Int. J. Engng Ed. Vol. 20, No. 2, pp. 212–225, 2004
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Neural Network Control for Teaching Purposes*

ALEJANDRO M. SUÁREZ
Electronic Engineering Department, Federico Santa María Technical University, Casilla 110V, Valparaíso, Chile

MANUEL A. DUARTE-MERMOUD, NICOLÁS H. BELTRÁN
Electrical Engineering Department, University of Chile, Casilla 412-3, Santiago, Chile
E-mail: mduartem@cec.uchile.cl

DANILO F. BASSI
Computer Science Department, University of Santiago of Chile, Casilla 5328, Santiago, Chile

The comparison of four control schemes based on neural networks for teaching purposes is studied in this paper. The inverse model, inverse model loop, direct scheme with and without models are investigated and compared. The study is based on computer simulations of linear and nonlinear single-input single-output (SISO) typical plants and take into account different aspects such as the training period, the training method (on-line, off-line, simple or with epochs, etc.) network configuration and training signals amongst others. This set of experiences can be used in an advanced control course in the Electrical Engineering curricula, to teach different aspects of neural control.

INTRODUCTION

THE NECESSITY of controlling complex dynamical systems under uncertainties motivates the use of neural networks (NN) due to their capacity of learning, approximate functions, classify patterns and the potential of using parallel hardware. Nowadays, the NN field has a very broad range of applications [1–6], making its study an important subject in Electrical Engineering Programs.

The most commonly used type of NN is the so-called feedforward multilayer, where no information is fedback during the operation. Certainly, there is feedback information during the training process. Supervised training methods are typically used when the NN is trained to learn about input/output patterns. Often, versions of the backpropagation learning algorithms are used to adjust the synaptic weights during the training process. Generally, this algorithm is slow and usually take a long time to converge [7]. The neural activation functions are usually sigmoid functions, but Gaussiann functions are also used [8].

The property of multilayer NN of approximating functions [9] is a key factor in the majority of control applications. Such a NN are able to generate input/output mappings that approximate any function with any degree of accuracy, having a suitable number of hidden layers. Any approximation can be done using a multilayer NN with only one hidden layer or two weighting layers [10].

* Accepted 30 September 2003.

In modeling the input/output behavior of a dynamical system the NN is trained using input/output data and the weights are adjusted using usually the backpropagation method [7]. Since typical applications involve nonlinear systems, the NN is trained for a class of inputs and initial conditions. The basic assumption is that the static map generated by the NN can suitably represent the behavior of the system in the operating range for a particular application.

When a multilayer NN is trained as a controller (either in open or closed loop) all the previous conclusions can also be drawn. The difference is that the NN desired output is not really available (the controller has to generate a suitable input to the plant) so it has to be induced from the knowledge of the plant desired output. To this extent, model-based approximations or NN models for the plant or its inverse are used. In the latter it is assumed that the inverse dynamics of the plant can be represented by a NN [8, 9, 11–13]. Several other control strategies based on NN have been lately proposed in the control literature [14–22] and recently, a special issue of *Journal Automatica* has been devoted to this important subject [23].

Since teaching advanced control strategies involve NN, a comparative study of four classical control schemes based on NN is performed in this work. The dynamical processes used in the study are linear and nonlinear plants of first and second order, and they are taken from the control literature [7]. Simulation tools were specially developed in MATLAB SIMULINK to work in a modular environment.

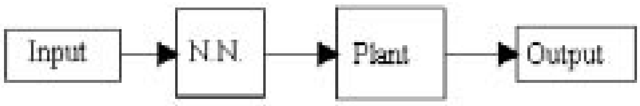

Fig. 1. Dynamic inverse control scheme; Method C1.

The study takes into account different aspects such as the training period, the training method (on-line, off-line, simple or with epochs) network configuration and training signals amongst others. This set of experiences are used to teach NN control in the course EL650 Advanced Control at the Electrical Engineering Department of the University of Chile.

NEURAL NETWORK CONTROL STRUCTURES

A brief description of the control structures based on NN that will be used in this study is given in what follows. The purpose of this description is to give the reader only the basis of these methods since a deeper understanding can be obtained directly from the references given in each case.

Representative examples of control structures using neural networks can be found in [11, 24] applied to linear and nonlinear plants. The network parameters are adjusted either off-line or on-line. These control schemes can be classified as direct or indirect control schemes. In the former the parameter adjustment of the NN is performed in such a way that a measure of the tracking error is minimized (difference between the actual plant output and the desired plant output). In the latter the neural networks are used to identify the plant and its inverse, and the learning objective is to minimize the identification error (difference between the actual plant output and the estimated plant output).

Dynamic inverse control scheme

The control schemes showed in [25, 26] and the generalized learning scheme mentioned in [12] are based on the identification of the plant inverse dynamic. That is to say, the controller is constructed to approximate the plant inverse dynamic (see Fig. 1). These networks are applicable to those cases where the plant can be exactly represented by a discrete model and its dynamics is invertible [25, 27]. For simulation purposes the block denoted by N.N. in Fig. 1 is a tool developed in SIMULINK which represent the inverse dynamic of the plant.

An important aspect of inverting a system dynamic is the availability of the state variables of the plant. In the discrete time case, where the plant is represented by NARMAX models [28], the inverse is defined in terms of the past values of the input and the output.

Dynamic inverse in the control loop

More conventional schemes using standard controllers of PID type and the system inverse have also been studied [13]. In this case the neural network represents, as approximate as possible, the dynamic inverse of the plant in the working range, so that the neural network in series with the plant are seen by the controller as a much simpler transfer function. (See Fig. 2).

Direct control scheme without model

In this scheme the parameters of the controller network are adjusted accordingly with the gradient of a cost function defined in terms of the output error. The derivatives of the output error with respect to the control signals are computed in an approximate fashion by finite differences [12]. The method to approximate the partial derivatives are given in the Appendix.

This scheme can be applied even in those cases where the dynamic of the system in not invertible (see Fig. 3). For simulation purposes the block called N.N. in Fig. 3 is a tool developed in SIMULINK which implements a controller with a neural network trained on-line.

