ANN-Based Adaptive Control of Robotic Manipulators With Friction and Joint Elasticity

Hicham Chaoui, *Student Member, IEEE*, Pierre Sicard, *Member, IEEE*, and Wail Gueaieb, *Senior Member, IEEE*

***Abstract*—This paper proposes a control strategy based on artificial neural networks (ANNs) for a positioning system with a flexible transmission element, taking into account Coulomb friction for both motor and load, and using a variable learning rate for adaptation to parameter changes and accelerate convergence. A control structure consists of a feedforward ANN that approximates the manipulator's inverse dynamical model, an ANN feedback control law, a reference model, and the adaptation process of the ANNs with a variable learning rate. A supervisor that adapts the neural network's learning rate and a rule-based supervisor for online adaptation of the parameters of the reference model are proposed to maintain the stability of the system for large variations of load parameters. Simulation results highlight the performance of the controller to compensate the nonlinear friction terms, particularly Coulomb friction, and flexibility, and its robustness to the load and drive motor inertia parameter changes. Internal stability, which is a potential problem in such a system, is also verified. The controller is suitable for DSP and very large scale integration implementation and can be used to improve static and dynamic performances of electromechanical systems.**

***Index Terms*—Adaptive control, flexible structures, intelligent control, manipulators, uncertain systems.**

I. Introduction

IN MOST existing robotic manipulators, maximizing stiffness to minimize vibration and achieve good position accuracy of robotic manipulators is a key element in their design. This high stiffness is achieved by using heavy material and a bulky design. Hence, the existing heavy rigid manipulators are shown to be inefficient in terms of power consumption and operational speed. In order to improve industrial productivity, reducing the weight of the arms and increasing their speed of operation are required. Therefore, flexible-joint manipulators have received a thorough attention lately, thanks to their lightweight, lower cost, larger work volume, better maneuverability, higher operational speed, power efficiency, and larger number of applications. However, controlling such systems still faces numerous challenges that need to be addressed before they can be used in abundance in everyday real-life applications. The severe nonlinearities, coupling stemming from the manipulator's flexibility, varying operating conditions, structured and unstructured dynamical uncertainties, and external disturbances, are among the typical challenges to be faced with when dealing with such often ill-defined systems.

Several studies have shown the destabilizing effect of flexibility and nonlinear friction on many control systems for high quality servomechanisms, which can lead to severe tracking errors and limit cycles, chattering, and excessive noise [1]–[5]. Flexible-joint manipulators are governed by complex dynamics [5]–[8] and hence are inevitably subject to the ubiquitous presence of high, particularly unstructured, modeling nonlinearities, such as Coulomb friction and external disturbances, for instance. Thus, modeling the system's dynamics based on presumably accurate mathematical models might lead to severe consequences in this case. This raises the urgency to consider alternative approaches for the control of this type of manipulator systems to keep up with their increasingly demanding design requirements. Many control laws that have been proposed for flexible joints, including robust and adaptive control laws, using techniques such as singular perturbations and energy methods [9], [10] generally consider (structured) parametric uncertainties only. Several models and compensation schemes have been proposed, and adaptive control techniques [9] have been regarded among the most rewarding research avenues for such type of problems. However, not much has been achieved yet for systems that exhibit both flexibility and severe nonlinearities [11], and generally, only parametric uncertainties were considered [12], [13].

The presence of high, particularly unstructured, nonlinearities [14], [15] such as in the form of Coulomb friction on a manipulator driven through a flexible joint significantly changes the system's dynamics, as opposed to when the load is driven with a rigid joint [7], [10], [16], [17], and [19]. In this case, solving the inverse dynamics of the system is not realizable since the discontinuity in the motor position would be needed to exactly track the manipulator's desired position while the motor position is not uniquely defined at standstill. This last condition also illustrates that the motor state cannot be observed continuously from the load output. Henceforth, only an approximate inverse model can be realized. Thus, modeling the system's dynamics based on presumably accurate mathematical models cannot be applied efficiently in this case.

Over the years, researchers attempted various techniques to control flexible-joint manipulators [19]. De Luca *et al.* [20] and Khorasani [21] proposed feedback-linearization-based controllers that depend on joint acceleration and jerk signal

Manuscript received November 14, 2008; revised March 15, 2009. First published June 10, 2009; current version published July 24, 2009.

H. Chaoui is with the Machine Intelligence, Robotics, and Mechatronics Laboratory, School of Information Technology and Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada, and also with Envitech Energy Inc., Pointe-Claire, QC H9R 5P9, Canada (e-mail: h.chaoui@uOttawa.ca).

P. Sicard is with the Research Group on Industrial Electronics, Department of Electrical and Computer Engineering, Université du Québec à Trois-Rivières, Trois-Rivières, QC G9A 5H7, Canada (e-mail: pierre.sicard@uqtr.ca).

W. Gueaieb is with the Machine Intelligence, Robotics, and Mechatronics Laboratory, School of Information Technology and Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: wgueaieb@site.uOttawa.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2009.2024657

measurements. These signals are excessively noisy and, hence, unreliable in most real-life robotic systems. Even as such, the suggested controllers require the full *a priori* knowledge of the system's dynamics. To relax this requirement, several adaptive control schemes were introduced [22]–[24]. Most of these control techniques capitalize on the singular perturbation theory to extend the adaptive control theory developed for rigid bodies to flexible bodies [25]–[28]. Although many of these controllers are shown to be quite performant in theory, they failed to address important issues that might stand against their practical implementation, like basing the control laws on joint torques and their derivative [29], [30], which are known to be extremely noisy in real-life applications. In addition, such type of control algorithms approximates the system's dynamics using continuous online estimation of a set of the plant's physical parameters through well-defined adaptation laws. For it to provide a satisfactory performance, a typical adaptive control algorithm assumes that the dynamic model is perfectly known and free of significant external (unmodeled) disturbances. In other words, the controller is only robust to parametric or structured (also called modeled) uncertainties and possibly to minor unstructured uncertainties. Moreover, the unknown physical parameters must have a constant or slowly varying nominal values. An explicit linear parameterization of the uncertain dynamic parameters must also exist, and even if it does, it might not be trivial to derive, particularly with complex dynamic systems. Although the latter condition is guaranteed for robotic systems, it might not be the case for many other dynamic models. Although some conventional adaptive control techniques, proposed in the literature, did indeed tackle external disturbance attenuation, in addition to the compensation for parametric uncertainties, they rarely take into consideration the effects of modeling uncertainties [22].

On another aspect, tools of computational intelligence, such as artificial neural networks (ANNs) and fuzzy logic controllers, have been credited in various applications as powerful tools capable of providing robust controllers for mathematically ill-defined systems that may be subjected to structured and unstructured uncertainties [31]–[33]. The universal approximation theorem has been the main driving force behind the increasing popularity of such methods, as it shows that they are theoretically capable of uniformly approximating any continuous real function to any degree of accuracy. This has led to recent advances in the area of intelligent control [34], [35]. Various neural network models have been applied in the control of flexible-joint manipulators, which have led a satisfactory performance [36]. Chaoui *et al.* [37], [38] used a control approach inspired by sliding mode control that learns the system's dynamics through a feedforward neural network. A time-delay neuro-fuzzy network was suggested in [39]. In this scheme, a linear observer was used to estimate the joint velocity signals and eliminated the need to measure them explicitly. Subudhi and Morris [40] presented a hybrid architecture composed of a neural network to control the slow dynamic subsystem and an H_∞ to control the fast subsystem. A feedback linearization technique using a Takagi–Sugeno neuro-fuzzy engine was adopted in [41]. However, despite the success witnessed by neural-network-based control systems, they remain incapable of incorporating any humanlike expertise already acquired about the dynamics of the system in hand, which is considered as one of the main weaknesses of such soft computing methodologies.

In [42], de Wit proposed a robust control scheme for friction overcompensation due to uncertainties in friction models. Spong [43] used a singular perturbation technique to reduce a flexible-joint manipulator model to a standard rigid manipulator model as the joint stiffness tends to infinity. This model has been widely used by many researchers to achieve better tracking performance. For example, Ghorbel *et al.* [9] used a rigid manipulator's conventional method as a slow controller and a fast feedback control law to damp out the oscillations of the joint flexibility modes. In a similar approach, Khorasani [44] illustrated how standard adaptive control schemes for rigid robots may be generalized for flexible-joint manipulators under a certain set of assumptions.

The contribution of this paper is to propose an ANN-based control strategy, taking advantage of the learning and approximation capabilities of ANNs to approximate the system's dynamics. This scheme consists of feedforward and feedback neural-network-based adaptive controllers to form a model reference adaptive controller. The reference model is implemented in a similar way as a sliding hyperplane in variable structure control, and its output, which can be interpreted as a filtered error signal, is used as an error signal to adapt the weights of the feedback controller ANN_{FBK}. It consists of a first order model which defines the desired dynamics of the error between the desired and actual load positions and between motor and load velocity to assure internal stability. The feedforward ANN (ANN_{FF}) provides an approximate inverse model for the positioning system, while ANN_{FBK} corrects its residual errors, ensuring the manipulator's internal stability as well as a fast response of the controller. The weakness of this structure is that the feedback's learning rate depends on the load inertia. Henceforth, a supervisor is also proposed to adapt the learning rate of the ANNs to increase the stability region of the neural network controllers. The supervisor also improves the convergence properties of the adaptation process. To enlarge the stability region of the control structure, which was reduced by closing a control loop on the feedforward controller (its adaptation process) and to reduce the possibility of causing instabilities induced by parameter variation, we also propose a rule-based supervisor to adapt the parameters of the reference model. More precisely, we propose a strategy that incorporates a fuzzy controller to adapt the parameter ψ_i that represents the desired bandwidth of the closed loop system and a controller based on the frequency response of the load and motor velocities to adapt the parameter λ_i that trades off the output tracking performance for internal stability. The rest of this paper is organized as follows: Section II outlines the dynamical model of a flexible-joint manipulator. In Section III, we introduce a number of soft computing-based controllers, highlighting their advantages and relevance to the current work. The design of the proposed controller is detailed in Section IV. In Section VI, simulation results are reported and discussed. We conclude with a few remarks and suggestions for further studies pertaining to this important, yet complex, control problem.

