A Neural Network Compensator for Uncertainties of Robotic Manipulators

著者	石黒　章夫
journal or publication title	IEEE Transactions on Industrial Electronics
volume	39
number	6
page range	565-570
year	1992
URL	http://hdl.handle.net/10097/46525

doi: 10.1109/41.170976

A Neural Network Compensator for Uncertainties of Robotics Manipulators

Akio Ishiguro, Takeshi Furuhashi, Shigeru Okuma, *Member, IEEE,* and Yoshiki Uchikawa

Abstract—**Neural networks have been studied to control robotic manipulators. Most researches aimed to internalize inverse dynamic models of controlled objects. It has been difficult, however, to obtain true teaching signals of neural networks for learning unknown controlled objects. In the case of robotic manipulators, approximate models of the controlled objects can be generally derived. We believe that the neural networks perform best when they are not required to learn too much. Thus, in this paper, we propose neural networks that do not learn inverse dynamic models but compensate nonlinearities of robotic manipulators with the computed torque method. Furthermore, we show a method to obtain true teaching signals of the neural network compensators.**

I. Introduction

ROBOTIC manipulators have become increasingly important in the field of flexible automation. High-speed and high-precision trajectory tracking is one of the indispensable capabilities for versatile applications of the manipulators. Even in a well-structured setting for an industrial use, the manipulators are subject to structured and/or unstructured uncertainties. Structured uncertainty is characterized by a correct dynamical model with parameter uncertainty due to imprecision of the manipulator link properties, unknown loads, inaccuracies on the torque constants of the actuators, and so on. Unstructured uncertainty is characterized by unmodeled dynamics. Unmodeled dynamics result from the presence of high-frequency mode of the manipulator, neglected time-delays, nonlinear friction, and so on.

Adaptive approaches have been proposed to maintain the tracking performance of the robotic manipulators in the presence of the structured uncertainty [1], [2]. Although the adaptive control is effective to compensate the influence of the structured uncertainty, it is not clear that the adaptive means can reduce the effect of the unstructured uncertainty.

Neural networks [3] have versatile features such as learning capability, nonlinear mapping, and parallel processing. The error back-propagation algorithm [3] is the most prevalent technique for learning neural networks. Neural networks have been studied for use in the field of control, especially robotic manipulator control, as a new approach to adaptive control [4]–[8].

Most of these methods aimed at internalizing inverse dynamic models of controlled objects in the neural networks. Since the control inputs, which are to realize the desired responses of the unknown controlled objects cannot be derived *a priori*, it has been difficult to obtain true teaching signals of neural networks for the unknown objects. Therefore, neural networks have not learned correctly and it has been doubtful whether the learning converges.

Atkeson *et al.* [9], Jordan *et al.* [10], and Psaltis *et al.* [11] proposed interesting methods to obtain the inverse dynamic model of the controlled object in a neural network. These studies are aimed at internalizing inverse models by devising connection between neural networks and the controlled object. But these methods do not guarantee the desirable response of the system with the trained neural network, since the inputs of the controlled object that give the desired actual outputs cannot be known in the learning stage. As a result, it is doubtful whether the inverse dynamic models are constructed due to their imprecise teaching signals.

Another interesting approach is proposed by Li and Slotine [12] to internalize the inverse dynamic model of the controlled objects with self-learning. Since, in this method, the inputs of the controlled object that give the desired response are unknown, it is also doubtful whether the correct inverse dynamic model is internalized in the trained neural network.

In the case of the robotic manipulators, the approximate models of the controlled objects can be derived. And the model-based control is an effective approach to high-performance control. The computed torque method [13] using the model of the robotic manipulator is an effective means for the trajectory control. It has become widely recognized, however, that the tracking performance of the method in high-speed operations is severely affected by the uncertainties mentioned above. This is especially true for direct-drive robots without gearing that reduces the dynamic effects.

