Recurrent Neural Control of a Continuous Bioprocess Using First and Second Order Learning

Carlos-Román Mariaca-Gaspar, Julio-César Tovar Rodríguez, and Floriberto Ortiz-Rodríguez

Department of Communication and Electronic Engineering,
Higher School of Mechanic and Electrical Engineering - Zacatenco,
National Polytechnic Institute
Av. Juan de Dios Batiz s/n, Mexico D.F., 07738, Mexico
{cmariacag,jctovar,flortiz}@ipn.com

Abstract. The propose of this paper is to introduce a new Kalman Filter based in a Recurrent Neural Network topology (KFRNN) and a recursive Levenberg-Marquardt (L-M) algorithm. Such algorithm is able to estimate the states and parameters of a highly nonlinear continuous fermentation bioprocess in noisy environment. The control scheme is direct adaptive and also contains feedback and feedforward recurrent neural controllers. The proposed control scheme is applied for real-time identification and control of continuous stirred tank bioreactor model, taken from the literature, where a fast convergence, noise filtering and low mean squared error of reference tracking were achieved.

Keywords: Kalman filter, recurrent neural network, recurrent trainable neural network controller, real-time backpropagation learning, recursive Levenberg-Marquardt learning, real-time direct adaptive neural control, continuous stirred tank reactor bioprocess.

1 Introduction

In a FFNN the signals are transmitted only in one direction, starting from the input layer, subsequently through the hidden layers to the output layer, which requires applying a tap delayed global feedbacks and a tap delayed inputs to achieve a Nonlinear Autoregressive Moving Average (NARMAX) neural dynamic plant model. A new Kalman Filter Recurrent Neural Network (KFRNN) topology and the recursive Backpropagation (BP) type learning algorithm in vector-matrix form was derived and its convergence was studied, [4], [5]. But the recursive BP algorithm, applied for KFRNN learning, is a gradient descent first order learning algorithm which not permits to augment the precision and to accelerate the learning. So, the main contribution in this paper is to use a second order learning algorithm for the KFRNN like the Levenberg-Marquardt (L-M) algorithm [6]-[8], in order to obtain a fast convergence, avoid noise and the effect of unmodeled dynamics as much as possible. The KFRNN with L-M learning will be applied for a Continuous Stirred Tank Reactor (CSTR)

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 211–222, 2013.
© Springer-Verlag Berlin Heidelberg 2013

model, [9], identification and control. In [10], [11] a comparative study of linear, nonlinear and neural-network-based adaptive controllers for a CSTR is done. The papers proposed to use the neuro-fuzzy and adaptive nonlinear control systems design applying FFNNs (multilayer perceptron and radial basis functions NN). The proposed control gives a good adaptation to the nonlinear plants dynamics, better with respect to the other methods of control. The application of KFRNNs together with the recursive L-M could avoid these problems improving the learning and the precision of the plant states estimation.

2 Topology and Learning of the RTNN

This section is dedicated to the topology, the BP and the L-M algorithms of the Recurrent Trainable Neural Network, [5], [15]-[17] learning. The RTNN topology could be obtained from the KFRNN topology removing the output local and global feedbacks. The RTNN was used as a feedback and feedforward controller in direct adaptive neural control scheme.

2.1 RTNN Topology

The RTNN topology is described by the following vector-matrix equations:

$$X(k+1) = A_1 X(k) + BU(k) \tag{1}$$

$$B = \left[B_1; B_0\right];\ U^T = \left[U_1^T; U_2^T\right] \tag{2}$$

$$A = \text{block-diag}\left(A_i\right),\ \left|A_i\right| < 1 \tag{3}$$

$$Z_1(k) = G[X(k)] \tag{4}$$

$$C = \left[C_1; C_0\right];\ Z^T = \left[Z_1^T; Z_2^T\right] \tag{5}$$

$$V(k) = CZ(k) \tag{6}$$

$$Y(k) = F[V(k)] \tag{7}$$

Where: X, Y, U are vectors of state, output, and augmented input with dimensions N, L, (M+1), respectively, Z is an (L+1) –dimensional input of the feedforward output layer, where Z_1 and U_1 are the (Nx1) output and (Mx1) input of the hidden layer; the constant scalar threshold entries are Z_2 = -1, U_2 = -1, respectively; V is a (Lx1) pre-synaptic activity of the output layer; the super-index T means vector transpose; A is (NxN) block-diagonal weight matrix; B and C are [Nx(M+1)] and [Lx(N+1)]-augmented weight matrices; B_0 and C_0 are (Nx1) and (Lx1) threshold weights of the hidden and output layers; F[·], G[·] are vector-valued tanh(·) or sigmoid(·) –activation

functions with corresponding dimensions. Equation (3) represents the local stability condition imposed on all blocks of A. The dimension of the state vector X of the RTNN is chosen using the simple rule of thumb which is: N=L+M.

