International Journal of Fuzzy Logic and Intelligent Systems, vol. 1, no. 1, June 2001 pp. 29-34

Control of Flexible Joint Robot Using Direct Adaptive Neural Networks Controller

In-Yong Lee, Han-Ho Tack[**], Sang-Bae Lee and Boo-Kwi Choi[***]

Department of Electronic Engineering, Dong-A University
** Department of Electronic Engineering, Chinju National University
*** Department of Electronic and Comm. Engineering, Korea Maritime University

Abstract

This paper is devoted to investigating direct adaptive neural networks control of nonlinear systems with uncertain or unknown dynamic models. In the direct adaptive neural networks control area, theoretical issues of the existing backpropagation-based adaptive neural networks control schemes. The major contribution is proposing the variable index control approach, which is of great significance in the control field, and applying it to derive new stable robust adaptive neural networks control schemes. This new schemes possess inherent robustness to system model uncertainty, which is not required to satisfy any matching condition.

To demonstrate the feasibility of the proposed learning algorithms and direct adaptive neural networks control schemes, intensive computer simulations were conducted based on the flexible joint robot systems and functions.

Key Words : direct adaptive neural networks, flexible joint robot systems.

I. Introduction

The existence of flexibilities in the robot structure limits its ability to perform high precision

manipulator. Experimental results reveal that, for a wide variety of robots, joint flexibility is principal source contributing to overall robot flexibility[1]. Therefore, joint flexibility should be taken into account in the modeling and design of robot controllers if high performance is to be achieved.

Due to existence of modeling errors, most conventional control schemes for flexible joint robots have limited control precisions. The control problem of flexible joint robots with uncertain dynamic models attracted great interest from both academia and industry. Different control approaches have been proposed for flexible joint robots in the past decade. They can be classified into: 1) the exact model based control approach, 2) robust control approach, 3) adaptive control approach, 4) iterative learning control approach, 5) fuzzy control approach and 6) neural networks control approach.

Recently, the fuzzy control approach and neural networks control approach have been attracting more interest due to their potential in dealing with large structural and large parametric uncertainty. In general, the neural networks control approach is relatively new, and there are theoretical results and less practical implementation experience available. The main existing problems in the neural networks control approach are: 1) how to ensure the global convergence of the learning processes, and 2) how to determine the neural networks structure for a given problem. At present, most existing backprop-based neural networks control schemes are only suitable for off-line training of neural networks controllers, since there are several methods and heuristics to solve the above mentioned problems by off-line training. Few published papers deal with the neural networks control problem of flexible joint robots, due to their higher order dynamics.

This paper aims at developing stable robust direct adaptive neural networks controllers for flexible joint robots. The feedback signals are the joint and motor angular positions and velocities. Measurements of accelerations or jerks are not required. It is proved that all the signals in the closed-loop adaptive neural networks control systems can be made bounded and the output tracking errors can be guaranteed to be globally convergent to zero. Since the new direct adaptive neural networks control systems are robust to neural networks representation errors, the structure design of the neural networks can be simplified. Unlike the conventional adaptive controllers, the proposed direct adaptive neural networks control schemes allow the flexible joint robots to possess a general structure and unknown disturbances, and need the least a priori information about the dynamics. Any observable dynamics, some of which were regarded as unmodeled in conventional approaches, can be modeled by neural networks.

II. Neural Networks Control Algorithm

Figure 1 shows the configuration of the flexible joint robot. For the formulation of the dynamic model of a

Maniscript received March 15, 2001; revised May 1, 2001.

robot with flexible joints, several commonly used assumptions will be adopted [2]. A robot with flexible joints is usually modeled as a system of 2 n rigid bodies: n links and n actuators, connected by elastic transmissions. The robot dynamic model may be obtained by using Euler-Lagrange equations. The matrix form of equations of motion can be expressed as follows:

$$\begin{bmatrix} A(q) & 0 \\ 0 & I_m \end{bmatrix}\begin{bmatrix} \ddot{q} \\ \ddot{p} \end{bmatrix}+\begin{bmatrix} B(q,\dot{q}) \\ B_m\dot{p}+F_m \end{bmatrix}+K\begin{bmatrix} q-p \\ p-q \end{bmatrix}=\begin{bmatrix} 0 \\ u \end{bmatrix} \quad (1)$$

