Control and Cybernetics

vol. **27** (1998) No. 1

A stability based neural networks controller design method

by

Jun Song, Xiaoming Xu, Xing He

Department of Automation, Shanghai JiaoTong University,
People's Republic of China

Abstract: The use of neural networks in control systems can be seen as a natural step in the evolution of control methodology to meet new challenges. Many attempts have been made to apply the neural networks to deal with non-linearities and uncertainties of the control systems. Research in neural network applications to control can be classified according to the major methods depending on structures of the control system, such as NN-based Non-linear System Identification, NN-based Supervised Control, NN-based Direct Control, NN-based Indirect Control, NN-based Adaptive Control, NN-based Self-learning Control, NN-based Fuzzy Control, and NN Variable Structure Control.

All these control methods cannot, however, effectively guarantee system stability, i.e. none of these neural network controls, except for NN-based Variable Structure Control, is based on system stability. This also limits the application and development of the neural networks in control theory.

The paper shows the effort to solve this difficulty and give a way for the design method of the stability based neural networks controller using Lyapunov second stability theorem. This kind of controller can not only guarantee system stability, but also fully compensate for the influence of system uncertainties and non-linearities. Simulation results also show the effectiveness of the controller.

Keywords: neural networks control, nonlinear control, stability, sliding mode

1. Introduction

Neural networks control, as an important component of Intelligent Control, has widely been used in control engineering. Antsaklis (1990, 1992) made an important contribution to introduction of neural network in control theory. In his papers he summarized some good characteristics of neural networks control classifying them into three categories: the ability of self-learning, of performing massive parallel processing, and of significant fault tolerance. Such control therefore, can effectively meet the need of dealing with increasingly complex systems, the need to satisfy the increasingly demanding design requirements, and the need to meet these requirements with less precise advance knowledge of the plant and its environment – that is, the need to control under increasing uncertainty. Fukuda (1992) also systematically summarized the advance of applications of neural networks in control engineering. Thus, research in neural network applications to control can be classified into the major method groups depending on structures of the control system, e.g., NN-based Non-linear System Identification, Chu (1992), Chen, Khalil (1995), Song, Xu (1997); NN-based Supervised Control, Burns (1995), Bouslama (1993), NN-based Direct Control, Gomi; Kawato (1993), Sanger (1994), Chen, Khalil (1995), Yabuta, Yamada (1992), Yuh, Lakshmi (1993), Yuh (1990), Venugopal, Sudhakar, Pandya (1992); NN-based Indirect Control, Nguyen, Derrick, Widrow (1990); NN-based Adaptive Control, Sartori, Antsaklis (1992), Fukuda, Toshio, Shibata, Takanori (1992), Yabuta, Tetsuro, Yamada, Takayuki (1992), Song, Xu (1997); NN-based Self-learning Control, Chen, Fu-Chuang (1990), Chen, Khalil (1995), Gomi, Hiroaki, Kawato, Mitsuo (1993); NN-based Fuzzy Control, Bouslama, Faouzi, Ichikawa, Akira (1993), Song, Xu (1997); NN Variable Structure Control, Karakasoglu, Sundareshan, Malur (1995). All these papers introduced various kinds of control methods using neural networks in different engineering settings. The common major shortcoming of these methods is that is none of them, except for the NN-based variable structure control, can guarantee system stability by applying neural networks. Thus, neural networks controller designed under these methods loses its application value in practical engineering. The present paper is trying to solve this problems. In Section 2, we give a design method for a stability based neural networks controller for a non-linear system. Section 3 contains further discussion for this control method to improve control performance. In Section 4 we present the simulation results of application of this control method to a subwater robot control.

2. Stability based neural networks controller

Consider a non-linear control system,

$$\dot{X} = g(X) + \Delta g(X) + BU + \Delta B(U) + \Delta f(X, U, t) \tag{1}$$

where X is the system state variable, $g(X)$ is state matrix, $\Delta g(X)$ is state variation matrix, B is control matrix, U is control variable, $\Delta B(U)$ is control variation matrix, $\Delta f(X, U, t)$ is the system's internal and external disturbance. The form (1) can be further simplified as follows:

$$\dot{X} = g(X) + BU + f \tag{2}$$

where $f = \Delta g(X) + \Delta B(U) + \Delta f(X, U, t)$ is the overall uncertainty and disturbance of the system, whose values are bounded, $\|f\| < M_f$.