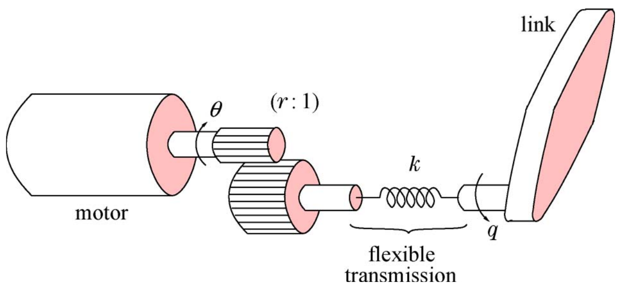

Fig. 1. Flexible-joint model.

II. Flexible-Joint Manipulator Dynamics

A. Flexible-Joint Manipulator Modeling

Typically, a flexible joint can be modeled as shown in Fig. 1. The actuator is coupled to a flexible transmission through an $r:1$ reduction gear. The transmission is directly linked to the load (e.g., manipulator link).

Consider a robot manipulator with n revolute flexible joints. Using Euler–Lagrange formulation and neglecting gyroscopic effects, the dynamic equations of the manipulator can be written as [22]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau_t - \tau_{fl} - \tau_{dl} \tag{1a}$$

$$J_m\ddot{\theta} = \tau_m - \frac{1}{r}\tau_t - \tau_{fm} - \tau_{dm} \tag{1b}$$

$$\tau_t = K\left(\frac{\theta}{r} - q\right) \tag{1c}$$

where

$q \in \mathbb{R}^n$	vector of links' positions;
$\theta \in \mathbb{R}^n$	vector of motors' positions;
$M(q) \in \mathbb{R}^{n\times n}$	manipulator's positive definite inertial matrix;
$C(q,\dot{q}) \in \mathbb{R}^{n\times n}$	matrix of Coriolis and centrifugal terms;
$G(q) \in \mathbb{R}^n$	vector of gravitational torques;
$J_m \in \mathbb{R}^{n\times n}$	motors' diagonal inertial matrix;
$\tau_t \in \mathbb{R}^n$	vector of transmission torques;
$\tau_m \in \mathbb{R}^n$	motors' generalized torque vector (control input);
$\tau_{fl} \in \mathbb{R}^n$	load friction vector;
$\tau_{fm} \in \mathbb{R}^n$	motors' friction vector;
$\tau_{dl} \in \mathbb{R}^n$	load's unmodeled dynamics and external disturbance vector;
$\tau_{dm} \in \mathbb{R}^n$	motors' unmodeled dynamics and external disturbance vector;
$K \in \mathbb{R}^{n\times n}$	diagonal matrix of joints' stiffness coefficients;
$r \in \mathbb{R}$	gear ratio.

Given the desired trajectories q_d and $\dot{q}_d$, we aim to design a control law τ_m which ensures that the manipulator's position q and velocity $\dot{q}$ track their desired trajectories under unknown dynamics and in the presence of external disturbances. The proposed controller uses q, $\dot{q}$, and $\dot{\theta}$ as system's measurable states, and the manipulator's parameters $M(q)$, $C(q,\dot{q})$, $G(q)$, J_m, τ_{fl}, τ_{fm}, τ_{dl}, and τ_{dm} are assumed to be unknown.

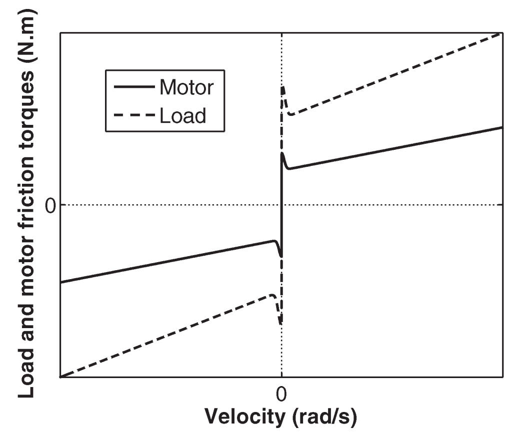

Fig. 2. Stribeck friction model.

The dynamics of a robotic manipulator is characterized by the following properties.

Property 1: The inertia matrix $M(q)$ is characterized by the following properties:

1) positive definite symmetric, i.e., $M^{\mathrm{T}}(q) = M(q)$ and $x^{\mathrm{T}}M(q)x > 0$ for any nonnull vector x;
2) upper and lower bounded, i.e., there exist two scalars $\alpha_1(q)$ and $\alpha_2(q)$ such that $\alpha_1(q)I \leq M(q) \leq \alpha_2(q)I$, where I is the identity matrix.

Property 2: The Coriolis and centripetal term $C(q,\dot{q})$ has the following properties.

1) Matrix $\dot{M}(q) - 2C(q,\dot{q})$ is skew symmetric, i.e.,

$$x^{\mathrm{T}}\left(\dot{M}(q) - 2C(q,\dot{q})\right)x = 0 \qquad \forall\, x \in \mathbb{R}^n.$$

2) $C(q,\dot{q})\dot{q}$ is quadratic in $\dot{q}$ and bounded, i.e., there exists a scalar $\alpha_3(q)$ such that $\|C(q,\dot{q})\dot{q}\| \leq \alpha_3(q)\|\dot{q}\|^2$.

Property 3: The gravity vector $G(q)$ is bounded, i.e., $\|G(q)\| \leq \alpha_4(q)$, for a scalar $\alpha_4(q)$.

Before we proceed further, we introduce the following realistic assumption.

Assumption 1: The norm of the unknown disturbance τ_d is upper bounded by a scalar b_d, i.e., $\|\tau_d\| \leq b_d$.

B. Friction Modeling

The complexity of system (1) is increased even further by adopting a highly nonlinear *a priori* unknown friction model that is composed of Coulomb, viscous, and static friction terms [4]. The model of such memoryless friction operating along a displacement x is described by

$$\tau_F = F_c\,\mathrm{sign}(\dot{x}) + F_v\dot{x} + F_s\,\mathrm{sign}(\dot{x})e^{-(\dot{x}/\eta_s)^2} \tag{2}$$

where F_c, F_v, and F_s are the Coulomb, viscous, and static friction parameters, respectively, and η_s is the rate of decay of the static friction term. This friction model is used to represent friction at each joint of the manipulator (Fig. 2).

Remark 1: It is important to point out that, although such a friction model results in a drastic increase in the system's nonlinear complexity, it generally provides a more accurate representation of the system's dynamics [5].

In a single stage speed reduction system, the assumed lumped flexibility model can be used if the dominant flexion appears in the gear teeth. In this case, the input gear's inertia is added to the motor's inertia J_m, while the output gear's inertia is added to the load inertia $M(q)$. As shown in [38] and the references therein, this model reduction method has been used in multistage reduction systems, such as planetary gears, and in multimass flexibility models, like harmonic drives.

III. Soft-Computing-Based Control

In spite of the recent advances in the area of nonlinear control systems, the common point still shared by the vast majority of conventional control techniques is their dependence on precise mathematical models of the systems to be controlled for them to provide satisfactory performance. In real life, and due to the typical high nonlinearities within the dynamics of flexible-joint manipulators, deriving a precise model for such systems could be a difficult undertaking. Although conventional adaptive control strategies compensate for the system's parametric uncertainties, they are still vulnerable in the face of unstructured modeling uncertainties. Expert controllers based on tools of soft computing, on the other hand, may not have such a limitation. In fact, computational intelligence tools, in general, have been credited in a number of applications to provide satisfactory results in the face of relatively large magnitudes of noise in the input signals, of dynamically variable parameters, and in the lack of a precise mathematical model of the system in hand [45], [46]. The computational intelligence tools that we are concerned with in this paper are ANNs and fuzzy logic systems (FLSs).

ANNs represent an important class of numerical learning tools known as connectionist modeling. An ANN is a set of interconnected computational nodes in which information is processed and transferred from one node to another through the means of weighted links with the purpose of mimicking the functionality of the neurons in the human brain. ANNs are characterized by their nonlinear behavior, parallel processing, and their automatic optimization and learning capabilities. These advantages have been behind the increasing popularity of ANNs for numerical modeling and control, particularly for systems in which little is known about their dynamics and operating environments. The neural network universal approximation theorem [47] guarantees that any sufficiently smooth function can be approximated to any degree of accuracy using a single-hidden-layer ANN [46]. Although several neural-network-based controllers have been proposed in the literature, the supervised multilayer perceptron scheme is among the simplest and most popular schemes, particularly in control systems' applications. The network's learning mechanism is often carried out as to minimize the network's output error based on a user-defined feedback signal [48].

