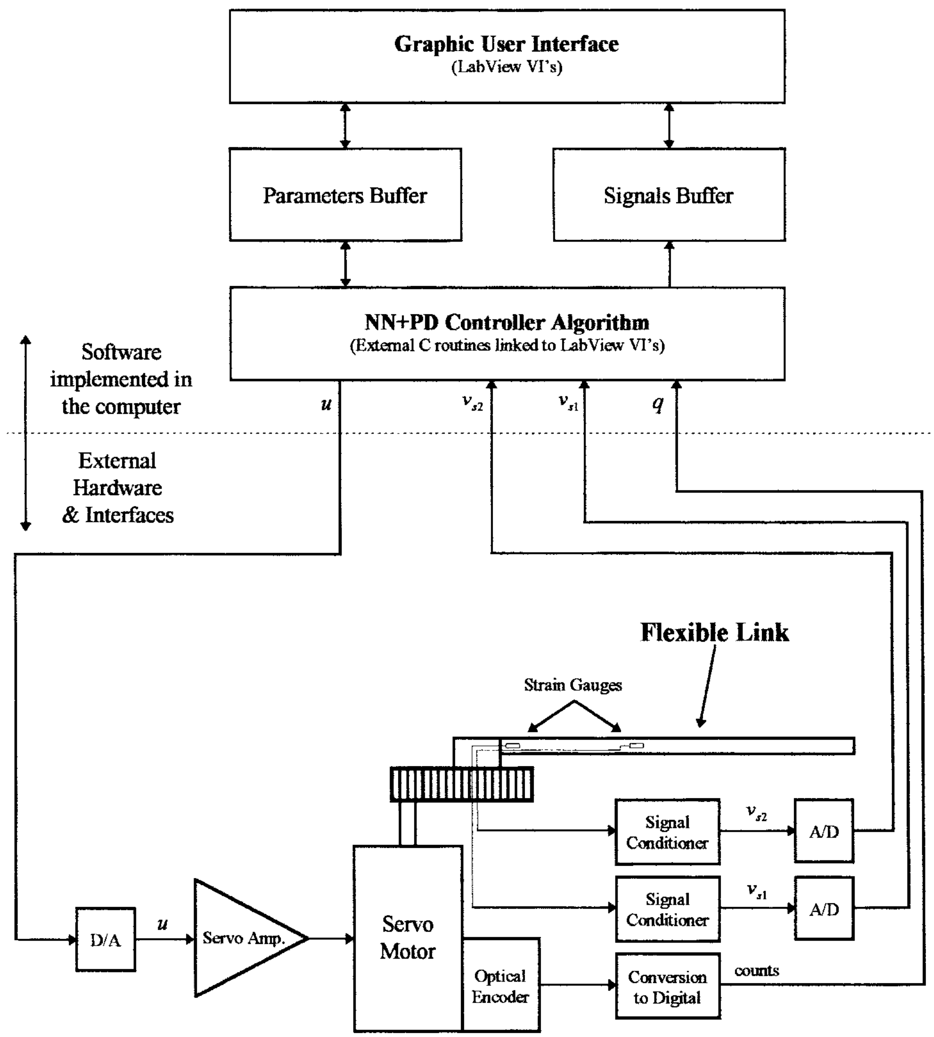


Fig. 4. Block diagram of NN controller implementation at ARRI's flexible-link test bed.

link is unknown. As shown by (9), the implementation of $\overline{\xi}$ would require the exact knowledge of the matrixes of the model.