Then, we define a hyperplane

$$S_o = KerC = \{X \,|CX = 0\} \tag{3}$$

where $C^\tau = [C_0, C_1, \cdots, C_{n-2}]$ is the coefficient matrix of the hyperplane S, and the selection principle of $C^\tau = [C_0, C_1, \cdots, C_{n-2}]$ satisfies the stability condition.

The only restriction on the choice of the hyperplane

$$S(x) = 0 \tag{4}$$

is that it has to be associated with stable dynamics in the sense that

$$S(x(t)) = 0, \text{ for all } t > t_0 \Rightarrow \lim_{t\to\infty} x(t) = 0 \tag{5}$$

for any initial conditions $x(t_0)$. The choice of a linear hyper surface gives:

$$S(x) = C^\tau X \tag{6}$$

By defining the Lyapunov function:

$$V(x) = \frac{1}{2}[S(x)]^2 \tag{7}$$

we guarantee that the hyper surface $S(x) = 0$ is reached in finite time by the condition:

$$S\dot{S} = -\xi_0^2(x)\,|S(x)| \text{ or } \dot{S} = -\xi_0^2(x)\, sgn(S) \tag{8}$$

where $\xi_0^2(x) = diag\left(\xi_{0,1}^2(x),\ \xi_{0,2}^2(x),\ \cdots,\ \xi_{0,n}^2(x)\right)$, and

$$y = sgn(x) = \begin{cases} +1 & x > \phi \\ \phi & |x| \le \phi \quad ,\phi > 0 \\ -1 & x < -\phi \end{cases}$$

Since $S(x) = C^\tau X$, we can use (8) and (2) to get:

$$C^\tau (g(X) + BU + f) = -\xi_0^2 sgn(S) \tag{9}$$

By knowing a bound $\xi^2(x)$ on the non-linearity such that

$$\xi^2(x) > \frac{\|C^\tau f\|}{\|C^\tau B\|} \tag{10}$$

for all x, the condition (8) with $\xi_0^2(x) = \xi^2(x) - C^\tau f$ can be satisfied by choosing the control input:

$$U = -(C^\tau B)^{-1} C^\tau g(X) - (C^\tau B)^{-1} \xi^2(x)\, sgn(S) \tag{11}$$

or

$$U = \hat{U} + \bar{U}$$

Hence, the feedback control law U is composed of two parts. The first,

$$\hat{U}(X) = -\left(C^{\tau}B\right)^{-1} C^{\tau} g(X) \tag{12}$$

is a non-linear feedback law which can compensate for the system disturbance, whereas the second,

$$\bar{U}(X) = -\left(C^{\tau}B\right)^{-1} \xi^2(x)\, sgn(S) \tag{13}$$

is also a non-linear feedback with its sign toggling between plus and minus according to which side of the hyper plane the system is located in. Two comments are in order here: first, $\bar{U}$ has to change its sign as the system crosses $S(x) = 0$. Secondly, it is $\bar{U}$ which is mainly responsible for driving and keeping the system onto the hyperplane $S(x) = 0$. Provided that the gain $\xi^2(x)$ has been chosen large enough, $\bar{U}$ can secure the required robustness due to momentary disturbances and unmodeled dynamics without any compromise in stability.

Since no information regarding non-linear characteristics of the control system dynamics exists, we have to use the neural networks to identify it. The identification model can be described as follows:

$$\dot{X}_p = \hat{g}(X_p) + BU \tag{14}$$

The control diagram is shown in Fig.1, where neural networks NN1 and NN2 have the same structure. $\hat{g}(X)$ is the non-linear map of the control system state matrix expressed by using neural networks. Then, the final feedback control law can be modified as

$$U = -\left(C^{\tau}B\right)^{-1} C^{\tau}\hat{g}(X) - \left(C^{\tau}B\right)^{-1} \xi^2(x)\, sgn(S) \tag{15}$$

This approach clearly constitutes a "worst case scenario" and enhances the robustness properties of the design.