An n-input m-output FLS can be regarded as a mapping from $U = U_1 \times U_2 \times \cdots \times U_n$ into $V = V_1 \times V_2 \times \cdots \times V_m$, where $U_i \subset \mathbb{R}$, $V_j \subset \mathbb{R}$, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. The output $y = (y_1, \ldots, y_m)^{\mathrm{T}}$ of an n-input m-output FLS with a center-average defuzzifier, sum–product inference, and singleton output fuzzifier is given by

$$y_j = \frac{\sum_{l=1}^{L} \bar{y}_j^{(l)} \left(\prod_{i=1}^{n} \mu_{A_i^{(l)}}(x_i) \right)}{\sum_{l=1}^{L} \left(\prod_{i=1}^{n} \mu_{A_i^{(l)}}(x_i) \right)}, \qquad j = 1, \ldots, m \tag{3}$$

where $x = (x_1, \ldots, x_n)^{\mathrm{T}} \in U$ is the input vector of FLS, $\mu_{A_i^{(l)}}$ denotes the membership functions of the fuzzy sets $A_i^{(l)}$, $\prod$ and $\sum$ denote the fuzzy t-norm and t-conorm operations used, respectively, l is the rule index from a total of L rules, and $\bar{y}_j^{(l)}$ is the point in V_j at which $\mu_{B_j^{(l)}}$ achieves its maximum value, which is assumed to be one. In this paper, we use the "min" and "max" operators as the t-norm and t-conorm, respectively. FLS is capable of uniformly approximating any well-defined nonlinear function over a compact set U to any degree of accuracy.

Theorem 1 (Universal Approximation Theorem): For any given real continuous function g on the compact set $U \subset \mathbb{R}^n$ and arbitrary $\epsilon > 0$, there exists a function $f(x)$ in the form of (3) such that

$$\sup_{x \in U} \|g(x) - f(x)\| < \epsilon.$$

The aforementioned universal approximation theorem [47] shows the power of FLSs in approximating continuous nonlinear functions. As such, ANNs and FLSs provide a natural alternative for tackling problems usually raised when attempting to model and design controllers for complex systems with ill-defined dynamics. This is quite convenient for the problem in hand, given the high degree of complexity and the ubiquitous uncertainties involved in flexible-joint manipulators' dynamics.

IV. Control Strategy

The control strategy is based on the design of an adaptive controller that not only leads to a precise tracking of the system's nominal desired signals but also improves the motors' internal stability.

A. Reference Model

Let $\Delta q = q_d - q$ and $\Delta\theta = \theta_d - \theta$ denote the links' and motors' position errors, respectively, with θ_d being the unknown desired time-dependent motor position vector. This latter vector can theoretically be evaluated from the desired link trajectory if all the parameters of the model are known. The presence of link Coulomb friction renders impossible to define this reference in a continuous fashion. Should the motors' desired position θ_d have been available, the control strategy would be based on tracking Δq and $\Delta\theta$ to zero. Since that is not the case, we define the following compounded velocity error signal:

$$\Delta\dot{\epsilon}_r = \dot{q}_d - \left(\Lambda\dot{q} + (1 - \Lambda)\frac{1}{r}\dot{\theta} \right) \tag{4}$$

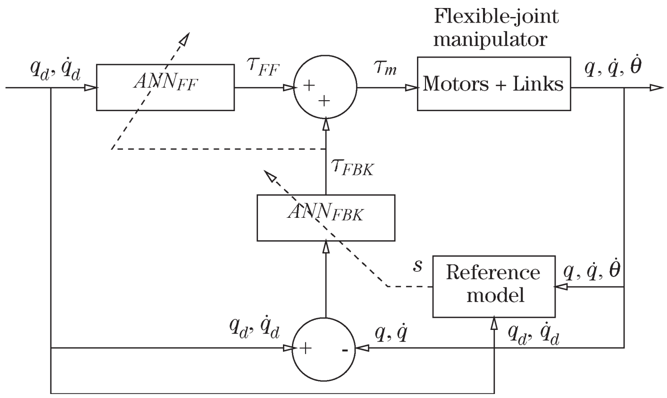

Fig. 3. Proposed control scheme.

for a diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ with $\lambda_i \in [0, 1]$, $i = 1, \ldots, n$. The feedback matrix gain Λ is introduced to provide a tradeoff between the link tracking performance and internal stability, due to the high nonlinear coupling between the two.

Define the reference model as

$$s = \Psi \Delta q + \Delta \dot{\epsilon}_r$$

where $\Psi = \mathrm{diag}(\psi_1, \psi_2, \ldots, \psi_n)$ with ψ_i being a positive constant, $i = 1, \ldots, n$, that defines the desired bandwidth of the closed loop system. A description of the justification for this form of s can be found in [38] for a similar representation.

B. Control Structure

ANNs make use of nonlinearities, learning, parallel processing, and generalization capabilities to provide interesting properties for application to advanced intelligent control. As ANNs have been proven to be universal approximators of nonlinear functions, they are very attractive for control law development and implementation, e.g., to learn the inverse dynamics of a system for inverse control. Henceforth, a feedforward signal can be obtained by building an inverse model of the positioning mechanism to compensate for the nonlinearities and flexion. Due to the presence of Coulomb friction and flexible coupling, the inverse model of the system does not exist in some operating conditions so that only an approximate model can be found. Given the designed control signals q_d and $\dot{q}_d$, a feedforward multilayer perceptron ANN (ANN_{FF}) is designed to learn the manipulator's inverse dynamics (1) online using these two inputs (Fig. 3). The system cannot follow exactly the aforementioned reference model, as it would entitle a reduction of the relative order of the system. Thus, our goal is asymptotic tracking and internal stability. Moreover, the feedforward cannot compensate for Coulomb friction since the zero crossings of the motor velocity cannot be predicted from the load's desired trajectory. Due to the iterative nature of the neural network's learning mechanism and because of the high complexity order of the system's dynamical model, the ANN_{FF} controller may take a relatively long time to converge, which may lead to unstable dynamics or unsatisfactory performance. Hence, an ANN_{FBK} is designed to correct the residual errors left from the feedforward stage and possibly slow convergence. The ANN_{FBK} takes two inputs Δq and $\Delta \dot{q}$ and provides a control action τ_{FBK} to force the link position error Δq and the compounded velocity error $\Delta \dot{\epsilon}_r$ to approach the hyperplane $\Delta q = 0$ and $\Delta \dot{\epsilon}_r = 0$. This is due to the fact that the reference model's output is used for the feedback weight adaptation so that ANN_{FBK} learns to give to the system the desired error dynamics; its output is then used, only when active, to adapt the weights of the ANN_{FF}. In the ideal case, ANN_{FF} would have learned the inverse model of the flexible joint, augmented by the reference model. The resultant control scheme is shown in Fig. 3. In the next section, we establish the online learning strategy for the neural networks. In the learning process, the objective was to have the simplest ANN structure that gives satisfying performance and guarantees the stability. Multilayer feedforward ANN was used.

C. ANN

The feedforward (ANN_{FF}) and feedback (ANN_{FBK}) are composed of three layers each: one input layer of two neurons, one hidden layer with six neurons, and one neuron for the output layer. The sigmoid function is used as an activation function for all neurons except for the output neuron which uses a linear function. To prevent limit cycles when the desired velocity is around zero, the online learning is stopped in both neural networks once the error is very small.

We can describe the relationship between the inputs and outputs of a neural network by

$$v_j^{(l)}(k) = \sum_{i=1}^{m} w_{ji}^{(l)}(k) x_i^{(l)}(k)$$

$$O_j^{(l)}(k) = \varphi_j^{(l)}\left(v_j^{(l)}(k)\right)$$

where $\varphi_j^{(l)}$ is the jth node's activation function of layer l, m is the number of layer nodes, and $x_i^{(l)}(k)$ and $O_j^{(l)}(k)$ are the input of node i and the output of node j of layer l at time index k, respectively. v_j is the net input of node j. $w_{ji}^{(l)}$ is the weight linking node i of layer $(l-1)$ to node j of layer l. Then, the ANN_{FF} and ANN_{FBK}'s outputs can be expressed as

$$\begin{aligned} y(k) &= \sum_{j=1}^{n} w_{1j}^{(2)}(k) O_j^{(1)}(k) \\ &= \sum_{j=1}^{n} w_{1j}^{(2)}(k) \varphi\left(\sum_{i=1}^{m} w_{ji}^{(1)}(k) x_i(k)\right) \end{aligned} \tag{5}$$

with m and n being the number of input and hidden nodes, respectively. The signal $y(k)$ defines the output of the neural network. The new weights matrices are computed by the rules inspired from [49]

$$w_{1j}^{(2)}(k) = \frac{y(k) + \eta e(k-1) - e(k)}{O_j^{(1)}(k-1) + \delta_1} \tag{6}$$

$$w_{ji}^{(1)}(k) = \frac{1}{m x_i(k) + \delta_2} \varphi^{-1}\left(\frac{y(k) + \eta e(k-1) - e(k)}{n w_{1j}^{(2)}(k) + \delta_3}\right) \tag{7}$$

where $0 < \eta < 1$ is the network's learning rate. For the feedforward network ANN_{FF}, the input, output, and error signals are taken as $x_i = [q_d \quad \dot{q_d}]$, $y = \tau_{\mathrm{FF}}$, and $e = \tau_{\mathrm{FBK}}$. As for the feedback network ANN_{FBK}, these signals are chosen to be $x_i = [\Delta q \quad \Delta\dot{q}]$, $y = \tau_{\mathrm{FBK}}$, and $e = s$. The constant δ_z, $z = 1, \ldots, 3$, is introduced to avoid numerical singularity problems and is set to zero when $x_i \neq 0$, $w_{1j}^{(2)} \neq 0$, and $O_j^{(1)} \neq 0$.

