Neuro-control and Its Applications to Electric Vehicle Control

Sigeru Omatu*

Graduate School of Engineering, Osaka Prefecture University,
Nakaku, Sakai, 599-8531, Osaka, Japan
omatu@cs.osakafu-u.ac.jp
http://www.osakafu-u.ac.jp

Abstract. Neuro-control which adopts neural network architectures to synthesis of control has been summarized and its application to electric vehicle control is developed in this paper. The neuro-control methods adopted here is based on proportional-plus-integral-plus-derivative (PID) control, which has been adopted to solve process control or intelligent control. In Japan about eighty four per cent of the process industries have used the PID control. Using the learning ability of the neural network, we will show the self- tuning PID control scheme (neuro-PID) and the real application to an electric vehicle control. environment.

Keywords: neuro-control, self-tuning PID, electric vehicle control.

1 Introduction

In applying conventional control theory to practical problems, we have to model the plant or system. The modelling is done by using a set of linear differential or difference equations, in which unknown parameters are included. But the range of applicability is not so wide to cover real control problems. In real world, the plant and its environment are very complex and difficult to be described by such linear models. For example, in a robotic control system, it may have many sensors providing inputs that cannot necessarily be interpreted as state variables. Furthermore, the models of the system may be unknown and interact with unknown changing environments.

Therefore, it is beneficial to consider new methods of control. They may not be so rigorous mathematically so that it can work in a wide range of domains and under more dynamic and more realistic conditions. One of the powerful methods is neuro-control based on the neural networks since the neural networks have preferable properties to overcome the difficult problems stated above. Some of them are 1) learning by experience (training), i.e., human-like learning behavior, 2) generalization ability, i.e., mapping ability of similar inputs to similar outputs, 3) nonlinear mapping ability, 4) parallel distributed processing, allowing fast

* This work was supported by Research Project-in-Aid for Scientific Research (2008) No. 20360178 in JSPS, Japan.

S. Omatu et al. (Eds.): IWANN 2009, Part II, LNCS 5518, pp. 1–12, 2009.

computation for large scale systems, 5) robustness for noise and environmental change, 6) self-organizing property, etc. These properties make neuro-control suitable for applications to real control problems.

In this paper, we will survey the neuro-control architectures developed until now. Then we describe the self-tuning PID control based on neural networks which has been proposed by the authors. After that, we will show the real application to electrical vehicle torque control and speed control problems.

2 Historical Review of Neuro-control

The first neuro-control was discussed by Widrow and Smith [1] who used ADALINE to stabilize and control the pole balancing act. Other early research on neur0-control could also found in Waltz and Fu [2], Michie and Chambers [3], and Barto et al. [4].

Neuro-control research has begun sharp increase around 1987 when the first IEEE Conference on Neural Networks has held in San Diego. These papers have demonstrated that neuro-control methods can be applied successfully to control unknown nonlinear systems while conventional control approaches based on linear dynamical system theory could not solve such control problems. Many neuro-control structures were also proposed. Typical neuro-control methods are 1) feedback error learning by Kawato et al. [5], 2) neuro-internal model control by Hunt and Sbarbaro [6], 3) neuro-predictive control by Willia et al. [7], 4) Forward and inverse modelling by Jordan et al. [8]), 5) generalized and specialized learning by Psaltis et al. [9], 6) Self-tuning neuro-control by Omatu [10]). More information on neuro-control could be obtained by the books by D.A. White and D.A. Sofge [11], W. T. Miller III et al. [12], S. Omatu et al. [13], P.M. Mills et al. [14], and N.W. NG [15].

3 Error Back-Propagation Algorithm

The error back-propagation (BP) algorithm has been well-known since it was proposed by Rumerhart et al. [20] in 1985. The self-tuning PID being described in detail later is based on the derivation of this algorithm. First, we will explain the derivation of the BP algorithm in compact way. The form of a neural network described by Fig. reflayerednn is called a layered neural network since they have more than three layers which are called input layer, hidden layer, and output layer. Outputs of neurons in the input layer are the input data which should be processed. We assume that numbers of neurons in the input, hidden, and output layers are I, J, and K, respectively. In Fig. 1, large circles denote neurons and each neuron, for example, neuron j can be described by the following nonlinear input-output relation:

$$O_j = f(\mathrm{net}_j), \qquad \mathrm{net}_j = \sum_{i=1}^{I} w_{ji} O_i - \theta_j, \qquad f(x) = \frac{1}{1+\exp(-x)} = sigmoid(x). \tag{1}$$