3. Further discussion of the controller

From the above analysis of the controller we know that it has two outstanding good characteristics: it can not only guarantee system stability, but also can effectively eliminate system uncertainty, disturbance and model deficiency. These good points are due to the contribution of the non-linear feedback control law $\bar{U}$ in (13). Rewrite the control law as follows,

$$\bar{U}(X) = -\left(C^{\tau}B\right)^{-1} \xi^2(x)\, sgn(S)$$

Why this control law can effectively compensate for system disturbances? The fact is that this control law is a negative feedback control law with a big feedback gain which can guarantee system stability and eliminate disturbance. It entails, though, also some negative influence: generation of big "dithers" on the hyper plane. Theoretically speaking, the system should not generate

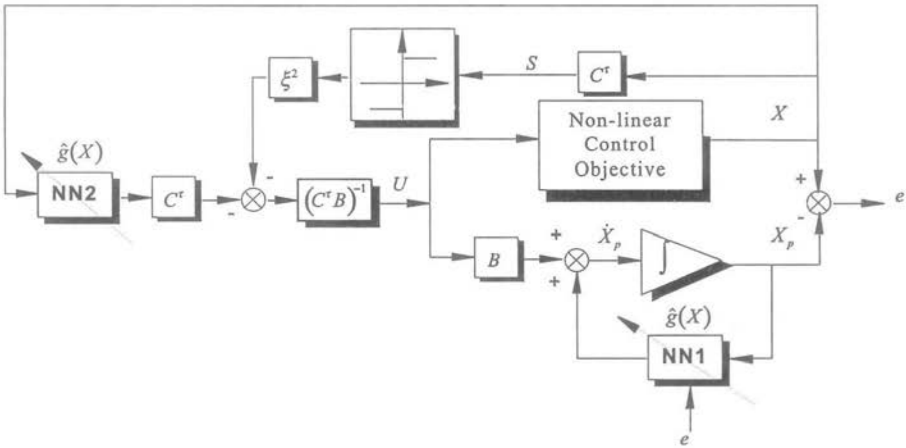

Figure 1. Control system diagram

"dithers" on the hyper plane. But the practical system is not ideal, it has inertia and relay factors, and dithers can be generated. In particular, the bigger the feedback control gain ξ^2, the worse the dither. Therefore, decreasing of the dithers is of great practical significance.

In the practical control process, when the system is far from the hyperplane, the uncertainties of the system are also large, and we have to use a high gain negative control law to guarantee system stability. On the contrary, when the system comes close to the hyperplane, the uncertainties of the system are also small, and in order to decrease the dithers we use a small gain negative control law to guarantee system stability and compensate for the disturbance. Then, by modifying system stability conditions (8) in the following manner we can reduce the dithers:

$$\dot{S} = -\xi^2 (x)\, satsgn\,(S) \text{ or } \dot{S} = -\xi^2 \tan (S) \tag{16}$$

where $satsgn\,()$ is a saturation function and $\tan ()$ is a tangent function (both shown in Fig. 2).

We can select other functions instead of (16). For a example, we may choose the following stability function:

$$\dot{S} = n (S) \tag{17}$$

In order to meet Lyapunov condition, we choose $\begin{cases} n(S) < 0 & S > 0 \\ 0 & S = 0 \\ n(S) > 0 & S < 0 \end{cases}$, then define