Theorem 2: The learning algorithm (6) and (7) with a learning rate η satisfying $0 < \eta < 1$ guarantees the asymptotic stability of both ANN_{FF} and ANN_{FBK} and the convergence of their respective errors to zero.

Proof: The proof and the weight adaptation laws are extensions of the results presented in [49] for a general multilayer perceptron learning to approximate a mapping through backpropagation.

Consider the following candidate Lyapunov function:

$$V(k) = \frac{1}{\eta} e^2(k)$$

$$\Delta V(k) = V(k) - V(k-1)$$

$$= \frac{1}{\eta} e^2(k) - \frac{1}{\eta} e^2(k-1).$$

Let $e(k) = y(k) - d(k)$, where $d(k)$ represents the desired output of the neural network at instant k

$$\Delta V(k) = \frac{1}{\eta}\left(y(k) - d(k)\right)^2 - \frac{1}{\eta} e^2(k-1).$$

Substitute $y(k)$ from (5)

$$\Delta V(k) = \frac{1}{\eta}\left(\sum_{j=1}^{n} w_{1j}^{(2)}(k)\varphi\left(\sum_{i=1}^{m} w_{ji}^{(1)}(k)x_i(k)\right) - d(k)\right)^2 - \frac{1}{\eta} e^2(k-1).$$

Substitute $w_{ji}^{(1)}(k)$ from (7)

$$\Delta V(k) = \frac{1}{\eta}\left(\sum_{j=1}^{n} w_{1j}^{(2)}(k)\varphi\left(\sum_{i=1}^{m} \frac{1}{mx_i(k)+\delta_2}\varphi^{-1}\left(\frac{y(k)+\eta e(k-1)-e(k)}{nw_{1j}^{(2)}(k)+\delta_3}\right) \times x_i(k)\right) - d(k)\right)^2 - \frac{1}{\eta} e^2(k-1).$$

Assume that $x_i(k) \neq 0$ and $w_{1j}^{(2)}(k) \neq 0$. Then, $\delta_z = 0$. Hence, the derivation of $\Delta V(k)$ is given below. Recall $e(k) = y(k) - d(k)$. Thus, $d(k) = y(k) - e(k)$

$$\Delta V(k) = \left(\eta - \frac{1}{\eta}\right) e^2(k-1) < 0.$$

If, for some k, it happened that $x_i(k) = 0$ or $w_{1j}^{(2)}(k) = 0$, then

$$\Delta V(k) = -\left(d(k)^2 + \frac{1}{\eta} e^2(k-1)\right) < 0.$$

Hence, the neural network is asymptotically stable. This also shows that $V(k)$, and thus $e(k)$, converges to a finite positive constant as $k \to \infty$. Thus, $\Delta V(k) - \Delta V(k-1)$ is bounded $\forall k \geq 0$. From Barbalat's lemma, it follows that $\lim_{k\to\infty} \Delta V(k) = 0$. $(\eta - (1/\eta)) \neq 0$ for $0 < \eta < 1$; therefore, $\lim_{k\to\infty} e(k) = 0$. ■

The learning algorithm tends to bring the output of the neural networks to the best approximation of their desired outputs. For ANN_{FBK}, s is meant to represent the best approximation available of the error on the motor torque. If s is positive, it means that the tendency of the feedback torque should be to increase (velocity is too low and/or set point position was not reached). For ANN_{FF}, the best approximation of the desired feedforward torque available is $ANN_{\mathrm{FF}}^{\mathrm{actual}} + ANN_{\mathrm{FBK}}$, i.e.,

$$\Delta V(k) = \frac{1}{\eta}\left(\sum_{j=1}^{n} w_{1j}^{(2)}(k)\varphi\left(\sum_{i=1}^{m} \frac{1}{mx_i(k)}\varphi^{-1}\left(\frac{y(k)+\eta e(k-1)-e(k)}{nw_{1j}^{(2)}(k)}\right) x_i(k)\right) - d(k)\right)^2 - \frac{1}{\eta} e^2(k-1)$$

$$\Delta V(k) = \frac{1}{\eta}\left(\sum_{j=1}^{n} w_{1j}^{(2)}(k)\varphi\left(\sum_{i=1}^{m} \frac{1}{m\cancel{x_i(k)}}\varphi^{-1}\left(\frac{y(k)+\eta e(k-1)-e(k)}{nw_{1j}^{(2)}(k)}\right) \cancel{x_i(k)}\right) - d(k)\right)^2 - \frac{1}{\eta} e^2(k-1)$$

$$\Delta V(k) = \frac{1}{\eta}\left(\sum_{j=1}^{n} w_{1j}^{(2)}(k)\cancel{\varphi}\left(\cancel{\varphi^{-1}}\left(\frac{y(k)+\eta e(k-1)-e(k)}{nw_{1j}^{(2)}(k)}\right)\right) - d(k)\right)^2 - \frac{1}{\eta} e^2(k-1)$$

$$\Delta V(k) = \frac{1}{\eta}\left(\sum_{j=1}^{n} \cancel{w_{1j}^{(2)}(k)}\left(\frac{y(k)+\eta e(k-1)-e(k)}{n\cancel{w_{1j}^{(2)}(k)}}\right) - d(k)\right)^2 - \frac{1}{\eta} e^2(k-1)$$

$$\Delta V(k) = \frac{1}{\eta}\left(y(k)+\eta e(k-1)-e(k)-d(k)\right)^2 - \frac{1}{\eta} e^2(k-1)$$

ANN_{FBK} represents the error on the feedforward torque. Again, the tendency of the learning algorithm is to make the output of the network to converge to $ANN_{\mathrm{FF}}^{\mathrm{actual}} + ANN_{\mathrm{FBK}}$.

V. Supervisory Approach

A. Learning Rate Supervisor

The cost function for the synaptic weights is

$$\xi_n = \frac{1}{2}e_n^2$$

where e_n represents the output error. The standard backpropagation algorithm using a fixed learning rate cannot handle all the error surfaces. In other words, an optimal learning rate for a given synaptic weight is not necessarily optimal for the rest of the network's synaptic weights. Henceforth, every adjustable network parameter of the cost function should have its own individual learning rate parameter. Moreover, every learning rate parameter should be allowed to vary at each iteration because the error surface typically behaves differently along different regions of a single weight dimension [48]. The current operating point in the weight space may lie on a relatively flat portion of the error surface along a particular weight dimension. In such a situation, where the derivative of the cost function with respect to that weight maintains the same algebraic sign, which means pointing in the same direction, for several consecutive iterations of the algorithm, the learning rate parameter for that particular weight should be increased. When the current operating point in the weight space lies on a portion of the error surface along a weight dimension of interest that exhibits peaks and valleys (i.e., the surface is highly irregular), then it is possible for the derivative of the cost function with respect to that weight to change its algebraic sign from one iteration to the next. To prevent the weight adjustment from oscillating, the learning rate parameter for that particular weight should be deemed down appropriately when the algebraic sign of the derivative of the cost function with respect to a particular synaptic weight alternates for several consecutive iterations of the algorithm [48]. It is noteworthy that the use of a different time-varying learning rate parameter for each synaptic weight in accordance to this approach modifies the standard backpropagation algorithm in a fundamental way.

B. Fuzzy Logic Controller

The bandwidth parameter ψ_i has an effect on the control chattering. With the flexible joint, it may have a negative effect on motor oscillations. A fuzzy controller is introduced to overcome this problem. The idea is to design a nonlinear gain ψ_i that varies with the dynamics of the system and assumes large gain values only when it is really needed based on the magnitude of parameter $s^* = \Psi_{\mathrm{nom}}\Delta q + \Delta\dot{q}$, where Ψ_{nom} is the nominal value of Ψ. The fuzzy rules were chosen heuristically (Table I) and can be refined by an expert. The main idea is based on the following three hypotheses. 1) When s^* is far from its nominal zero-valued surface, then parameter ψ_i takes a high value, giving a large weight to the error in position. 2) When s^* approaches its nominal zero-valued surface, the gain is adjusted to a smaller value for a smoother approach. 3) Once s^* is close or equal to zero, then parameter ψ_i takes a high value for a fast convergence requirement in $s = 0$. The center of area method is used for defuzzification. Scaling in Table I is chosen to obtain compatible requirements as in s, neglecting joint flexibility.

TABLE I
Adaptation Rules of ψ_i

	Δq				
$\Delta\dot{q}$	NL	NS	Z	PS	PL
PL	L	S	L	L	L
PS	Z	L	S	S	L
Z	S	Z	L	Z	S
NS	L	S	S	L	Z
NL	L	L	L	S	L

C. Tuning Algorithm

The parameter λ_i defines the degree of compromise between the link and actuator stability according to (4). This parameter should be tuned when the frequency response of the motor velocity is out of a predetermined frequency range. The critical issue is to obtain a reliable frequency response. The design strategy is that, once the frequency response is established, it is compared to the previous frequency response data. If the variation exceeds a predetermined limit, the parameter λ_i is automatically tuned; otherwise, no action is taken.

