International Journal of Fuzzy Logic and Intelligent Systems, vol. 7, no. 2, June 2007 pp. 96-101

An Adaptive Tracking Control for Robotic Manipulators based on RBFN

Min-Jung Lee[1] and Tae-Seok Jin[2]

1 RIC for Ubiquitous Appliance, Dongseo University, Busan, 617-716, Korea
2 Dept. of Mechatronics Eng., Dongseo University, Busan, 617-716, Korea

Abstract

Neural networks are known as kinds of intelligent strategies since they have learning capability. There are various their applications from intelligent control fields; however, their applications have limits from the point that the stability of the intelligent control systems is not usually guaranteed. In this paper we propose an adaptive tracking control for robot manipulators using the radial basis function network (RBFN) that is a kind of neural networks. Adaptation laws for parameters of the RBFN are developed based on the Lyapunov stability theory to guarantee the stability of the overall control scheme.

Filtered tracking errors between actual outputs and desired outputs are discussed in the sense of the uniformly ultimately boundedness(UUB). Additionally, it is also shown that parameters of the RBFN are bounded. Experimental results for a SCARA-type robot manipulator show that the proposed adaptive tracking controller is adaptable to the environment changes and is more robust than the conventional PID controller and the neuro-controller based on the multilayer perceptron.

Key Words : Neural Network, RBFN, Lyapunov stability, Robot Manipulators, Uniformly Ultimately Boundedness

1. Introduction

Generally, robot manipulators used as industrial automatic elements are known as systems with high nonlinearities that are often unknown and time-varying. Therefore, if we want to design a controller for robot manipulators, we should consider that the exact trajectory tracking performance for reference input and the robustness for the existence of system's nonlinearites and external disturbances. The conventional feedback controllers, including proportional integral derivative (PID) controller, are commonly used in the field of industries because their control architectures are very simple and easy to implement. But when these conventional feedback controllers are directly applied to nonlinear systems, they suffer from the poor performance and robustness due to the unknown nonlinearities and the external disturbances. During decades, various control strategies to deal with the unknown nonlinearities and the external disturbances are proposed such as automatic tuning of PID control, variable structure control known as nonlinear robust control, feedback linearization, model reference adaptive control, direct adaptive control, and intelligent control approaches, etc [1],[2],[3],[4].

An adaptive control strategy has found many applications in such areas as robot manipulators, ship steering, aircraft control, and process control, because it can continuously adjust parameters of a controller to accommodate changes in system dynamics and disturbances [3]. In these applications, an on-line adaptation law is usually used to estimate the unknown parameters of the system and then, an appropriate controller is designed to control the plant to satisfy a desired performance. To apply an adaptive control method to robot manipulators as a main controller, the a priori knowledge about the robot manipulator is required: the linearity about unknown parameters and skew-symmetry features, etc. Additionally it takes a long time to calculate the regression matrix used in the algorithm.

Recently, neural networks that are kinds of intelligent control strategy have been investigated to control a nonlinear system because they have a feature of making a flexible controller for unknown nonlinearities by learning process. But there is a problem that neural network as well as intelligent control is difficult to guarantee the stability of control systems mathematically. Therefore, some researchers try to connect neural network to a conventional controllers and to guarantee the stability also.

These approaches are that Naraendra and Parthasarthy discussed identification and control of dynamical system using neural networks. Khalil et al. and Abido et al. designed the output feedback controller using radial basis function networks (RBFN). Choi et al. presented a neural network compensator for conventional control systems to improve the control performance without hardware modifications. Jagannathan et al. as well as some researchers showed good tracking performance through a Lyapunov stability approach in their model reference adaptive control using multilayer neural networks. And Slotine et al. approached the direct adaptive control using Gaussian

Manuscript received Apr. 3, 2007; revised Jun. 11, 2007.

networks. But these studies are used neural networks for compensate for the unknown nonlinearities. Lewis et al. tried to connect neural networks to direct adaptive controller for robot manipulators, but they used lots of multilayer neural networks to approximate inertia matrix and Coriolis/ Centrifugal matrix. In case of multilayer neural networks, the structures are complicated and it takes long time to compute the output of multilayer neural networks. Lee et al. applied an adaptive neurocontroller based on the RBFN to robot manipulators [5],[6],[7],[8],[9],[10],[11],[12],[13], [14],[15],[16],[17][18].