This paper presents a new neural network controller that assists the computed torque method. The neural network is used not to learn the inverse dynamic model but to compensate the uncertainties of robotic manipu-

Manuscript received April 2, 1991; revised April 21 and June 12, 1992.
The authors are with the Department of Electronic-Mechanical Engineering, Nagoya University, Nagoya 464-01, Japan.
IEEE Log Number 9202509.

lators. A comparison of the performance of the proposed neural network controller with that of the adaptive controller proposed by Craig [1] is shown. And the effectiveness of the proposed neural network controller in compensating the unstructured uncertainties is clarified. Furthermore, this paper presents a method to obtain true teaching signals for the neural network. Convergence of endpoint tracking error of a robotic manipulator is remarkably improved. Feasibility of the proposed method is confirmed by simulations.

II. Computed Torque Method

The motion equation of the manipulator consisting of a set of n moving rigid links connected in a serial chain is given by

$$\tau = \boldsymbol{M}(\boldsymbol{\theta})\boldsymbol{u} + \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{1}$$

where τ represents the $n \times 1$ vector of joint torques of the manipulator supplied by the actuators. $\boldsymbol{M}$ and $\boldsymbol{h}$ are the $n \times n$ manipulator inertia matrix and the $n \times 1$ vector of the centrifugal and Coriolis terms, respectively. And $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, and $\ddot{\boldsymbol{\theta}}$ are, respectively, the $n \times 1$ vectors of joint angles, angular velocities, and angular accelerations.

The control system with the computed torque method is shown in Fig. 1. The nonlinear compensation portion is

$$\tau = \hat{\boldsymbol{M}}(\boldsymbol{\theta})\boldsymbol{u} + \hat{\boldsymbol{h}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{2}$$

and $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{h}}$ are the estimated values of the true parameters $\boldsymbol{M}$ and $\boldsymbol{h}$, respectively. The servo portion is

$$\boldsymbol{u} = \ddot{\theta}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}. \tag{3}$$

In (3), the servo error $\boldsymbol{e} = [e_1, e_2, \cdots, e_n]^{\mathrm{T}}$ is defined as

$$\boldsymbol{e} = \boldsymbol{\theta}_d - \boldsymbol{\theta} \tag{4}$$

where $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ represent $n \times n$ constant, diagonal-gain matrices that consist of positive parameters on the diagonals, and $\boldsymbol{\theta}_d$ denotes the desired joint angle θ. When the structured and unstructured uncertainties do not exist, the error dynamics of the system is derived from (1) to (4) as:

$$\hat{\boldsymbol{M}}\left(\ddot{\boldsymbol{\theta}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e}\right) + \hat{\boldsymbol{h}} = \boldsymbol{M}\ddot{\boldsymbol{\theta}} + \boldsymbol{h} \tag{5}$$

and (5) is rewritten by using $\hat{\boldsymbol{M}} = \boldsymbol{M}$ and $\hat{\boldsymbol{h}} = \boldsymbol{h}$ as

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = 0. \tag{6}$$

If parameters $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ are chosen in the favorable situation, the error will be asymptotically zero. In general, however, the precise parameters of the manipulator such as link properties are difficult to identify. Hence when the uncertainties exist, the dynamics of the manipulator is given by

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{F} \tag{7}$$

where $\boldsymbol{F}$ is the unstructured uncertainty. As a result, if $\hat{\boldsymbol{M}}^{-1}$ exists, the error equation is rewritten as

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = \hat{\boldsymbol{M}}^{-1}(\Delta\boldsymbol{M}\ddot{\boldsymbol{\theta}} + \Delta\boldsymbol{h} + \boldsymbol{F}) \tag{8}$$

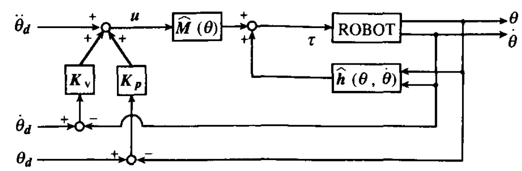

Fig. 1. Controller using the computed torque method.

where $\Delta\boldsymbol{M}$ and $\Delta\boldsymbol{h}$ are the errors caused by the structured uncertainties and are expressed as

$$\Delta\boldsymbol{M} = \boldsymbol{M} - \hat{\boldsymbol{M}}$$

$$\Delta\boldsymbol{h} = \boldsymbol{h} - \hat{\boldsymbol{h}}. \tag{9}$$

Equation (8) leads to the fact that the steady-state error will exist even if the gain parameters are chosen favorably under the presence of the structured and unstructured uncertainties.