Implementing a self-tuning strategy requires the data to be continuously monitored. The frequency response is obtained from the I/O data by performing spectral analysis. The most critical component in developing a self-tuning control system is accurately computing the frequency response of a plant. The data conversion from the time domain to the frequency domain must occur in real time to allow for a self-tuning system to operate automatically. The calculation of the plant frequency response must also be accurate and not too sensitive to the system noise or the computational inaccuracy. Theoretically, the frequency response of the plant can be obtained by taking the ratio of the Fourier transform of the input and output data. Such an approach is known to suffer from poor numerical properties and is sensitive to noise. A more reliable method to determine the plant frequency response from the input and output data is to use a ratio of the cross and auto spectra of the input and output time history G_N [48] as

$$\hat{G}_N(e^{iw}) = \frac{\hat{\Phi}_{yu}^N(w)}{\hat{\Phi}_u^N(w)}$$

where $\hat{\Phi}_{yu}^N(w)$ and $\hat{\Phi}_u^N(w)$ are the cross spectra of the input and output signals and the auto spectra of the input signal, respectively.

VI. Simulation Results and Discussion

A. Experimental Setup

To demonstrate the performance of the proposed controller, a set of numerical experiments is carried out on a single

TABLE II
MANIPULATOR'S PHYSICAL PARAMETERS

Parameter	Link	Motor
rotational inertia (kg·m^2)	$I = 5.05 \cdot 10^{-2}$	$J_m = 4 \cdot 10^{-3}$
viscous friction coefficient (N·m·s/rad)	$F_{vl} = 7 \cdot 10^{-3}$	$F_{vm} = 3 \cdot 10^{-3}$
Coulomb friction coefficient (N·m)	$F_{cl} = 1 \cdot 10^{-2}$	$F_{cm} = 6 \cdot 10^{-3}$
static friction coefficient (N·m)	$F_{sl} = 5 \cdot 10^{-3}$	$F_{sm} = 2 \cdot 10^{-3}$
static friction decreasing rate (rad/s)	$\eta_{st} = 9 \cdot 10^{-2}$	$\eta_{sm} = 5 \cdot 10^{-3}$
link's mass (kg)	$m = 0.41$	
link's length (m)	$l = 0.3$	

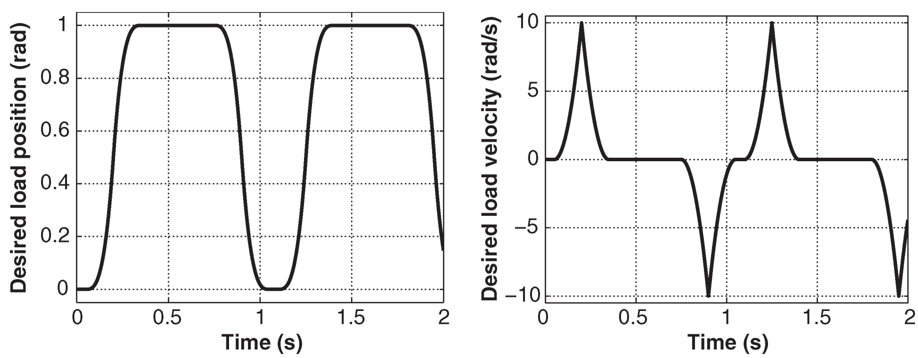

Fig. 4. Manipulator's position and velocity reference signals.

link flexible-joint manipulator. The manipulator's dynamics in terms of its physical parameters is defined by $M(q) = I$, $C(q, \dot{q}) = 0$, and $G(q) = mgl\sin(q)$, where m is the link's mass, g is the gravity constant, and l is the link's length. Table II summarizes the manipulator's physical parameters along with their respective values. The stiffness coefficient and gear ratio are assumed to be $K = 5$ N $\cdot$ m/rad and $r = 1$. The controller is set to operate at a bandwidth of 100 Hz.

B. Experimental Results

The manipulator's desired position and velocity trajectories are shown in Fig. 4.

A first experiment is carried out to study the behavior of the controller with nominal parameters. For this experiment, the system's response is studied, taking into account the desired versus actual manipulator's positions, the manipulator's position and velocity errors, the system's internal stability, the reference model output s, the adaptive parameter Ψ, and the controller's output torques, i.e., $\tau_m = \tau_{\mathrm{FF}} + \tau_{\mathrm{FBK}}$. As Fig. 5 shows, a small manipulator's position error is achieved while the system's internal stability is maintained. The feedforward torque τ_{FF} is able to provide most of the control torque signal, while the feedback controller τ_{FBK} provides only torque impulses when the motor velocity crosses zero to compensate for the highly nonlinear friction around the zero velocity neighborhood, which contributes the most to the error observed on the load trajectory. The feedforward control signal τ_{FF} alone cannot compensate for these effects since the zero crossings of the motor velocity cannot be predicted from the load's desired trajectory.

Three other experiments are carried out to study the different aspects of the controller's behavior. For each experiment, the system's response is studied, taking into account the desired versus actual manipulator's positions, the manipulator's position and velocity errors, the system's internal stability, and the controller's output torque.

In the second experiment, the effect of initial conditions on the proposed controller is studied. Hence, experiment 1 is repeated with initial conditions ($q = 0.25$ and $\theta = 0.2$ rad). Note that these initial conditions implicitly imply a nonzero initial torque τ_t at the flexible transmission [see (1)]. The controller provides a satisfactory transient response, and acceptable overshoot is obtained [Fig. 6(b)] with small activity in the motor's velocity and control torque. The envelope of the time response of the resulting position error shows that its convergence rate corresponds to the magnitude of the bandwidth parameter ψ_i in the reference model. After the transient period, the manipulator's position error remains within a negligible amplitude while maintaining the system's internal stability by making the motor's speed converge to zero. The controller is compensating for the highly nonlinear friction terms by providing a smooth control torque signal. As can be observed in Fig. 6(f), the feedback control signal is quite significant at the beginning of the simulation to compensate for the system's initial error. The controller's characteristics beyond the first unit step are similar to those of experiment 1 (Fig. 5).

The next experiment is meant to show the modularity of the proposed controller. For that, experiment 1 is run again but, this time, with twice the link's nominal inertia. Increasing the load inertia might destabilize the system and cause limit cycles. The feedforward controller shows more excitation and seems to better learn the inverse model of the system. However, this learning can be viewed as an overfitting problem which the feedforward network cannot fully cope with. The results are shown in Fig. 7. As can be seen, the controller maintains a

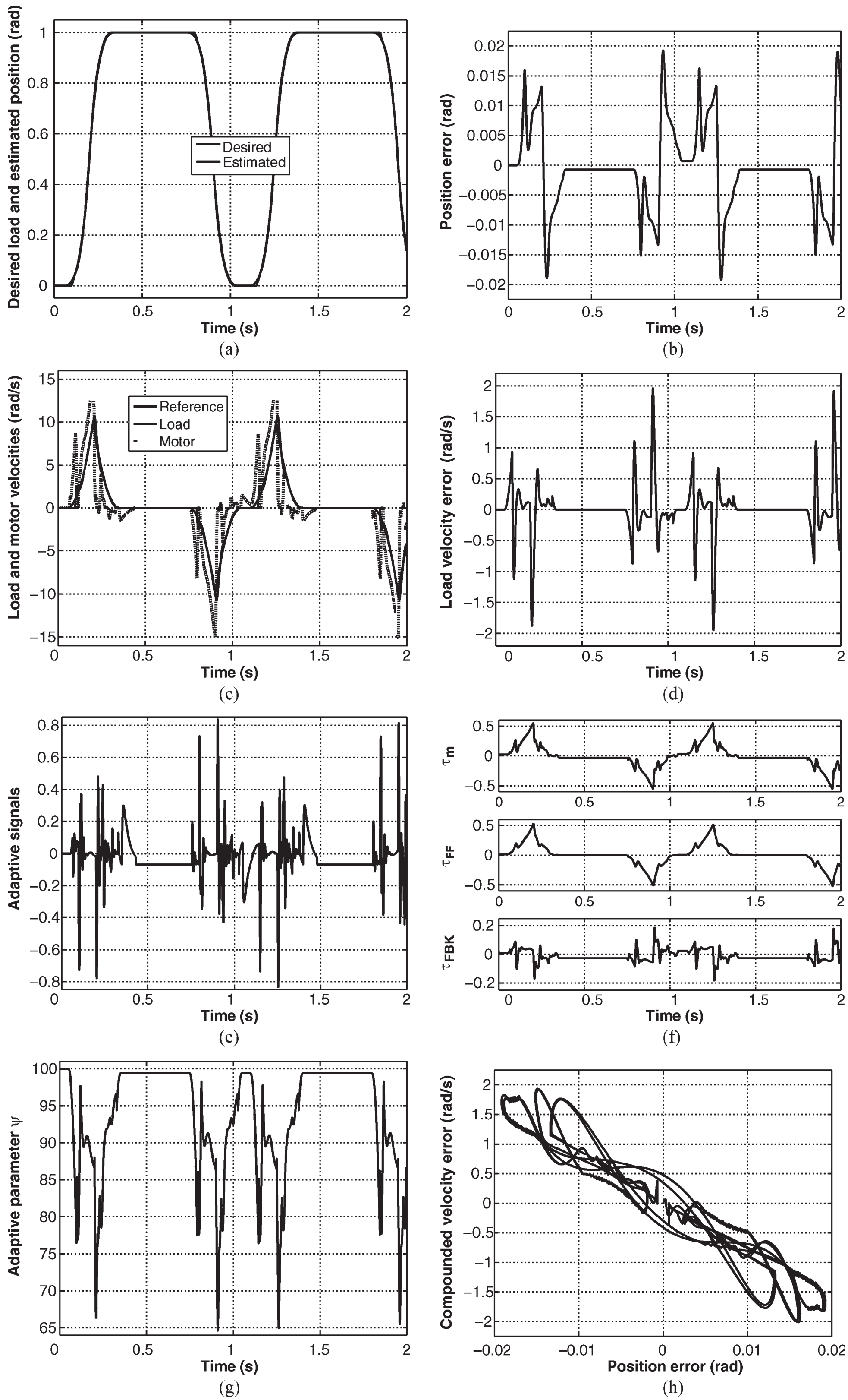

Fig. 5. Controller's response. (a) Desired and estimated manipulator's positions. (b) Position error. (c) Motor's velocity versus manipulator's velocity. (d) Manipulator's velocity error. (e) Adaptive signal s. (f) Controller's output torque $\tau_m = \tau_{\mathrm{FF}} + \tau_{\mathrm{FBK}}$. (g) Adaptive parameter Ψ. (h) Compounded velocity versus manipulator's velocity errors.