In this paper, the RBFN is used as a tracking controller of robot manipulators [9],[10],[11]. The proposed controller has a parallel structure that consists of a fixed gain PD controller and a RBFN controller. The PD controller is used to control robot manipulators during the initial learning stage of RBFN. The role of the PD controller is reduced after the learning stage of the RBFN. The on-line learning method is employed to adjust parameters of the RBFN. Adaptive law is used to update parameters of the RBFN and constructed to guarantee the stability of the total control system on the basis of Lyapunov stability theory.

By comparing with the experimental results for a SCARA type robot manipulator about two different control strategies, such as PID controller and multilayer neural networks with backpropagation, we proved the validation that the proposed adaptive tracking controller for robot manipulators is more adaptable to the unknown nonlinearities and the external disturbances than two difference controllers.

2. Dynamics and Structural Properties of Robot Manipulators

For control design purpose, it is necessary to have a mathematical model that reveals the dynamical behaviour of a system. Therefore, we derive the dynamical equations of motion for robot manipulators and we summarize the structural properties of robot dynamics.

Using the Euler-Lagrangian formulation, the equations of motion of an n-degree-of-freedom manipulator can be written as [1],[2][3], [11]

$$D(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+\tau_d=\tau \tag{1}$$

Where $q\in R^n$ is the generalized coordinates (joint position); $D(q)\in R^{n\times n}$ is the symmetric, bounded positive definite inertial matrix; vector $C(q,\dot{q})\in R^n$ presents the Coriolis and Centrifugal torques; $G(q)\in R^n$ is the vector of gravitational torques; $\tau_d\in R^n$ is the disturbance torque vector; and $\tau\in R^n$ is the vector of applied joint torques.

The dynamics of robot manipulators in the form of (1) is characterized by the following structural properties.

Property 1: An inertial matrix D is symmetric and positive definite and there exist scalars m_1 and m_2, such that $m_1 I\le D(q)\le m_2 I$.

Property 2: The Coriolis/Centrifugal matrix $C(q,\dot{q})$ is bounded by $c_b(q)\|\dot{q}\|^2$ with $c_b(q)\in C^1(S)$. S is a simply connected compact set of R^n.

Property 3: The matrix $\dot{D}-2C$ is skew-symmetric, so the matrix is satisfied with $x^T(\dot{D}-2C)x=0$ for $\forall x\in R^n$.

Property 4: The norm of the unknown disturbance τ_d has a positive upper bound b_d.

3. Radial Basis Function Network

The RBFN proposed by Moody, Darken, Powell, Broomhead and Lowe, is used to approximate a non-linear function and has a faster convergence time than the multilayer neural networks. Additionally, The RBFN has a similar feature to the fuzzy inference system. First, the output value is calculated using the weighted sum or weighted average method. Second, the number of hidden layer's node of the RBFN is the same as the number of if-then rules in the FIS. Third, the radial basis functions are similar to the membership functions of FIS' premise part [4]. Fig. 1 shows the structure of the RBFN. The RBFN consists of hidden layer and output layer. The number of hidden layers is determined by designer. Gaussian function, triangular function and trapezoidal function are usually employed as basis functions of the RBFN. Gaussian function is frequently used as a basis function.

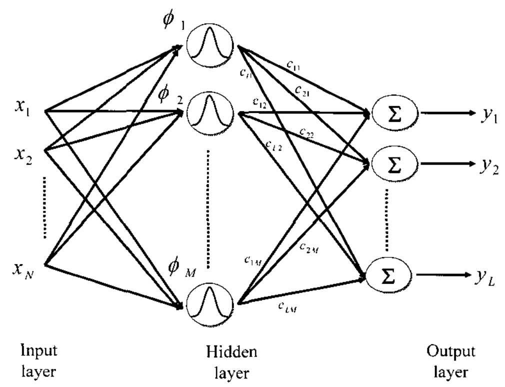


Fig. 1. Architecture of the RBFN with M receptive file units

If we select the Gaussian function as the basis function and use the weighted sum method to calculate the output of the RBFN, then the output becomes

$$y_i=\sum_{j=1}^{M}c_{ij}\phi_j \qquad i=1,2,\cdots,l \tag{2}$$

$$\phi_j(x)=\exp\left(\frac{\|x-u_j\|^2}{\sigma_j^2}\right) \tag{3}$$

where M and l are the number of node and output, respectively. c_{ij} is the j-th weight of RBFN, $\phi_j(x)$ is the j th basis function, $u_j \in R^m$ is the j th center vector and $\sigma_j \in R^m$ is the j th standard deviation.

