NEURAL NETWORK SENSITIVITY TO INPUTS AND WEIGHTS, AND ITS APPLICATION TO FUNCTIONAL IDENTIFICATION OF ROBOTICS MANIPULATORS

S. Khanmohammadi
Department of Electrical Engineering
University of Tabriz
Tabriz, Iran

Abstract Neural networks are applied to the system identification problems using adaptive algorithms for either parameter or functional estimation of dynamic systems. In this paper the neural networks' sensitivity to input values and connections' weights, is studied. The Reduction-Sigmoid-Amplification (RSA) neurons are introduced and four different models of neural network architecture are proposed and analyzed. A two degree-of-freedom manipulator is considered as a case study and the functional dynamics for computed torque are identified using the proposed models. The simulation results are studied and analyzed for different models.

Key Words NNT Sensitivity, RSA Neurons, Sigma-PI Neurons, Robotic Manipulator, Identification, Computed Torque

چکیده شبکه های عصبی با استفاده از روند خودسازگاری برای بازشناسی سیستمها، تخمین تابعی سیستمهای پویا یا بازشناسی پارامترها، بکار می رود. در این مقاله حساسیت شبکه های عصبی نسبت به مقادیر داده ها و وزن اتصالها مطالعه می شود. نرونهای کاهشی-سیگموید-افزایشی (RSA)، معرفی شده و چهار نوع مدل مختلف برای ساختار شبکه های عصبی پیشنهاد و تجزیه و تحلیل می گردد. یک عمل کننده با دو درجه آزادی بعنوان مطالعه نوعی در نظر گرفته شده و پویش تابعی گشتاور محاسبه شده با استفاده از مدلهای پیشنهادی بازشناسی می شود. نتایج شبیه سازی برای مدلهای مختلف مطالعه و مورد تجزیه و تحلیل قرار می گیرد.

INTRODUCTION

An artificial neural network is a system of interconnected computational elements operating in parallel and arranged in pattern similar to biological nets [1]. Recently the remarkable capability of neural networks has led to their application in many areas of engineering and science. The fact that the neural networks' specially multilayer feed-forward networks (FFN) are widely used is due to the following reasons [2]: i) it can easily be trained by the generalized delta rule (GDR) [1, 3 - 6]; ii) it is able to learn any function with arbitrarily desired accuracy [7,8].

Neural networks have also received attention for their potential applications in the identification and control of adaptive dynamic systems, where various approaches are introduced and studied by researchers. Kumpati S. Narendra and Kannan Parthasaramy have studied different configurations of generalized neural networks in the identification and control of dynamical systems [9], using in their studies a diagrammatic representation of back propagation [10, 11]. H. J. Sira-Ramirez and S. H. Zak have studied the utilization of neural networks for the inverse dynamics identification of systems [12]; and recently D.A. Hoskins, J.N. Hwang and J. Vagners have used an iterative constrained inversion technique to find the control inputs of the plant in an adaptive control system. The interesting work of Yoshiaki Ichikawa and Toshiyuki Sawa on the application of neural networks for direct feedback controllers is represented as a learning algorithm for a neural network whose outputs and inputs are directly connected to the plant like ordinary feedback controllers [13].

One of the important applications of neural networks in adaptive control is the control of robot manipulator to follow a desired trajectory. A considerable number of papers have been published in this field, where different aspects of neural networks' applications for adaptive identification of plants and their controls have been studied [14 - 16].

Neurocomputing techniques are normally applied to the system identification problems using adaptive algorithms for either parameter or functional estimation of dynamic systems [17]. In this work, the sensitivity of neural networks to input values or connection weights, is analyzed. The effects of reductions (amplifications) of the input (output) values of the neurons in the sensitivity of the global network are studied. The analyses are used for the design of neural networks for system identification in robotics. A simple two degree-of-freedom manipulator is considered as a numerical example. It is assumed that the exact values of the computed torques are available for back propagation. This assumption is not in contradiction with the applicability of the methods introduced in this work. However, it leads to a fine and more exact comparison between the proposed models of the neural networks, avoiding the failures due to simulation techniques used for inverse dynamics.