III. Neural Network Controller

A. System Configuration

Various methods can be considered to incorporate neural networks into the computed torque method. For example, a method using two neural networks is supposed to be effective. These networks are installed in place of the estimated models $\hat{\boldsymbol{M}}, \hat{\boldsymbol{h}}$. One of the networks is to identify the inertia matrix $\boldsymbol{M}$ and the other one is to identify the centrifugal, Coriolis $\boldsymbol{h}$ and such unstructured uncertainties as friction forces $\boldsymbol{F}$. But the neural networks in this method have to learn not only the uncertainties but also the structures of the robotic manipulators. We believe that the neural networks perform effectively when they are not required to learn too much.

Thus, in this paper, we propose a method to use a neural network together with the models $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{h}}$ to compensate only the uncertainties in the control system as a nonlinear compensator of the manipulator. The proposed neural network controller is shown in Fig. 2. The neural network just compensates the structured and unstructured uncertainties. We think that it is the best way to compensate for unknown uncertainties in an otherwise known plant. Furthermore, the structure of the controller is simple and small.

B. True Teaching Signals

Since the proposed control system shown in Fig. 2 makes the neural network just learn the structured and unstructured uncertainties of the manipulator, the neural network can be used effectively. To make the neural network compensate the uncertainties completely, it is necessary to obtain the teaching signal corresponds to the desired compensating one of the torque demand of the robotic manipulator. In this section, we propose a method to obtain true teaching signals for the neural network compensator.

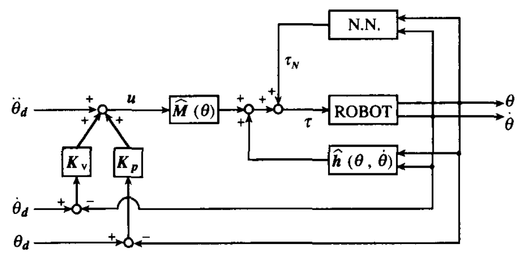

Fig. 2. Controller using the computed torque method with neural network.

Letting the output of the neural network compensator be τ_N, the error equation of the control system is given by

$$\hat{M}\left(\ddot{\theta}_d + K_v\dot{e} + K_p e\right) + \hat{h} + \tau_N = M\ddot{\theta} + h + F \quad (10)$$

Equation (10) is rewritten as

$$\ddot{e} + K_v\dot{e} + K_p e = \hat{M}^{-1}\left(\Delta M\ddot{\theta} + \Delta h + F - \tau_N\right). \quad (11)$$

If the neural network uses the following teaching signals τ_t:

$$\tau_t = \Delta M\ddot{\theta} + \Delta h + F \quad (12)$$

the right-hand side of (11) becomes zero.

Fig. 3 shows the control system that realizes the true teaching signals for the neural network compensator. We prepare the following dynamics in the nonlinear compensation portion in the figure:

$$\tau_m = \hat{M}\ddot{\theta} + \hat{h}. \quad (13)$$

The relationship between the inputs and the outputs of the controlled object is given by

$$\tau = M\ddot{\theta} + h + F. \quad (14)$$

Thus, by subtracting (13) from (14),

$$\begin{aligned}\tau - \tau_m &= M\ddot{\theta} + h + F - (\hat{M}\ddot{\theta} + \hat{h}) \\ &= \Delta M\ddot{\theta} + \Delta h + F. \quad (15)\end{aligned}$$

As a result, the teaching signals in (12) can be obtained. The neural network compensator is to realize nonlinear mapping from the variables θ, $\dot{\theta}$, and $\ddot{\theta}$ to τ_t expressed in (12). Here, $\ddot{\theta}$ is approximated by the difference of $\dot{\theta}$ as follows:

$$\ddot{\theta}(n) = \frac{\dot{\theta}(n) - \dot{\theta}(n-1)}{T_S} \quad (16)$$

where n denotes the time index and T_S denotes the sampling period. The structure of the neural network is shown in Fig. 4.