Direct control scheme with model

In this control scheme, the parameters of the controller network are adjusted in the same way as in the previous scheme accordingly with the gradient of a cost function. The derivatives of the output error with respect to control signals are computed via back propagation of the error through the neural model of the process (identification network represented by N.N.2 in Fig. 4) [7, 25]. This scheme can also be applied in those cases where the dynamic of the plant is not invertible. For simulation purposes the identification

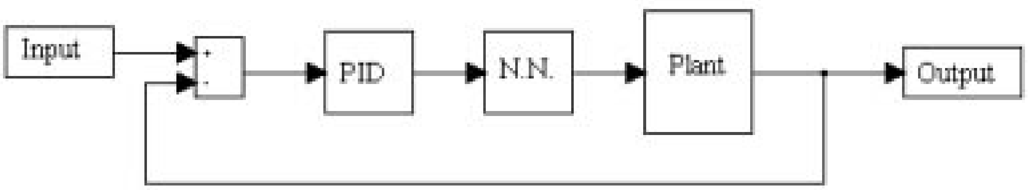

Fig. 2. Dynamic inverse in the control loop scheme; Method C2.

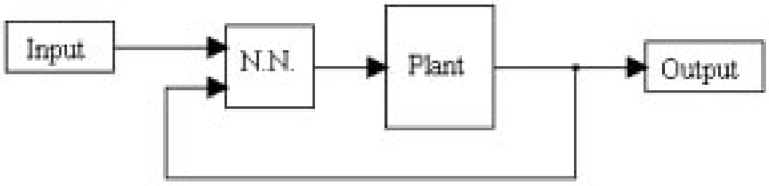

Fig. 3. Direct control scheme without model; Method C3.

network (N.N.2) and the controller network (N.N.1) in Fig. 4 are tools developed in SIMULINK, which represent the dynamic of the plant, trained either off-line or on-line, and the controller of the plant respectively.

COMPARISON OF THE CONTROL SCHEMES

To evaluate the behavior of the four different control schemes mentioned in the previous section, they were applied to control first and second order linear and nonlinear processes. The processes are described by equations or transfer functions (1) through (4) and some of them were taken from reference [7], since they have been thoroughly analyzed using different NN controllers.

Linear systems:

$$H(z) = \frac{.00995}{z - .99} \tag{1}$$

$$H(z) - \frac{.00487z + .00474}{z^2 - 1.91z + .923} \tag{2}$$

Non-linear systems:

$$y_p(k+1) = \frac{y_p(k)}{1 + y_p(k)^2} + u^3(k) \tag{3}$$

$$y_p(k+1) = \frac{y_p(k)\, y_p(k-1)\left[y_p(k) + 2.5\right]}{1 + y_p^2(k) + y_p^2(k-1)} + u(k) \tag{4}$$

NEURAL NETWORK CHARACTERISTICS

The networks used in this study are of feedforward type with linear activation function in the output layer and tanh activation functions in the hidden layers. This type of neural networks is quite known and can be easily implemented in real time.

The feedforward networks with at least one hidden layer are able to approximate any nonlinear function with an arbitrary degree of accuracy [9]. For nonlinear cases it is enough to consider neural networks with three layers of synaptic weights: an input stage, two hidden layers and one output layer. With two hidden layers a better convergence of the training process is achieved.

To describe the topology of the feedforward multilayer neural networks, the following notation is used

$$N^N_{i_1, i_2, \ldots i_{N+1}}$$

where

N: number of layer with synaptic weights
i_1: number of inputs
i_{N+1} number of outputs
$i_2, i_3, \ldots, i_N$: number of neurons in the (N – 1) hidden layers.

For example $N^3_{5,20,10,1}$ denotes a neural network with 3 layers with synaptic weights, 5 inputs, 20 neurons in the first hidden layer, 10 in the second and 1 in the output layer.

Denoting by **u** the input and **y** the output, with **n** and **m** representing the number of their past values

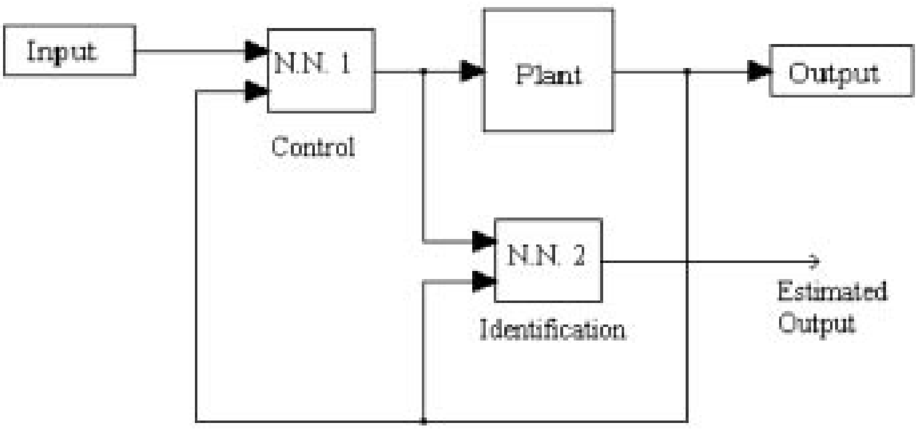

Fig. 4. Direct control scheme with model; Method C4.

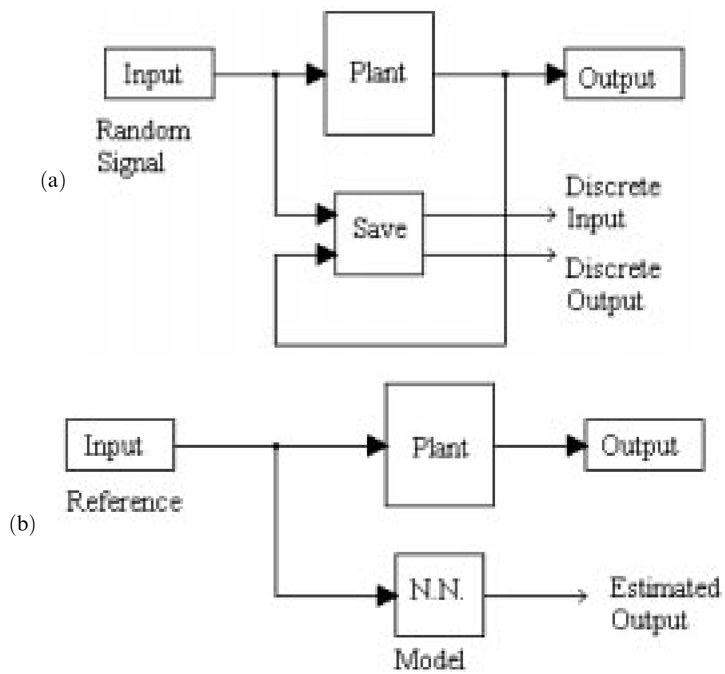

Fig. 5. Direct dynamics; a) data storage; b) network testing.