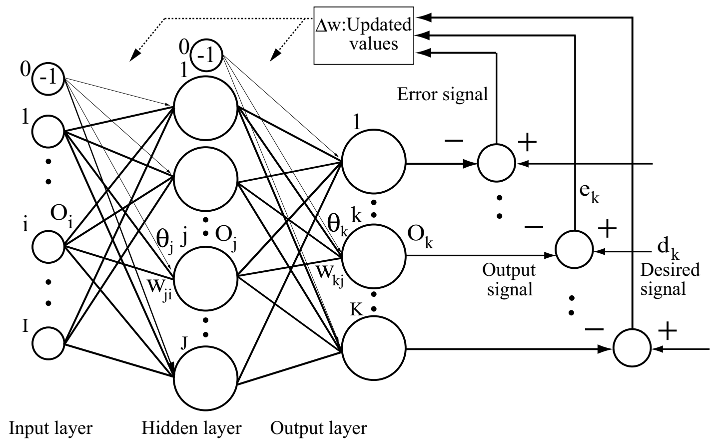

Fig. 1. Structure of a layered neural network

where O_j denotes the output of neuron j, w_{ji} denotes the connection weight from a neuron i to a neuron j, θ_j is a threshold value of neuron j.

Note that the output of a neuron is limited within 0 to 1 since $f(x) \in [0,1]$. If we assume that $O_0 = -1$ and $w_{j0} = \theta_j$, then we can rewrite net_j as follows:

$$O_j = f(\mathrm{net}_j), \qquad \mathrm{net}_j = \sum_{i=0}^{I} w_{ji} O_i, \qquad f(x) = \frac{1}{1+\exp(-x)}. \tag{2}$$

From now on, we assume that threshold θ_j has been included in the weighting function and use the expression Eq.(2) instead of Eq.(1).

When the input data $\{O_i, i = 0, 1, \ldots, I\}$, connection weights w_{ji} from a neuron i in the input layer to a neuron j in the hidden layer where $\{j = 1, 2, \ldots, J, i = 0, 1, \ldots, I\}$, and connection weights w_{kj} from a neuron j in the hidden layer to a neuron k in the output layer where $\{k = 1, 2, \ldots, K, j = 0, 1, \ldots, J\}$, we can get the output values of the neural network by the following equation:

$$O_k = f(\mathrm{net}_k), \qquad \mathrm{net}_k = \sum_{j=0}^{J} w_{kj} O_j, \qquad f(x) = \frac{1}{1+\exp(-x)}. \tag{3}$$

Then we will compare the output $\{O_k\}$ with the desired value $\{d_k\}$ for each $k, k = 1, 2, \ldots, K$ and if there are large discrepancies, we will correct the weighting functions, $w_{ji} and w_{kj}$ such that the following error function E will be decreased.

$$E = \frac{1}{2}\sum_{k=1}^{K} e_k^2, \qquad e_k = d_k - O_k. \tag{4}$$

Using the gradient search, the minimizing cost of E is given by the following relation(the error back-propagation algorithm):

$$\Delta w_{kj} = w_{kj}(\mathrm{new}) - w_{kj}(\mathrm{old}) = \eta \delta_k O_j \tag{5}$$

$$\delta_k = e_k O_k (1 - O_k). \tag{6}$$

$$\Delta w_{ji} = w_{ji}(\text{new}) - w_{ji}(\text{old}) = \eta \delta_j O_i \tag{7}$$

$$\delta_j = \sum_{k=1}^{K} \delta_k w_{kj} O_j (1 - O_j) \quad k = 1, 2, \ldots, K, \quad j = 0, 1, \ldots, N \tag{8}$$

Since the output O_k is limited within [0,1], we should modify the form when we need the value of $(-\infty, \infty)$, for example, $f(x) = x$, $f(x) = A(\frac{1}{2} - sigmoid(x))$, etc. Furthermore, to speed up the convergence of the gradient algorithm, we use an additional term as follows:

$$\Delta w_{kj}(\text{new}) = \eta \delta_k O_j + \alpha \Delta w_{kj}(\text{old}) \ , \ \ j = 0, 1, \ldots, N, \ \ k = 1, 2, \ldots, K \tag{9}$$

$$\Delta w_{ji}(\text{new}) = \eta \delta_j O_i + \alpha \Delta w_{ji}(\text{old}) \ , \ \ i = 0, 1, \ldots, M, \ \ j = 1, 2, \ldots, N \tag{10}$$

where the first term and second term of (9) and (10) are called the learning term and the momentum terms, respectively and η and α are called learning rate and momentum rate, respectively.