And the RBFN is used to design adaptive tracking controller for robot manipulators in this paper because the structure is simpler than multiplayer perceptron and its mathematical expression can be shown the linearity about the connection weights. As the design of adaptive control, radial basis function is used as nonlinear approximator. In general, the approximated system motel can be described as

$$y = c^T\Phi + \varepsilon \tag{4}$$

where

$$y = \begin{bmatrix} y_1 & y_2 & \cdots & y_l \end{bmatrix}^T \quad \Phi = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_l \end{bmatrix}^T$$

$$c^T = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{l1} & c_{l2} & \cdots & c_{lm} \end{bmatrix}^T \quad \varepsilon = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_l \end{bmatrix}^T$$

In (4), there are approximation errors, ε, because we just considered the finite dimensional hidden nodes of RBNF. But approximation errors, ε, have very small values and their norm value is bounded by a known constant value according to the approximation theorem [5],[11], such that

$$\|\varepsilon\| \le \varepsilon_N \tag{5}$$

We will use (5) to calculate the control gains.

4. Design of an Adaptive Tracking Controller

4.1 Architecture

This paper attempts to connect neural networks to adaptive tracking control schemes in order to solve the difficult problems such as the stability in neural network control systems and the requirement of the model structure in the adaptive control scheme. We choose the RBFN since its architecture is simple and mathematically tractable. Fig. 2 shows the structure of the adaptive tracking controller. The proposed adaptive tracking controller consists of two parts: a nonlinear function approximator and an auxiliary controller. In the nonlinear function approximator, the RBFN represents the nonlinear robot dynamics expressed in terms of the filtered tracking errors. The adaptation laws for updating the weights of the RBFN and the centers and widths of Gaussian functions are derived to guarantee the stability of control system. Next we have the auxiliary controller to guarantee the stability and robustness under the existence of nonlinearities and external disturbances.

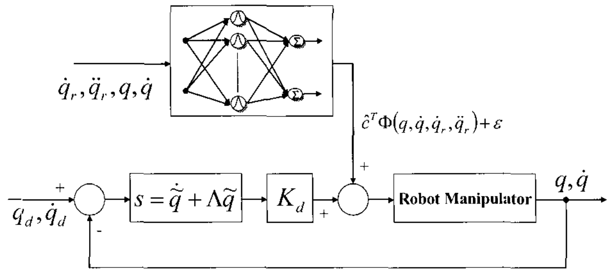


Fig. 2. Structure of the proposed controller

4.2. Stability and Robustness

If the reference trajectory $q_d \in R^n$ is given, the tracking error $\tilde q$ is defined as

$$\tilde q = q_d - q \tag{6}$$

The filtered tracking errors s and the control input τ are also defined as

$$s = \dot{\tilde q} + \Lambda\tilde q = \dot q_r - \dot q \tag{7}$$

$$\tau = \hat c^T\Phi + K_d s \tag{8}$$

Where $x = [q, \dot q, \dot q_r, \ddot q_r]$, $\dot q_r = \dot q_d + \Lambda\tilde q$, $\Lambda = \Lambda^T > 0$, and K_d is diagonal and positive definite.

According to the Lyapunov stability analysis, the system is stable if the Lyapunov function is positive definite and its derivative is negative semi-definite. Therefore, to guarantee the stability of the total control system, a positive-definite Lyapunov function candidate function is selected as follows:

$$V = \tfrac{1}{2}s^T D s + \tfrac{1}{2}tr\left(\tilde c^T\Gamma_1^{-1}\tilde c\right) + \tfrac{1}{2}tr\left(\tilde u^T\Gamma_2^{-1}\tilde u\right) + \tfrac{1}{2}tr\left(\tilde\sigma^T\Gamma_3^{-1}\tilde\sigma\right) \tag{9}$$

where $\tilde c = c^* - \hat c$ is an error between the optimal weight c^* and estimated weight $\hat c$ of RFFN in (8). $\tilde u = u^* - \hat u$ and $\tilde\sigma = \sigma^* - \hat\sigma$ are a center error and a standard deviation error in (3), respectively. Γ_1, Γ_2, and Γ_3 are diagonal, symmetric, and positive-definite matrices, and $tr(\bullet)$ denotes trace.