where $q=[q_1,\ldots,q_n]^T$ is the vector of the joint angle; $A(q)$ is inertia matrix including the stator masses and also the rotor masses in their translational portion; $B(q,\dot{q})$ is the vector of centrifugal, Coriolis and gravitational terms; $p=[p_1,\ldots,p_n]^T$ is the vector of rotational angles of actuator rotors; the moment of inertia of the rotor: $I_m=diag[I_1,\ldots,I_n]$; the viscous friction coefficient of the joint: $B_m=diag[B_{m1},\ldots,B_{mn}]$,; the friction torque acting on the rotor: $F_m=diag[F_{m1},\ldots,F_{mn}]^T$; the elastic constant of the joint: K; and u is the vector of drive torques applied to the actuator rotors.

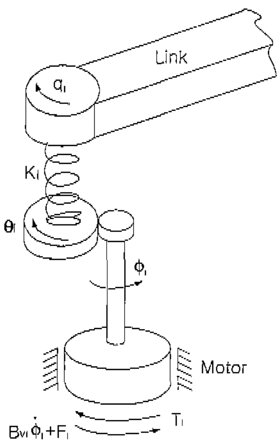

Fig. 1. Model of the flexible joint robot.

Based on these facts, the following general model for an n-link flexible joint robot manipulator is proposed:

$$A(q)\ddot{q}+f_1(q,\dot{q})=K(q-p)p \quad (2)$$

$$I_m\ddot{p}+f_2(q,p,\dot{p})=u \quad (3)$$

where $f_1(q,\dot{q})$ and $f_2(q,p,\dot{p})$are unknown function vector; and $K(q-p)$ is the unknown symmetric positive definite joint stiffness matrix and is nonlinear in $q-p$.

Assume that the desired trajectory of the end-effector of a robot manipulator is expressed in joint coordinates as $q_d \in C^4$, and the corresponding desired trajectory of the rotational angles of actuator rotors is expressed as p_d, $\dot{p}_d$, $\ddot{p}_d$, which are to be determined. Only the measurements of $q(t)$, $\dot{q}(t)$, $p(t)$, and $\dot{p}(t)$ are available.

Define the tracking error vectors as:

$$e=q-q_d\,,\ \delta=p-p_d \quad (4)$$

Let $s_1=\dot{e}+\Lambda_1 e$ and $s_2=\dot{\delta}+\Lambda_2\delta$. Therefore,

$$s_1=\dot{q}-\dot{q}_r \quad with \quad \dot{q}_r=\dot{q}_d-\Lambda_1 e \quad (5)$$

$$s_2=\dot{p}-\dot{p}_r \quad with \quad \dot{p}_r=\dot{p}_d-\Lambda_2\delta \quad (6)$$

where q_r and p_r are called the reference trajectories and are computable from measured signals; Λ_1 and Λ_2 are diagonal positive definite matrices. With the new notions, the generalized error equations of the flexible joint robotic systems can be expressed as follows:

$$A(q)\dot{s}_1+[A(q)\ddot{q}_r+f_1(q,\dot{q})]=Kp \quad (7)$$

$$I_m\dot{s}_2[I_m\ddot{p}_r+f_2(q,p,\dot{p})]=u \quad (8)$$

Define new supplementary vectors of s_i with dead-zone Δ_i as follows:

$$s_{\Delta i}=[s_{\Delta i1},\ldots,s_{\Delta in}]^T \quad i=1,\ 2,\ldots,n \quad (9)$$

with

$$s_{\Delta ij}=s_{ij}-\Delta_i sat(\frac{s_{ij}}{\Delta_i}) \quad j=1,\ 2,\ldots,n \quad (10)$$

where $sat(.)$ is the saturation function.