NEURAL NETWORKS' SENSITIVITIES

A neural network is basically composed of many neurons and interconnections with a particular architecture. The basic elements of an artificial neural network are neurons, which are considered as processing units (PUS) where the computations are performed [2]. In the following sections the processing units will be called neurons or units interchangeably.

The neural network models used in this work consist of L layers (L=2) of processing units and an input layer. Each neuron i of the l^{th} layer generates an output, which is a sigmoid function of the weighted sum of the inputs to the neuron, given the following iterative equations:

For $1<i<N_{l+1}$ $1<l<L$

$$u_i(l) = \sum_{j=1}^{N_{l-1}} w_{ij}(l)\, a_j(l-1) + \theta_i(l) \tag{1}$$

$$a_i(l) = \frac{1}{1+e^{-u_i(l)}} \tag{2}$$

where $a_j(l-1)$ is the output of the j^{th} neuron of the $(l-1)^{th}$ layer, $\theta_i(l)$ is the bias value of the i^{th} neuron at the l^{th} layer, and $w_{ij}(l)$ denotes the interconnection weight between the i^{th} neuron at the l^{th} layer and the j^{th} neuron at the $(l-1)^{th}$ layer [18].

A special proposed type of neuron is the Sigma Pi (SP) neuron introduced by F. Crick and C. Asanuma [19]; where the output values of two (or more) units are multiplied before entering into the sum. In this case:

$$u_i(l) = \sum_{j=1}^{m} w_{ij}(l) \prod_{q=1}^{k} a_{jq}(l-1) \tag{3}$$

where, $a_{jq}(l-1)$s are the outputs of the k conjunct neurons connected to the J^{th} gate of the SP unit. If the SP unit contains only one gate, the input value will be simply the weighted product of the outputs of the conjunct units, as follows:

$$u_i(l) = w_i(l) \prod_{q=1}^{k} a_{iq}(l-1) \tag{4}$$

where, w_i denotes the weight of the connection of the conjunct units to i and the $a_{iq}(l-1)$s are the outputs of these units.

In this work, another version of the SP neurons, called the Reduced-Sigmoid-Amplified (RSA) neuron, is introduced. The input value of each RSA neuron is multiplied by the bias value $0<\theta<1$ before entering to the unit; and the output value is amplified by $\alpha>1$. In this way:

$$a_i(l) = \frac{\alpha}{1+e^{-\theta u_i(l)}} \tag{5}$$

The schematic diagram of a RSA unit is displayed in Figure 1.

The output value a_i, computed by Equation 5, is shown in Figure 2 for different values of α and θ. Clearly the sensitivity region of the neuron to the input value is enlarged for the lesser values of θ. It is also seen from Figure 2 that the sensitivity value (shape of the curve) increases for greater values of α.