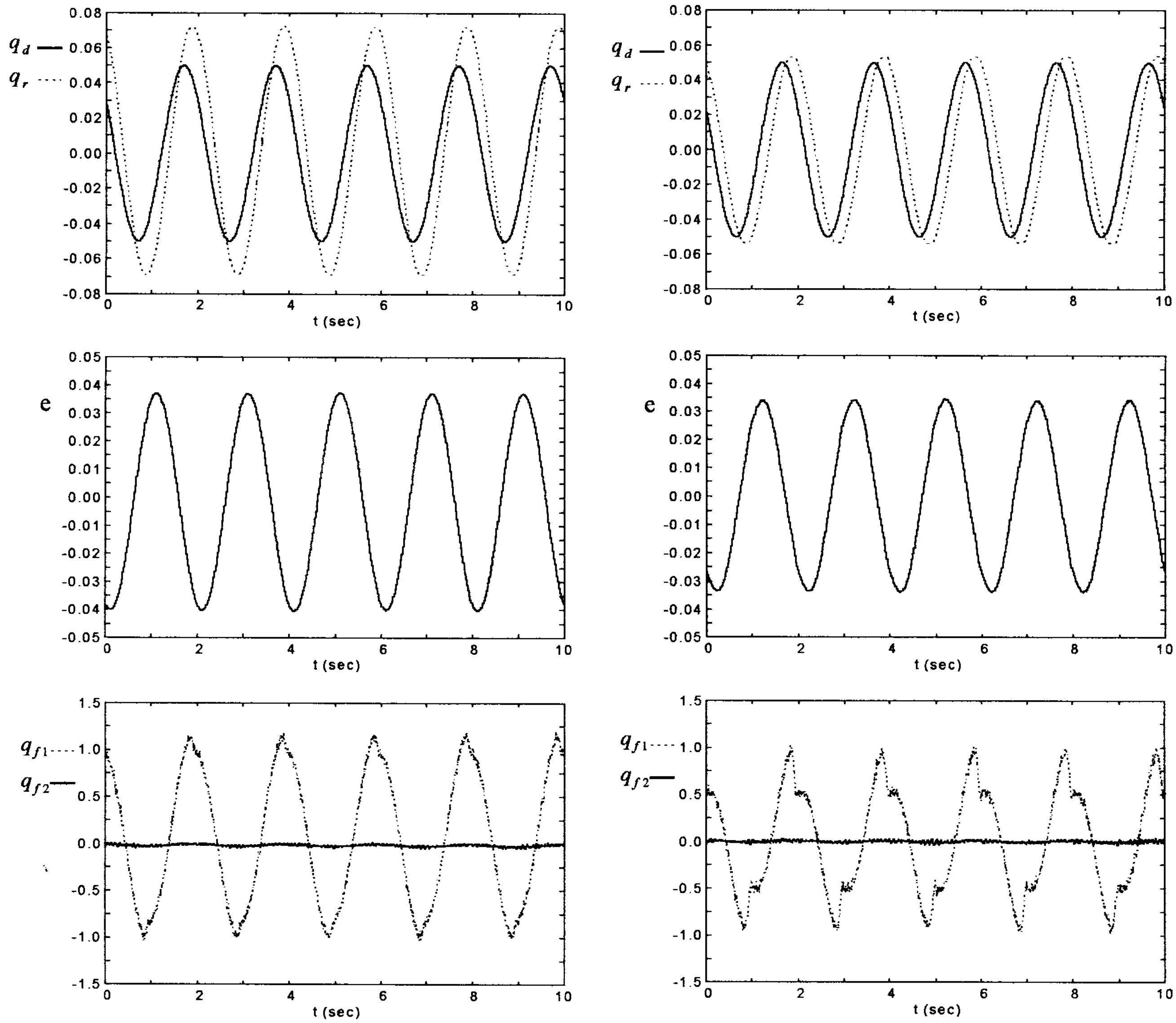
- Even though the dynamics for the flexible link with 1 degree of freedom is linear, an extra nonlinear friction term was added to check the capability of the controller to compensate for the nonlinearities in the model.
- The NN is composed of ten neurons in the hidden layer, with five inputs ($x = [e\ \dot{e}\ \overline{q}_d\ \dot{\overline{q}}_d\ \ddot{\overline{q}}_d]^T$) and one output $\hat{h}(x)$.
- The controller defined by (6), (17), (33), and (35) was discretized with a sampling period of 5 ms. In the discretization process, the differential equations in (35) were solved on line using trapezoidal integration.

A block diagram describing the practical implementation of the controller is shown in Fig. 4. The hardware includes the interface cards and external components necessary for the measurement of the angular position of the link q_r and the flexible modes q_{f1} and q_{f2} (optical encoder, strain gauges, signal conditioners, and analog-to-digital converters). Estimated values of $\dot{q}_r, \dot{q}_{f1}, \dot{q}_{f2}$ are calculated based on consecutive samples of q_r, q_{f1}, and q_{f2}, respectively. Besides, there is a digital-to-analog converter connected to the servo amplifier that drives the servo motor for the link.

Notice in Fig. 8(a), without the extra friction term, and Fig. 8(b), with the extra friction term, that the same controller learns the model of the link, readapting to changes in it (changes in the model like changes in friction characteristics). Without changing the parameters of the controller, the NN controller is able to take the tracking error to almost zero in both cases.

The software was implemented in LabView and C. The routines that perform the control action in real time are implemented in C. The execution of these external routines is fired periodically by the computer timer routines. The control routines sample the external signals and use the parameters defined in the parameters buffer to calculate the control signal u. Some of the signals are stored in the signals buffer allowing the LabView virtual instruments (VI's) to monitor them.

The LabView VI's work as a graphic user interface that allows one to start the controller, change the mode of operation, define the reference signals, change the parameters of the controller, and monitor the signals through charts and graphics.



(a) (b)

Fig. 5. Performance of the PD control. (a) Without additional friction. (b) With additional friction.

These VI's are linked to the external C routines which run in the background in real time. The communication between the external C routines and the LabView VI's is accomplished through some VI's that read from and write to the buffers using code interface nodes (CIN's).

B. Experimental Results in the Flexible-Link Test Bed

Standard PD and PID controllers were implemented and tested in the flexible-link test bed to compare their performance with the PD+NN controller. This comparison allows us to show the advantages of the proposed controller over the standard controllers.