3 Backpropagation RTNN Learning

Following the same procedure as for the KFRNN, it was possible to derive the following updates for the RTNN weight matrices:

$$E_1(k) = F'[Y(k)]E(k); E(k) = Y_p(k) - Y(k) \tag{8}$$

$$\Delta C(k) = E_1(k)Z^T(k) \tag{9}$$

$$E_3(k) = G'[Z(k)]E_2(k); E_2(k) = C^T E_1(k) \tag{10}$$

$$\Delta B(k) = E_3(k)U^T(k) \tag{11}$$

$$\Delta A(k) = E_3(k)X^T(k) \tag{12}$$

$$\Delta vA(k) = E_3(k) \oplus X(k) \tag{13}$$

Where ΔA, ΔB, ΔC are weight corrections of the learned matrices A, B, and C, respectively; E, E_1, E_2, and E_3 are error vectors; X is a state vector; F'(·) and G'(·) are diagonal Jacobean matrices, whose elements are derivatives of the tanh activation functions . Equation (12) represents the learning of the full feedback weight matrix of the hidden layer. Equation (13) gives the learning solution when this matrix is diagonal vA, which is the present case. The initial values of the weight matrices during the learning are chosen as arbitrary numbers inside a small range. The stability of the RTNN model used as a direct controller is assured by the activation functions [-1, 1] bounds and by the local stability weight bound condition given by (3). The stability of the RTNN movement around the optimal weight point has been proved by one theorem (see the thesis of Mariaca [13] and the paper of Nava, [14]).

4 Stability Proof of the KFRNN BP Learning (Theorem of Stability)

The *Theorem of Stability of the BP RTNN* used as a system controller is proven in the PhD thesis of Mariaca [13], so it shall be given here in brief. Let the RTNN with Jordan Canonical Structure is given by equations (1)-(7). Under the assumption of RTNN identifiability made, the application of the BP learning algorithm for A(·), B(·), C(·), in general matricial form, (8)-(13) without momentum term, and the learning

rate η (k) (here it is considered as time-dependent and normalized with respect to the error) are derived using the following Lyapunov function:

$$L(k) = L_1(k) + L_2(k) \tag{14}$$

Where: L_1 (k) and L_2 (k) are given by:

$$L_1(k) = \tfrac{1}{2} e^2(k)$$

$$L_2(k) = tr\left(\tilde{W}_{A(k)} \tilde{W}^T_{A(k)}\right) + tr\left(\tilde{W}_{B(k)} \tilde{W}^T_{B(k)}\right) + tr\left(\tilde{W}_{C(k)} \tilde{W}^T_{C(k)}\right)$$

Where

$$\tilde{W}_{A(k)} = \hat{A}_{(k)} - A^*, \quad \tilde{W}_{B(k)} = \hat{B}_{(k)} - B^*, \quad \tilde{W}_{C(k)} = \hat{C}_{(k)} - C^*$$

are vectors of the estimation error and (A^*, B^*, C^*) and $(\hat{A}_{(k)}, \hat{B}_{(k)}, \hat{C}_{(k)})$ denote the ideal neural weight and the estimate of neural weight at the k-th step, respectively, for each case.

Let us define: $\psi_{\max} = \max_k \left\| \psi(k) \right\|$, and $\vartheta_{\max} = \max_k \left\| \vartheta(k) \right\|$. Here: $\psi(k) = \partial o(k) / \partial W(k)$, and $\vartheta(k) = \partial y(k) / \partial u(k)$, where W is a vector composed by all weights of the RTNN, used as a system controller, and $\|\cdot\|$ is an Euclidean norm in $\Re^n$.

Then the identification error is bounded, i.e.:

$$L(k+1) = L_1(k+1) + L_2(k+1) < 0 \tag{15}$$

$$DL(k+1) = L(k+1) - L(k) \tag{16}$$

Where the condition for L_1 (k+1) < 0 is that:

$$0 < \eta_{\max} < \frac{2}{\vartheta^2_{\max} \psi^2_{\max}}$$

and for L_2(k+1) < 0, we have:

$$\Delta L_2(k+1) < -\eta_{\max} \left| e(k+1) \right|^2 + \beta(k+1) \tag{17}$$

Note that $\eta_{\max}$ changes adaptively during the learning process of the network and:

$$\eta_{\max} = \max_{i=1}^{3} \{\eta_i\}$$

Where all: the unmodelled dynamics, the approximation errors and the perturbations, are represented by the β-term, and the complete proof of that theorem is given in [13] and the Lemma of convergence of (17) is given by Nava [14].