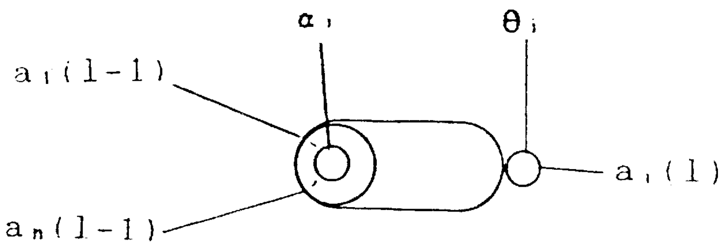

Figure1. The schematic diagram of an RSA neuron

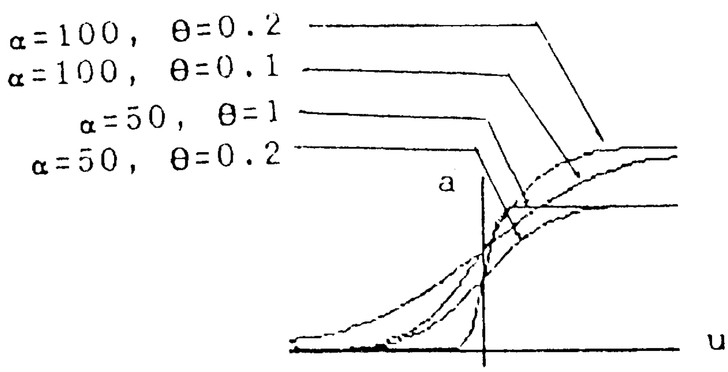

Figure 2. The sensitivity region of an RSA neuron as a function of α and θ

From the analytical view point, in a RSA type neuron with the sigmoid function

$$a = \frac{\alpha}{1 + e^{-\theta u}} \qquad (6)$$

the final value of the output is

$$\lim_{u \to \infty} a = \alpha \qquad (7)$$

and

$$\frac{\partial a}{\partial u} = \frac{\alpha \theta e^{-\theta u}}{(1 + e^{-\theta u})^2} \qquad (8)$$

if a $k\alpha$ ($0<k<1$) portion of the final value is considered as the saturation index, then

$$k\,\alpha = \frac{\alpha}{1 + e^{-\theta \hat{u}}} \qquad (9)$$

or

$$\hat{u} = \frac{1}{\theta} \ln \frac{k}{1 - k} \qquad (10)$$

where $\hat{u}$ is an input value which satisfies the saturation index. The goal is to choose an appropriate small value to θ, to have a large value to $\hat{u}$ before saturation. On the other hand

$$\frac{\partial a}{\partial u}\Big|_{\hat{u}} = k\,(1 - k)\,\alpha\,\theta \qquad (11)$$

That is, if the θ is chosen small, the gradient of the output at $\hat{u}$ decreases; and to save it at a desired value (say δ), α must be chosen proportional to the inverse of θ. That is

$$\alpha = \frac{\Delta}{\theta} \qquad (12)$$

where

$$\Delta = \frac{\delta}{k\,(1 - k)} \qquad (13)$$

One may ask why instead of introducing such neurons we don't use smaller values for the weights of the input connections and greater values for weights of each neuron. There are two reasons: i) the weight of connections will change during back propagation procedures; ii) using the second strategy for the l^{th} layer the sensitivity regions of the neurons of the $(l+1)^{th}$ layer diminishes, be cause of the large values of their inputs.

SENSITIVITY OF NETWORKS TO INPUTS AND CONNECTION WEIGHTS

To study the sensitivity of a neural network consisting of the RSA units, the class of function $\Pi^3_{14,14,14,2}$ is considered. The symbol $\Pi^L_{i_0, i_1, \ldots, i_L}$ denotes a network of L layers with i_0 inputs, i_l outputs and (L-1) sets of hidden layers, where the k^{th} layer contains i_k neurons [9].

The sensitivity curves (input-output relations) of four different models of neural networks, as functions of inputs and weights of interconnections, are shown in Figures 3 and 4. The following models of architectures are considered:

Model 1: Elementary neural network with the activation rule of Equation 2;

Model 2: Activation rule of Equation 5 for all the neurons with the same α and θ.

Model 3: Activation rule of Equation 5, with $\alpha_{(i+1)} > \alpha_i$, and $\theta_{(i+1)} < \theta_i$, where α_i and θ_i represent the α and θ values of the units of the i^{th} layer.

Model 4: Activation rule of Equation 5 with $\alpha_{(i+1)} < \alpha_i$, and $\theta_{(i+1)} > \theta_i$, where α_i and θ_i represent the α and θ values of the units of the i^{th} layer.