1) PD Control: A PD controller was implemented using the control law

$$u = \overline{u} + u_F$$

with

$$\overline{u} = K_v r = K_v(\dot{e} + \Lambda e)$$

$$u_F = -[K_{pF} \quad K_{dF}] \begin{bmatrix} q_f \\ \dot{q}_f \end{bmatrix}$$

using the following parameters:

$$K_v = 36$$

$$\Lambda = \frac{200}{36}$$

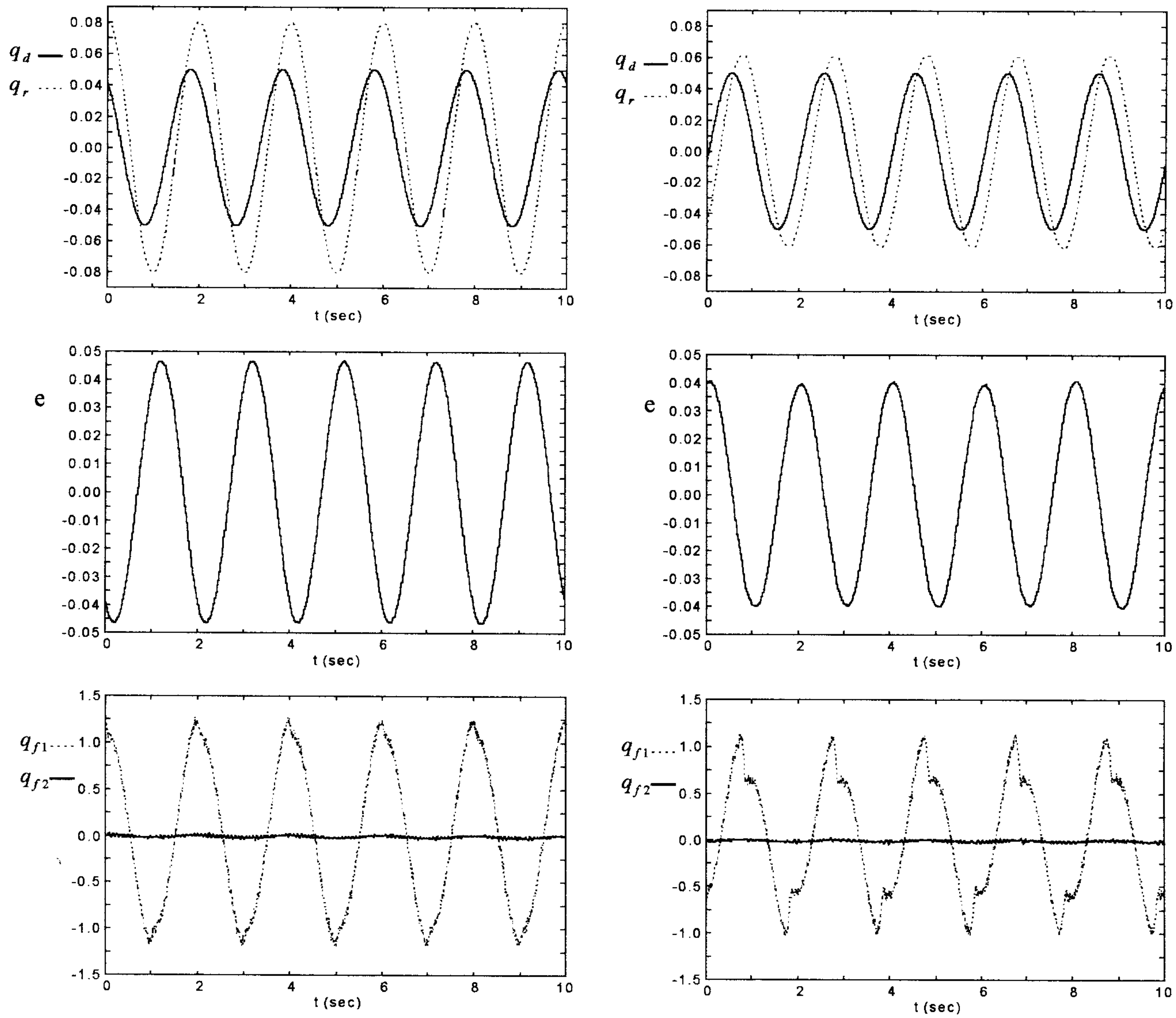
$$K_{pf} = [-8 \quad 10]$$

$$K_{df} = [0 \quad 0]$$

and the reference signal

$$q_d = 0.05 \sin(2\pi f t)$$

with frequency $f = 0.5$ Hz.



(a) (b)

Fig. 6. Performance of the PID control. (a) Without additional friction. (b) With additional friction.

The performance of the tracking PD control without the NN is illustrated in Fig. 5(a) without the extra friction term and in Fig. 5(b) with the extra friction term. Notice that the tracking error is very big; its magnitude is comparable to that of the reference signal. Even though the magnitude of the error decreases incrementing the controller gains, the tracking error is not eliminated. These characteristics are preserved in the presence of the extra friction term.

