15th Triennial World Congress, Barcelona, Spain

A NEURAL-NETWORK BASED OBSERVER FOR FLEXIBLE-JOINT MANIPULATORS [1]

H.A. Talebi * **R.V. Patel** ** **M. Wong** **

* *Dept. of Electrical Engineering, AmirKabir University, Tehran, Iran 15914*
** *Dept. of Electrical and Computer Engineering, University of Western Ontario, London, Ontario, Canada N6A 5B9*

Abstract: The problem of designing a nonlinear observer for flexible-joint manipulators using a neural network approach is considered in this paper. In the first instance, no *a priori* knowledge about the system dynamics is assumed in developing the basic structure of the neural observer. The recurrent neural network configuration is obtained by a combination of a multilayer feedforward network and dynamical elements in the form of stable filters. Next, partial knowledge about the manipulator dynamics is assumed. However, a model of the joint stiffness, stiction, and friction is assumed to be unknown. This modification greatly simplifies the original design and facilitates its real-time implementation. This scheme does not need any measurement from the output shaft of the manipulator. The neural networks are trained online. Simulation results for single and two-link manipulators are presented to demonstrate the effectiveness of the proposed approach.

Keywords: Nonlinear Observer, Neural Networks, State Estimation, Flexible-Joint Manipulators

1. INTRODUCTION

The flexibility of robot motor transmissions (joint flexibility) is often the dominant source of compliance in a robotic system. For example, in relative terms, a harmonic drive transmission is much more flexible than a conventional gear transmission. The unconventional gear-tooth meshing action of the harmonic drive, makes it possible to acquire higher gear ratios and high torque capabilities in a compact geometry. However, the flexibility of the joint causes difficulty in modeling manipulator dynamics and becomes a potential source of uncertainty that can degrade the performance of the manipulator and in some cases can even destabilize the system (Goldsmith *et al.*, 1999). Consequently, addressing this issue is essential for calibration as well as modeling and control of robot manipulators.

To compensate for joint flexibility, many sophisticated control algorithms have been proposed both in constrained (Spong, 1989; Ge and Woon, 1998) and unconstrained motions (Spong, 1987; Chang and Daniel, 1992; Zeman *et al.*, 1997; Hung, 1991; Al-Ashoor *et al.*, 1993). Most of these schemes assume the availability of both the link and the motor positions, a condition that may not always be satisfied or practically feasible. A Luenberger observer, a reduced-order high-gain observer, and Kalman filter based observer were reported to relax the requirement of measurements from both sides of the transmission device (Jankovic, 1995; Nicosia *et al.*, 1988; Nicosia *et al.*, 1989; Nicosia and Tornambe, 1989; Tomei, 1990; Jankovic, 1992). However, a fundamental assumption underlying all of these methods is that the system nonlinearities are completely known *a priori*.

[1] This research was supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada grant RGPIN-1345.

The nonlinear mapping properties of neural networks, their adaptive nature, and their ability to deal with uncertainty make them a powerful tool for the control and identification of nonlinear systems. In (Elanayar and Shin, 1994), a state estimator has been designed using a Radial Basis Function (RBF) neural network. The problem of state estimation for a class of discrete–time nonlinear systems was considered in (Levin and Narendra, 1996). A dynamic neural-network-based observer for a class of SISO nonlinear systems was considered in (Kim *et al.*, 1996). In (Ahmed and Riyaz, 2000), a Kalman filter based observer using neural networks was considered. The nonlinearities of the system were assumed to be known and the neural network was employed to approximate the Kalman filter gain.

In this paper, a reduced-order observer for flexible-joint robots when motor positions and velocities are available, is proposed. This choice of measured variables is most desirable. In Section 2, the dynamic model used for the simulation is defined. The proposed neural network observers are introduced in Section 3. As a first step, no *a priori* knowledge about the system dynamics is assumed. The recurrent neural network configuration is obtained by a combination of a multilayer feedforward network and dynamical elements in the form of stable filters. In the next step, partial knowledge about the manipulator dynamics is assumed, but joint stiffness, stiction, and friction are assumed to be unknown. This modification greatly simplifies the original design and facilitates its real-time implementation. This scheme does not require any measurement from the output shaft of the manipulator. Finally, in Section 4 simulation results are presented.