similar behavior as in experiment 1 but with slightly larger torque pulses from the feedback signal τ_{FBK}, which is expected due to the larger control effort needed to control a heavier load. It is important to note that, although the load's inertia is doubled, the manipulator's position and velocity errors remain within a small range as in experiment 1. The use of variable

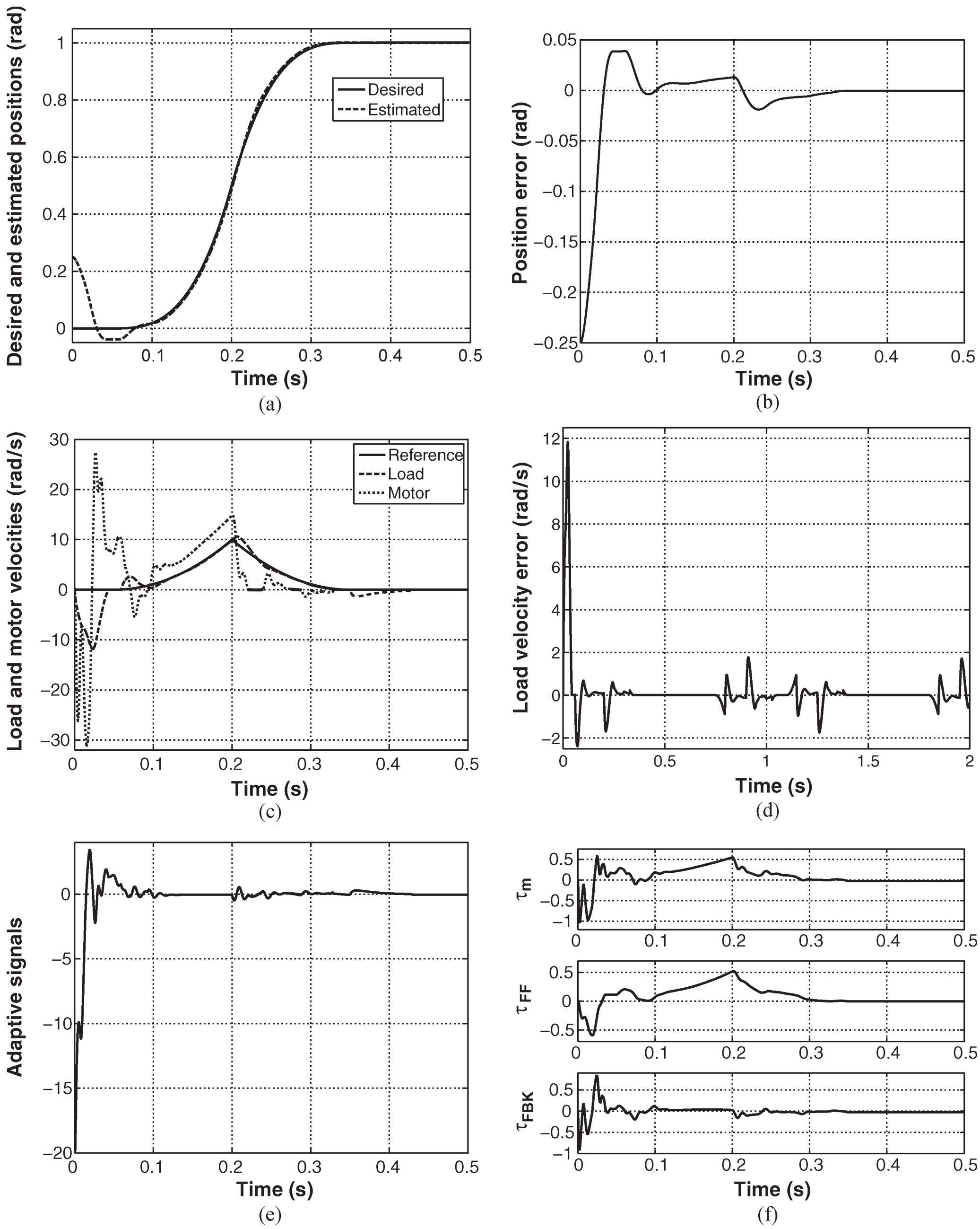

Fig. 6. Controller's response with initial conditions ($q = 0.25$ and $\theta = 0.2$ rad). (a) Desired and estimated manipulator's positions. (b) Manipulator's position error. (c) Motor's versus manipulator's velocities. (d) Manipulator's velocity error. (e) Adaptive signal s. (f) Controller's output torque $\tau_m = \tau_{\mathrm{FF}} + \tau_{\mathrm{FBK}}$.

learning rate and reference model parameters improved the robustness of the control structure, avoiding limit cycles and oscillation problems.

In the last experiment, external disturbance is introduced. Experiment 1 is thus repeated with the link being subjected to an external disturbance $\tau_{dl} = 0.1\sin(2\pi t) + 0.05$, where t is the time index. It is worth pointing out the fact that the introduced external disturbance is not dependent on the system's measurable states $(q, \dot{q}, \dot{\theta})$, and hence, it is not explicitly modeled in the design of the proposed controller. In general, external disturbance may significantly affect the precision of the positioning system and causes unacceptable high frequency oscillations on the motor's velocity and input torque. The controller's performance under such conditions is shown in Fig. 8. Again, the link's position error remains limited to the same range as in experiment 1. The role of the feedback signal in annihilating the effect of the external disturbance is clearly shown in Fig. 8(f).

VII. Conclusion

In this paper, we have developed an ANN-based control structure for flexible-joint manipulators with unknown or uncertain dynamics. The feedforward uses the neural network approximation power to approximate online the system's dynamics, while the feedback compensates for residual errors left from the forward stage. A first order reference model gives

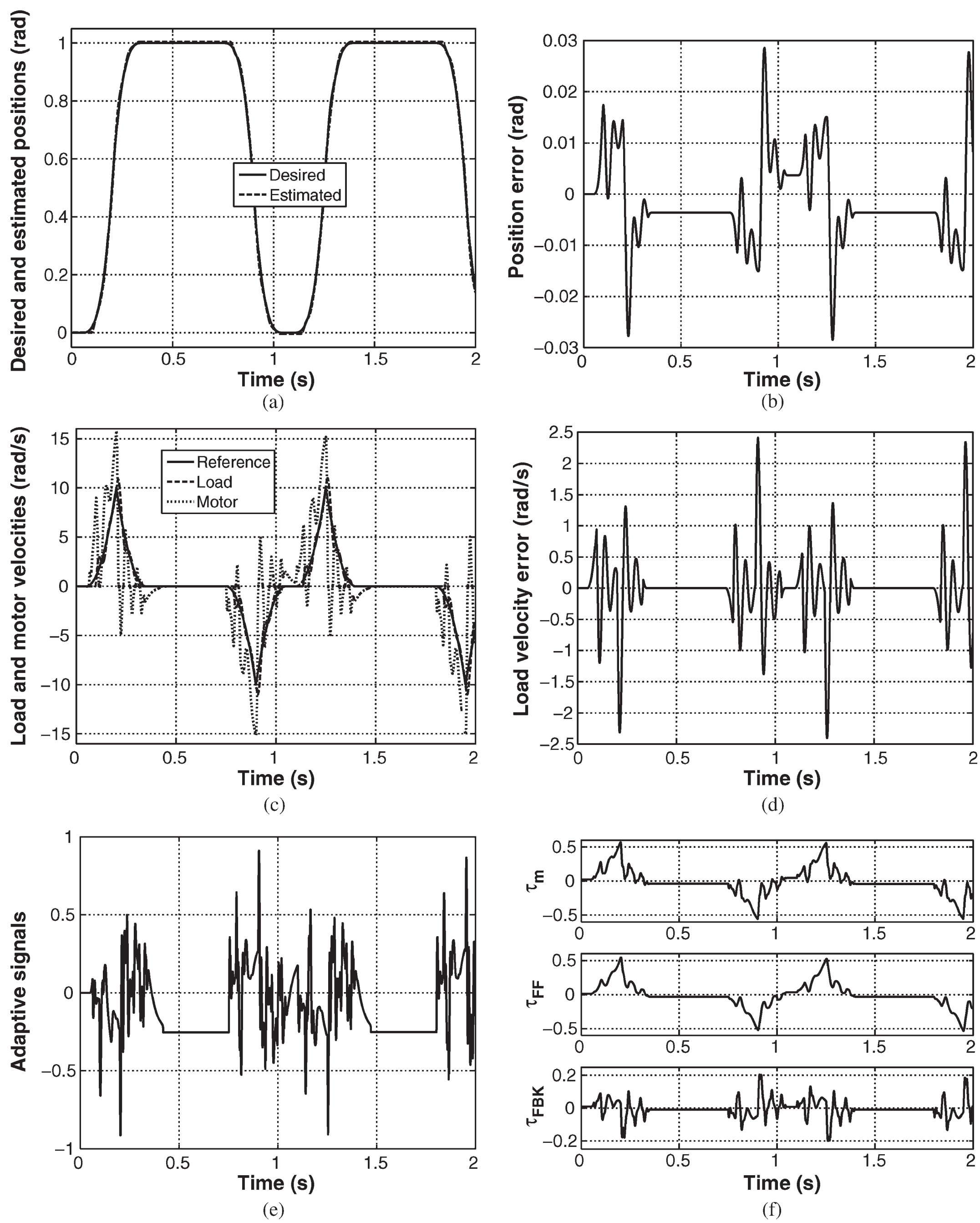

Fig. 7. Controller's response with double-the-load inertia. (a) Desired and estimated manipulator's positions. (b) Manipulator's position error. (c) Motor's versus manipulator's velocities. (d) Manipulator's velocity error. (e) Adaptive signal s. (f) Controller's output torque $\tau_m = \tau_{\mathrm{FF}} + \tau_{\mathrm{FBK}}$.