2) PID Control: A PID controller was implemented using the control law

$$u = \overline{u} + u_F$$

with

$$\overline{u} = K_v r + K_i \int e\, dt = K_v(\dot{e} + \Lambda e) + K_i \int e\, dt$$
$$u_F = -[K_{pF} \quad K_{df}] \begin{bmatrix} q_f \\ \dot{q}_f \end{bmatrix}$$

using the following parameters:

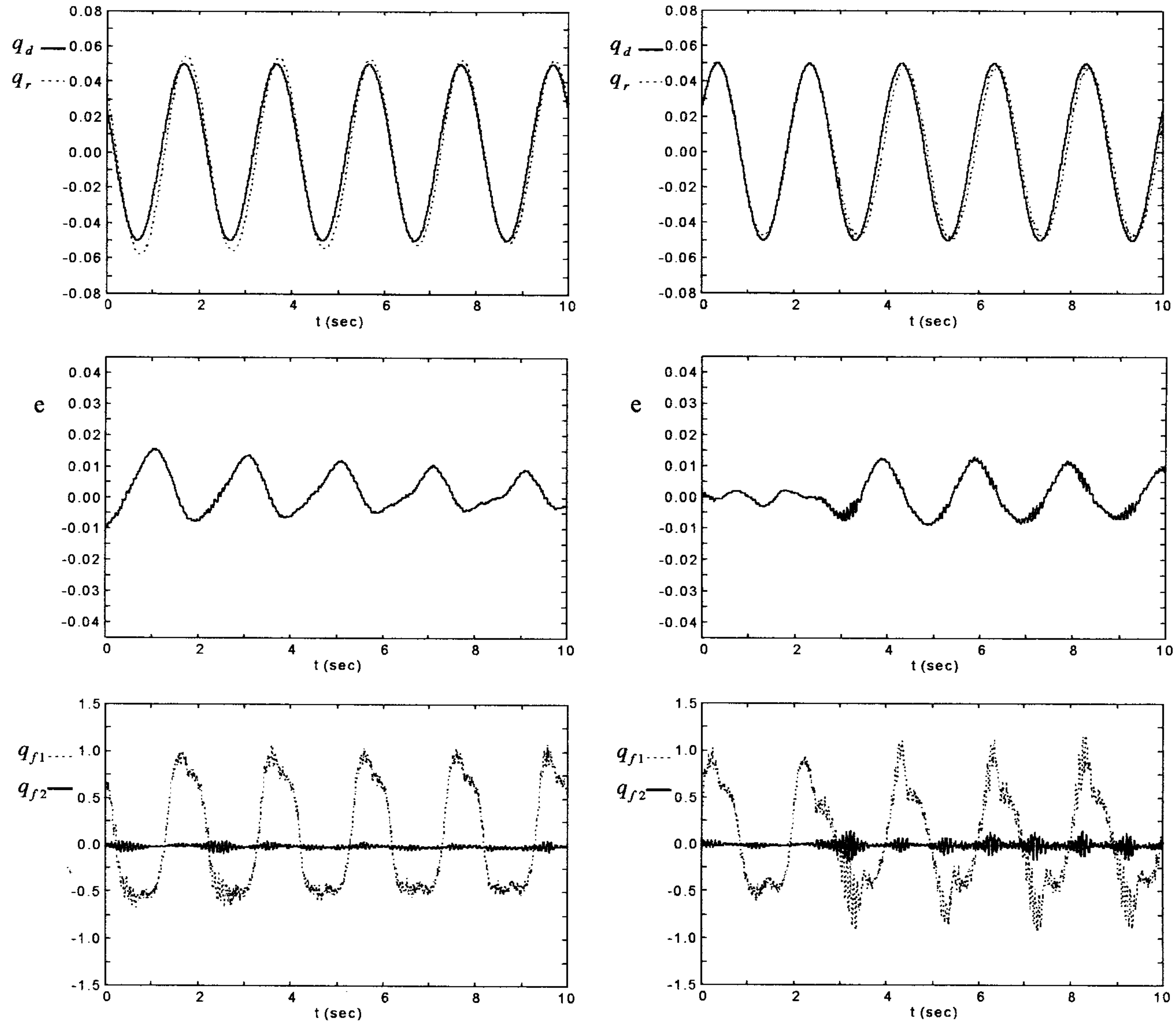
$$K_v = 36$$
$$\Lambda = \frac{200}{36}$$
$$K_i = 100$$
$$K_{pf} = [-8 \quad 10]$$
$$K_{df} = [0 \quad 0]$$

and the reference signal

$$q_d = 0.05 \sin(2\pi f t)$$

with frequency $f = 0.5$ Hz.

The performance of the tracking PID control is illustrated in Fig. 6(a) without the extra friction term and in Fig. 6(b) with the extra friction term. The integral part of the PID controller is supposed to eliminate the steady-state error, but only works for constant desired trajectories. In this



(a) (b)

Fig. 7. Performance of the PD+NN control before learning. (a) Without additional friction. (b) With additional friction.

case, with a varying desired trajectory, the tracking is even worse when the integral part is introduced (notice that the tracking error is bigger than with the PD control). As in the case of the PD controller, the PID controller is not able to compensate for the extra friction term.