2. MANIPULATOR MODEL

Manipulators with harmonic drive actuators or motors with long shafts tends to have inherent joint flexibilities. The most common way to model the joint flexibility is to consider a rotational spring between the input shaft (motor) and the output shaft (link) of the manipulator (Spong and Vidyasagar, 1989; Ortega *et al.*, 1998).

$$\begin{aligned} D_l(q_1)\ddot{q}_1 + C_1(q_1,\dot{q}_1) + g(q_1) + B_1\dot{q}_1 &= \tau_s, \\ J_a\ddot{q}_2 + \tau_s + B_2\dot{q}_2 &= \tau, \end{aligned} \tag{1}$$

where $q_1 \in \Re^n$ is the vector of link positions, $q_2 \in \Re^n$ is the vector of motor shaft positions, $g(q_1) \in \Re^n$ is the gravity loading force, $C_1(q_1,\dot{q}_1) \in \Re^n$ is the term corresponding to the centrifugal and Coriolis forces, $B_1 \in \Re^{n\times n}$ and $B_2 \in \Re^{n\times n}$ are the viscous damping matrices at the output and input shafts, $D_l(q_1) \in \Re^{n\times n}$ and $J_a \in \Re^{n\times n}$ are the robot and the actuator inertia matrices respectively, and τ is the input torque.

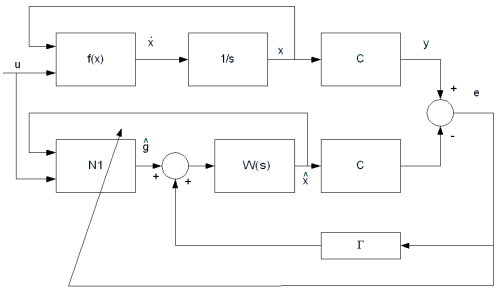

Fig. 1. Structure of the neural network observer.

The reaction torque τ_s from the rotational spring is often considered as

$$\tau_s = K(q_2 - q_1) + \beta(q_1,\dot{q}_1,q_2,\dot{q}_2) \tag{2}$$

where $K \in \Re^{n\times n}$ is the positive-definite stiffness matrix of the rotational spring attached between the input and the output shafts. In general, there is an unknown nonlinear force $\beta(q_1,\dot{q}_1,q_2,\dot{q}_2)$ which can be regarded as a combination of a nonlinear spring and the friction at the output shafts of the manipulator. The reaction torque τ_s cannot be modeled accurately and is assumed to be unknown for observer design and is included for simulation purposes only.

3. THE NEURAL NETWORK OBSERVER

Consider the nonlinear system

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= Cx \end{aligned} \tag{3}$$

where $u \in \Re^m$ is the input, $y \in \Re^m$ is the output and $x \in \Re^n$ is the state vector of the system. Now, by adding and subtracting Ax where A is a Hurwitz matrix, (3) becomes

$$\begin{aligned} \dot{x} &= Ax + g(x,u) \\ y &= Cx \end{aligned} \tag{4}$$

where $g(x,u) = f(x,u) - Ax$ (Kosmatopoulos *et al.*, 1992; Talebi *et al.*, 2000). Based on (4), a recurrent network model is constructed by parameterizing the mapping g by feedforward (static) neural network architectures, denoted by N_1. Therefore, the following model is considered for observer design.

$$\dot{\hat{x}} = A\hat{x} + \hat{g}(\hat{x},u,w) + \Gamma(y - C\hat{x}) \tag{5}$$

where w is the weight matrix of the neural network and Γ is selected such that $A - \Gamma C$ is a Hurwitz matrix (Kim *et al.*, 1996). The structure of the observer is shown in Figure 1. In this figure, $\hat{x}$ denotes the state of the observer (5). Corresponding to the Hurwitz matrix A, $W(s) := (sI - A)^{-1}$ is also shown which is an $n \times n$ matrix whose elements are stable transfer functions.