respectively, a neural network of those used in this study is represented as:

$$y(k+1) = \mathbf{N}[u(k), u(k-1), \ldots, u(k-n+1); \; y(k), y(k-1), \ldots, y(k-m+1)] \quad (5)$$

After a series of simulations tests it was determined that the most suitable characteristics of the NN used in each studied control configuration, for each process to be controlled, are the following:

1. Linear first-order equation:
   - Structure $N^1_{3,1}$
   - Inputs $[u(k), u(k-1), y(k)]$
2. Linear second-order equation:
   - Structure $N^1_{5,1}$
   - Inputs $[u(k), u(k-1), u(k-2), y(k), y(k-1)]$
3. Nonlinear first-order equation:
   - Structure $N^3_{3,20,10,1}$
   - Inputs $[u(k), u(k-1), y(k)]$
4. Nonlinear second-order equation:
   - Structure $N^3_{5,20,10,1}$
   - Inputs $[u(k), u(k-1), u(k-2), y(k), y(k-1)]$

The NN used to model the inverse or direct dynamic of the process was trained off-line, whereas the NN used to control were trained on-line.

Epochs of 300 pairs of data were used in the off-line training, obtained from a random noise uniformly distributed in the range [–2,+2] used as input signal. The training period was 10 000 epochs.

The signal sin(t) + sin(4t) was used as input signal for the on-line training of NN controller. Thus, the network is trained with similar type of signals used in the reference (sin(2t)) with a learning rate of 0.25. In all cases sampling period of 0.01 was used.

Generation of dynamical neural model

In the control schemes mentioned in Section 2, the direct or the inverse dynamic of the process is used. To implement these dynamics with neural networks, functional modular tools were developed in SIMULINK [29] to generate the neural networks, to generate and to store data for the training stage, to train and to test the dynamic once the NN is trained.

For example, Figs 5 and 6 show the data generation and data storage for training purposes and also for testing the dynamic (direct and inverse) of the process, once the NN has been trained. Blocks of noise generation, data storage, direct NN and inverse NN are employed.

SIMULATION RESULTS

The control configurations described in the second section with the structure defined in the third section, applied to each one of the processes described by equations (1) through (4), were tested by simulation using as reference input the signal r(t) = sin(2t).

For each case studied, several simulation tests were previously performed and the results shown for each plant in this section are the best found using all the information acquired from these tests. These previous tests allowed to define a series of parameters involved in the NN such as the training period, the training method (on-line, off-line,

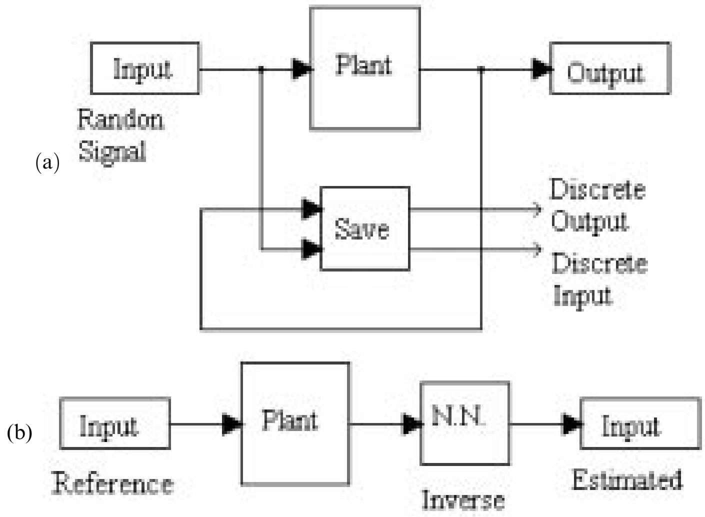

Fig. 6. Inverse dynamics: a) data storage; b) network testing.

simple or with epochs) network configuration, training signals. training rate, etc.

Linear first-order plant

Figures 7, 8, 9 and 10 show the behavior of the controlled systems when the four methods mentioned in the second section are applied to the first-order plant described by equation (1).

It is observed that schemes C1 and C2 behave quite well for all times and the true output y(t) reaches the reference signal r(t), since the NN have been trained off-line using a long period of time. Schemes C3 and C4 are trained on-line and the adjustment is done at each sampling period. This explains the difference between the true output and the reference observed during the first instants of time exhibited by control scheme C3, but finally the output reaches the reference. The control scheme C4 shows a bigger difference between y(t) and r(t) because of the NN controller and the NN identifier are adjusted on-line, and therefore the time needed to adjust both networks is much longer as shown in Fig. 10.

Linear second-order plant

Figures 11, 12, 13 and 14 present the response of the controlled linear second-order system (2) using the four schemes described in second section.

Control schemes C1 and C2 have the same training period of that used in the linear first order case, although the error between y(t) and r (t) is larger. Instead, in control schemes C3 and C4 the training period was augmented n_v times, where n_v is the number of vectors considered in one epoch, obtaining a reduction in the tracking error as shown in Figs 13 and 14.

Non-linear first-order plant

Figures 15, 16, 17 and 18 illustrate the behavior of the controlled nonlinear first order system applying the four schemes described above.