The units in each layer are connected through weights and the output O_j of a unit in the output and hidden layers is obtained by the following equations:

$$\begin{aligned}O_j &= f(n_j) \\ n_j &= \sum_i W_{ji} O_i \quad (17)\end{aligned}$$

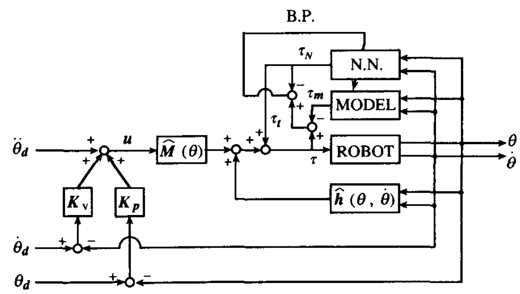

Fig. 3. Generation method of true teaching signals for the neural network compensator.

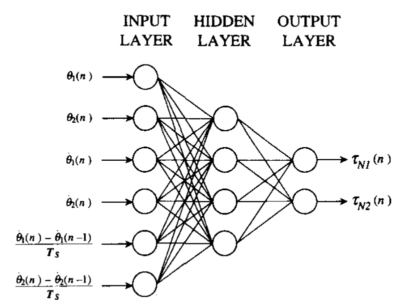

Fig. 4. Structure of the neural network compensator.

where n_j is the weighted sum of the outputs of the preceding layer. W_{ji} denotes the weight. The function $f(\cdot)$ is a semilinear activation function called a sigmoid function and is expressed as

$$f(x) = \frac{2}{1 + \exp(-x)} - 1. \quad (18)$$

The units in the input layer of the network just deliver their input signals to the units in the hidden layer. Each output of the unit in the output layer is condensed within ± 0.3.

Learning proceeds with the error back-propagation algorithm using the teaching signal τ_t expressed in (12). The weight between the ith and jth units is updated by the following equation at every sampling period:

$$\Delta W_{ji} = \eta \delta_j O_i \quad (19)$$

where η is a small positive number called a learning rate and δ_j is given by the following equations:

$$\text{for output units:} \quad \delta_j = (\tau_t - O_j) f'(n_j) \quad (20)$$

$$\text{for other units:} \quad \delta_j = \left(\sum_l W_{lj}\delta_l\right) f(n_j) \quad (21)$$

where $f'(\cdot)$ is the derivative of the function $f(\cdot)$ and l denotes the number of the unit of the succeeding layer.

IV. Simulation Results

Simulations were done to verify the proposed neural network controller compensating uncertainties. We use a

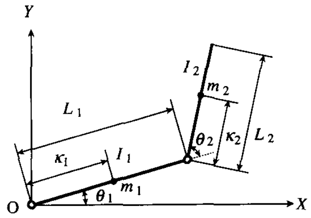

Fig. 5. Two-degree-of-freedom robotic manipulator.

TABLE I
PARAMETERS OF MANIPULATOR

		Link 1	Link 2	Unit
Arm length	L	0.25	0.16	m
Link center of gravity	k	0.20	0.14	m
Mass	m	9.5	5.0	kg
Inertia	I	4.3×10^{-3}	6.1×10^{-3}	$kg \cdot m^2$
Gear ratio	N	40	30	
Coulomb friction coefficient	T	0.20	0.20	$N \cdot m$
Motor inertia	J_m	4.61×10^{-5}	2.65×10^{-5}	$kg \cdot m^2$
Motor damping coefficient	D_m	3.84×10^{-3}	1.39×10^{-3}	$N \cdot s \cdot m^{-1}$