The objective function for training the neural network is selected as $J = \frac{1}{2}(e^T e)$, where $e = y - C\hat{x}$. The weight adjustment mechanism is based on the steepest descent method, namely

$$\dot{w} = -\eta(\frac{\partial J}{\partial w})^T,$$

where η is the learning rate. Note that $\frac{\partial J}{\partial w}$ is computed according to

$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial e}\frac{\partial e}{\partial w} = e^T \frac{\partial e}{\partial w}$$

Since $e = y - C\hat{x}$, we get

$$\frac{\partial J}{\partial w} = -e^T C \frac{\partial \hat{x}}{\partial w} \tag{6}$$

Now, by using static approximation of $\frac{\partial \hat{x}}{\partial w}$ as $-A^{-1}\frac{\partial \hat{g}}{\partial w}$, we may write $\frac{\partial J}{\partial w}$ as

$$\frac{\partial J}{\partial w} = (e^T C A^{-1})\frac{\partial \hat{g}}{\partial w},$$

where $\frac{\partial \hat{g}}{\partial w}$ can be computed using the backpropagation method.

3.1 *Reduced Order Observer*

For the case of a flexible-joint manipulator, there are 4 states corresponding to each joint (positions and velocities of the input and output shafts). Consequently, for a manipulator with n joints, the dynamical system is a $4nth$ order system. Hence, the neural network N_1 should have $4n$ outputs. This makes the observer computationally expensive especially if the neural network observer is to be employed and trained online. One solution to this problem is to consider partial knowledge about the dynamics of the system (basically the rigid-body model) and employ the neural network to approximate the stiffness dynamics (the reaction torque τ_s in equations (1,2)). The manipulator dynamics (1) can be rearranged as

$$M(q)\ddot{q} + C(q,\dot{q}) + \kappa\tau_s = U, \tag{7}$$

where $q = [q_1^T, q_2^T]^T$, $M(q) = \begin{bmatrix} D_l(q_1) & 0 \\ 0 & J \end{bmatrix}$, $C(q,\dot{q}) = \begin{bmatrix} C_1(q_1,\dot{q}_1) + g(q_1) + B_1\dot{q}_1 \\ B_2\dot{q}_2 \end{bmatrix}$, $U = \begin{bmatrix} 0 \\ \tau \end{bmatrix}$, $\kappa = \begin{bmatrix} -I_n \\ I_n \end{bmatrix}$, and I_n is the $n \times n$ identity matrix. Now, by defining $x = [q^T, \dot{q}^T]^T$ and following the same steps as in the previous section, equation (7) can be written as

$$\begin{aligned} \dot{x} &= Ax + g(x,u) + T_s \\ y &= Cx \end{aligned} \tag{8}$$

where, $T_s = \begin{bmatrix} 0 \\ M^{-1}\kappa\tau_s \end{bmatrix}$ and $y = [q_2^T, \dot{q}_2^T]^T$. The observer model is now given by

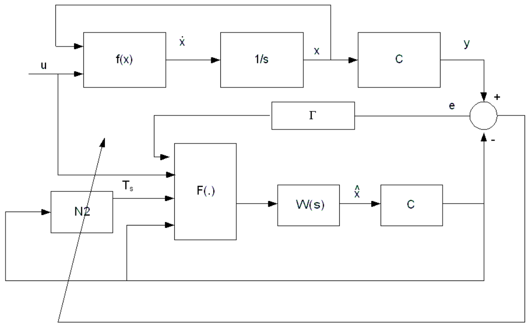

Fig. 2. Structure of the reduced-order neural network observer, where $F(.)$ is realized according to (9)

$$\dot{\hat{x}} = A\hat{x} + g(\hat{x},u) + \hat{T}_s(x,w) + \Gamma(y - C\hat{x}), \tag{9}$$

where

$$\hat{T}_s(x,w) = \begin{bmatrix} 0 \\ M^{-1}\kappa\hat{\tau}_s(\hat{x},w) \end{bmatrix} \tag{10}$$

and $\hat{\tau}_s(\hat{x},w)$ is the mapping performed by the neural network (N_2). The structure of the observer is shown in Figure 2.