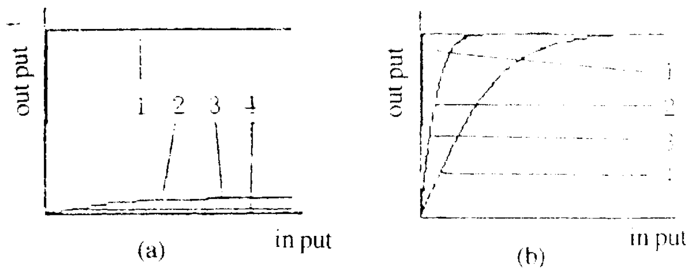

Figure 3. Sensitivity of the network's output to input values. All inputs are considered to have the same values for simplicity. a) Outputs of different models. b) Outputs are normalized to have the same steady state values for a better comparison.

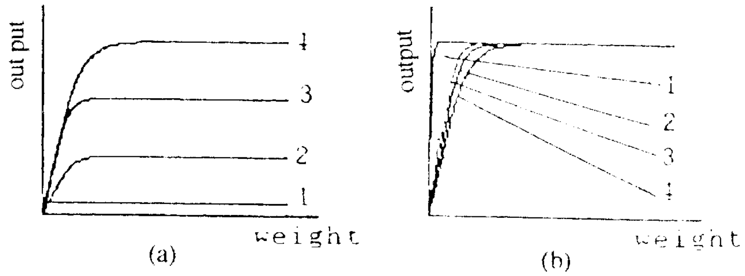

Figure 4. Sensitivity of the output to the weights of connections. All the connections are considered to have the same values for simplicity. a) Outputs of different models. b) Outputs are normalized to have the same steady state values for a better comparison.

As may be seen from Figure 3, in the neural network of the first model, after some increases in the input values, the output reaches its saturatrion value. The 2nd and 3rd models have nearly the same sensitivity shapes; but the 3rd model is sensible to a wider range of input values. The sensitivity range of the 4th model is larger but its shape is smoother than the others.

As it is shown in Figure 4, the sensitivity range of the 3rd model to the weights of connections is wider than that of the other models. The sensitivities of the 2nd and 3rd models are practically the same; but the shape of the sensitivity of the 3rd model is higher than the 2nd one. The sensitivity range of the 4th model is larger but its shape is smoother than the others.

All the four models are learnt to follow a constant output with zero inputs. The results are shown in Figure 5. As can be seen from the figure, in this case, the 3rd model follows the constant output faster than the others but the overshoot phenomenon is significant.

Considering these results it may be predicted that for small ranges of variations of inputs or weights, the 3rd model will respond better than the others; but in the case of some applications where instantaneous oscillations at the start time are not permissible, the 4th model may be preferred.

SIMULATION RESULTS FOR A ROBOT MANIPULATOR

In this section the simulation results of the four neural network models are studied. A simple two degree-of-freedom manipulator, as shown in Figure 6, is considered as a numerical example [20, 21]. This manipulator is modeled as two rigid links of lengths (l_1, l_2) with point masses at the distal ends of the links (m_1, m_2). It moves in a vertical plane with gravity acting. The equations of motions for this manipulator are:

$$\tau_1 = m_2 l_2^2(q_1+q_2) + m_2 l_1 l_2 \cos q_2 (2q_1+q_2) + (m_1+m_2) l_1^2 q_1$$

$$- m_2 l_1 l_2 \mathrm{Sin} q_2^2 - 2 m_2 l_1 l_2 (\sin q_2) q_1 q_2 + m_2 l_2 g \sin(q_1+q_2)$$

$$+ (m_1+m_2) l_1 g \sin q_1 + v_1 q_1 + k_1 \mathrm{sgn}(q_1) \qquad (14)$$

$$\tau_2 = m_2 l_1 l_2 (\cos q_2) q_1 + m_2 l_1 l_2 (\sin q_2) q_1^2 + m_2 l_2 g \sin(q_1+q_2)$$

$$+ m_2 l_2^2 (q_1+q_2) + v_2 q_2 + k_2 \mathrm{sgn}(q_2) \qquad (15)$$

where q_is are the angular positions of links, and v_i and k_i are the viscous and coulomb frictions respectively.