3) NN+PD Control: The NN tracking controller was implemented as described in Section IV-A using the same parameters of the simulation in Section III-C, except that, in this case

$$F = 2$$
$$G = 20$$
$$\kappa = 0.000\,001.$$

This value of F in the practical implementation was enough. A bigger value produced a very oscillatory response.

The reference signal was

$$q_d = 0.05 \sin(2\pi f t)$$

with frequency $f = 0.5$ Hz.

The performance of the NN tracking control is illustrated in Fig. 7 before the learning is complete and in Fig. 8 after the learning is complete. The training of the NN takes less than 1 min, after which the tracking error is reduced to almost zero. The learning is really active all the time (on-line training), but we refer to learning as being complete to the instant when the NN has learned the model of the link under the actual conditions, reducing the tracking error to almost zero.

In practice, it was noticed that a change in the reference signal increased the tracking error momentarily, requiring a readaptation of the NN. However, after some time, when the NN learned the new conditions, it was able to get rid of the tracking error.

C. Comparison Between Different Approaches

Comparing the tracking performance of the different controllers shown in Fig. 5 for the PD controller, Fig. 6 for the PID controller, and Figs. 7 and 8 for the NN controller, the superiority of the last one is clear. Even the PID cannot be

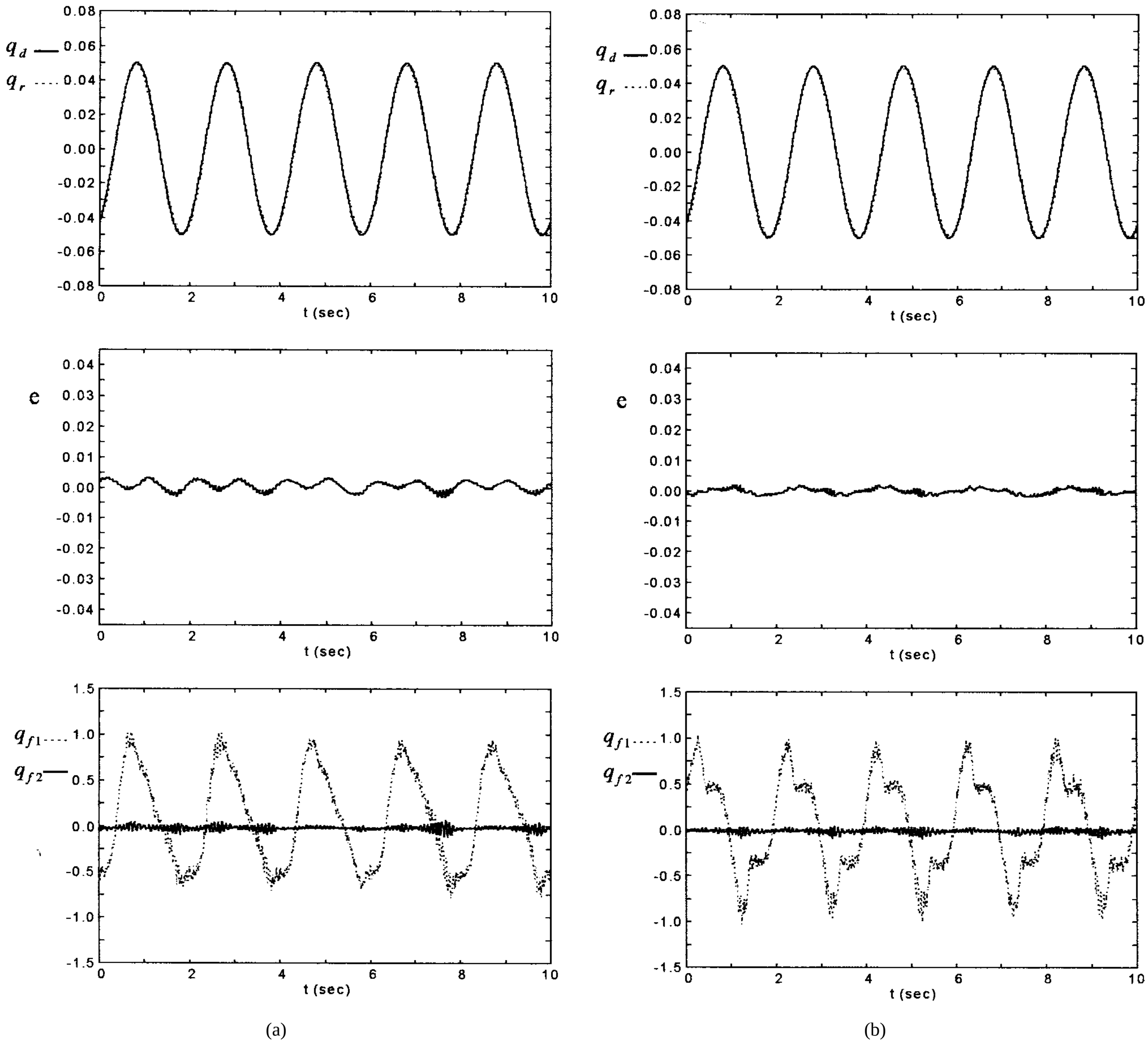


Fig. 8. Performance of the PD+NN control after learning. (a) Without additional friction. (b) With additional friction.

a better tracking controller than the NN controller under a varying desired trajectory. In this situation, the PD controller is better than the PID, but not as good as the NN controller.