The objective function for training the neural network is selected as before, namely $J = \frac{1}{2}(e^T e)$, where $e = y - C\hat{x}$. The steepest descent method is also used to adjust the weights of the neural network,

$$\dot{w} = -\eta(\frac{\partial J}{\partial w})^T,$$

Note that $\frac{\partial J}{\partial w}$ in this case is computed according to

$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial e}\frac{\partial e}{\partial w} = e^T \frac{\partial e}{\partial w}$$

Since $e = y - C\hat{x}$, we get

$$\begin{aligned} \frac{\partial J}{\partial w} &= -e^T C \frac{\partial \hat{x}}{\partial w} \\ \frac{\partial \hat{x}}{\partial w} &= \frac{\partial \hat{x}}{\partial \hat{\tau}_s}\frac{\partial \hat{\tau}_s}{\partial w} \end{aligned} \tag{11}$$

Now, by using the static approximation of $\frac{\partial \hat{x}}{\partial \hat{\tau}_s}$ as $-A^{-1}\frac{\partial \hat{T}_s}{\partial \hat{\tau}_s}$, we can write $\frac{\partial J}{\partial w}$ as

$$\frac{\partial J}{\partial w} = (e^T C A^{-1})\frac{\partial \hat{T}_s}{\partial \hat{\tau}_s}\frac{\partial \hat{\tau}_s}{\partial w},$$

where $\frac{\partial \hat{T}_s}{\partial \hat{\tau}_s}$ can be computed using (10) and $\frac{\partial \hat{\tau}_s}{\partial w}$ can be computed using the backpropagation method.

4. SIMULATION RESULTS

The performance of the proposed observer is first investigated on a single flexible-joint manipulator whose parameters are $J = 1.16kg.m^2$, $m = 1Kg$, $l = 1m$, $K = 100N/m$, where J is the motor inertia, m is

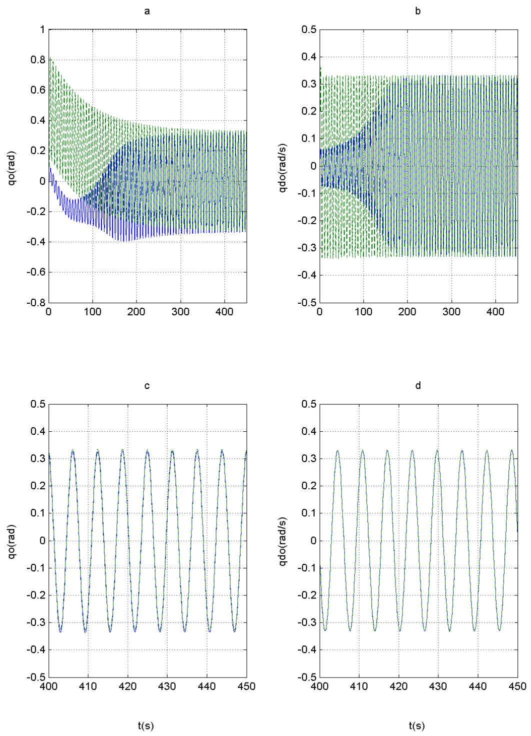

Fig. 3. The responses of the open-loop system to $\sin(t)$ reference trajectory with initial random weights for neural network: (a) link position, (b) link velocity, (c) link position after learning, (d) link velocity after learning. The dashed lines correspond to the states of the manipulator and the solid lines correspond to the states of the observer.