Let

$$g_1(q,\ \dot{q},\ \ddot{q}_r,s_{\Delta 1})=[A(q)\ddot{q}_r+f_1(q,\ \dot{q})]-\frac{1}{2}\dot{A}(q)s_{\Delta 1} \quad (11)$$

$$g_2(q,p,\ \dot{p},\ \ddot{p}_r)=I_m\ddot{p}_r+f_2(q,p,\ \dot{p}) \quad (12)$$

$$g_3(q-p)=[k_{11},\ldots,K_{1n};\ldots;K_{n1},\ldots,K_{nn}]^T \quad (13)$$

Then equation(7) and (8) become:

$$A(q)\dot{s}_1+\frac{1}{2}\dot{A}(q)s_{\Delta 1}=Kp-g_1(q,\ \dot{q},\ \ddot{q}_r,s_{\Delta 1}) \quad (14)$$

$$I_m\dot{s}_2=u-g_2(q,p,\ \dot{p},\ \ddot{p}_r) \quad (15)$$

Assume that the corresponding approximation error vectors of the neural networks with finite number of neurons are $d_1(t)$, $d_2(t)$, and $d_3(t)$. Then $g_1(q,\ \dot{q},\ \ddot{q}_r,s_{\Delta 1})$, $g_2(q,p,\ \dot{p},\ \ddot{p}_r)$, and $g_3(q-p)$ can be expressed as:

$$g_1(q,\ \dot{q},\ \ddot{q}_r,s_{\Delta 1})=g_{NN1}(q,\ \dot{q},\ \ddot{q}_r,s_{\Delta 1})+d_1(t) \quad (16)$$

$$g_2(q,p,\ \dot{p},\ \ddot{p}_r)=g_{NN2}(q,p,\ \dot{p},\ \ddot{p}_r)+d_2(t) \quad (17)$$

$$g_3(q-p)=g_{NN3}(q-p)+d_3(t) \quad (18)$$

$$K(q-p)=K_{NN}(q-p)+K_d(t) \quad (19)$$

where $g_{NNi}(\cdot))\ (i=1,2,3)$ are realized by neural networks. By proper neural networks design, $d_1(t)$, $d_2(t)$, and $d_3(t)$ are reducible and can be confined to be bounded by any specified small constant ε_{ij}.

$$|\,d_{ij}(t)\,|<\varepsilon_{ij},\quad i=1,2,\ j=1,\ldots,n \quad (20)$$

$$|\,d_{3j}(t)\,|<\varepsilon_{3j},\quad j=1,2,\ldots,n^2 \quad (21)$$

Here, two-layer neural networks, the localized polynomial networks with competitive lateral inhibitory(CLI) cells, are adopted due to the possibility of deriving global convergent learning algorithms.

The localized receptive fields can be represented at least in two ways: 1) as a set of superboxes and 2) as a set of Gaussian potential function. Such realizations define the function of the CLI cells, which resemble the function of biological Golgi cells in the cerebellum. Take the case of using Gaussian potential functions.

If the bounded working region $\Omega \subset R^N$ is divided into J smaller subregions, $\Omega_1, \ldots, \Omega_J$, then $f(x)$ can be represented in each subregion Ω_j by a much lower order polynomial with the same given precision.

$$f(x) \doteq P(x) = \sum_{l=1}^{b} w_l^j \phi_l(x) \equiv f_j(x) \quad x_j \in \Omega \tag{22}$$

Define an input receptive field selection function for each subregion as follows:

$$s_j(x) = \begin{bmatrix} 1 & x \in \Omega_j \\ 0 & x \notin \Omega_j \end{bmatrix} \tag{23}$$

Then $f(x)$ can be expressed in the following localized representation:

$$f(x) = \sum_{j=1}^{J} s_j(x) \left(\sum_{l=1}^{b} w_l^j \phi_l(x) = \sum_{j=1}^{J} s_j(x) f_j(x) \right. \tag{24}$$

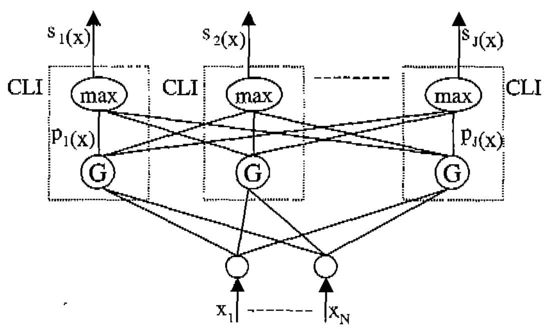

Fig. 2. Localized receptive field division using CLI cells.

The inputs to the CLI cells are $x = [x_1, \ldots, x_n]^T \in \Omega \subset R^N$, as shown in Figure 2.