$$n (S) = \tilde{n} (S) + C^\tau f, \tag{18}$$

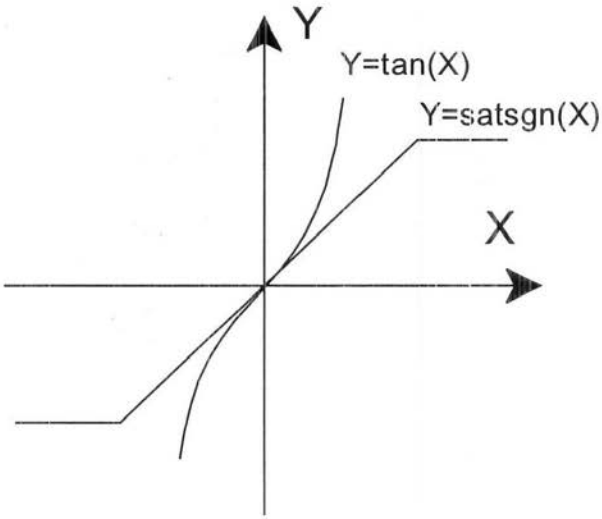

Figure 2. Non-linear function curves

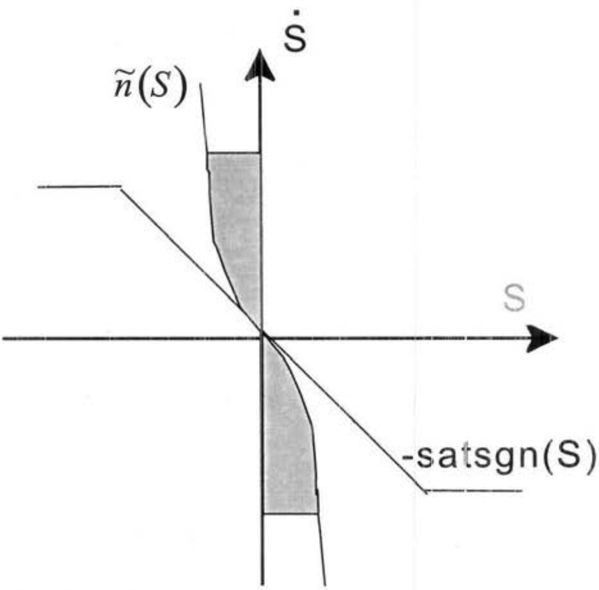

Figure 3. Non-linear control function

and in order to eliminate system uncertainty, select

$$\|\tilde{n}(S)\| > \frac{\|C^{\tau} f\|}{\|C^{\tau} B\|} \tag{19}$$

and $\begin{cases} \tilde{n}(S) < 0 & S > 0 \\ 0 & S = 0 \\ \tilde{n}(S) > 0 & S < 0 \end{cases}$, where $\tilde{n}(S)$ is a monotone decreasing function (shown in Fig. 3).

From (18), we have

$$\tilde{n}(S) = n(S) - C^{\tau} f \tag{20}$$

Considering (19), we get that

$$\|n(S) - C^{\tau} f\| > \frac{\|C^{\tau} f\|}{\|C^{\tau} B\|} \tag{21}$$

Then

$$\|n(S)\| \geq \frac{\|C^{\tau} f\|}{\|C^{\tau} B\|} - \|C^{\tau} f\| \tag{22}$$

Therefore, the control law can be adjusted as

$$U = -\left(C^{\tau} B\right)^{-1} C^{\tau} \hat{g}(X) + \left(C^{\tau} B\right)^{-1} \tilde{n}(S) \tag{23}$$

4. Simulation results

4.1. Fast back-propagation algorithm

The neural network we used here is the feed-forward neural network, while the back-propagation (BP) algorithm is a typical network learning method, but traditional BP algorithm's convergence is too slow when the learning error is small. So, we will present a fast back-propagation algorithm in our simulation algorithm.

Rewrite the traditional weights updating formula as follows

$$\Delta W_{ji}(t+1) = \eta \delta_{pj} o_{pj} \tag{24}$$

The sigmoid function is selected as follows

$$f(x) = \frac{(1 - \exp(-x))}{(1 + \exp(-x))} \tag{25}$$

the related function is defined as

$$Net_{pj} = \sum_{k} W_{ji} o_{pj} + \theta_j \tag{26}$$

$$o_{pj} = \frac{1 - \exp\left\{-\sum_i W_{ji} o_{pi} - \theta_j\right\}}{1 + \exp\left\{-\sum_i W_{ji} o_{pi} - \theta_j\right\}} = \frac{1 - e^{-Net_{pj}}}{1 + e^{-Net_{pj}}} \tag{27}$$

and

$$\frac{\partial o_{pj}}{\partial Net_{pj}} = \left(1 - o_{pj}^2\right) \tag{28}$$

For the output unit, it has

$$\delta_{pj} = \left(t_{pj} - o_{pj}\right)\left(1 - o_{pj}^2\right) \tag{29}$$

For the hidden layer, it has

$$\delta_{pj} = \left(1 - o_{pj}^2\right) \sum_k \delta_{pk} W_{kj} \tag{30}$$

In order to accelerate convergence, a momentum term is added and weight changes are smoothed:

$$\Delta W_{ji}(t+1) = \eta \delta_{pj} o_{pj} + \alpha \Delta W_{ji}(t) + \beta \Delta^2 W_{ji}(t) \tag{31}$$

here α is the momentum term, and it reflects how the last weight change affects current weights change, β is the derivative momentum term and η is the learning rate. In many cases, if the learning rate, η, is too small, the number of iterations required for arriving at a solution of the weight vector may be exceedingly large. On the other hand, the weights many oscillate during iterations when η is too large. If $\eta(k)$ is not a constant, but adjusted at each k to overcome this problem, it is called a dynamic learning rate. Several schemes have been developed for adaptive adjustment of $\eta(k)$. We dynamically adjust the learning rate as follows