In this case the four schemes hardly control the

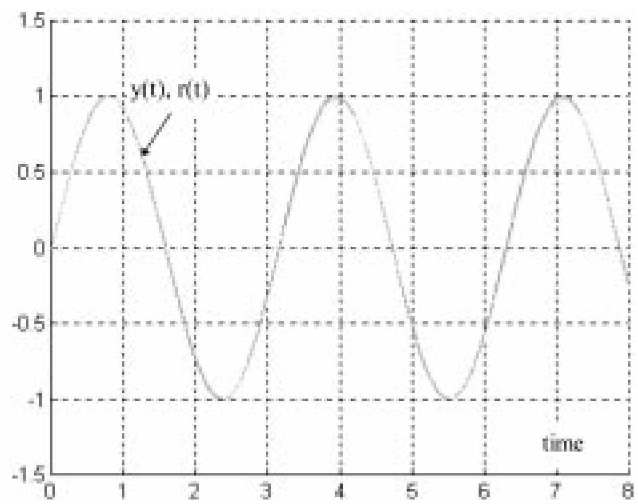

Fig. 7. Dynamic inverse method applied to a linear first-order plant.

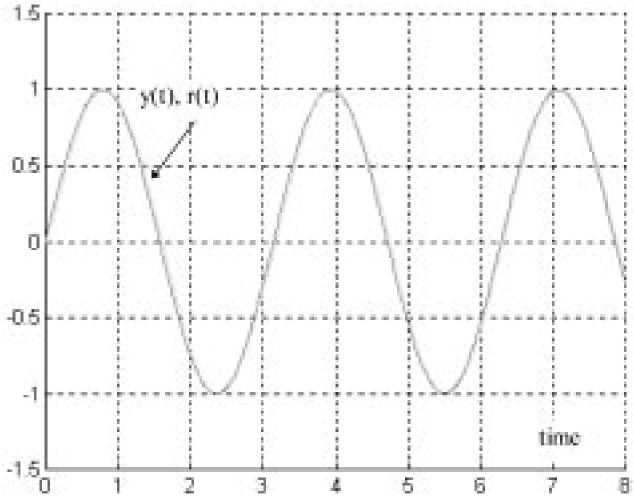

Fig. 8. Dynamic inverse in the loop method applied to a linear first-order plant.

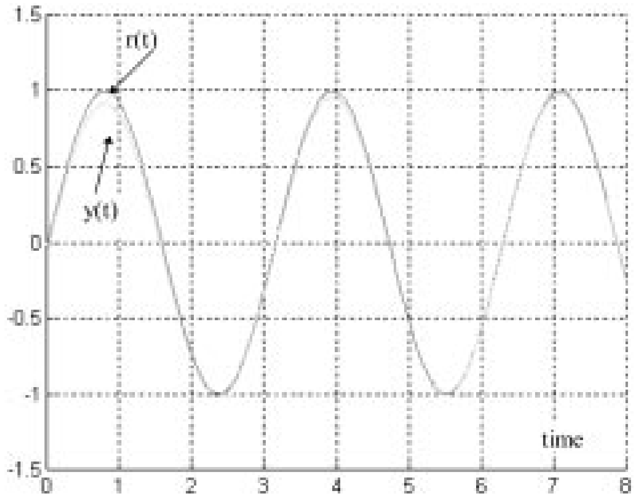

Fig. 9. Direct scheme without model applied to a linear first-order plant.

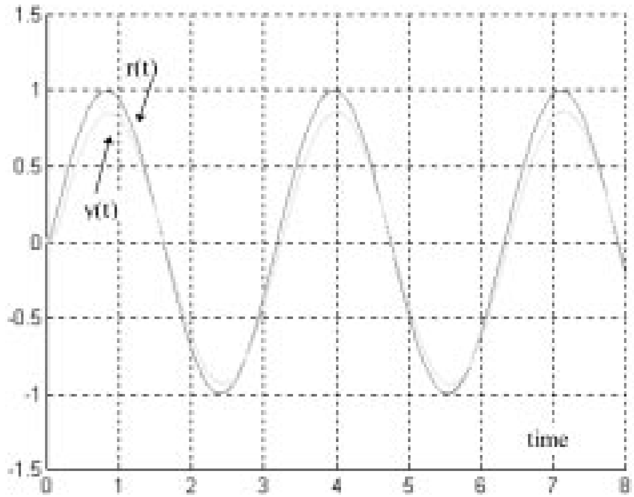

Fig. 10. Direct scheme with model applied to a linear first-order plant.

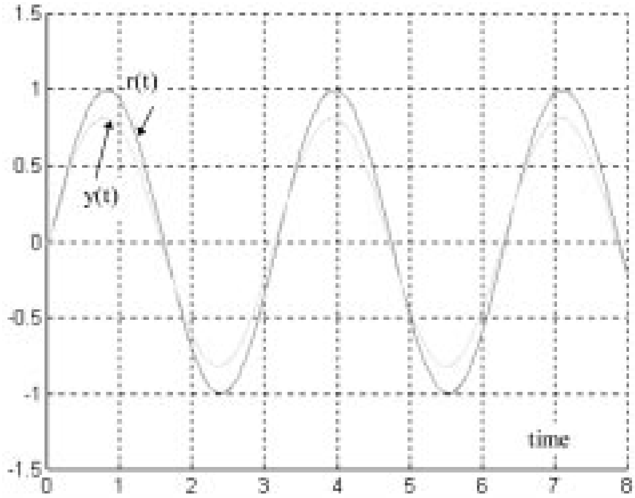

Fig. 11. Dynamic inverse method applied to a linear second-order plant.

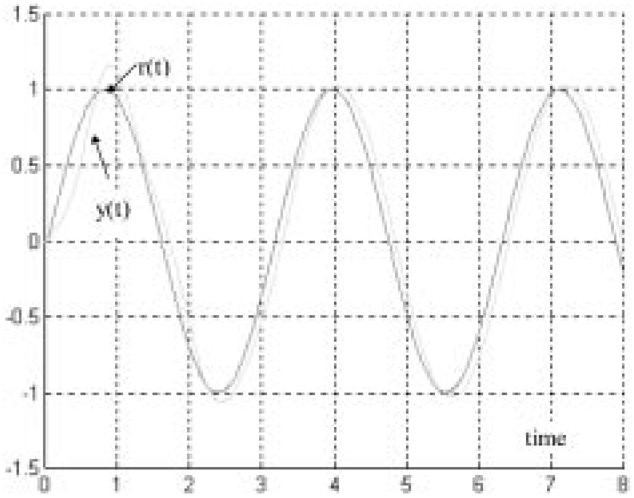

Fig. 12. Dynamic inverse in the loop method applied to a linear second-order plant.