Differentiating (9) with respect to time, we get

$$\dot V = s^T D\dot s + \tfrac{1}{2}s^T\dot D s - tr\left(\tilde c^T\Gamma_1^{-1}\dot{\hat c}\right) - tr\left(\tilde u^T\Gamma_2^{-1}\dot{\hat u}\right) - tr\left(\tilde\sigma^T\Gamma_3^{-1}\dot{\hat\sigma}\right) \tag{10}$$

If we differentiate s with respect to time, the robot dynamics (1) may be written in terms of the filtered tracking errors as follows:

$$D\dot s = D\ddot q_r + C\dot q_r + \tau_d - \tau - Cs \tag{11}$$

Substituting (11) into (10), we have

$$\dot V = \tfrac{1}{2}s^T\left(D\ddot q_r + C\dot q_r + \tau_d - \tau\right) + \tfrac{1}{2}s^T\left(\dot D - 2C\right)s - tr\left(\tilde c^T\Gamma_1^{-1}\dot{\hat c}\right) - tr\left(\tilde u^T\Gamma_2^{-1}\dot{\hat u}\right) - tr\left(\tilde\sigma^T\Gamma_3^{-1}\dot{\hat\sigma}\right) \tag{12}$$

Using the property 2, $s^T(D-2C)s = 0 \;\; \forall s \in R^n$ (6) and approximating the filtered robot dynamics using the RBFN with finite hidden nodes as shown in (4), (12) becomes

$$\dot{V} = -s^T K_d s + tr\left\{\tilde{c}^T\left(\Phi s^T - \Gamma_1^{-1}\dot{\hat{c}}\right)\right\} + s^T\left(\tau_d + \varepsilon\right) - \tfrac{1}{2}tr\left(\tilde{u}^T \Gamma_2^{-1}\dot{\hat{u}}\right) - \tfrac{1}{2}tr\left(\tilde{\sigma}^T \Gamma_3^{-1}\dot{\hat{\sigma}}\right) \tag{13}$$

Since it is desired to have $\dot{V}$ at least negative semi-definite, let us have the following adaptation laws:

$$\dot{\hat{c}} = \Gamma_1 \Phi s^T \tag{14}$$

$$\dot{\hat{u}} = -\alpha \Gamma_2 \|s\| \hat{u} \tag{15}$$

$$\dot{\hat{\sigma}} = -\beta \Gamma_3 \|s\| \hat{\sigma} \tag{16}$$

Then, (13) becomes

$$\dot{V} = -s^T K_d s + s^T\left(\tau_d + \varepsilon\right) + \alpha\|s\| tr\left\{\tilde{u}^T\left(u^* - \hat{u}\right)\right\} + \beta\|s\| tr\left\{\tilde{\sigma}^T\left(\sigma^* - \hat{\sigma}\right)\right\} \tag{17}$$

From the property of the Frobenius norm [11],

$$tr\left\{\tilde{u}^T\left(u^* - \hat{u}\right)\right\} \le \left\langle \tilde{u}, u^* \right\rangle_F - \|\tilde{u}\|_F^2 \le \|\tilde{u}\|_F \|u^*\|_F - \|\tilde{u}\|_F^2 \tag{18}$$

$$tr\left\{\tilde{\sigma}^T\left(\sigma^* - \hat{\sigma}\right)\right\} \le \left\langle \tilde{\sigma}, \sigma^* \right\rangle_F - \|\tilde{\sigma}\|_F^2 \le \|\tilde{\sigma}\|_F \|\sigma^*\|_F - \|\tilde{\sigma}\|_F^2 \tag{19}$$

Substituting (18) and (19) into (17), we get

$$\dot{V} \le -K_{d\min}\|s\|^2 + \|\tau_d + \varepsilon\|\|s\| + \alpha\|s\|\|\tilde{u}\|_F\left(u_{\max} - \|\tilde{u}\|_F\right) + \beta\|s\|\|\tilde{\sigma}\|_F\left(\sigma_{\max} - \|\tilde{\sigma}\|_F\right) \tag{20}$$

Where $u_{\max}$ and $\sigma_{\max}$ are the maximum values of the Frobenius norm of the center and standard deviation vector in the RBFN.