In all cases, it is noted that the flexible modes are well damped out by the controller. It is impossible to eliminate the bending of the link (represented by q_{f1}) under a varying desired trajectory because that is part of its physics, but the compensation of the higher frequency mode q_{f2} is evident.

Even though the NN tracking controller was not designed to track a step function [see assumption given by (32)], it was tested with a step desired trajectory for purposes of comparison with the PD and PID controllers described above. These controllers were tested with a step desired trajectory with and without the extra friction term. The results are plotted in Fig. 9.

Notice that the PD controller has good transient response [Fig. 9(a)] but is not able to get rid of the steady-state error, and it gets worse in the case of the extra friction [Fig. 9(b)]. Increasing the gains in this case improves the steady-state error, but makes the transient response more oscillatory.

The PID controller presents a worse transient response with a higher overshot [Fig. 9(c)], but tries to eliminate the steady-state error, even though it is very slow [Fig. 9(d)]. It is possible to increase the speed of the PID controller by increasing K_i, but that produces a bad transient response with a big overshot and it is very oscillatory; in addition, the tracking performance gets worse.

The NN controller presents a response which is a little oscillatory, but the overshot is not too high [Fig. 9(e)], being comparable to that of the PD controller, and it always takes the steady-state error to zero, and it even is able to compensate for the extra friction term [Fig. 9(f)]. In general, the NN controller acts as a smart nonlinear integrator which is able to compensate for the nonlinear dynamics of the link (learned by the NN), taking the steady-state error to almost zero, even in the presence of hard nonlinearities like friction.