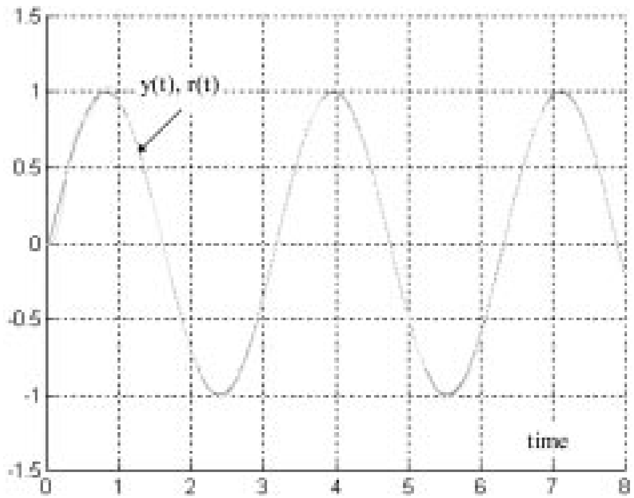

Fig. 13. Direct scheme without model applied to a linear second-order plant.

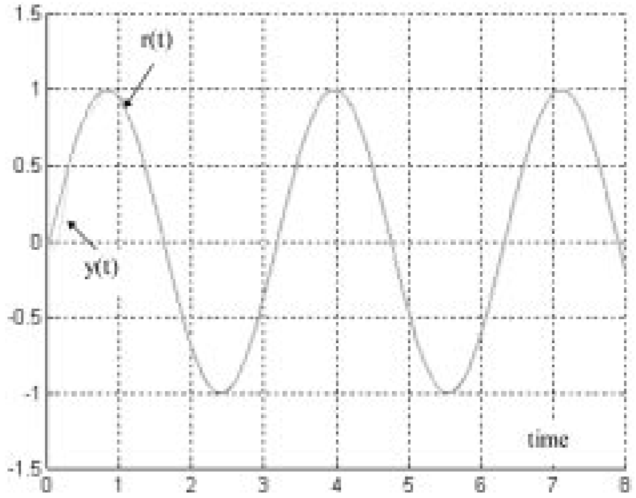

Fig. 14. Direct scheme with model applied to a linear second-order plant.

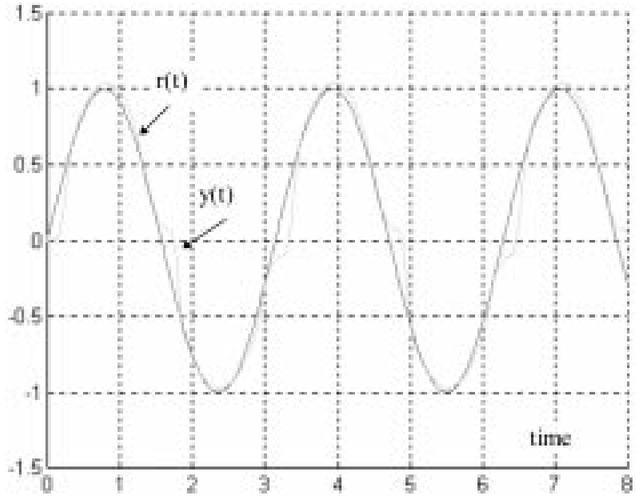

Fig. 15. Dynamic inverse method applied to a nonlinear first-order plant.

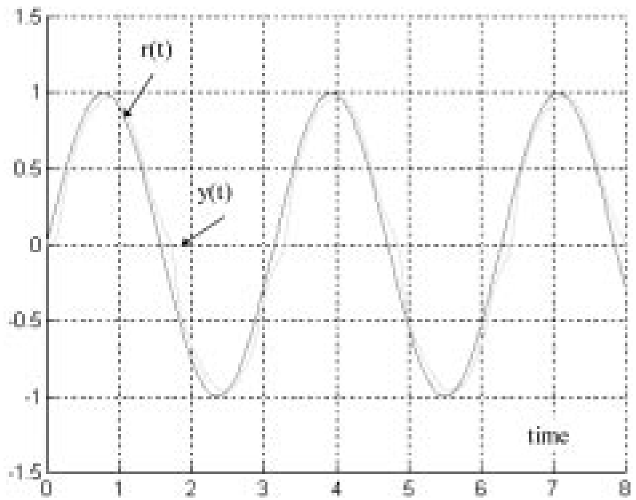

Fig. 16. Dynamic inverse in the loop method applied to a nonlinear first-order plant.

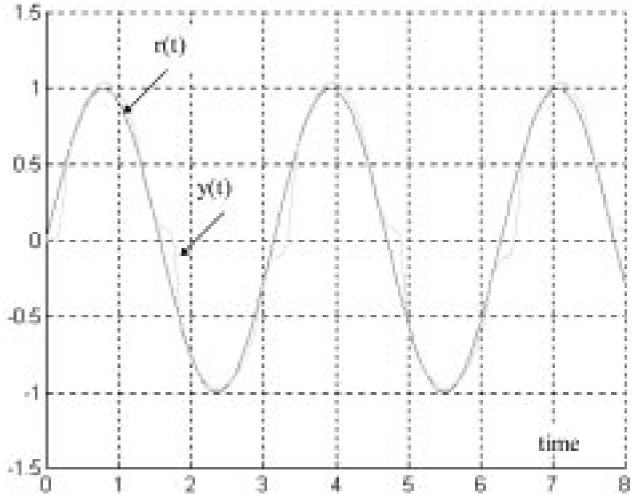

Fig. 17. Direct scheme without model applied to a nonlinear first-order plant.