4 Feedback Control System Algorithm

We show the neuro-control scheme. The general control system can be described in Fig. 2 where FFC and FFB stand for feed-forward controller and feedback controller, respectively and FB is feedback. The aim of the controller is to find the suitable plant input u in order to follow the plant output y to the plant specification by adjusting the FFB and FFD.

The neuro-control is to determine the control input by using neural networks. Three types of neuro-controllers were proposed [10],[16],[13]. They are 1) series type, 2) parallel type, and 3) self-tuning type as shown in Fig. 3. We will consider those types in what follows.

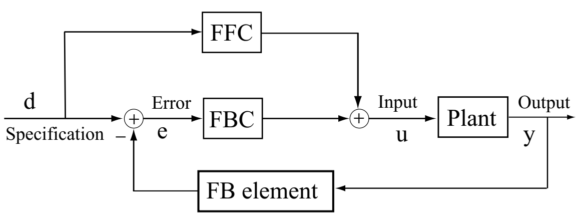

Fig. 2. General structure of control system

4.1 Series Type Neuro-control

This is to use the neural network directly such that the plant output will approach to reference signals as much as possible. The basic configuration is shown

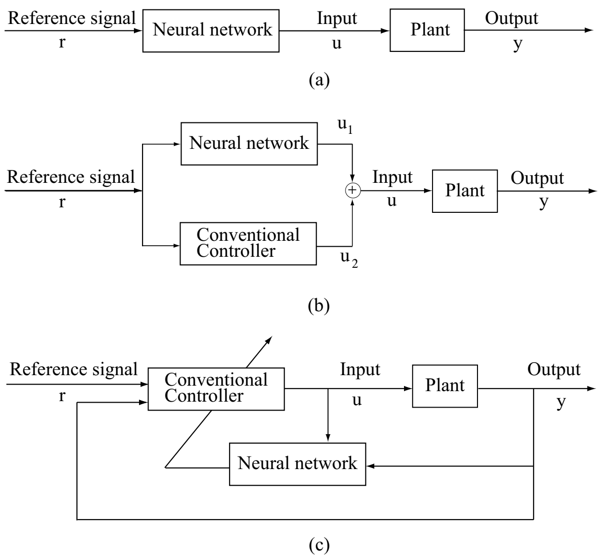

Fig. 3. Three types of neuro-control system

in Fig. 4 where (a) is the original structure, (b) is the series neuro-controller with an emulator, and (c) is the inverse dynamical structure. More detail algorithms have been explained in [17], [18], [19].

This approach is direct application of the layered neural network to find the control input and it is powerful for process control without so many fluctuations. But we need the emulator of the plant and it takes much time to find a stable parameter set of the neural network.

4.2 Parallel Type Neuro-control

A parallel neuro-control architecture is shown in Fig. 3(b). For any conventional control scheme, we can use this type and the neural network works as the compensator of the adopted control scheme. If we take a feedback controller, this control becomes to the feedback error learning structure proposed by Kawato et al. [5].

Control engineers design an excellent controller at the laboratory or factory which is given by u_1 but when it is set at the real working place in an industrial factory, the control engineers must adjust the control input level such that it is suitable for real production under several environments. The adjustment is u_2 given by neuro-control in Fig. 3(b). This means that a well-trained cook at the restaurant could provide a delicious dinner for customers but on each table there are pepper and salt to be added to suit the taste of each individual dish. For detail algorithms see [13].