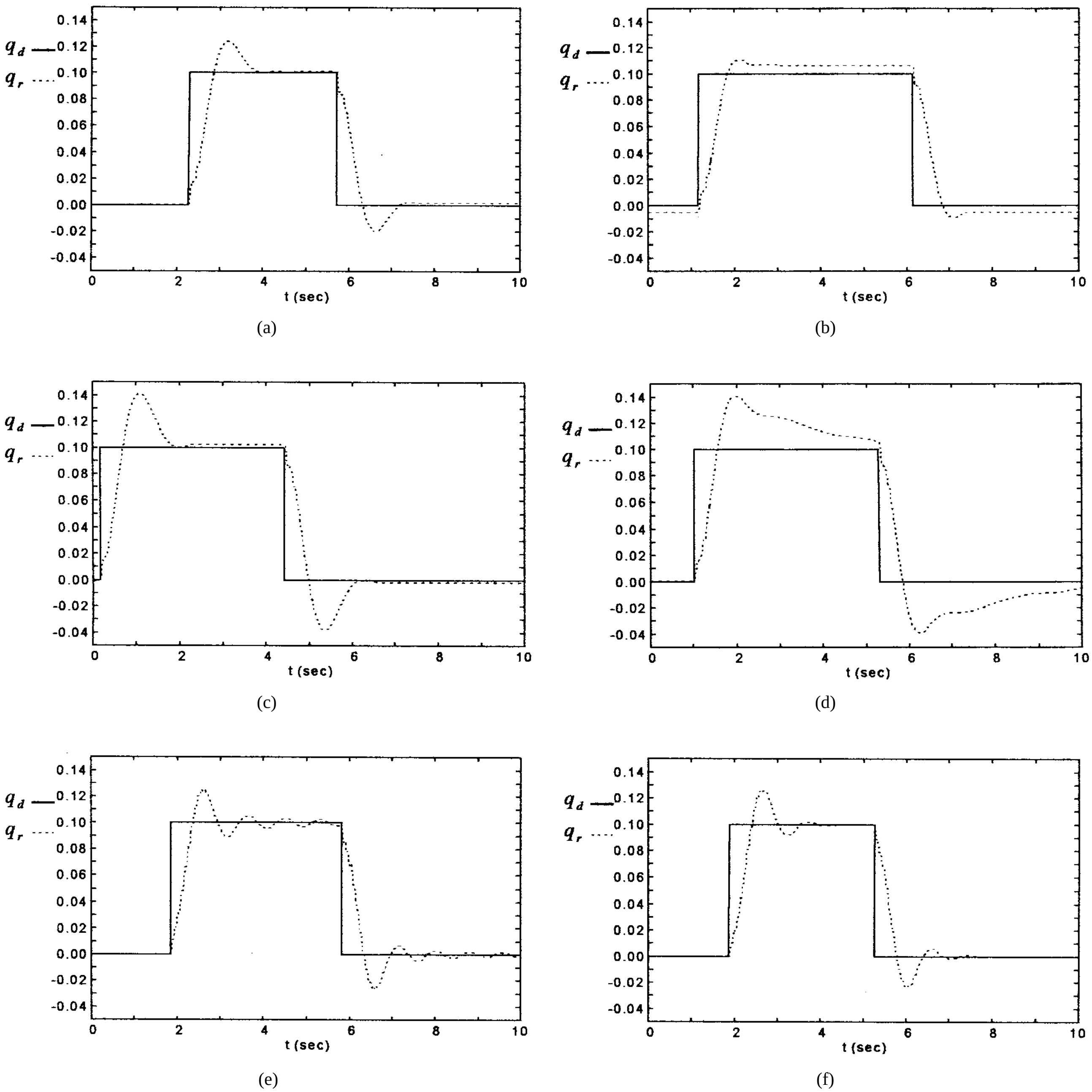


Fig. 9. Step response of the controllers. (a) PD control. (b) PD control with extra friction. (c) PID control. (d) PID control with extra friction. (e) PD+NN control. (f) PD+NN control with extra friction.

V. Conclusions

The practical implementation of a multiloop nonlinear NN tracking controller for a single flexible link has been tested and its performance compared to that of the standard PD and PID controllers. An extra friction term was added in the implementation to show the ability of the NN controller to learn and compensate for the nonlinearities.

The controller includes an outer PD tracking loop, a singular perturbation inner loop for stabilization of the fast dynamics, and an NN inner loop used to feedback linearize the slow dynamics. This NN controller requires no off-line learning phase, the NN weights are easily initialized, and it guarantees boundedness of the tracking error and control signal.

The practical results corroborate the simulations showing that standard PD or PID controllers are not able to track a varying desired trajectory, while the NN controller takes the tracking error to almost zero, readapting to any changes in the model of the link (extra friction terms).

References

[1] H. Asada, Z.-D. Ma, and H. Tokumaru, "Inverse dynamics of flexible robot arms: modeling and computation for trajectory control," *J. Dynam. Syst. Measur. Contr.*, vol. 112, pp. 177–185, June 1990.

[2] W. J. Book, "Modeling, design, and control of flexible manipulator arms: A tutorial review," in *Proc. 29th IEEE Conf. Decision and Control*, Dec. 1990, pp. 500–506.

[3] S. Centikunt, B. Siciliano, and W. J. Book, "Symbolic modeling and dynamic analysis of flexible manipulators," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Oct. 1986, pp. 798–803.

[4] F.-C. Chen and C.-C. Liu, "Adaptively controlling nonlinear continuous-time systems using multilayer neural networks," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1306–1310, June 1994.

[5] A. De Luca and B. Siciliano, "Closed-form dynamic model of planar multilink lightweight robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 826–839, July/Aug. 1991.

[6] G. G. Hastings and W. J. Book, "A linear dynamic model for flexible robotic manipulators," *IEEE Contr. Syst. Mag.*, vol. 7, pp. 61–64, Apr. 1987.

[7] G. G. Hastings and W. J. Book, "Verification of a linear dynamic model for flexible robotic manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, Apr. 1986, pp. 1024–1029.

[8] S. Haykin, *Neural Networks*. New York, NY: IEEE Press, 1994.

[9] F. Khorrami, "Analysis of multi-link flexible manipulators via asymptotic expansions," in *Proc. 28th IEEE Conf. Decision and Control*, Dec. 1989, pp. 2089–2094.

[10] P. V. Kokotovic, H. K. Kalil, and J. O'Reilly, *Singular Perturbation Methods in Control: Analysis and Design*. London, U.K.: Academic, 1986.

[11] P. V. Kokotovic, "Applications of singular perturbation techniques to control problems," *SIAM Rev.*, vol. 26, no. 4, pp. 501–550, Oct. 1984.

[12] P. V. Kokotovic, R. E. O'Malley Jr., and P. Sannuti, "Singular perturbations and order reduction in control theory—An overview," *Automatica*, vol. 12, no. 3, pp. 123–132, Mar. 1976.

[13] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 6, pp. 703–715, May 1995.

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

[15] F. L. Lewis, K. Liu, and A. Yeildirek, "Neural net robot controller: Structure and stability proofs," in *Proc. 32nd IEEE Conf. Decision and Control*, Dec. 1993, pp. 2785–2791.

[16] J. Lin and F. L. Lewis, "Dynamic equations of a manipulator with rigid and flexible links: Derivation and symbolic computation," in *Proc. American Control Conf.*, June 1993, pp. 2868–2872.

[17] S. H. Lin, S. Tosunolu, and D. Tesar, "Control of a six-degree-of-freedom flexible industrial manipulator," *IEEE Contr. Syst. Mag.*, vol. 10, pp. 24–30, Apr. 1990.

[18] K. Liu and F. L. Lewis, "Hybrid feedback linearization/fuzzy logic control of a flexible link manipulator," *J. Intell. Fuzzy Syst.* , vol. 2, no. 4, pp. 325–336, 1994.

[19] S. K. Madhavan and S. N. Singh, "Inverse trajectory control and zero dynamics sensitivity of an elastic manipulator," in *Proc. 1991 American Control Conf.*, June 1991, pp. 1879–1884.