the desired dynamics of the error between the desired and actual load positions and between motor and load velocities to assure internal stability. Its output is used for the neural-network-based feedback controller weight adaptation, and the feedback's output is then used (only when active) to adapt the weights of the feedforward. It is noteworthy that the controllers and adaptation process only use load position, load velocity, and motor velocity measurements and that accelerations are not required as with many results in adaptive control literature. Measurement of load state is required to assure accuracy in positioning. Measurement of the motor velocity, while adding an extra sensor, is a common practice for geared motors to damp flexible modes.

We have implemented a supervisor to adapt the learning rate by a measure of internal stability of the learning process. We have also proposed a rule-based supervisor to adapt the reference model parameters by a fuzzy controller and a tuning algorithm based on the frequency response of the system. The supervisor approach was instrumental in obtaining good robustness of stability and performance in cases that caused instability with a fixed learning rate or with fixed reference model parameters. The performance of the resultant control strategy has been shown through numerical simulations considering different test scenarios. The results highlight the quality of compensation of the nonlinear friction terms and flexibility, and accurate tracking of the desired load trajectory. The control

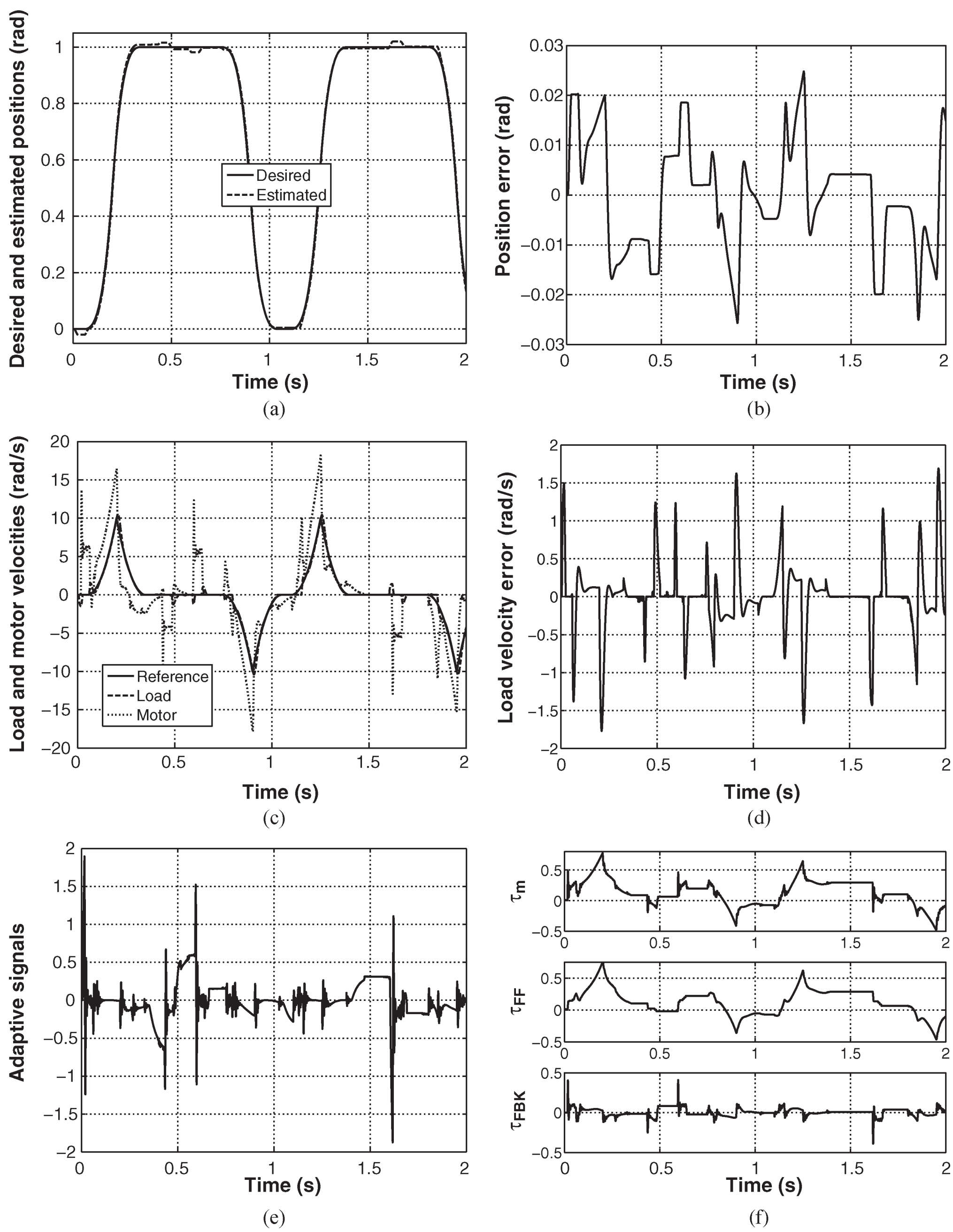

Fig. 8. Controller's response with disturbance. (a) Desired and estimated manipulator's positions. (b) Manipulator's position error. (c) Motor's versus manipulator's velocities. (d) Manipulator's velocity error. (e) Adaptive signal s. (f) Controller's output torque $\tau_m = \tau_{\mathrm{FF}} + \tau_{\mathrm{FBK}}$.

structure is a good candidate for real-time and very large scale integration (VLSI) implementation. Henceforth, this controller is suitable for DSP and VLSI implementation and can be used to improve the static and dynamic performance of electromechanical systems.

REFERENCES

[1] Y. Pan, U. Ozguner, and O. H. Dagci, "Variable-structure control of electronic throttle valve," *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3899–3907, Nov. 2008.

[2] A. Chatterjee, R. Chatterjee, F. Matsuno, and T. Endo, "Augmented stable fuzzy control for flexible robotic arm using LMI approach and neuro-fuzzy state space modeling," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1256–1270, Mar. 2008.

[3] L. Sweet and M. Good, "Redefinition of the robot motion-control problem," *IEEE Control Syst. Mag.*, vol. 5, no. 3, pp. 18–25, Aug. 1985.

[4] B. Armstrong and C. C. de Wit, "Friction modeling and compensation," in *The Control Handbook*, vol. 77. Boca Raton, FL: CRC Press, 1996, pp. 1369–1382.

[5] H. Olsson, K. Astrom, C. C. de Wit, M. Gafvert, and P. Lischinsky, "Friction models and friction compensation," *Eur. J. Control*, vol. 4, no. 3, pp. 176–195, 1998.

[6] S. Katsura and K. Ohnishi, "Force servoing by flexible manipulator based on resonance ratio control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 539–547, Feb. 2007.

[7] D. Seidl, S.-L. Lam, J. Putman, and R. Lorenz, "Neural network compensation of gear backlash hysteresis in position-controlled mechanisms," *IEEE Trans. Ind. Appl.*, vol. 31, no. 6, pp. 1475–1483, Nov./Dec. 1995.

[8] S. Katsura, J. Suzuki, and K. Ohnishi, "Pushing operation by flexible manipulator taking environmental information into account," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1688–1697, Oct. 2006.

[9] F. Ghorbel, J. Hung, and M. Spong, "Adaptive control of flexible-joint manipulators," *IEEE Control Syst. Mag.*, vol. 9, no. 7, pp. 9–13, Dec. 1989.

[10] M.-C. Chien and A.-C. Huang, "Adaptive control for flexible-joint electrically driven robot with time-varying uncertainties," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 1032–1038, Apr. 2007.

[11] A. Hace, K. Jezernik, and A. Sabanovic, "SMC with disturbance observer for a linear belt drive," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3402–3412, Dec. 2007.

[12] K. Kyoungchul, T. Masayoshi, M. Hyosang, H. Beomsoo, and J. Doyoung, "Mechanical design and impedance compensation of SUBAR (Sogang University's Biomedical Assist Robot)," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2008, pp. 377–382.

[13] J.-P. Hauschild and G. R. Heppler, "Control of harmonic drive motor actuated flexible linkages," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3451–3456.