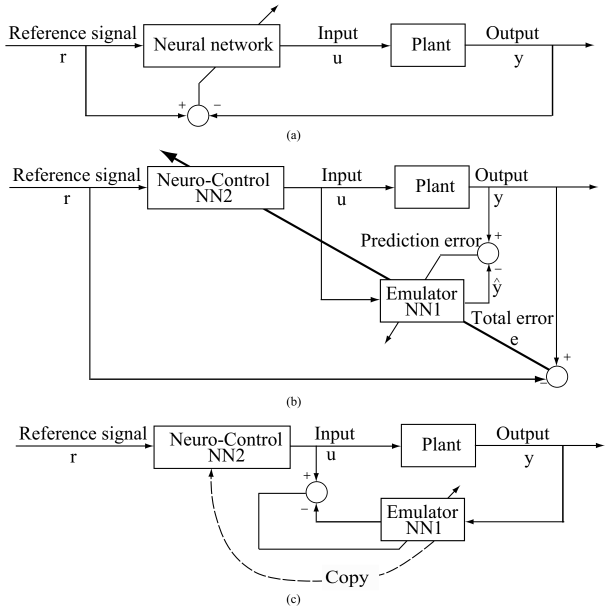

Fig. 4. Series type neuro-control structure

4.3 Self-Tuning Type Neuro-control

The self-tuning neuro-control scheme is illustrated in Fig. 3(c) where a neural network is used to tune the parameters of a conventional control method like a human operator in the factory. The transfer function of PID controller is given by Eq.(11).

$$G_c(s) = \frac{U(s)}{E(s)} = k_c\left[1 + \frac{1}{T_i s} + T_d s\right] \tag{11}$$

where $U(s)$and $E(s)$ are input and error between the desired value and output. Here, k_c, T_i, and T_d are called as proportional gain, integral time, and derivative time, respectively. In time domain, it can be written as follows:

$$u(t) = k_c\left[e(t) + \frac{1}{T_i}\int_{-\infty}^{t} e(\tau)d\tau + T_d\frac{d}{dt}e(t)\right] \tag{12}$$

$$e(t) = d(t) - y(t) \tag{13}$$

Therefore, in the PID control it is essential to find a suitable PID gains. Many researchers have tried to determine them as precise as possible. The most famous method was proposed by Ziegler-Nichols and to determine them by the following relations(Ziegler-Nichols method).

$$k_c = \frac{1.2}{RL}, \qquad T_i = 2L, \qquad T_d = \frac{L}{2} \tag{14}$$

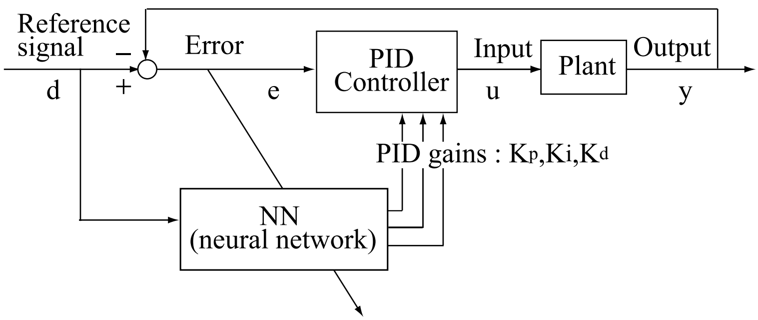

Fig. 5. Three types of neuro-control system

where R and L are maximum slope of the step response and the equivalent delay of the step response, respectively.

By rapid progress of computer, digital control has become common approach in control method and discrete PID control is also discussed. By descritizing Eq.(12) using trapezoidal rule for numerical integration, we obtain the following relation.

$$\begin{aligned} u(n) = u(n-1) &+ K_p \big((e(n) - e(n-1)\big) + K_i e(n) \\ &+ K_d \big(e(n) - 2e(n-1) + e(n-2)\big) \end{aligned} \tag{15}$$

$$K_p = k_c - \frac{1}{2} K_i, \quad K_i = k_c \frac{T}{T_i}, \quad K_d = k_c \frac{T_d}{T} \tag{16}$$

As in the continuous-time case, Ziegler-Nichols method in the discrete-time case has become as follows:

$$\begin{aligned} &K_p = k_c - \frac{K_i}{2}, \qquad K_i = \frac{1.2}{RL}\frac{T}{2L} = \frac{0.6}{(\frac{L}{T})^2 (RT)} = \frac{0.6}{G_0 L_0^2}, \\ &K_d = k_c \frac{T_d}{T} = \frac{0.6}{G_0}, \qquad G_0 = \max_n \big(y(n) - y(n-1)\big), \qquad L_0 = \frac{L}{T} \end{aligned} \tag{17}$$

Ziegler-Nichols method is helpful to find the rough estimation of PID gains, it is not so good in any case. Therefore, in the process control the operators are adjusting these gains based on their experience and knowledge in trial and error.