$$\begin{cases} \eta_{ij}(k) = q\eta_{ij}(k-1), \quad q > 1 \\ \qquad \text{when } sign\left(\frac{\partial E}{\partial W_{ij}(k)}\right) = sign\left(\frac{\partial E}{\partial W_{ij}(k-1)}\right) \\ \eta_{ij}(k) = d\eta_{ij}(k-1), \quad 0 < d < 1 \\ \qquad \text{when } sign\left(\frac{\partial E}{\partial W_{ij}(k)}\right) = -sign\left(\frac{\partial E}{\partial W_{ij}(k-1)}\right) \end{cases}$$

The back-propagation training algorithm with this dynamic learning rate is capable to speed up the training process and achieve high recognition accuracy.

4.2. Nonlinear control system simulation

4.2.1. Nonlinear control system model

As we know, a subwater robot is a complex nonlinear control system, in which some traditional control policies such as optimal control, have been used. All these methods need an approximate nonlinear system model so as to ensure

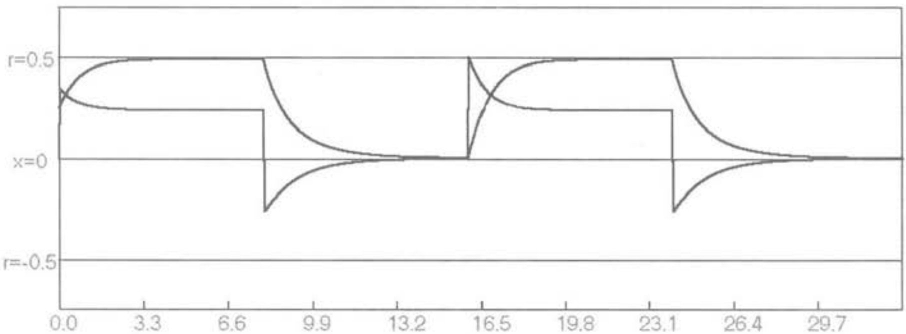

Figure 4. The tracking path with no disturbance

accurate control, but the practical system features heavily coupled and complicated nonlinear dynamics, and its work environment generates many internal and external disturbances. All these factors make it impossible to develop a proper model for the subwater robots, and so traditional control policy cannot deal with these problems. But neural network control, which does not need to know a priori the system model, and has the on-line learning ability, motivates us to apply it in subwater robot control.

A typical subwater robot is given in Healey, Lienard (1993), with the following forward speed control dynamics

$$\dot{u}(t) = \alpha_h(t)\, u(t)\, |u(t)| + (\alpha_h(t)\, \beta_h(t))\, n(t)\, |n(t)| + f$$

where $u(t)$ is forward speed, $n(t)$ the propeller rotate speed, f the non-linear disturbance, $\alpha_h(t)$, $\beta_h(t)$ the hydrodynamic parameters.

4.2.2. Simulation results

The neural network applied is a typical *feed-forward network* with the structure $N \in N^3_{1,5,1}$. The network learning algorithm is *fast error back-propagation algorithm*. The parameter values selected are $\eta(0) = 0.45$, $\alpha = 0.5$, $\beta = 0.0035$, $q = 1.1131$, $d = 1/1.0011$.

We use sufficient input/output data to train the network off-line, and after system error goes down to 0.001, we consider that the neural network has fully matched the controlled model.

The controlled value is subwater robot's forward speed. The initial value is 0.25m/s, the desired tracking path is a square wave of 0-0.5m/s. From Fig. 4, we can see that system output can accurately track the desired value.