5 Recursive Levenberg-Marquardt Algorithm of RTNN Learning

The general recursive L-M algorithm of learning, [4], [6]-[8], [17], is given by the equations (18)-(21), where W is the general weight matrix (A, B, C), under modification; Y is the RTNN output vector which depends of the updated weights and the input; E is an error vector; Yp is the plant output vector which is in fact the target vector. Using the RTNN adjoint block diagram, [4], [5], [15]-[17] we could obtain the values of DY for each updated weight propagating D = I through it.

$$W(k+1) = W(k) + P(k)\nabla Y[W(k)]E[W(k)] \tag{18}$$

$$Y[W(k)] = g[W(k), U(k)] \tag{19}$$

$$E^2[W(k)] = \{Y_p(k) - g[W(k), U(k)]\}^2 \tag{20}$$

$$DY[W(k)] = \frac{\partial}{\partial W} g[W, U(k)]\Big|_{W=W(k)} \tag{21}$$

Applying equation (21) for each element of the weight matrices (A, B, C) to be updated, the corresponding gradient components are obtained as follows:

$$DY[C_{ij}(k)] = D_{1,i}(k) Z_j(k) \tag{22}$$

$$D_{1,i}(k) = F_i'[Y_i(k)] \tag{23}$$

$$DY[A_{ij}(k)] = D_{2,i}(k) X_j(k) \tag{24}$$

$$DY[B_{ij}(k)] = D_{2,i}(k) U_j(k) \tag{25}$$

$$D_{2,i}(k) = G_i'[Z_i(k)] C_i D_{1,i}(k) \tag{26}$$

So the Jacobean matrix could be formed as:

$$DY[W(k)] = [DY(C_{ij}(k)), DY(A_{ij}(k)), DY(B_{ij}(k))]$$

Next the given up topology and learning is applied for CSTR system control.

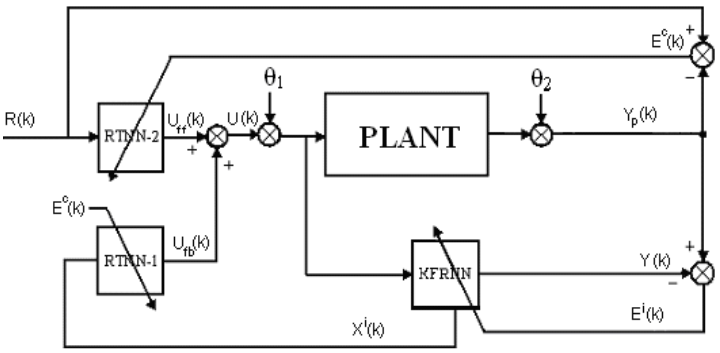

Fig. 1. Block-diagram of the closed-loop neural control

6 Direct Adaptive Neural Control of CSTR System

This part described the direct adaptive CSTR control using KFRNN as plant identifier and RTNN as two plant controllers (feedback and feedforward). The block-diagram of the control system is given on Fig. 1, [5], [15]. Next we described the linearized model of the control system. Let us to write the z-transfer function representations of the plant, the state estimation part of the KFRNN, the feedback / feedforward controllers:

$$W_p(z) = C_p\left(zI - A_p\right)^{-1} B_p \tag{27}$$

$$P_i(z) = \left(zI - A_i\right)^{-1} B_i \tag{28}$$

$$Q_1(z) = C_c\left(zI - A_c\right)^{-1} B_{1_c} \tag{29}$$

$$Q_2(z) = C_c\left(zI - A_c\right)^{-1} B_{2_c} \tag{30}$$

The control systems z-transfer functions (27)-(30) are connected by the following equation, derived from the Fig. 4, and given in z-operational form:

$$Y_p(z) = W_p(z)\left[I + Q_1(z)P_i(z)\right]^{-1} Q_2(z)R(z) + \theta(z) \tag{31}$$

$$\theta(z) = W_p(z)\theta_1(z) + \theta_2(z) \tag{32}$$

Where θ(z) represents a generalized noise term. The RTNN and the KFRNN topologies were controllable and observable, and the BP algorithm of learning was convergent, [5], so the identification and control errors tended to zero:

$$E_i(k) = Y_p(k) - Y(k) \to 0;\ k \to \infty \tag{33}$$

$$E_c(k) = R(k) - Y_p(k) \to 0;\ k \to \infty \tag{34}$$

This means that each transfer functions given by equations (27)-(30) is stable with minimum phase. The closed-loop system is stable and the RTNN-1 feedback controller compensates the plant dynamics. The RTNN-2 feedforward controller dynamics is an inverse dynamics of the closed-loop system one, which assure a precise reference tracking in spite of the presence of process and measurement noises.