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

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

[22] K. S. Narendra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Automat. Contr.*, vol. AC-32, pp. 134–145, Feb. 1987.

[23] M. M. Polycarpou and P. A. Ioannou, "Identification and control using neural network models: Design and stability analysis," Dept. Elect. Eng. Syst., Univ. Southern California, Los Angeles, CA, Tech. Rep. 91-09-01, Sept. 1991.

[24] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, pp. 400–412, May 1994.

[25] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 4, pp. 982–988, Nov. 1993.

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

[27] B. Siciliano and W. J. Book, "A singular perturbation approach to control of lightweight manipulators," *Int. J. Robot. Res.*, vol. 7, no. 4, pp. 79–90, Aug. 1988.

[28] P. B. Usoro, R. Nadira, and S. S. Mahil, "A finite element/Lagrange approach to modeling lightweight flexible manipulators," *J. Dynam. Syst., Measur. Contr.*, vol. 108, pp. 198–205, Sept. 1986.

[29] M. W. Vandegrift, F. L. Lewis, and S. Zhu, "Flexible-link robot arm control by a feedback linearization/singular perturbation approach," *J. Robot. Syst.*, vol. 11, no. 7, pp. 591–603, 1994.

[30] D. Wang and M. Vidyagasar, "Transfer functions for a single flexible link," *Int. J. Robot. Res.*, vol. 10, no. 5, pp. 540–549, Oct. 1991.

[31] A. Yesildirek, M. W. Vandegrift, and F. L. Lewis, "A neural network controller for flexible-link robots," in *Proc. IEEE Int. Symp. Intelligent Control*, Aug. 1994, pp. 63–68.

L. B. Gutiérrez (M'93) was born in Medellín, Colombia, in 1966. He received the Electronics Engineer degree from the Universidad Pontificia Bolivariana, Medellín, Colombia, in 1989 and the M.Sc. degree in electrical engineering from The University of Texas at Arlington, Fort Worth, TX, in 1996, as a Fulbright-LASPAU Scholar.

He joined the School of Engineering, Universidad Pontificia Bolivariana, as an Auxiliary Professor in 1989, becoming an Assistant Professor in 1991 and an Associate Professor in 1994. He teaches courses in signals and systems and for the specialization in automatica. His current research interests are control theory, intelligent control, neural networks, and signal processing. He is the author of the book *Sistemas y Señales.*

F. L. Lewis (S'78–M'81–SM'86–F'94) was born in Würzburg, Germany. He received the B.S. degree in physics/electrical engineering and the M.E.E. degree from William Marsh Rice University, Houston, TX, in 1971, the M.S. degree in aeronautical engineering from the University of West Florida, Pensacola, in 1977, and the Ph.D. degree from Georgia Institute of Technology, Atlanta, in 1981.

He served for six years in the U.S. Navy, as Navigator aboard the frigate USS Trippe (FF-1075) and as Executive Officer and Acting Commanding Officer aboard the USS Salinan (ATF-161). He was with Georgia Institute of Technology from 1981 to 1990. He was awarded the Moncrief-O'Donnell Endowed Chair in 1990 at the Automation and Robotics Research Institute, The University of Texas at Arlington, Fort Worth, TX. His current interests include robotics, intelligent control, nonlinear systems, and manufacturing process control. He is the author or coauthor of numerous journal papers, refereed conference papers, and books, including *Optimal Control*, *Optimal Estimation*, *Applied Optimal Control and Estimation*, *Aircraft Control and Simulation*, *Control of Robot Manipulators*, and *Robot Control*. He is a member of the Editorial Boards of the *International Journal of Control* and *Circuits, Systems, and Signal Processing*.

Dr. Lewis is a Registered Professional Engineer in the State of Texas. He is the recipient of an NSF Research Initiation Grant, a Fulbright Research Award, the American Society of Engineering Education F. E. Terman Award, three Sigma Xi Research Awards, the UTA Halliburton Engineering Research Award, the UTA University-Wide Distinguished Research Award, and the IEEE Control Systems Society Best Chapter Award (as Founding Chairman). He was selected as Engineer of the Year in 1994 by the Fort Worth IEEE Section.

J. Andy Lowe received the B.S.E.E. degree from Rose-Hulman Institute of Technology, Terre Haute, IN, in 1988 and the M.S.E.E. degree from Georgia Institute of Technology, Atlanta, in 1989.

He is a real-time digital control system software designer and a specialist in electromechanical control system implementation with the Automation and Robotics Research Institute, the University of Texas at Arlington, Fort Worth, TX. He has designed, developed, and implemented several real-time controllers and assisted in the design and implementation of digital control systems for original-equipment manufacturers that produce semiconductor wafer fabrication equipment, material handling manipulator systems, shipboard and mobile land-based antenna pointing manipulators, and autonomous vehicles. He has recently developed and implemented several PC-based control systems which have been demonstrated on a range of electromechanical systems, including a hydraulic Stewart platform, a flexible-link (gun barrel) pointing device, and an automatically programmed CAD-driven robotic surface finishing system.