simple two-degree-of-freedom SCARA type manipulator as shown in Fig. 5 for simulations of the neural network controller. The meanings of the symbols in Fig. 5 are listed in Table I. The unstructured uncertainty in this paper was the Coulomb friction defined as follows:

$$\begin{aligned} \boldsymbol{F}_1 &= \boldsymbol{T}_1 \cdot \operatorname{sgn}(\dot{\theta}_1) \\ \boldsymbol{F}_2 &= \boldsymbol{T}_2 \cdot \operatorname{sgn}(\dot{\theta}_2). \end{aligned} \tag{22}$$

The link inertia and the link centroid position are the main parameters that are difficult to measure. Here, the parameter uncertainty was assumed to be the deviation of the link centroid position as follows:

$$\begin{aligned} \hat{\boldsymbol{k}}_1 &= 0.9 \cdot \boldsymbol{k}_1 \\ \hat{\boldsymbol{k}}_2 &= 0.9 \cdot \boldsymbol{k}_2. \end{aligned} \tag{23}$$

$\hat{\boldsymbol{k}}_1$ and $\hat{\boldsymbol{k}}_2$ were used for calculating $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{h}}$ for the computed torque method. The Coulomb friction was not taken into account in the control law.

Fig. 6 shows the simulation results of the trajectory control with the neural network controller. The figure shows the 1st and 100th trial of writing a circle on the X-Y plane. At the outset of the simulation, the connection weights of the neural network $\boldsymbol{W}_{ji}$ were randomly initialized. The sampling period is 2 ms, and it takes about 3 s to write a circle on the X-Y plane. Each diagonal element of the proportional gain matrix is 20, and that of the differential gain matrix is 5. The number of the units in the hidden layer of the neural network is 4. As learning of the neural network proceeds, the trajectory of the manipulator well follows the desired one. At the 100th trial, the tracking error converges to a small value. Fig. 7 shows a comparison between the obtained trajectory by the neural network controlled at the 100th trial and that by the computed torque method with the adaptive scheme [1] at the 10th trial. The reason for the comparison under the different trial number is that the convergence of the adaptive scheme does not proceed after the 10th trial.

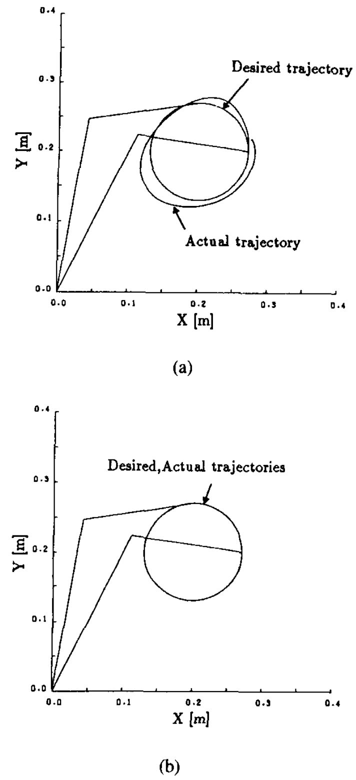

Fig. 6. Simulation results. (a) 1st trial. (b) 100th trial.

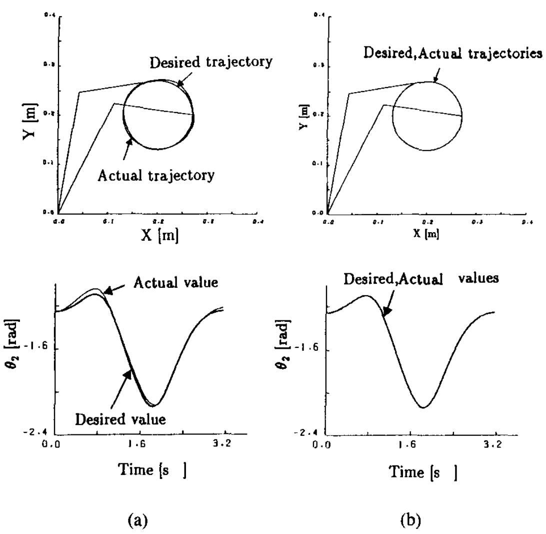

Fig. 7. Comparison between the proposed neural network controller and the adaptive scheme. (a) Adaptive scheme. (b) Proposed scheme.