The approximation error term ε is limited by the upper bound ε_N as shown in (5). If gain K_d is selected to satisfy the following inequality:

$$\|s\| \ge \frac{\|\tau_d + \varepsilon\| + \frac{1}{4}u_{\max} + \frac{1}{4}\sigma_{\max}}{K_{d\min}} \tag{21}$$

then, we have

$$\dot{V} \le 0 \tag{22}$$

From (9) and (25), the control system is stable based on the Lyapunov stability.

5. Experimental Results

To prove the validation of the performance of the proposed controller, we adopt the SCARA type robot manipulator as our test-bed.

Fig. 3 shows the configuration of the robot system. The setup consists of an IBM-PC computer, a DSP processor board equipped with a TMS320C40 chip to calculate the control inputs on-line, a DIO board to acquire the error signals and position data, and a D/A board to send command signals to robot manipulators.

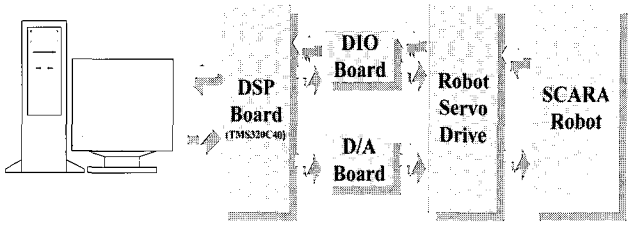


Fig. 3. Configuration of the robot system

And we did the experiments for two different control strategies to compare with the proposed controller. The first control scheme is a PID type conventional control method, with the gains such as K_{p1}=24.23, K_{I1}=7.27, K_{D1}=0.60, K_{P2}=39.9, K_{I2}=8.34 and K_{D2}=1.78 for the joint 1 and 2. And the second control scheme is multilayer neural networks with backpropagation algorithms for updating the connection weights of neural networks. The second control scheme has the same architecture as the proposed controller but the updating law is different from the proposed controller. To implement the proposed controller, we choose the control parameters shown in Table 1. And the number of node of the RBFN is set to be 10. Sampling time is 5[msec].

Table 1. Parameters of the Proposed Controller

Control Parameters	Values
λ	[56 60]T
Kd	[4 3]T
Γ	diag(0.018,0.018)
α, β	0.15, 0.005

And to compare the performance with the different control strategies, we considered three control conditions. The first condition is that the reference trajectories for the manipulator are given as q_{d1}=0.4cos(3.76t)[rad] and q_{d2}=0.4{sin(3.76t)+1}[rad] for the joints 1 and 2, respectively. Fig. 4 shows the tracking errors of the PID controller, multilayer neural networks with backpropagation and the proposed controller. In Fig. 4, the errors multilayer neural network controller is decreasing because the connection weights are updating on-line. But the decreasing rate of the errors are very slow than the proposed controller.

The second condition is that the new reference input trajectories are given q_{d1}=0.4cos(5.65t)[rad] and q_{d2}=0.4{sin(5.65 t)+1}[rad] for the joint 1 and 2 in order to consider the environment change. Fig. 5 shows the tracking errors of three controllers mentioned above. We can find that the performance of the PID controller is deteriorated. However, the performance of the multilayer neural network controller and the proposed controller are still good due to the learning process. The tracking errors of multilayer neural network control slowly approach the similar magnitude of the tracking errors of proposed method.