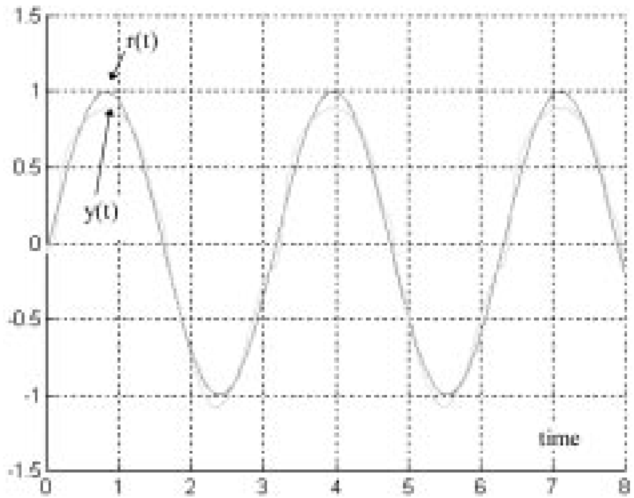

Fig. 18. Direct scheme with model applied to a nonlinear first-order plant.

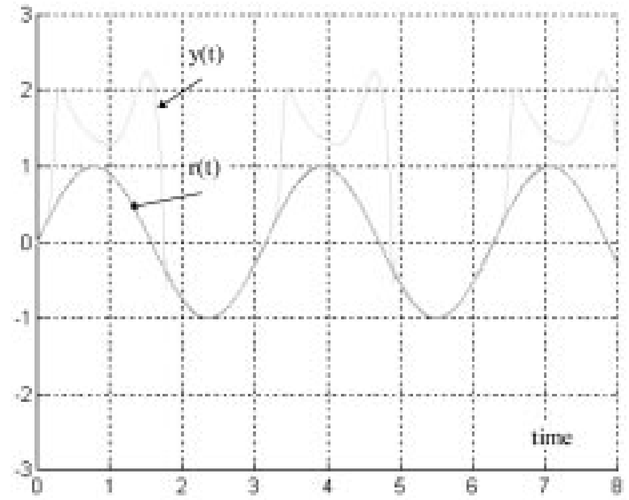

Fig. 19. Dynamic inverse method applied to a nonlinear second-order plant.

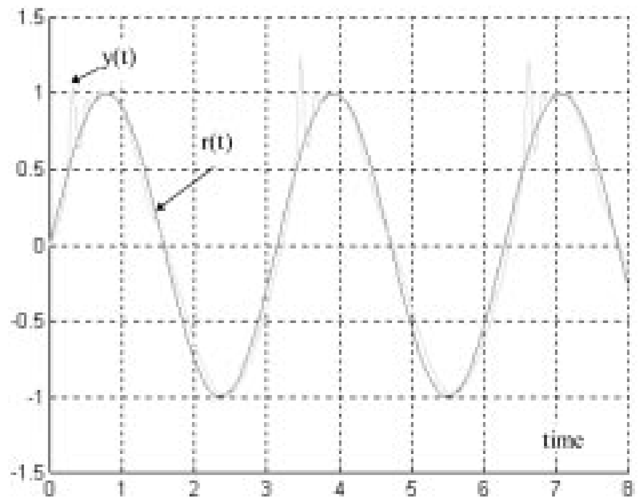

Fig. 20. Dynamic inverse in the loop method applied to a nonlinear second-order plant.

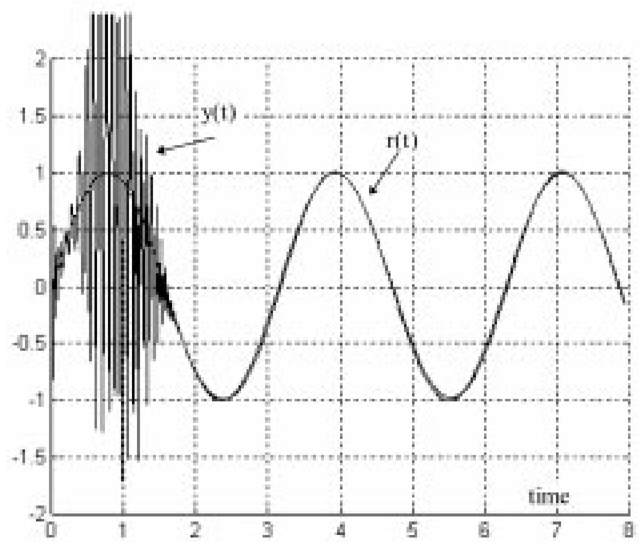

Fig. 21(a). Direct scheme with model applied to a nonlinear second-order plant; transient stage.

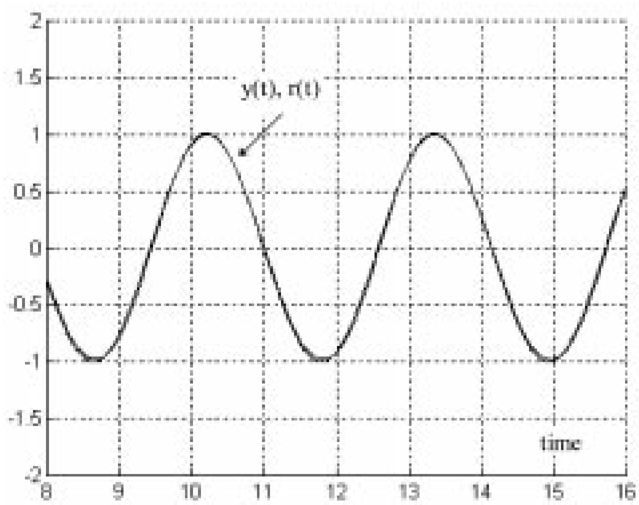

Fig. 21(b). Direct scheme with model applied to a nonlinear second-order plant; stationary stage.

process in the zone where the nonlinearity is more severe. The scheme C4 exhibits the best response of all.

Nonlinear second-order system

Figures 19, 20, 21(a) and 21(b) indicate the behavior of the controlled nonlinear second-order system employing three of the four schemes shown above.

The response of the system using the scheme C3 is not shown since the overall response is quite oscillatory with a large amplitude. Figure 21(a) shows the transient part of the response using scheme C4, whereas Fig. 21(b) shows the stationary part. Scheme C4 exhibits the best response of all schemes applied in this case.

CONCLUSIONS

Teaching advanced control strategies is an important topic in the automatic control area. Electrical Engineering students who decide to specialize in control have many different choices to control linear and nonlinear plants, depending upon their complexity. One of the most challenging methods to control industrial plants is based on NN. From the set of experiences based on NN control studied in this paper, we can draw the following conclusions.