We have developed a self-tuning PID controller. The control structure is shown in Fig. 5.

Using the learning ability of the neural networks, we have developed a self-tuning method of the PID control gains, automatically although we have stared the PID gains given by Ziegler-Nichols. From our experience, the power of the tuning and the improvement of cost function are excellent compared with the results by the Ziegler-Nichols method. The detail derivation will omit here, the following papers or books will be helpful to construct the self-tuning PID controller, [16], [17], [18], [19], [13], [15], [14], etc.

5 Appplication to Electric Vehicle Control

Due to environmental problems the automobile industry is currently venturing into producing electric vehicles. At the Shikoku Electric Power Company, Japan, a new type of electric car which is called PIVOT has been developed in 1993. The specification is shown in Table 1 and the overview and specific characteristics are illustrated in Fig. 6.

This PIVOT has equipped four wheels and each wheel has been made with in-wheel motor. Therefore, the wheels can be steered more than 90 degrees opposed to the body. This newly developed function accounts for universal drive performance such as lateral drive and rotation at a point. Another advantage is high-accuracy residual battery capacity indicator based on neural networks. A small and high accurate indicator has been developed. The residual battery capacity is calculated by a computer using voltage and current while driving.

The third one is an automatic battery exchange system. By the development of an automatic battery exchange system, the battery, having little residual capacity, is removed and a charged battery is installed within approximately five minutes, making refueling as easy as a gasoline-engine vehicle.

The fourth one is an energy-saving technology. Development of a regenerative braking system to convert kinetic energy to electrical energy and charge the battery during deceleration. Adoption of a lightweight frame/body and low air resistance body configuration and development of a lightweight heat-pump type air conditioning system are also equipped.

In 1993 when PIVOT was completed in Japan, there was no permission to drive any electric vehicle on the road in raw and it is difficult to do the real driving experiment under various load change or load conditions, we have made experimental simulator as shown in Fig. 7. This can be written in Fig. 8 where DDC is direct digital controller which has been equipped with PID controllers, ACM is an alternative current motor which produces torque of OIVOT, DC is a direct current motor which produces any load with various specifications, T is a torque meter, and UFAS denoted a universal factory automation system.

Table 1. Specification of PIVOT

Specification	**Performance**
length	4,126 mm
width	1,671 mm
height	1,603 mm
dry weight	2,200 Kg
passengers	4 persons
maximum speed	100 Km/h
range	200 Km(at a constant cruiing spped of 40km/h
acceleration	Approximate 20 secs. from 0 m to 400 m
grand climb ability	30%
battery type	lead battery
equipment	power steering, heat-pump type sir conditioning