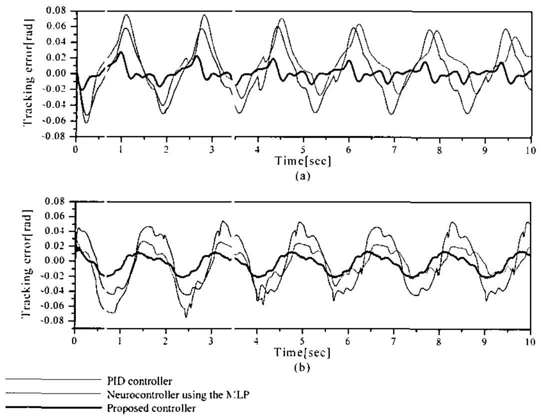


Fig. 4. Tracking errors under reference inputs with $\omega = 3.76\, rad/\mathrm{sec}$ (a) Joint 1 (b) Joint 2

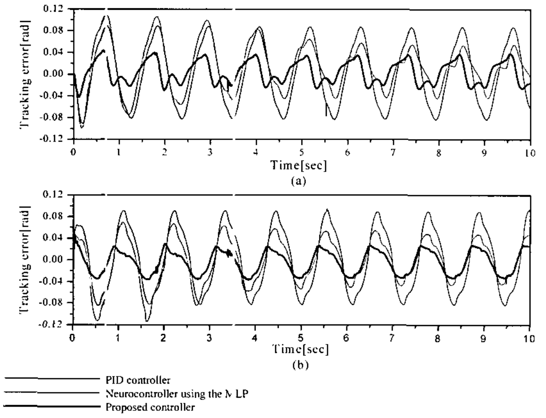


Fig. 5. Tracking errors under reference inputs with $\omega = 5.65\, rad/\mathrm{sec}$ (a) Joint 1 (b) Joint 2

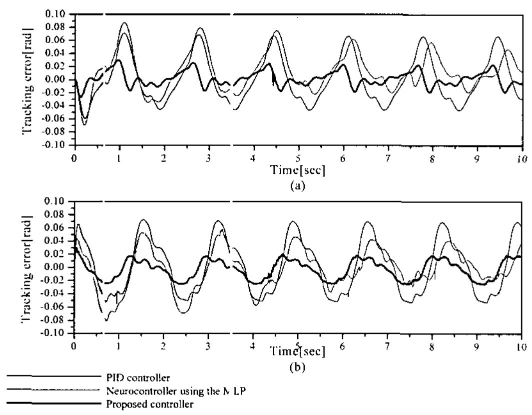


Fig. 6. Tracking errors under disturbance. (a) Joint 1. (b) Joint 2

The final condition is the existence of 4kg load under the first condition. Fig. 6 shows the tracking errors of three different controllers. In Fig. 6, the joint 2 errors of PID controller is affected by the load and the tracking errors of controllers with learning laws are reduced during the learning process.

Experimental results mentioned above show that the proposed controller is more adaptable to the environment changes and is more robust than the conventional PID controller and multilayer neural network controller with backpropagation algorithms.

6. Conclusions

This paper presents a adaptive tracking controller for robot manipulators based on the RBFN. The proposed controller has a parallel structure that consists of PD controller with a fixed gain and the RBFN. And the parameters (weight, center of basis function, and standard deviation) of the RBFN are adjusted with on-line method. The learning law is constructed on the basis of Lyapunov stability theory.

This paper shows that the tracking errors are bounded uniformly and ultimately under the existence of the disturbances and modeling error, mathematically. The SCARA type robot manipulator is employed as a test-bed to apply the proposed controller. And we compared the proposed controller with two different controllers. The experimental results show that the proposed controller is adaptable to the environment changes and is more robust that the conventional PID controller and the multilayer neural network controller with backpropagation algorithms.

References

[1] R. K.S.Fu, R.C.Gonzalez, and C.S.G.Lee : *Robotics*, McGraw-Hill International Editions, 1987.

[2] M.W.Spong and M.Vidyasagar : *Robot Dynamic and Control*, John Wiley & Sons, 1989.

[3] J.-J.E.Slotine and W. Li : *Applied Nonlinear Control*, Prentice Hall, 1991.

[4] J.-S.R.Jang, C.-T.Sun, and E.Mizutani : *Neuro-Fuzzy and Soft Computing*, Prentice Hall, 1997.

[5] K.S.Narendra and K.Parthasarathy : "Identification and control of dynamical systems using neural networks," *IEEE Trans. On Neural Networks*, vol. 1, no. 1, March 1990.