The CLI cells perform the following computation:

(1) Gaussian potential function computation to determine the potential of the current inputs in each localized receptive field:

$$p_j(x) = \exp\left[-\sum_{n=1}^{N} \frac{(x_n - c_{jn})^2}{\sigma_{jn}^2}\right] \quad j = 1, \ldots, J \tag{35}$$

(2) CLI to select a unique excitatory receptive field:

$$s_j(x) = \begin{bmatrix} 1 & \text{if } p_j(x) = \max[p_1(x), \ldots, p_J(x)] \\ 0 & \text{if } p_j(x) < \max[p_1(x), \ldots, p_J(x)] \end{bmatrix} \tag{26}$$

Because only one $s_j(x)$ is allowed to be nonzero for each input vector x, randomly select one $s_j(x)$ to be excitatory and set the other $s_j(x)$ to be zero when there is a tie. This reflects the lateral inhibition property.

Based on the CLI cells, the structure of the localized polynomial neural networks with CLI cells is defined as shown in Figure 3 to realize the representation (24), where $f_j(x)$ is changed into $f_{jm}(x)$in the multi-output case and realized by a low-order local polynomial neural networks, and $s_j(x)$ is realized by the CLI cells. The interconnection lines between different CLI cells are omitted in the figure. The outputs of the overall multi-output network are:

$$y_m = \sum_{j=1}^{J} s_j(x) f_{jm}(x) \quad m = 1, \ldots, M \tag{27}$$

The number of the CLI cells for each output, J, is related to the order of each subpolynomial neural networks, L. When J is large, L can be small.

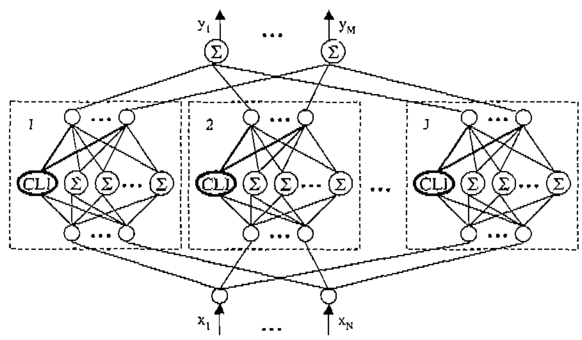

Fig. 3. The structure of localized polynomial networks with CLI cells.

Using linear parameterization, we can express $g_{NNi}(\cdot)$ $(i=1,2,3)$ as follows:

$$g_{NN1}(q, \dot{q}, \ddot{q}_r, s_{\Delta 1}) = \Phi_1(q, \dot{q}, \ddot{q}_r, s_{\Delta 1})^T \theta_1 \tag{28}$$

$$g_{NN2}(q, p, \dot{p}, \ddot{p}_r) = \Phi_2(q, p, \dot{p}, \ddot{p}_r)^T \theta_2 \tag{29}$$

$$g_{NN3}(q-p) = \Phi_3(q-p)^T \theta_3 \tag{30}$$

where $\theta_i (i=1,2,3)$ is the unknown output weight vector of the neural networks; and $\Phi_i(\cdot)$ is defined according to the structure of the used neural networks. Assume that the flexible joint robotic system (2) and (3) are time-invariant or slowly time-varying. Let the estimated θ_i and $\hat{\theta}_i$, and the estimation error vector be $\tilde{\theta}_i = \hat{\theta}_i - \theta_i$. The outputs of the neural networks at time instant t can be expressed as:

$$\hat{g}_{NN1}(q, \dot{q}, \ddot{q}_r, s_{\Delta 1}) = \Phi_1(q, \dot{q}, \ddot{q}_r, s_{\Delta 1})^T \hat{\theta}_1 \tag{31}$$

$$\hat{g}_{NN2}(q, p, \dot{p}, \ddot{p}_r) = \Phi_2(q, p, \dot{p}, \ddot{p}_r)^T \hat{\theta}_2 \tag{32}$$

$$\hat{g}_{NN3}(q-p) = \Phi_3(q-p)^T \hat{\theta}_3 \tag{33}$$

Thus, the following direct adaptive neural networks control algorithm is proposed:

$$u = -D_2 s_2 + \hat{g}_{N2}(q, p, \dot{p}, \dot{p}_r) \tag{34}$$

$$p_d = \hat{K}_N^{-1}[D_1 s_1 + \hat{g}_{M1}(q, \dot{q}, \ddot{q}_r, s_{\Delta 1})] \tag{35}$$

where D_1 and D_2 are diagonal positive definite matrices; $\hat{K}_N$ is derived from equation (13) and (33). Define:

$$\bar{d}_1(t) = K_d(t)p - d_1(t) \tag{36}$$

If the neural networks weights are updated as:

$$\dot{\hat{\theta}}_1 = -\Gamma_1 \Phi_1(q, \dot{q}, \ddot{q}_r, s_{\Delta 1}) s_{\Delta 1} \tag{37}$$

$$\dot{\hat{g}}_2 = -\Gamma_2 \Phi_2(q, p, \dot{p}, \ddot{p}_r) s_{\Delta 2} \tag{38}$$

$$\dot{\hat{g}}_3 = \Gamma_3 \Phi_3(q-p) s_3(s_{\Delta 1}, p) \tag{39}$$

where $\Gamma_i (i = 1, 2, 3)$ are symmetric positive definite matrices; and $s_3(s_{\Delta 1}, p)$ is defined from re-ordering the right-hand side of the following equation:

$$g_3(q-p)^T s_3(s_{\Delta 1}, p) = s_{\Delta 1}^T K(q-p)p \tag{40}$$

III. Simulation Results

The new direct adaptive neural networks control system for flexible joint robots can be illustrated by Figure 4. The neural networks control algorithm consists of the following steps: 1) error signal computing; 2) neural networks weights updating; 3) desired motor trajectory estimation; and 4) control torque computing.

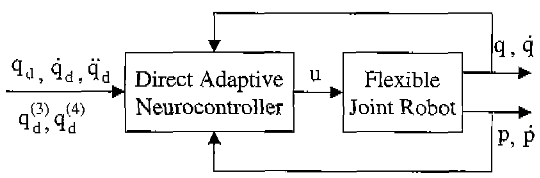

Fig. 4. Block diagram of the direct adaptive neural networks control system.

In order to evaluate the proposed control algorithms, simulation were performed based on the dynamic model of the flexible joint robot. The terms in equation (1) are determined as follows:

$$A(q) = \begin{bmatrix} (a_1 + 2a_3 c_2) & (a_2 + a_3 c_2) \\ (a_2 + a_3 c_2) & a_2 \end{bmatrix} \tag{41}$$

$$H(q, \dot{q}) = \begin{bmatrix} -a_3 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) s_2 + B_{l1} \dot{q}_1 \\ a_3 \dot{q}_1^2 s_2 + B_{l2} \dot{q}_2 \end{bmatrix} \tag{42}$$

A set of nominal values of the parameters are $a_1 = 2.087$, $a_2 = 0.084$, $a_3 = 0.216$, $l_1 = 0.4m$, $l_2 = 0.35m$, $I_{m1} = 0.1224 kgm^2$, $I_{m2} = 0.0168 kgm^2$, $B_{m1} = 1.254 Nms/rad$, $B_{m2} = 0.119 Nms/rad$, $K_1 = 125.56 Nm/rad$, $K_2 = 31.27 Nm/rad$, $B_{l1} = 2.041 Nms/rad$, $B_{l2} = 0.242 Nm/rad$, $F_{m1} = 3.5 sgn(\dot{p}_1) Nm$, $F_{m2} = 1.2 sgn(\dot{p}_2) Nm$, $d_m = 0.018m$. The relation between the joint angles and the end-effect position is:

$$x = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \tag{43}$$

$$y = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \tag{44}$$

In the following, we assume that the model and the above parameters of the flexible joint robot are unknown. The direct adaptive neural networks control laws (34) and (35) with equation (37), (38), and (39) as the learning algorithms are applied to control the end-effect trajectory. Due to their simplicity, the Gaussian radial basic function networks are used to model the robot dynamics. Some a priori information about the structure design. To ensure the approximation precision and task space coverage, 2000 neurons are used in the neural networks controller.