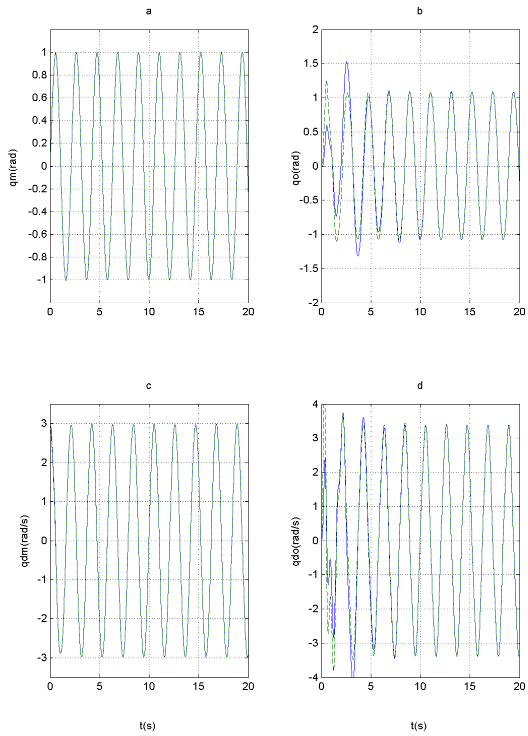

Fig. 4. The responses of the closed-loop system to $\sin(3t)$ reference trajectory: (a) motor position, (b) link position, (c) motor velocity, (d) Link velocity. The solid lines correspond to the states of the observer.

the link mass, l is the link length, and K is the stiffness of the joint. For the neural network based observer, a three-layer neural network was used with 4 neurons in the input layer, 5 neurons in the hidden layer, and 1 neuron in the output layer. The inputs to the network are $\hat{q}_1, \hat{\dot{q}}_1, \hat{q}_2$, and $\hat{\dot{q}}_2$. The hidden layer neurons have sigmoidal transfer functions and the output neurons use linear activation functions. The initial weights of the network are selected as small random numbers. when training the neural network, $\beta(q_1, \dot{q}_1, q_2, \dot{q}_2)$ in (2) is considered to be zero and later included in the model to show the ability of the neural network in dealing with uncertainties.

Figure 3 shows the simulation results for this case. Figures 3–a and 3–b show the responses of q_1 and $\dot{q}_1$ to a $sin(t)$ input signal when the learning evolves. After the learning has been completed, the responses of q_1 and $\dot{q}_1$ are shown in Figures 3–c and 3–d, respectively. As can be seen, the neural network observer has learned the system dynamics and its states accurately track the flexible-joint states. In the next simulations, the weights of the neural network are taken from the previous simulation. To evaluate the ability of the neural observer under different conditions, the frequency of the reference trajectory was changed to $3Hz$ and the control loop was also closed using a PD controller. Note that, the weights of the network are still updated in this case. The simulation results in this case are shown in Figures 4–a to 4–d. As can be seen, the neural network observer is still able to follow the states of the flexible-joint manipulator. Next, $\beta(q_1, \dot{q}_1, q_2, \dot{q}_2)$ was changed to $\beta(q_1, \dot{q}_1, q_2, \dot{q}_2) = 10000(q_1 - q_2)^3$ and the performance of the observer was tested. The simulation results are shown in Figures 5–a to 5–d. It follows that the states of the observer track the states of the system after some transient period. This shows the advantage of the proposed observer in using an online training scheme. In general, the initial condition of the system is not available to the observer. Figures 6–a to 6–d show the simulation results when the initial condition of the system was selected randomly. It takes a longer time for the states of the observer to converge, but eventually they follow the states of the system. Finally, simulation results for a two-link planar manipulator are given. The parameters are $J = diag([1.16, 1.16])kg.m^2$, $m_1 = m_2 = 1Kg$, $l_1 = l_2 = 1m$, $K = diag([100, 100])N/m$. Figures 7–a to 7–d show the system and observer state responses. After the learning has been completed, the responses of the system are shown in Figures 7–e to 7–h. It can be observed that the neural network was also able to learn the system states in this case.