[14] R. Martinez, J. Alvarez, and Y. Orlov, "Hybrid sliding-mode-based control of underactuated systems with dry friction," *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3998–4003, Nov. 2008.

[15] F.-J. Lin, Y.-C. Hung, and S.-Y. Chen, "FPGA-based computed force control system using Elman neural network for linear ultrasonic motor," *IEEE Trans. Ind. Electron.*, vol. 56, no. 4, pp. 1238–1253, Apr. 2009.

[16] A. Merabet and J. Gu, "Robust nonlinear predictive control with modeling uncertainties and unknown disturbance for single-link flexible joint robot," in *Proc. 7th World Congr. Intell. Control Autom.*, Jun. 2008, pp. 1516–1521.

[17] J. S. Yeon and J. H. Park, "Practical robust control for flexible joint robot manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 3377–3382.

[18] D. Seidl, T. Reineking, and R. Lorenz, "Neural network compensation of gear backlash hysteresis in position-controlled mechanisms," in *Conf. Rec. IEEE IAS Annu. Meeting*, Oct. 4–9, 1992, vol. 2, pp. 1937–1944.

[19] M. Benosman and G. L. Vey, "Control of flexible manipulators: A survey," *Robotica*, vol. 22, no. 5, pp. 533–545, Oct. 2004.

[20] A. D. Luca, A. Isidori, and F. Nicolo, "Control of robot arm with elastic joints via nonlinear dynamic feedback," in *Proc. IEEE Conf. Decision Control, Symp. Adaptive Pro*, Fort Lauderdale, FL, 1985, pp. 1671–1679.

[21] K. Khorasani, "Nonlinear feedback control of flexible joint manipulators: A single link case study," *IEEE Trans. Autom. Control*, vol. 35, no. 10, pp. 1145–1149, Oct. 1990.

[22] C. Ott, A. Albu-Schaffer, and G. Hirzinger, "Comparison of adaptive and nonadaptive tracking control laws for a flexible joint manipulator," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, Sep. 2002, vol. 2, pp. 2018–2024.

[23] R. Al-Ashoor, R. Patel, and K. Khorasani, "Robust adaptive controller design and stability analysis for flexible-joint manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 2, pp. 589–602, Mar./Apr. 1993.

[24] F. Ghorbel and M. W. Spong, "Adaptive integral manifold control of flexible joint robot manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nice, France, Apr. 1992, vol. 1, pp. 707–714.

[25] M. W. Spong, "Modeling and control of elastic joint robots," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 109, no. 4, pp. 310–319, Dec. 1987.

[26] S. S. Ge and I. Postlethwaite, "Adaptive neural network controller design for flexible joint robots using singular perturbation technique," *Trans. Inst. Meas. Control*, vol. 17, no. 3, pp. 120–131, 1995.

[27] H. Taghirad and M. Khosravi, "Design and simulation of robust composite controllers for flexible joint robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, Sep. 2003, vol. 3, pp. 3108–3113.

[28] L. Huang, S. Ge, and T. Lee, "Adaptive position/force control of an uncertain constrained flexible joint robots—Singular perturbation approach," in *Proc. SICE Annu. Conf.*, Sapporo, Japan, Aug. 2004, pp. 1693–1698.

[29] C. Ott, A. Albu-Schaffer, A. Kugi, S. Stramigioli, and G. Hirzinger, "A passivity based Cartesian impedance controller for flexible joint robots—Part I: Torque feedback and gravity compensation," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, Apr. 2004, vol. 3, pp. 2659–2665.

[30] L. Tian and A. Goldenberg, "Robust adaptive control of flexible joint robots with joint torque feedback," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nagoya, Japan, May 1995, vol. 1, pp. 1229–1234.

[31] H. Chaoui and W. Gueaieb, "Type-2 fuzzy logic control of a flexible-joint manipulator," *J. Intell. Robot. Syst.*, vol. 51, no. 2, pp. 159–186, Feb. 2008.

[32] F. Karray, W. Gueaieb, and S. Al-Sharhan, "The hierarchical expert tuning of PID controllers using tools of soft computing," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 1, pp. 77–90, Feb. 2002.

[33] C. W. de Silva, *Intelligent Control Fuzzy Logic Applications*. Boca Raton, FL: CRC Press, 1995.

[34] E. Kim, "Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 3, pp. 368–378, Jun. 2004.

[35] W. Gueaieb, F. Karray, and S. Al-Sharhan, "A robust adaptive fuzzy position/force control scheme for cooperative manipulators," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 4, pp. 516–528, Jul. 2003.

[36] H. Chaoui, W. Gueaieb, M. Yagoub, and P. Sicard, "Hybrid neural fuzzy sliding mode control of flexible-joint manipulators with unknown dynamics," in *Proc. 32nd Annu. IEEE IECON*, Paris, France, Nov. 7–10, 2006, pp. 4082–4087.

[37] H. Chaoui, P. Sicard, and A. Lakhsasi, "Reference model supervisory loop for neural network based adaptive control of a flexible joint with hard nonlinearities," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, Niagara Falls, ON, Canada, May 2004, vol. 4, pp. 2029–2034.

[38] H. Chaoui, P. Sicard, A. Lakhsasi, and H. Schwartz, "Neural network based model reference adaptive control structure for a flexible joint with hard nonlinearities," in *Proc. IEEE Int. Symp. Ind. Electron.*, Ajaccio, France, May 2004, vol. 1, pp. 271–276.

[39] D. Hui, S. Fuchun, and S. Zengqi, "Observer-based adaptive controller design of flexible manipulators using time-delay neuro-fuzzy networks," *J. Intell. Robot. Syst., Theory Appl.*, vol. 34, no. 4, pp. 453–466, Aug. 2002.

[40] B. Subudhi and A. Morris, "Singular perturbation based neuro-H infinity control scheme for a manipulator with flexible links and joints," *Robotica*, vol. 24, no. 2, pp. 151–161, Mar. 2006.

[41] C.-W. Park, "Robust stable fuzzy control via fuzzy modeling and feedback linearization with its applications to controlling uncertain single-link flexible joint manipulators," *J. Intell. Robot. Syst., Theory Appl.*, vol. 39, no. 2, pp. 131–147, Feb. 2004.

[42] C. de Wit, "Robust control for servo-mechanisms under inexact friction compensation," *Automatica*, vol. 29, no. 3, pp. 757–761, May 1993.

[43] M. Spong, "Modeling and control of elastic joint robots," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 109, pp. 310–319, Dec. 1987.

[44] K. Khorasani, "Adaptive control of flexible-joint robots," *IEEE Trans. Robot. Autom.*, vol. 8, no. 2, pp. 250–267, Apr. 1992.

[45] F. Karray and C. W. de Silva, *Soft Computing and Intelligent Systems Design, Theory, Tools and Applications*. Essex, U.K.: Addison-Wesley, 2004. [Online]. Available: http://pami.uwaterloo.ca/soft_comp/textbook.html

[46] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[47] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[48] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[49] Z. Man, H. R. Wu, S. Liu, and X. Yu, "A new adaptive backpropagation algorithm based on Lyapunov stability theory for neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1580–1591, Nov. 2006.

Hicham Chaoui (S'01) received the B.Sc. degree in electrical engineering from the Institut Supérieur du Génie Appliqué, Casablanca, Morocco, in 1999, the M.A.Sc. degree in electrical engineering from the Université du Québec à Trois-Rivières, Trois-Rivières, QC, Canada, in 2002, and the M.Sc. degree in computer science and the Project Management Institute accredited graduate degree "Diplôme d'Études Supérieures Spécialisées" in project management from the Université du Québec en Outaouais, Gatineau, QC, in 2004 and 2007, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering in the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada, where he is a member of the Machine Intelligence, Robotics, and Mechatronics Research Group.

Since 2007, he has been with Envitech Energy Inc., Pointe-Claire, QC, where he is currently a Project Leader. His research interests include modeling and control of nonlinear dynamic systems, friction identification and compensation, adaptive control, neural networks, fuzzy systems, robotics, power electronics, real-time embedded systems, and FPGA implementation.

Mr. Chaoui is a member of the Ordre des Ingénieurs du Québec.

Pierre Sicard (S'84–M'85) received the B.S. degree in technology of electricity from the École de Technologie Supérieure, Montréal, QC, Canada, in 1985, the M.S. degree in industrial electronics from the University du Québec à Trois-Rivières, Trois-Rivières, QC, in 1990, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1993.

In 1992, he joined the Université du Québec à Trois-Rivières as a Professor on electrical and computer engineering, where he held the Hydro-Québec Research Chair on Power and Electrical Energy from 1999 to 2003 and he has been the Director of the Research Group on Industrial Electronics, Department of Electrical and Computer Engineering, since 2004. His current research interests include power quality, modeling, controller and observer design for nonlinear systems, control of power electronics and multidrive systems, passivity-based control, adaptive control, and neural networks.

Wail Gueaieb (M'04–SM'06) received the B.S. and M.S. degrees in computer engineering and information science from Bilkent University, Ankara, Turkey, in 1995 and 1997, respectively, and the Ph.D. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 2001.

He is currently an Associate Professor in the School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, ON, Canada. He is also the Founder and Director of the Machine Intelligence, Robotics, and Mechatronics Laboratory, SITE. His research interests span the fields of intelligent mechatronics, robotics, and computational intelligence. He worked in industry from 2001 to 2004, where he contributed to the design and implementation of a new generation of smart automotive safety systems. He is the author/coauthor of four patents and more than 60 articles in highly reputed international journals and conference proceedings.