The simulation is conducted to track the following circle trajectory:

$$x_d = 0.64 + 0.8\cos(\omega t)[m] \tag{45}$$

$$y_d = 0.8\sin(\omega t)[m] \tag{46}$$

The sampling period is 2ms, The actual end-effector trajectory of the robot the learning stage is plotted in Figure 5. After about 1400 steps of learning, the tracking error becomes less than 3mm. after the learning converges, the tracking error remain small as shown in Figure 6. The simulation results show that the direct adaptive neural networks control scheme is feasible and robust to the neural networks representation errors.

The second simulation is conducted to track the rectangle trajectory with its four verties as (0.69, 0.2), (0.54, 0.2), (0.54, -0.2), (0.69, -0.2). The sampling period is 2ms. The actual end-effector trajectory of the robot during the learning stage is plotted in Figure 7. After about 2000steps of learning, the tracking error becomes less than 2mm. After the learning converges, the tracking error remains small as shown in Figure 8.

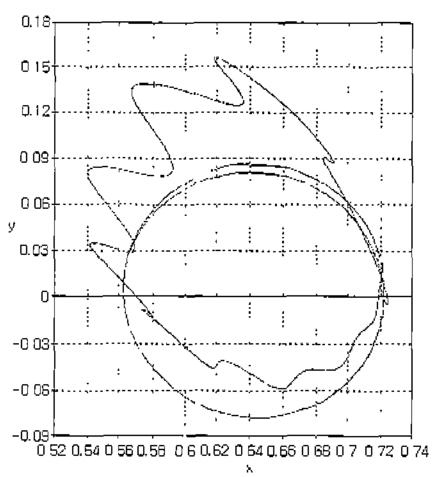

Fig. 5. The circle trajectory tracking: learning stage.

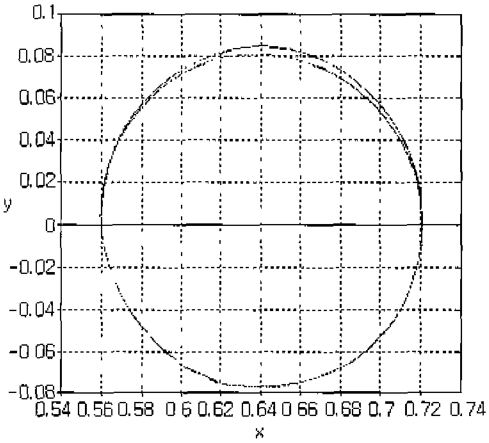

Fig. 6. The circle trajectory tracking: working stage.

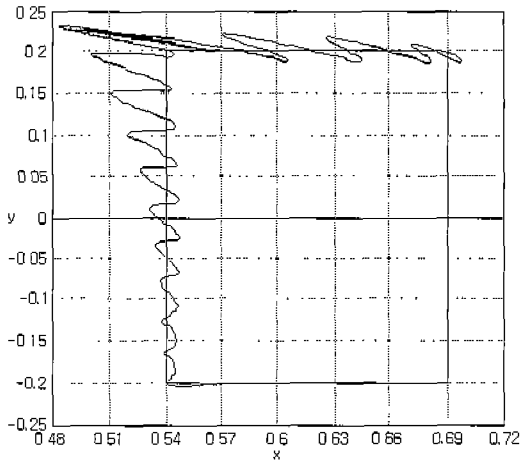

Fig. 7. The rectangle trajectory tracking: learning stage.

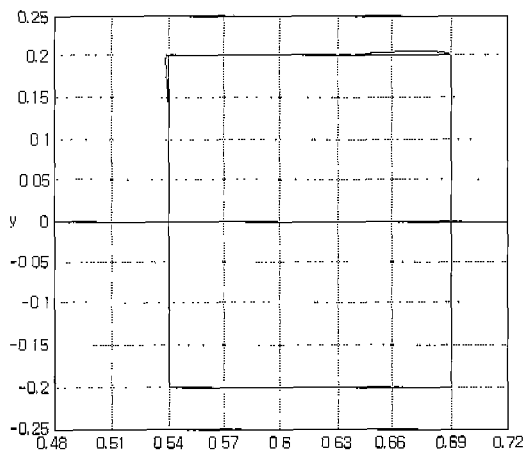

Fig. 8. The rectangle trajectory tracking: working stage.

IV. Conclusions

This paper presents a novel direct adaptive neural networks control scheme for general flexible joint robots. The direct adaptive neural networks controller is robust to the representation errors of the neural networks with a finite number of neurons and bounded additive external disturbance.