For each of the parameters l, m, k and v, two upper and lower bounds are assumed. These values in conjunction with the desired positions, velocities and accelerations of two links are considered as the inputs of the neural networks.

The position, velocity and acceleration parameters are obtained from the second order polynomial trajectory [22]. Figure 7 displays the desired trajectories for q_1, q_2 and the end-effector movement with the required torques of two links.

The computed torques by the four neural network models are shown and can be compared in Figure 8. The initial values of the connection weights are chosen randomly in [-0.5, 0.5] for all the models. The results obtained by the 3rd model are considerably near to the exact curves of the required torques to follow the desired trajectories.

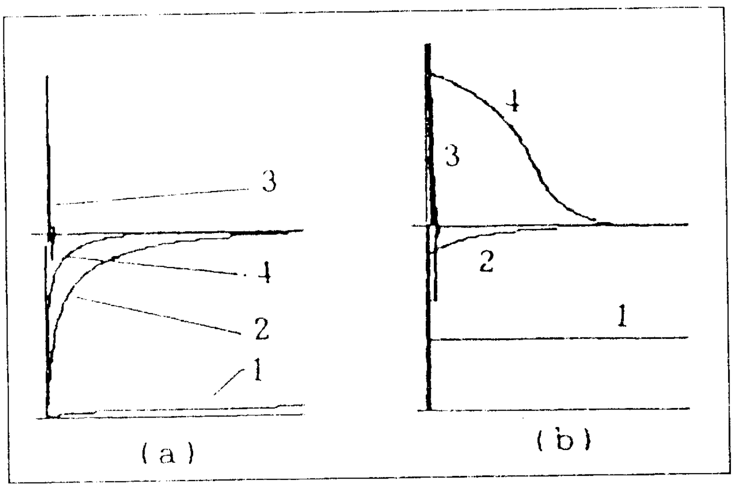

Figure 5. The outputs of different models during the learning phase for a constant output. a) $-0.5<w_{ji}<0.5$ b) $1<w_{ji}<2$

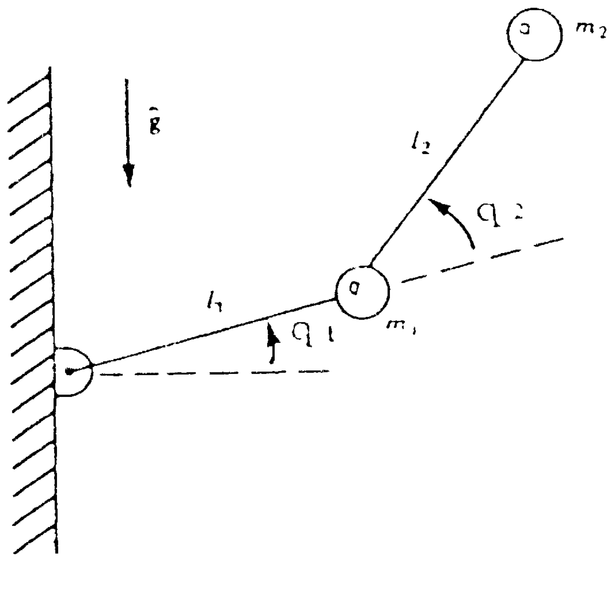

Figure 6. Two degree-of-freedom manipulator, from [21]

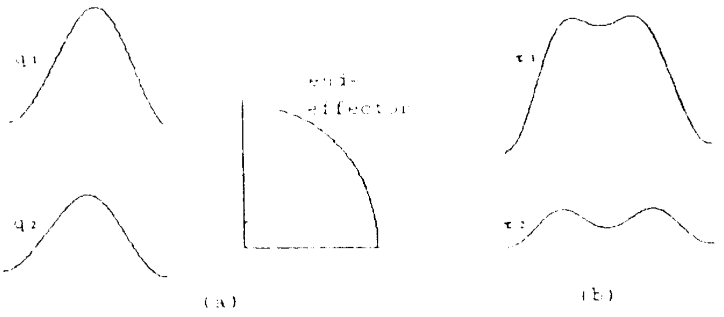

Figure 7. (a) Desired trejectories of q_1, q_2 and end-effector
(b) required torques of two links