The control scheme C1 gives in general a good response in almost all the processes studied depending only on the training of the neural networks. Similar to all other control schemes, this scheme does not perform well when faced to control the nonlinear second-order system. The main drawback of this scheme is the fact that external perturbations or parameter variations cannot be properly handled since no feedback route is available.

The control scheme C2 is the one which exhibits the best overall performance in all the cases studied. External perturbation and parameter variations can be handled since a feedback route is present. If the input data to NN has a range beyond the range used during the training period or parameter variations are large enough, the response is heavily affected since the NN has a limited range of generalization.

The control scheme C3 has a poor behavior when faced to control nonlinear systems of order two or higher with severe nonlinearities. This is mainly due to the fact that derivatives of errors with respect to the control signals (approximated by finite difference [12]) are used in the adjustment.

The control scheme C4 has a poor behavior in the transient stage when it is used to control nonlinear second-order (or higher) plants, but in steady-state the results are quite acceptable. A longer training period on-line with respect to scheme C3 is needed since the identification error given by the NN identifier is used. The NN controller uses only the present value of the error to adjust the parameters on-line.

If process parameters do not change with time (or the rate of change is slow) control scheme C2 still exhibits a reasonably good behavior. Control scheme C2, with its on-line adjustment, presents a quite good response since the scheme is adjusted when changes in plant parameters are produced.

Furthermore, a new alternative scheme can be obtained based on control scheme C4 by adjusting the synaptic weight using the present value of the error as well as the past and the future values. This can be done introducing a third NN being able to predict the future values of the error. The main challenge here is to consider the error propagation of present and past values between the NN identifier and the NN controller.

Based on the set of experiences developed in this paper, a computer simulation laboratory has been designed for the course EL650 Advanced Control, lectured at the Electrical Engineering Department of the University of Chile since Fall 1999. Student motivation for knowing and applying NN control strategies has noticeably increased since the laboratory implementation in the course.

Aknowledgments—The results reported in this work haven been supported by CONICYT-CHILE through grants FONDECYT 1970351 and FONDECYT 1941002. The partial support of DICYT-USACH 69619BA is also greatly appreciated.

REFERENCES

1. H. Hashimoto, T. Kubota, M. Kudou, and F. Harashima, Self-organizing visual servo system based on neural networks, *IEEE Control Systems Magazine*, **12**(2) April 1992, pp. 31–6.
2. C. Aldrich, J. Van Deventer, and M. Reuter, The application of neural nets in the metallurgical industry, *Mineral Engineering*, **7**(5/6) 1994, pp. 793–809.
3. F. Flament, J. Thibault and D. Hodouin, Neural network-based control of mineral grinding plants, *Mineral Engineering*, **6**(3) 1993, pp. 235–249.
4. K. Okata and T. Inoue, Mineral process control by neural network, *Proc. IFAC Workshop on Expert Systems in Mineral and Metal Processing*, Espoo, Finland, August 1991, pp. 167–171.
5. L. Behera, M Gopal and S. Chaudhury, On adaptive trajectory tracking of as robot manipulator using inversion of its neural emulator, *IEEE Trans. on Neural Networks*, **7**(6) 1996, pp. 1401–1414.
6. F. Sun, Z. Sun and P. Y. Woo, Neural network-based adaptive controller design of robotic manipulators with an observer, *IEEE Trans. on Neural Networks*, **12**(1) 2001, pp. 54–67.
7. K. S. Narendra and K. Parthasarathy, Identification and control of dynamical system using neural networks, *IEEE Transactions on Neural Networks* , **1**(1) March 1990, pp. 4–27.

8. K. J. Hunt and D. Sbarbaro, Neural networks for nonlinear internal model control, *IEE Proceedings-D*, **138**(5), September 1991, pp. 431–438.
9. K. Hornik, M. Stincombe, and H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *IEEE Trans. on Neural Networks*, **1**(3) 1990, pp. 211–223.
10. S. Chen, S. A. Billings and P. M. Grant, Nonlinear system identification using neural networks, *Int. J. Control*, **51**(6) 1990 pp. 1191–1214.
11. K. Hunt and D. Sbarbaro, Studies in neural networks based control, in *Neural Networks for Control and Systems*, K. Warwick, G. Irwin, and K. Hunt, (eds), Vol. 46, IEE Control Engineering Series, Peter Perigrinus Ltd., 1992. Ch. 6, pp. 94–122.
12. D. Psaltis, A. Sideris, and A. Yamamura, Neural controllers, *Proc. First IEEE Int. Conf. Neural Networks*, San. Diego, 1987, Vol. 4, pp. 551–558.
13. R. Murray-Smith, D. Sbarbaro, and D. Neumerkel, Neural networks for modeling and control of a non-linear dynamic systems, *Proc. IEEE Symp. Intelligent Control*, Glasgow, Scotland, August 1992, pp. 122–127.
14. D. V. Prokhorov, Adaptive critic design, *IEEE Trans. on Neural Networks*, **8**(5) 1997, pp. 997–1007.
15. K. S. Narendra and S. Mukhopadhyay, Adaptive control using neural networks and approximate models, *IEEE Trans. on Neural Networks*, **8**(3) 1997, pp. 475–485.
16. J. R. Noriega and H. Wang, Direct adaptive neural network control for unknown nonlinear system and its application, *IEEE Trans. on Neural Networks*, **9**(1) 1998, pp. 27–34.
17. M. Zhihong, H. R. Wa and M. Palaniswami, An adaptive tracking controller using neural networks for a class of nonlinear systems, *IEEE Trans. on Neural Networks*, **9**(5) 1998, pp. 947–955.
18. J. A. Farrell, Stability and approximation convergence in nonparametric nonlinear adaptive control, *IEEE Trans. on Neural Networks*, **9**(5) 1998, pp. 1008–1020.
19. R. B. McLain, M. A. Henson and M. Pottmann, Direct adaptive control of partially known nonlinear systems, *IEEE Trans. on Neural Networks*, **10**(3) 1999, pp. 714–721.
20. D. Shukka, Multiple neural-networks-based adaptive controller using orthonormal activation function neural networks, *IEEE Trans. on Neural Networks*, **10**(6) 1999, pp. 1494–1501.
21. S. Seshagiri and H. K. Khalil, Output feedback control of nonlinear systems using RBF neural networks, *IEEE Trans. on Neural Networks*, **11**(1) 2000, pp. 69–79.
22. S. Fabri and V. Kadirkamanathan, Dynamic structure neural networks for stable adaptive control of nonlinear systems, *IEEE Trans. on Neural Networks*, **7**(5), 1996, pp. 1151–1167.
23. K. S. Narendra and F. L. Lewis, Special Issue on neural networks feedback control, *Automatica*, **37**(8) August 2001, pp. 1147–1301.
24. F. C. Chen, Backpropagation neural networks for nonlinear self-tuning adaptive control, *IEEE Control Systems Magazine*, **10**(3) April 1990, pp. 44–48.
25. W. Li and J. J. E. Slotine, Neural network control of unknown nonlinear systems, *Proc. 1989 American Control Conference*, Pittsburgh, PA, June 1989, pp. 1136 –1141.
26. W. Miller, R. Hewes, F. Glanz, and L. Kraft, Real-time dynamic control of an industrial manipulator using a neural networks-based learning controller, *IEEE Trans. Robotics and Automation*, **6**(1) January 1990, pp. 1–9.
27. M. Jordan and D. Rumelhart, Forward models: supervised learning with distal teacher, *Cognitive Science*, **16**, 1992, pp. 307–354.
28. S. Chen and S. Billings, Representation of nonlinear systems: the NARMAX model, *Int. J. Control*, **49**(3) 1989, pp. 1012–1032.
29. *Neural Network TOOLBOX for use with MATLAB. User's Guide*, The MATH WORKS Inc. (1997).