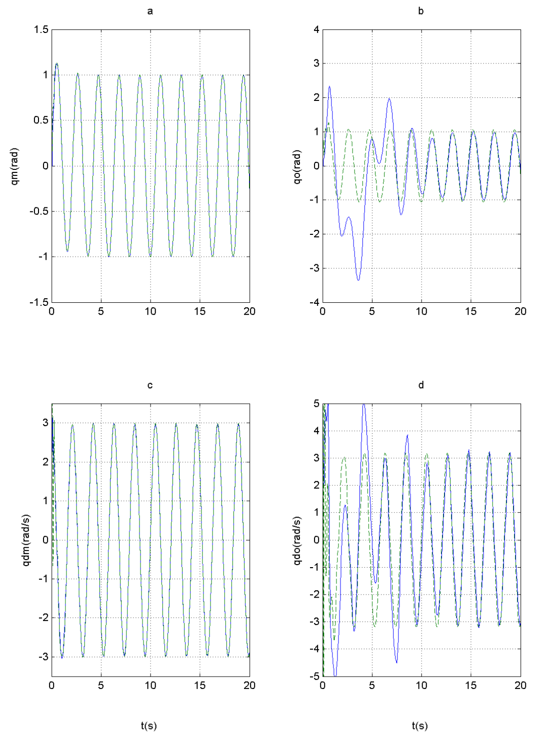

Fig. 5. The responses of the closed-loop system to $\sin(3t)$ reference trajectory with nonlinear model for stiffness: (a) motor position, (b) link position, (c) motor velocity, (d) link velocity. The solid lines correspond to the states of the observer.

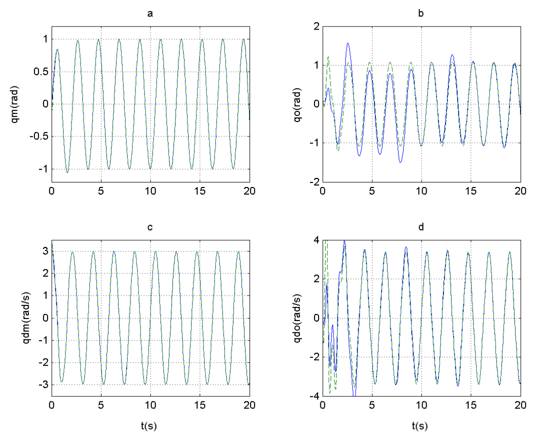

Fig. 6. The responses of the closed-loop system to $\sin(3t)$ reference trajectory with nonlinear model for stiffness and random initial conditions of the system: (a) motor position, (b) link position, (c) motor velocity, (d) link velocity. The solid lines correspond to the states of the observer.

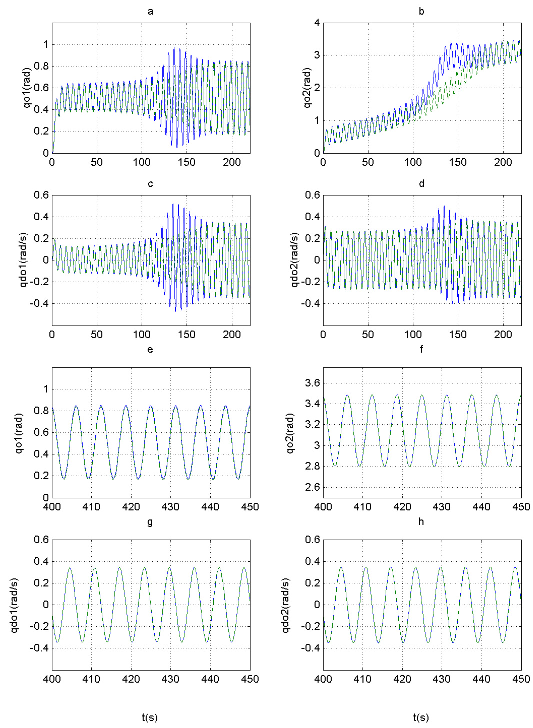

Fig. 7. The responses of the two-link manipulator to $\sin(t)$ reference trajectory with nonlinear model for stiffness, during (a–d) and after (e–h) the learning period: (a & e) position of the first link, (b & f) position of the second link, (c & g) velocity of the first link, (d & h) velocity of the second link. The solid lines correspond to the states of the observer.