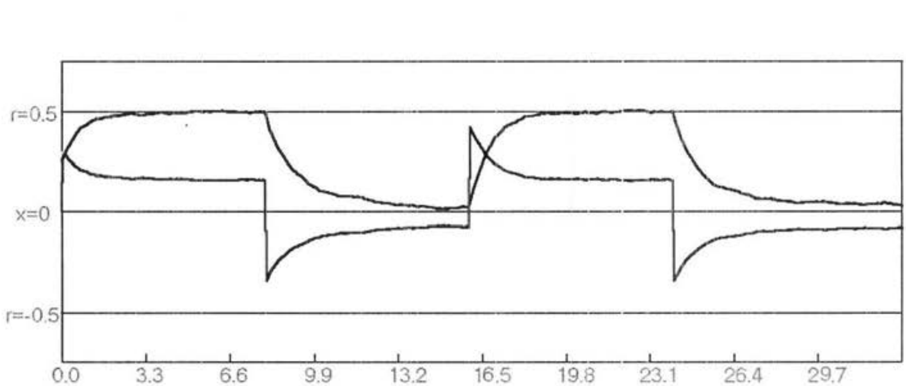

Figure 5. With ± 0.1 random disturbance

We added a random disturbance taking the values of -0.1 and $+0.1$, and the response process is given in Fig. 5. It still secures a good tracking properties of the system.

Then, we impose a periodic disturbance $+0.1 \cdot \sin(u \cdot t)$ on the system, and its influence is larger than in the previous case (Fig. 6).

Now, we add these two disturbances simultaneously on the system, and watch its output response (Fig. 7). Of course, the system output is heavily affected, but the response is still satisfying, and the neural network should be updated (note that the disturbance amounts to 40% of the track signals).

So, these simulation results prove again the conclusion that our controller is robust enough to deal with system uncertainties, nonlinearities, and disturbances.

5. Conclusion

Although neural networks have been widely used in control engineering, none of the control methods to date displayed the capability of dealing with the disturbance without losing stability. This paper provides an effective control scheme to cope with these problems. We applied this control method to the subwater robot speed control, and demonstrated the effectiveness of the controller both in terms of theory and computer simulation results.

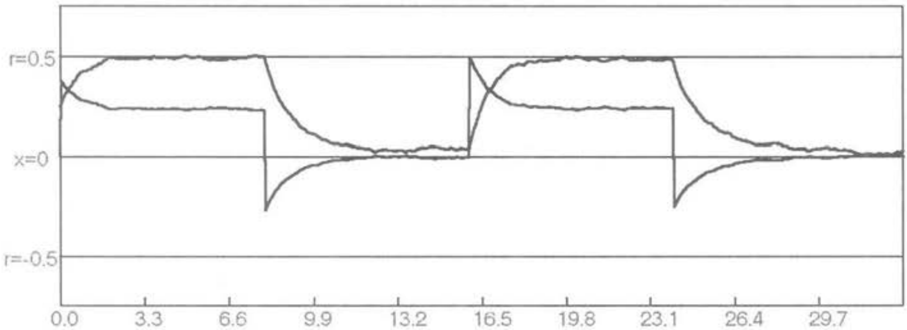

Figure 6. With $0.1 \cdot \sin(u \cdot t)$ disturbance

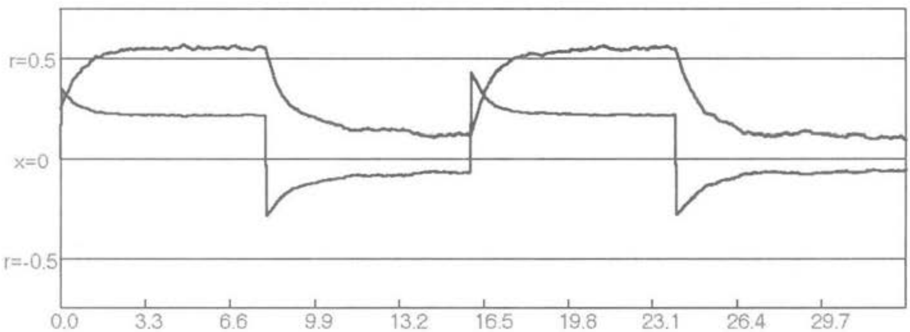

Figure 7. With $0.1 \cdot \sin(u \cdot t)$ and ± 0.1 random disturbance

References

ANTHONY, N.M., FARRELL, J.A. (1990) Associative memories via artificial neural networks. *IEEE Control Systems Magazine*, **12**, 2, 6-17.

ANTHONY, J.H., LIENARD, D.A.S. (1993) Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, **18**, 3, 327-339.

ANTSAKLIS, P.J. (1990) Neural networks in control systems. *IEEE Control System Magazine*, **10**, 3, 3-5.

ANTSAKLIS, P.J. (1990) Neural networks in control systems. *IEEE Control Systems Magazine*, **12**, 2, 8-10.

BOUSLAMA, F., ICHIKAWA, A. (1993) Application of neural networks to fuzzy control. *Neural Networks*, **6**, 6, 791-799.