Further research should be made along the lines of providing the effects of the direct adaptive neural networks control scheme by consecutive simulation and experiments.

References

[1] E. I. Rivin, "Effective rigidity of robot structure: Analysis and enhancement," in Proc. *Amer. Control Conf.* (Boston, MA), pp. 381-382, 1985.

[2] M. W. Spong, "Modeling and control of elastic joint robots," ASME J. Dynam. Syst. Meas. Control., vol. 109, pp. 310-319, 1987.

[3] R. A. Al-Ashoor, R. V. Patel, and K. Khorasani, "Trajectory following robust adaptive control of flexible-joint manipulators", *IEEE Trans. Syst.*, Man, and Cybern., vol. 23, pp. 589-602, 1993.

[4] H. Asada and S. Liu, "Transfer of human skills to neural net robot controller", *Proc. IEEE Int.* Conf. on Robotics and Automation, Sacramento, CA, pp. 2442-2448, 1991.

[5] X. Cui and K. G. Shin, "Intelligent Coordination of Multiple Systems with Neural Networks", *IEEE Trans. Syst.*, Man, Cybern., vol. 21, no. 6, pp. 1488-1497, 1991.

[6] M. Sekiguchi, T. Sugasaka and S. Nagata, "Control of Multivariable System by a Neural Network", *IEEE International Conf.* on Robotics and Automation, pp. 2644-2649, 1991.

In-Yong, Lee

In-Yong, Lee was born Agust 10, 1958. He received the B.S. degree in Department of Electronic Engineering from Pukyung National University and received the M.S. degree in Department of Electronic Engineering from Dong-A University, Pusan, Korea, in 1987 and 1993. He is currently pursuing the Ph.D degree at Dong-A University. Since 1996, he has been president and CEO of Samjin Tech Co.,Ltd.. His research interests are Neuro-Fuzzy and Adaptation Algorithm, Intelligent Control and Robot System and Fault-Tolerance Control etc. He is a member of KIPE, KIEE, ICASE, IEEK and KFIS.

Phone : +82-51-818-0571
Fax : +82-51-818-7052
E-mail : samjintech@jsamjintech.com

Han-Ho Tack

Han-Ho Tack was born july 6. 1959. He received the B.S. degree in Department of Electronic Engineering from Pukyung National University, Pusan, Korea, in 1987 He received the M..S. degree in Department of Electronic Engineering from Dong-A University, Pusan, Korea, in 1992. He received Ph. D. degree in Department of Electronic & Communication Engineering from the Korea Maritime University, Pusan, Korea, in 1998. From 1987 to 1989, he was a Researcher at the Laboratory of Hung Chang Co. Ltd. Since 1991, he has been a faculty member of the Electronic Engineering at the Chinju National University, where he is currently a associate Professor. His research interests are Neural Network, Fuzzy System, Robotics, Factory Automation, Mechanical Vibration, Trans-fortation, and Multimedia System etc. He is a member of IEEE, KIMISC, KMS, KIEE, and KFIS.

Phone : +82-55-751-3332
Fax : +82-55-751-3339
E-mail : fmtack@cjcc.chinju.ac.kr

Sang-Bae Lee

vol. 8, no. 7, Reference
Present : Processor, Department of Electronic & Communication Engineering, Korea Maritime University, Pusan, Korea,

Phone : +82-51-410-4317
E-mail : leesb@hanara.kmaritime.ac.kr

Boo-Kwi, Choi

Boo-Kwi Choi was born December 19, 1937. He received the B.S. degrees in Department of Electronical Engineering from Yonsei University and received the M.S degrees in Department of Electronic Enginnering from Kyung-Hee University, Seoul, Korea, in 1960 and 1980 respectively. Since 1970, he has been a faculty member of the Electronic Engineering at the Dong-A University, where he is currently a Professor. His research interests are Artificial Intelligent control System, Neuro-Fuzzy and Adaptation Algorithm, Intelligent Control and Robot System etc. He is a member of KIEE, ICASE, KICS, KIIEE, and KFIS.

Phone : +82-51-200-7704
Fax : +82-51-200-7712
E-mail : bkchoi@daunet.donga.ac.kr