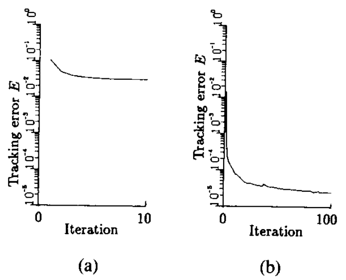

Fig. 8. Transition of tracking error. (a) Adaptive scheme. (b) Proposed scheme.

The top figures are the trajectories on the X-Y plane. The bottom figures are the trajectories of the link angles of the link 2 that were clearly affected by the uncertainty. Errors are observed in the left-hand side trajectories.

Fig. 8 shows a comparison of the transition of the tracking error between the proposed scheme and the conventional adaptive one. The tracking error E is defined as follows:

$$E = \sum_{i=1}^{N} \left\{ (X_{di} - X_i)^2 + (Y_{di} - Y_i)^2 \right\} \tag{24}$$

where N denotes the sampling number within one trial. X_{di}, Y_{di} are the demands of the trajectory on the X-Y plane at the sampling point i, and X_i, Y_i are the actual trajectories. In the figure, the steady-state error of the proposed scheme is remarkably reduced compared with that of the conventional adaptive scheme. Therefore, the neural network controller is effective for not only the structured uncertainty but also the unstructured one. The distinctive feature of the neural network controller over the conventional controller is that the neural network controller needs no information about the unstructured uncertainty which was assumed to be F_i in (22) in this paper. In this method, the generalization of the neural network is not guaranteed, but it does not become to be a demerit. Because the learning proceeds quickly due to its small structure of the neural network compared with other methods, which intended to construct the inverse dynamic model of the controlled object.

V. Conclusions

This paper presented a new neural network controller for trajectory control of robotic manipulators. The neural network is used not to internalize the inverse dynamic model of the controlled object but to compensate only the uncertainties of the robotic manipulator. A comparison of its performance with the conventional adaptive scheme in compensating the unmodeled effects was done. As the result, the adaptive capability of the neural network controller to the unstructured effects was shown, although the conventional scheme had no capability to reduce the unmodeled effects. Furthermore, a learning method of the neural network compensator with true teaching signals was shown. The tracking error of the robotic manipulator was remarkably reduced.

References

[1] J. J. Craig, *Adaptive Control of Mechanical Manipulators*. Reading, MA: Addison-Wesley, 1988.

[2] J. E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Automat. Contr.*, vol. AC-33, no. 11, pp. 995–1003, Nov. 1988.

[3] D. E. Rumelhart *et al.*, *Parallel Distributed Processing, vols.* 1 *and* 2. Cambridge, MA: The MIT Press, 1986.

[4] B. Bavarian, Introduction to neural networks for intelligent control, *IEEE Control Systems Mag.*, pp. 3–7, April 1988.

[5] M. Kawato *et al.*, "Hierarchical neural network model for voluntary movement with application to robotic," *IEEE Control Systems Mag.*, pp. 8–16, April 1988.

[6] H. Miyamoto *et al.*, "Feedback-error-learning neural network for trajectory control of a robotic manipulator," *Neural Networks*, vol. 1, no. 3, pp. 251–265, 1988.

[7] M. Kawato *et al.*, "A hierarchical neural-network model for control and learning of voluntary movement," *Biol. Cybern.*, vol. 57, pp. 169–185, 1987.

[8] T. Setoyama, M. Kawato, and R. Suzuki, "Manipulator control by inverse-dynamic model learned in multi-layer neural network," Japan IEICE Tech. Rep., vol. MBE87-135, pp. 249–256, 1987.