BURNS, R.S. (1995) Use of artificial neural networks for the intelligent optimal control of surface ships. *IEEE Journal of Oceanic Engineering*, **20**, 1, 65-72.

CHEN, F.-C. (1990) Back-propagation neural networks for nonlinear self-tuning adaptive control. *IEEE Control Systems Magazine*, **10**, 3, 44-48.

CHEN, F.-C., KHALIL, H.K. (1995) Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Transactions on Automatic Control*, 40, 5, 791-801.

CHEN, Y., LU, H. (1991) *Neural Networks Theory and Its Application in Control Engineering*. Xian: Northwestern Polytechnical University Press.

CUI, X., SHIN, K.G. (1993) Direct control and coordination using neural networks. *IEEE Transactions on Systems, Man and Cybernetics*, **23**, 3, 686-697.

COOPER, D.J. (1992) Disturbance pattern classification and neuro-adaptive control. *IEEE Control Systems Magazine*, **12**, 2, 42-48.

FAROTIMI, O., DEMBO, A., KAILATH, T. (1991) A general weight matrix formulation using optimal control. *IEEE Transactions on Neural Networks*, **2**, 3, 378-394.

FUKUDA, T., SHIBATA, T. (1992) Theory and applications of neural networks for industrial control systems. *IEEE Transactions on Industrial Electronics*, **39**, 6, 472-489.

FUKUDA, T., SHIBATA, T., TOKITA, M., MITSUOKA, T. (1992) Neuromorphic control: Adaptation and learning. *IEEE Transactions on Industrial Electronics*, **39**, 6, 497-503.

GEORGE, W.R. (1993) A self-organizing network for computing a posterior conditional calss probability. *IEEE Transactions On Systems, Man, and Cybernetics*, **23**, 6, 1672-1682.

GOMI, H., KAWATO, M. (1993) Neural network control for a closed-loop system using feedback-error-learning. *Neural Networks*, **6**, 7, 933-946.

HAJIME, K. (1992) Solving a placement problem by means of an analog neural network. *IEEE Transactions On Industial Electronics*. **39**, 6, 543-551.

HIDEKI, H. (1992) Self-Organizing visual servo system based on neural networks. *IEEE Control Systems Magazine.* **12**, 2, 31-36.

REIGI, H.S.M., LI, C.J. (1993) Learning algorithm for neural networks based on quasi-newton methods with self-scaling. *Journal of Dynamic Systems Measurement and Control*, **115**, 38-43.

HSINCHIN H. (1995) An adaptive training algorithn for back-propagation neural networks. *IEEE Transactions on Systems, Man and Cybernetics.* **25**, 3, 512-514.

HUANG, S.-C., HUANG, Y.-F. (1990) Learning algorithms for perceptions using back-propagation with selective updates. *IEEE Control System Magazine*, **10**, 3, 56-61.

JIN, L., NIKIFORUK, P.N., GUPTA, M.M. (1995) Fast neural learning and control of discrete-time nonlinear systems. *IEEE Transactions on Systems, Man and Cybernetics*, **25**, 3, 478-488.

KRAFT, L.G., CAMPAGNA, D.P. (1990) Comparison between CMAC neural network control and two traditional. *IEEE Control Systems Magazine*, **10**, 3, 36-43.

KOSKO, B.A. (1991) Structural stability of unsupervised learning in feedback neural. *IEEE Transactions on Automatic Control*, **36**, 7, 785-792.

KARAKASOGLU, A., SUNDARSHAN, M.K. (1995) Recurrent neural network-based adaptive variable structure model following control of robotic manipulators. *Automatica*, **31**, 10, 1495-1507.

LANE, S.H., HANDELMAN, D.A., GELFAND, J.J. (1992) Theory and development of higher-order CMAC neural networks. *IEEE Control Systems Magazine*, **12**, 2, 23-30.

LEVIN, A.U., NARENDRA, K.S. (1993) Control of nonlinear dynamical systems using neural networks controllability and stabilization. *IEEE Transactions on Neural Networks*, **4**, 2, 192-206.

LOU, K.-N. and RONALD, A.P. (1995) A novel appication of artificial neural networks to structural analysis. *Artificial Intelligence in Engineering*, **9**, 211-219.

LUISE, R. and XAVIER, J.R.A. (1992) Hierarchical neuro controller architecture for robotic manipulation. *IEEE Control Systems Magazine*, **12**, 2, 37-41.