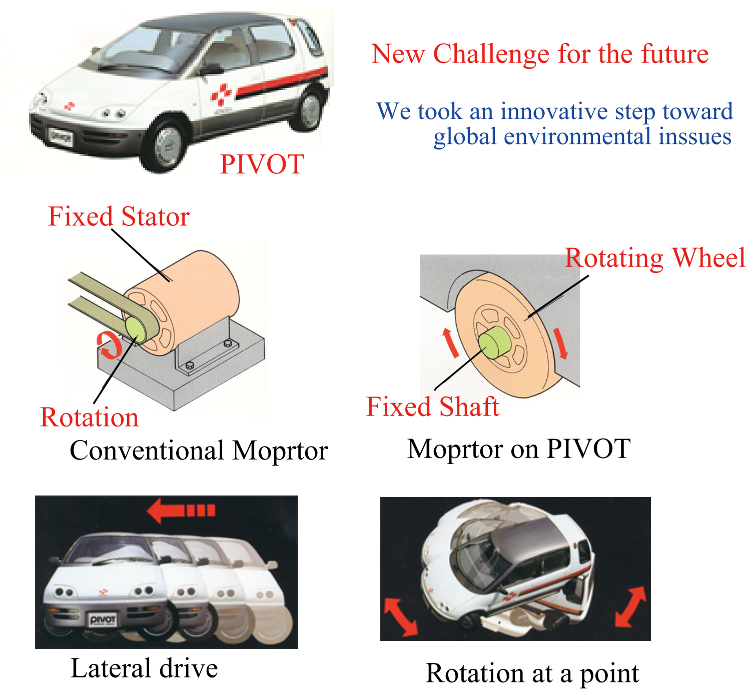

Fig. 6. PIVOT system

For training the neural networks for various loads and various speeds, we have obtained the input and output data using the physical simulator illustrated in Fig. 7. This simulator can be modelled as shown in Fig. 8 where DCM produces any kinds of loads and ACM outputs the corresponding control inputs by an ac motor. From our many experiences, we have used the neuro-control structure as shown in Fig. 9 where NNC means neuro-controller to adjust the PID gains and NNM was used to model the system emulator which is necessary to find the PID

Fig. 7. Experimental simurator

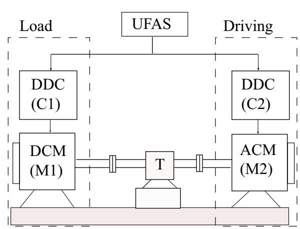

Fig. 8. PIVOT system

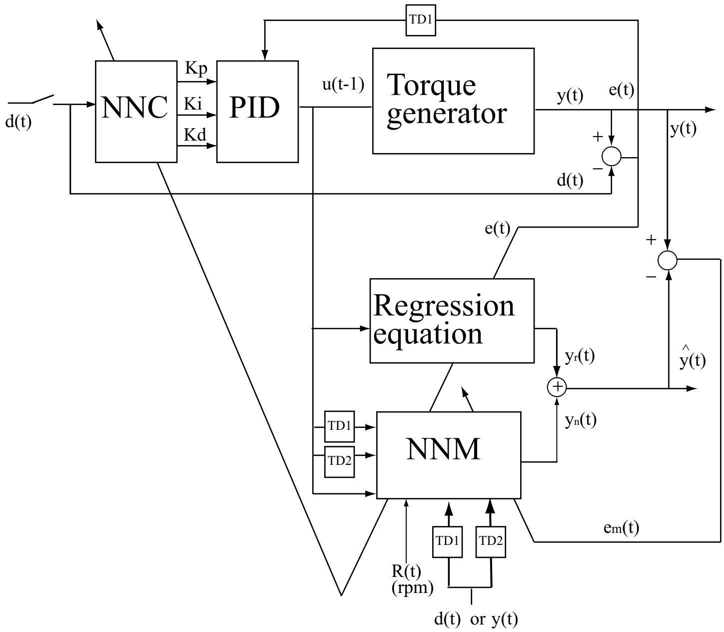

Fig. 9. Experimental simurator where TD1 and Td2 are time-delay elements with one and two steps delays, respectively

gains in NNC. Here, we use the parallel type emulator with regression model in order to speed up the modelling convergence and also used rotation number of motors. The notation $\hat{y(t)}$ means the estimated value of $y(t)$, $y_r(t)$ and $y_n(t)$ are estimated value of $y(t)$ by regression method and neural networks, respectively, $e(t) = d(t) - y(t)$, and $e_m(t) = y(t) - \hat{y}(t)$.

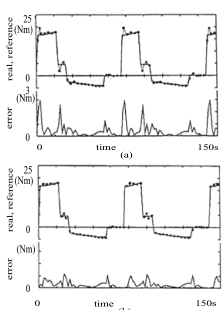

Fig. 10. PIVOT system

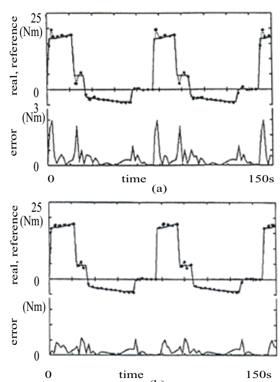

Fig. 11. Experimental simurator

Figs. 10 and 11 are simulation results where (a) denotes control results when we used the parameters by the experts who designed the electric vehicle simulator starting from parameters obtained by the Ziegler-Nichols method and (b) shows the results obtained by our approach after training. In these simulations were developed underthe following situations that learning parameter is $\eta = 0.001 \sim 0.05$ and the momentum coefficient is $\alpha = 0.9$.