[9] C. G. Atkeson *et al.*, "Using associative content-addressable memories to control robots," *IEEE Conf. Decision Contr.*, 1988.

[10] M. I. Jordan, "Supervised learning and systems with excess degrees of freedom," COINS Tech. Rep. 88-27, p. 1–41, 1988.

[11] D. Psaltis *et al.*, "A multilayer neural network controller," *IEEE Control System Mag.*, pp. 17–21, April 1988.

[12] W. Li and J. E. Slotine, "Neural network control for unknown nonlinear systems," in *Proc. 1989 Amer. Contr. Conf.*, pp. 1136–1141.

[13] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators," *J. Dynamic Syst., Meas. Contr.*, vol. 102, pp. 69–76, June 1980.

Akio Ishiguro was born in Kochi, Japan, on May 18, 1964. He received the B.S., M.S., and Ph.D. degrees in electronic-mechanical engineering from Nagoya University, Japan, in 1987, 1989, and 1991, respectively.

Since 1991, he has been with the Department of Electronic-Mechanical Engineering of Nagoya University as an Assistant Professor. He is engaged in research on robotics, neural networks, and inverse problems.

Dr. Ishiguro is a member of the Institute of Electrical Engineers of Japan, the Society of Instrument and Control Engineers of Japan, and Institute of Electronics, Information and Communication Engineers, and the Robotics Society of Japan.

Takeshi Furuhashi was born in Shizuoka, Japan, in 1954. He received the B.E., M.E., and Ph.D. degrees in electrical engineering from Nagoya University, Japan, in 1980, 1982 and 1985, respectively.

He was with Toshiba Corporation from 1985 to 1988. From 1988 to 1990, he was with Nagoya University as an Assistant Professor in the Department of Electrical Engineering. Since 1990, he has been an Associate Professor of the Department of Electronic-Mechanical Engineering of Nagoya University. His research interests are in fuzzy theory, neural networks, and knowledge acquisition in artificial intelligence.

Dr. Furuhashi is a member of the Institute of Electrical Engineers of Japan, the Society of the Instrument and Control Engineers of Japan, and the Japan Society for Fuzzy Theory and Systems.

Shigeru Okuma (M'82) was born in Gifu, Japan, in 1948. He received the B.E., M.E., and Ph.D. degrees in electrical engineering from Nagoya University, Japan, in 1970, 1972 and 1978, respectively. He also received the M.E. degree in system engineering from Case Western Reserve University, Cleveland, OH, in 1974.

From 1977 to 1984, he was with the Department of Electrical Engineering of Nagoya University as an Assistant Professor. From 1985 to 1990, he was with the Department of Electronic-Mechanical Engineering of Nagoya University as an Associate Professor. Since December 1990, he has been a Professor of the Department of Electronic-Mechanical Engineering of Nagoya University. His research interests are in the areas of power electronics, robotics, and micro/nano technology.

Dr. Okuma is a member of the Institute of Electrical Engineers of Japan, the Society of Instrument and Control Engineers of Japan, and the Robotics Society of Japan.

Yoshiki Uchikawa was born in Aichi, Japan, in 1941. He received the B.E., M.E., and Ph.D. degrees in electrical engineering from Nagoya University, Japan, in 1964, 1966, and 1971, respectively. From 1969 to 1984, he was with the Department of Electrical Engineering of Nagoya University as an Assistant Professor. Since 1985, he has been with the Department of Electronic-Mechanical Engineering of Nagoya University, where he is currently a Professor. His research interests are in the areas of electronic mechanics, electron microscopy, fuzzy modeling and neural networks.

Dr. Uchikawa is a member of the Institute of Electrical Engineers of Japan, the Society of Instrument and Control Engineers of Japan, the Robotics Society of Japan, the Japan Society of Electron Microscopy, the Japan Society of Applied Physics, the Japan Society of Applied Electromagnetics, and the Japan Society of Precision Engineering.