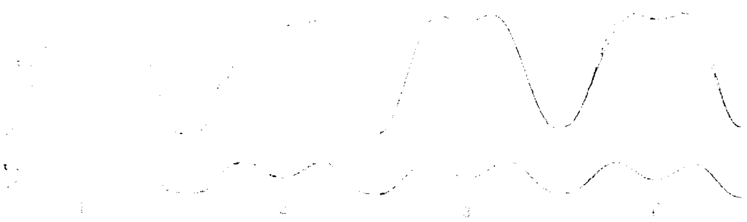

Figure 8. computed torques of different models, compared by the exact curve of necessary torques

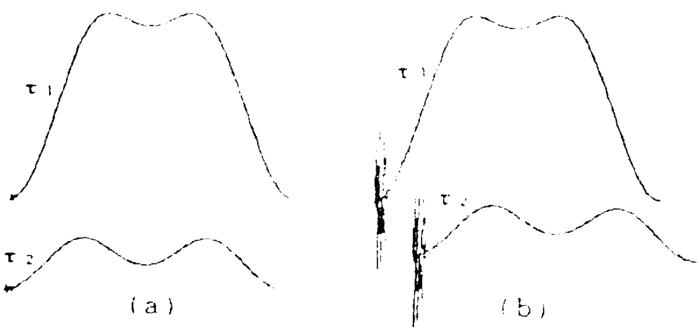

Figure 9. computed torque by the 3rd model. a) Only thedesired values of position, elocities and acceleration are cosidered as the neutal network. b) Initial weights chosen in the range [1,2]

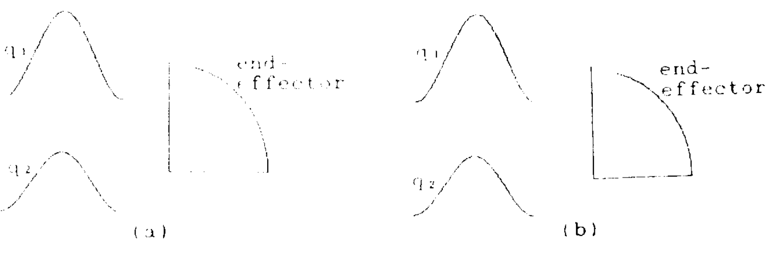

Figure 10. (a) Desired trajectories
(b) The plant's outputs when only the computed torque by the neural network of the 3rd model is applied to the system

The 3rd model was simulated in different conditions and the results were always satisfactory. Figure 9 shows the simulation results for two different conditions.

These results suggests that the 3rd model can be an efficient alternative for the computed torque of robotics manipulators. The computed torque's curve is so near to the exact values that the feedback error will in fact be small and sometimes negligible for feedback controllers, as shown in Figure 10.

CONCLUSIONS

The sensitivity of a simple neuron to the input values was studied. The RSA neuron was introduced and four different models of neural networks' architectures were proposed. The sensitivities of these models to inputs and connection weights were analyzed. A two degree-of-freedom manipulator was considered as a case study and the four models were used as system identifiers (for computed torque) for this system. The simulation results show that the 3rd model is a powerful alternative which responds effectively in different conditions.