5. CONCLUSIONS

The problem of designing a nonlinear observer for flexible-joint manipulators using a neural network approach has been addressed in this paper. No *a priori* knowledge about the joint stiffness, stiction, and friction were assumed. Simulation results for single and two-link manipulators were presented that demonstrate the effectiveness of the proposed algorithm. The possibility of applying the proposed scheme for calibration of a seven degrees-of-freedom manipulator is currently being investigated.

6. REFERENCES

Ahmed, M.S. and S.H. Riyaz (2000). Dynamic observers–a neural net approach. *J. Intelligent and Fuzzy Systems* **9**, 113–127.

Al-Ashoor, R.A., R.V. Patel and K. Khorasani (1993). Robust adaptive controller design and stability analysis for flexible-joint manipulators. *IEEE Transactions on Systems, Man, and Cybernetics* **23**, 589–602.

Chang, Y.Z. and R.W. Daniel (1992). On the adaptive control of flexible joint robots. *Automatica* **28**, 969–974.

Elanayar, S. and Y.C. Shin (1994). Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. *IEEE Trans. Neural Network* **5**, 594–603.

Ge, S.S. and L.C. Woon (1998). Adaptive neural network control of flexible joint manipulators in constrained motion. *Trans. Inst. Meas. Control* **20**, 37–46.

Goldsmith, P.B., B.A. Francis and A.A. Goldenberg (1999). Stability of hybrid position/force control applied to manipulators with flexible joints. *International Journal of Robotics and Automation* **14**, 146–160.

Hung, J.Y. (1991). Control of industrial robots that have transmission elasticity. *IEEE Trans. Indust. Electron.* **38**, 421–427.

Jankovic, M. (1992). Exponentially stable observer for elastic joint robots. In: *Proceedings of the 31st IEEE Conference on Decision and Control*. pp. 323–324.

Jankovic, M. (1995). Observer based control for elastic joint robots. *IEEE Trans. Robot. and Automat.* **11**, 618–623.

Kim, Y.H., F.L. Lewis and C.T. Abdallah (1996). Nonlinear observer design using dynamic recurrent neural networks. In: *Proceedings of the 31st IEEE Conference on Decision and Control*. pp. 949–954.

Kosmatopoulos, E.B., P.A. Ioannou and M.A. Christodoulou (1992). Identification of nonlinear systems using new dynamical neural network structures. In: *Proceedings of the 31st IEEE Conference on Decision and Control*. pp. 20–25.

Levin, A.U. and K.S. Narendra (1996). Control of nonlinear dynamical systems using neural networks– part ii: Observability, identification, and control. *IEEE Trans. Neural Network* **7**, 30–42.

Nicosia, S. and A. Tornambe (1989). High-gain observer in the state and parameter estimiation of robots having elastic joints.. *System and Control Letters* **13**, 331–337.

Nicosia, S., P. Tomei and A. Tornambe (1988). A nonlinear observer for elastic robots. *IEEE Trans. Robot. and Automat.* **4**, 45–52.

Nicosia, S., P. Tomei and A. Tornambe (1989). An approximate observer for a class of nonlinear systems. *System and Control Letters* **12**, 43–51.

Ortega, R., A. Loria, P.J. Nicklasson and H. Sira-Ramirez (1998). *Passivity-based Control of Euler-Lagrange Systems*. Springer-Verlag. London.

Spong, M.W. (1987). Modeling and control of elastic joint robots. *ASME J. Dyn. Syst., Meas. and Control* **109**, 310–319.

Spong, M.W. (1989). On the force control problem for flexible joint manipulators. *IEEE Trans. Automotic Control* **34**, 107–111.

Spong, M.W. and M. Vidyasagar (1989). *Robot Dynamics and Control*. John Wiley and Sons.

Talebi, H.A., R.V. Patel and H. Asmer (2000). Neural network based dynamic modeling of flexible-link manipulators with application to the ssrms. *Journal of Robotic Systems* **17**, 385–401.

Tomei, P. (1990). An observer for flexible joint robots. *IEEE Trans. Automotic Control* **35**, 739–743.

Zeman, V., R.V. Patel and K. Khorasani (1997). Control of a flexible-joint robot using neural networks. *IEEE Transactions on Control Systems Technology* **5**, 453–462.