APPENDIX

In the training stage of the control scheme C3 it is necessary to know the variation of the process output with respect to the control signals, i.e. the partial derivatives.

The partial derivative approximation between the output and the input of a process can be done using a Taylor series expansion. First, the derivative of a function with respect to time is approximated using its past values as follows:

Let us define

h time delay

$$l_o = f(t) \tag{A1}$$

$$l_1 = f(t-h) \tag{A2}$$

$$l_2 = f(t-2h) \tag{A3}$$

$$l_3 = f(t-3h) \tag{A4}$$

$$x_1 = f'(t) \tag{A5}$$

$$x_2 = f''(t) \tag{A6}$$

$$x_3 = f'''(t) \tag{A7}$$

Using the Taylor series expansion of function f(t), considering the first four terms and replacing the definitions stated in equations (A1) through (A7), the following set of equations is obtained:

$$l_1 = l_0 - hx_1 + \frac{h^2}{2}x_2 - \frac{h^3}{6}x_3 \tag{A8}$$

$$l_2 = l_0 - 2hx_1 + 4\frac{h^2}{2}x_2 - 8\frac{h^3}{6}x_3 \tag{A9}$$

$$l_3 = l_0 - 3hx_1 + 9\frac{h^2}{2}x_2 - 27\frac{h^3}{6}x_3 \tag{A10}$$

Solving the set of equations for x_1 we get:

$$f'(t) = \frac{17f(t) - 18f(t-h) + 9f(t-2h) - 8f(t-3h)}{6h} \tag{A11}$$

Using the notation:

$$T = h \tag{A12}$$

$$t = kT \tag{A13}$$

and replacing in equation (A11) we obtain:

$$f'(k) = \frac{17f(k) - 18f(k-1) + 9f(k-2) - 8f(k-3)}{6T} \tag{A14}$$

We know that:

$$\frac{\partial f(t)}{\partial x(t)} = \frac{\frac{df(t)}{dt}}{\frac{dx(t)}{dt}} = \frac{f'(t)}{x'(t)} \tag{A15}$$

Then, the approximation of the partial derivative can be obtained by replacing $f'(t)$ and $x'(t)$ in (A15) by the approximations given by (A14):

$$\frac{\partial f(t)}{\partial x(t)} \approx \frac{f'(k)}{x'(k)} = \frac{17f(k) - 18f(k-1) + 9f(k-2) - 8f(k-3)}{17x(k) - 18x(k-1) + 9x(k-2) - 8x(k-3)} \tag{A16}$$

Alejandro M. Suárez received the Electronic Civil Engineer and the M.Sc degrees from the Federico Santa Maria Technical University (Chile) in 1974 and 1983 respectively. He got the Ph.D. degree in Automatic Control from the University of Chile (Chile) in 1998. From 1975 to 1978 he worked at the Electrical Engineering Department of University of Concepción as Assistant Professor. In 1978 he joined the Electronics Engineering Department of Federico Santa María Technical University where he is currently an Associate Professor. His main research interests are in neural control and signal processing.

Manuel A. Duarte-Mermoud received the degree of Civil Electrical Engineer from the University of Chile (Chile) in 1977 and the M.Sc., M.Phil. and the Ph.D. degrees, all in electrical engineering, from Yale University (USA) in 1985, 1986 and 1988 respectively. From 1977 to 1979, he worked as Field Engineer at Santiago Subway. In 1979 he joined the Electrical Engineering Department of University of Chile, where he is currently Professor. His main research interests are in robust adaptive control (linear and nonlinear systems) and system identification. Dr Duarte is member of the IEEE and past President of ACCA, Chilean National Member Organization of IFAC.

Nicolas H. Beltran received the degree of Civil Electrical Engineer from the University of Chile (Chile) in 1974 and the Master of EE and Doctor in Applied Sciences degrees from the Katholieke Universiteit Leuven (Belgium) in 1981 and 1985, respectively. Since 1985 he is with the Electrical Engineering Department at the University of Chile, teaching and conducting research, where he is currently an Associate Professor. His areas of interest are

gas sensor technologies and instrumentation systems. Dr Beltran is member of the IEEE and past President of the Chile Section. He is also member of ACCA (Chilean Automatic Control Association).

Danilo F. Bassi received the Civil Electrical Engineer and the M.Sc. degrees from the University of Chile (Chile), both in 1986. He got the Ph.D. in Computer Science from University of Southern California (USA) in 1990. From 1991 to 1995 he worked at the Industrial Engineering Department of University of Chile, as an Assistant Professor. In 1995 he joined the Computer Science Department of the University of Santiago, where he is currently an Associate Professor. His main research interests are in neural networks, robotics and applications.