REFERENCES

1. D. E. Rumelhart, J. L. Mc Clelland and PDP Research Group, "Parallel Distributed Processing, Explorations in the Microstructure of Cognition", MIT press, 1986 (1988).
2. Si-Zhao Qin, Hong-Te Su and Thomas J. McAvoy, "Comparison of Four Neural Net Learning Methods for Dynamic System Identification": *IEEE trans. on Neural Networks*, Vol. 3, No. 1, January, (1992), pp. 122-130.
3. P. K Simpson, "Artificial Neural Systems Foundations, Paradigms, Applications and Implimentations", Pergamon Press Inc., (1990).
4. R. Hecht-Nielsen, "Theory of Back Propagation Neural Networks", in Proc. IJCMN (Washington, DC), June 18-22, (1989), pp. 593-608.
5. P. J. Webros, "Applications of Advances in Nonlinear Sensitivity Analysis", in *System Modelling and Optimization*, R. F. Drenick and F. Kozin, (Eds.), Berlin, Heidelberg, New York: Springer-Verlay, (1982).
6. P. J. Webros: "Generalization of Back Propagation with Application to a Recurrent Gas Market Model": *Neural networks*, Vol. 1, No. 4, (1988), pp. 339-358.
7. G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function": *Mathematics of Controls, Signals and Systems*, Vol. 2, (1989), pp. 303-314.
8. K. Hornik, M. Stinchcombe and H. White: "Multilayer Feed-Forward Neural Networks and Universal Approximations": *Neural Networks*, Vol. 2, No. 5, (1989), pp. 359-366.

9. 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), pp. 4-27.
10. K. S. Narendara and K. Parthasarathy, "A Diagrammatic Representation of Back propagation", Center for Syst. Sci., Dept. of Electrical Eng., Yale University, New Haven, CT, tech. rep. 8815, Aug., (1988).
11. K. S. Narendra and K. Parthasarathy: "Back Propagation in Dynamical Systems Containing Neural Networks", Center for Syst. Sci., Dept. of Electrical Eng., Yale University, New Haven, CT, tech. rep. 8905, Mar. (1989).
12. H. J. Sira-Ramirez and S.H. Zak, "The Adaptation of Perceptrons with Applications to Inverse Dynamics Identification of Unknown Dynamic Systems": *IEEE Trans. on Systems, man, and Cybernetics*, Vol. 21, No 3, May/June (1991), pp. 634-643.
13. Y. Ichikawa and T. Sawa, "Neural Network Application for Direct Feedback Controllers": *IEEE trans. on Neural Networks*, Vol. 3, No. 2, March (1992), pp. 224-231.
14. T. M. Martinetz, H. J. Ritter and K. J. Schulten, "Three-Dimensional Neural Net for Learning Visuomotor Coordination of a Robot Arm": *IEEE Trans. on Neural Networks*, Vol. 1, No 1, March (1990), pp. 131-136.
15. M. B. Leahy Jr., M.A. Johnson and S.K. Rogress, "Neural Network Payload Estimation for Adaptive Robot Control": *IEEE Trans. on Neural Networks*, Vol. 2, No. 1, January (1991), pp. 93-100.
16. M. Kuperstein and J. Wang, "Neural Controller for Adaptive Movements with Unforeseen Payloads": *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, March, (1990), pp. 137-142.
17. M. F. Tenorio and Wei-Tsih Lee, "Self-Organizing Network for Optimum Supervised Learning": *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, March, (1990), pp 100-109.
18. D. A. Hoskins, J. N. Hwang and J. Vagners, "Iterative Inversion of Neural Networks and its Application to Adaptive Control": *IEEE Trans. on Neural Networks*, Vol. 3, No. 2, March, (1992), pp. 292-301
19. F. Crik and C. Asanuma, "Certain Aspects of the Anatomy and Physiology of the Cerebral Cortex in Parallel Distributed Processing", D. E. Rumelhart, J. L. McCelland, (Eds.), MIT Press, 1986 (1988).
20. J. J. Crage, "*Introduction to Robotics.* Mechanics and Control," Reading, Mass. Addison-Wesley, (1986).
21. J. J. Crage, "Adaptive Control of Mechanical Manipulators", Addison-Wesley, (1988).
22. M. W. Spong and M. Vidyasagar, "Robot Dynamics and Control", John Wiley & Sons, (1989).