NAMIK, O.M. (1991) An analysis of a neural network with a fixed memory span. *IEEE Transactions On Systems, Man, and Cybernetics*, **21**, 3, 683-690.

NGUYEN, D.H., WIDROW, B. (1990) Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, **10**, 3, 18-23.

NAGATA, S., SEKIGUCHI, M., ASAKAWA, K. (1990) Mobile robot control by a structured hierarchical neural network. *IEEE Control Systems Magazine*, **10**, 3, 69-76.

NARENDRA, K.S., MUKHOPADHYAY, S. (1992) Intelligent control using neural networks. *IEEE Control Systems Magazine*, **12**, 2, 11-18.

NAVEEN, V.B. (1990) Modelling chemical processing systems via neural comutation. *IEEE Control Systems Magazine*, **10**, 3, 24-30.

NICOLAOS, B.,K. (1993) Efficient learning algorithms for neural networks. *IEEE Transactions on Systems, Man and Cybernetics*, **23**, 5, 1372-1383.

RAHMAT, S. (1991) The mystique of intelligent control. *IEEE Control Systems Magazine*, **11**, 33.

RICHARD, E.N., MECKL, H. (1993) An analytical comparsion of a neural network and a model-based adaptive controller. *IEEE Transactions on Neural Networks*, **4**, 4, 685-694.

REYNOLD, C.S. (1990) Neural networks for system identification. *IEEE Control Systems Magazine*, **10**, 3, 31-35.

SAMIR, R.D. (1991) On the synthesis of nonlinear continuous neural networks. *IEEE Transactions On Systems, Man, and Cybernetics*, **21**, 2, 413-418.

SANGER, T.D. (1994) Neural network learning control of robot manipulators using gradually increasing task difficulty. *IEEE Transactions on Robotics and Automation*, **10**, 3, 323-333.

SANNER, R.M., AKIN, D.L. (1990) Neuromorphic pitch attitude regulation of an underwater telerobot. *IEEE Control Systems Magazine*, **10**, 3, 62-68.

SARTORI, M.A., ANTSAKLIS, P.J. (1992) Implementations of learning control systems using neural networks. *IEEE Control Systems Magazine*, **12**, 2, 49-57.

SCHIFFMANN, W.H., GEFFERS, W.H. (1993) Adaptive control of dynamic systems by back propagation networks. *Neural Networks*, **6**, 4, 517-524.

SHIGO, A. (1996) Convergence acceleration of the hopfield neural network by optimizing integration step sizes. *IEEE Transactions on Systems, Man and Cybernetics*, **26**, 1, 194-201.

SHOURESHI, R. (1991) Learning and decision making for intelligent control systems. *IEEE Control Systems Magazine*, **11**, 1, 34-37.

SINNASAMY, R.N. (1990) Use of neural networks for sensor failure detection in a control system. *IEEE Control Systems Magazine*, **10**, 3, 49-55.

SONG, J., XU, X., HE, X. (1997) A stable neural networks controller for dynamic system. *Proceedings of the IEEE Singapore international symposium on control theory and applications, Singapore, July 29-30*, 130-134.

SUTTON, R.S., BARTO, A.G., WILLIAMS, R.J. (1992) Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, **12**, 2, 19-22.

TANOMARU, J., OMATU, S. (1992) Process control by on-line trained neural controllers. *IEEE Transactions on Industrial Electronics*, **39**, 6, 511-521.

VENUGOPAL, K.P., SUDHAKAR, R., PANDYA, A.S. (1992) On-line learning control of autonomous underwater vehicles using feedforward neural networks. *IEEE Journal of Oceanic Engineering*, **17**, 4, 308-319.

WERBOS, P.J. (1991) An overview of neural networks for control. *IEEE Control Systems Magazine*, **11**, 1 40-41.

YABUTA, T., YAMADA, T. (1992) Neural network controller characteristics with regard to adaptive control. *IEEE Transactions on Systems, Man and Cybernetics*, **22**, 1, 170-177.

YUH, J., LAKSHMI, R. (1993) Intelligent control system for remotely operated vehicles. *IEEE Journal of Oceanic Engineering*, **18**, 1, 55-62.

YUH, J. (1990) Neural net controller for underwater robotic vehicles. *IEEE Journal of Oceanic Engineering*, **15**, 3, 161-166.