[6] Sridhar Seshagiri and Hassan K. Khalil : "Output feedback control of nonlinear systems using rbf neural networks," *IEEE Trans. on Neural Networks*, vol. 11, no. 1, January 2000.

[7] M.A.Abido and Y.Abdel-Magid : "On-line identification of synchronous machines using radial basis function neural networks," *IEEE Trans. on Power Systems*, vol. 12, no. 4, November 1997.

[8] Y.-K. Choi, M.-J. Lee, S. Kim, and Y.-C. Kay : "Design and implementation of an adaptive neural-network compensator for control system," *IEEE Trans. on Industrial Electronics*,

vol. 48, no. ?, Apr. 2001.

[9] S. Jagannathan, F. L. Lewis and O. Pastravanu : "Model reference adaptive control nonlinear dynamical systems using multilayer neural networks," in *Proc. of IEEE Int. Conf. on Neural Networks*, vol. 7, pp. 4766-4771, 1994.

[10] A.S.Morris and S.Khemaissia : "A neural network based adaptive robot controller," *Journal of Intelligent and Robotic Systems*, vol. 15, pp. 3-10, 1996.

[11] D.Y.Meddah and A.Benallegue : "A stable neuro-adaptive controller for rigid robot manipulators," *Journal of Intelligent and Robotic Systems*, vol. 20, pp. 181-193, 1997.

[12] R.Carelli and E.F.Camacho : "A neural network based feedforward adaptive controller for robots," *IEEE Trans. On Systems, Man and Cybernetics*, vol. 25, no. 9, pp. 1281-1288, September 1995.

[13] M.Zhihong, H.R.Wu, and M.Palaniswame : "An adaptive tracking controller using neural networks for a class of nonlinear systems," *IEEE Trans. on Neural Networks*, vol. 9, no. 5, September. 1998.

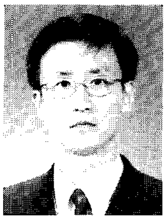
[14] Robert M. Sanner and Jean-Jacques E. Slotine : "Gaussian networks for direct adaptive control," *IEEE Trans. on Neural Networks*, vol. 3, no. 6, November 1992.

[15] H. D. Patirio and Derong Liu : "Neural network-based model reference adaptive control system," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 30, no. 1, February 2000.

[16] R. M. Scanner and J.-J. E. Slotine : "Gaussian networks for direct adaptive control," *IEEE Trans. on Neural Networks*, vol. 3, Nov. 1992.

[17] Frank L. Lewis, Kai Liu, and Aydin Yesildirek : "Neural-net robot controller with guaranteed tracking performance," *IEEE Trans. on Neural Networks*, vol. 6, no. 3, May 1995.

[18] M.-J. Lee and Y.-K. Choi : "An Adaptive Neurocontroller Using RBFN for Robot Manipulators," *IEEE Trans. on Industrial Electronics*, vol. 51, no. 3, June 2004.



Min-Jung Lee

He received the B.S. degree from Pukyong National University, Busan, Korea, in 1996, and the M.S. and Ph. D. degrees from Pusan National University, Busan, Korea, in 1998 and 2001, respectively, all in electrical engineering.

He was a Lecturer at Kyungnam College of Information and Technology, Busan, Korea, from 2003 to 2004, and a Principal Researcher at Kyungnam Technopark, Changwon, Korea, from 2005 to 2006. He is currently a Research Professor in the Dongseo University. His research interests are intelligent control, nonlinear control, robotics, power electronics, and home network.

Phone : +82-51-320-1936
Fax : +82-51-320-1940
E-mail : mnjlee@gdsu.dongseo.ac.kr

Tae-Seok Jin

He received the Ph.D. degrees from Pusan National University, Busan, Korea, in 2003, in electronics engineering.

He is currently a full-time lecturer at Dongseo University. From 2004 to 2005, he was a Postdoctoral Researcher at the Institute of Industrial Science, The University of Tokyo, Japan. His research interests include Network sensors fusion, mobile robots, computer vision, and intelligent control. Dr. Jin is a Member of the KFIS, IEEK, ICASE, and JSME.

Phone : +82-51-320-1541
Fax : +82-51-320-1751
E-mail : jints@dongseo.ac.kr