From these results, the neuro-control methods could be applied to several real control problems.

References

1. Widrow, B., Smith, F.W.: Pattern-Recognizing Control Systems. In: Computer and Information Sciences Symposium Proceedings, Spartan, Washington, DC, pp. 288–317 (1963)
2. Walt, M.D., Fu, K.S.: A Heuristic Approach to Reinforcement Learning Control Systems. IEEE Transactions on Automatic Control AC-10(4), 390–398 (1965)
3. Michie, D., Chambers, R.A.: An Experiment in Adaptive Control. In: Tou, J.T., Wilcox, R.H. (eds.) Machine Intelligence, Edinburgh, Oliver and Boyd, pp. 137–152 (1968)
4. Barto, A.G., Sutton, R.S., Anderson, C.W.: Neuronlike Adaptive Elements that Cn Solve Difficult Learning Control Problems. IEEE Transactions on Systems Man and Cybernetics 13(5), 834–846 (1983)
5. Kawato, M., Furukawa, K., Suzuki, R.: A Hierarchical Neural Network Model for Control and Learning of Voluntary Movement. Biological Cybernetics 57, 169–185 (1978)
6. Hunt, K.J., Sbarbaro, R.: Neural Networks for Non-Linear Internal Model Control. IEE Proceedings Control Theory and Applications 138(5), 431–438 (1991)
7. Willis, M.J., Montague, G.A., Dimassimo, C., Tham, M.T., Morris, A.J.: Artificial Neural Networks in Process Estimation and Control. Automatica 28(6), 1181–1187 (1992)
8. Jordan, M.I., Jacobs, R.A.: Learning to Control an Unstable System with forward Modelling. In: Lippmann, R.P., Moody, S.E., Touretzky, D.S. (eds.) Advances in Neural Information Processing Systems, San Mateo. Morgan Kaufmann, San Francisco (1990)
9. Psaltis, D., Sideris, A., Yamamura, A.: A Multilayered Neural Network Controller. IEEE Control Systems Magazine 8(2), 17–21 (1988)
10. Omatu, S.: Learning of Neural-Controllers in Intelligent Control Systems. In: Zurada, J.M., Marks II, R.J., Robinson, C.J. (eds.) Computational Intelligence Imitating Life. IEEE Press, New York (1994)
11. White, D.A., Sofge, D.A. (eds.): Handbok of Intelligent Control. Van Nostrand Reinhold, New York (1992)
12. Miller III, W.T., Sutton, R.S., Werbos, P.J. (eds.): Neural Netoworks for Control. MIT Press, Massachusetts (1990)
13. Omatu, S., Maruzuki, K., Rubiyah, Y.: Neuro-Control and Its Applications. Springer, London (1996)
14. Mills, P.M., Zomaya, A.Y., Tade, M.O.: Neuro-Adaptive Process Control. John Wiley & Sons, Chichester (1996)
15. Ng, G.W.: Application of Neural Networks to Adaptive Control of Nonlinear Systems. Research Studies Press, New York (1997)

16. Omatu, S.: Neuro-Control Applications in Real-World Problems. In: Proceedings of the 10th Yale Workshop on Adaptive and Learning Systems, pp. 92–97. Yale University, New Haven (1998)
17. Tanomal, J., Omatu, S.: Process Control by On-Line Trained Neural Controllers. IEEE Transactions on Industrial Electronics 39(6), 511–521 (1992)
18. Maruzuki, K., Omatu, S., Rubiyah, Y.: Temperature Regulation with Neural Networks and Alternative Control Schemes. IEEE Transactions on Neural Networks 6(3), 572–582 (1992)
19. Maruzuki, K., Omatu, S., Rubiyah, Y.: MIMO Furnace Control with Neural Networks. IEEE Transactions on Control Systems Technology 1(4), 238–245 (1993)
20. Rumelhart, D.E., McClelland, J.L.: PDP Group: Parallel Distributed Processing, Explorations in the Microsteucture of Cognition, vol. 1. MIT Press, Massachusetts (1987)