7 Description of the CSTR Bioprocess

The CSTR model given in [10], [11], [18] was chosen as a realistic example for application of the KFRNN and the RTNN for solution of system identification and control problems.

The CSTR is described by the following continuous time nonlinear system of ordinary differential equations:

$$\frac{dC_A(t)}{dt} = \frac{Q}{V}\left(C_{Af} - C_A(t)\right) - k_0 C_A(t)\exp\left(-\frac{E}{RT(t)}\right) \tag{35}$$

$$\begin{aligned}\frac{dT(t)}{dt} &= \frac{Q}{V}(T_f - T(t)) + \frac{(-\Delta H)C_A(t)}{\rho C_p}\exp\left(\frac{E}{RT(t)}\right) \\ &+ \frac{\rho_c C_{pc}}{\rho C_p V} Q_c(t)\left[1 - \exp\left(\frac{-hA}{Q_c(t)\rho_c C_{pc}}\right)\right](t_{ef} - T(t))\end{aligned} \tag{36}$$

In this model is enough to know that within the CSTR, two chemicals are mixed and that they react in order or produce a product compound A at a concentration $C_A(t)$, and the temperature of the mixture is T(t). The reaction is exothermic and produces heat which slows down the reaction. By introducing a coolant flow-rate $Q_c(t)$, the temperature can be varied and hence the product concentration can be controlled. Here C_{Af} is the inlet feed concentration; Q is the process flow-rate; T_f and T_{ef} are the inlet feed and coolant temperatures, respectively; all of which are assumed constant at nominal values. Likewise, k_0, E/R, V, ΔH, ρ, C_{pc}, C_p , and ρ_c are thermodynamic and chemical constants related to this particular problem. The quantities Q_{c0}, T_0, and C_{A0}, shown in Table 1, are steady values for a steady operating point in the CSTR. The objective was to control the product compound A by manipulating $Q_c(t)$. The operating values were taken from [7] and [8], where the performance of a NN control system is reported.

Table 1. Parameters and operating conditions of the CSTR

Parameters	Parameters	Parameters
$Q = 100 \quad (L / \min)$	$E / R = 9.95 \times 10^3 \quad (K)$	$Q_{c0} = 103.41 \quad (L / \min)$
$C_{Af} = 1.0 \quad (mol / L)$	$-\Delta H = 2 \times 10^5 (cal / mol)$	$hA = 7 \times 10^5 \quad (cal / \min K)$
$T_f = T_{fC} = 350 \quad (K)$	$\rho \cdot \rho_c = 1000 \quad (g / L)$	$T_0 = 440.2 \quad (K)$
$V = 100 \quad (L)$	$C_p C_{pc} = 1 \quad (cal / gK)$	$k_0 = 7.2 \times 10^{10} \quad (1 / \min)$

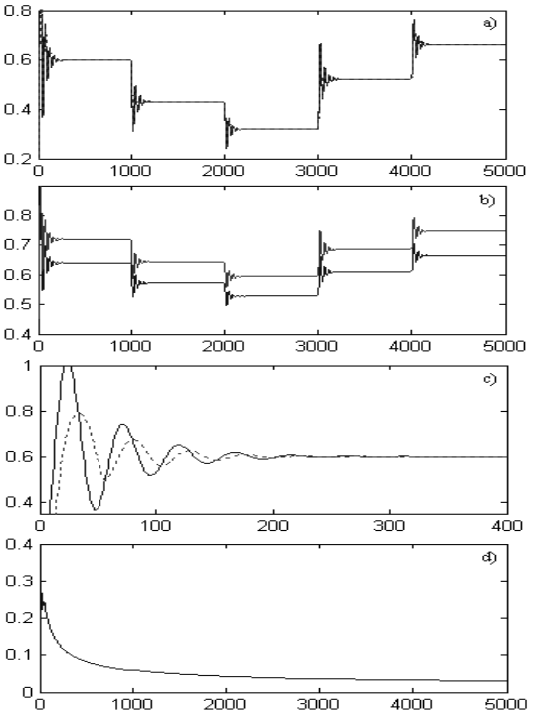

Fig. 2. Graphical results of identification using BP KFRNN learning. a) Comparison of the plant output (continuous line) and KFRNN output (pointed line); b) state variables; c) comparison of the plant output (continuous line) and KFRNN output (pointed line) in the first instants; d) MSE% of identification.

8 Simulation Results

Detailed comparative graphical simulation results of CSTR KFRNN plant identification by means of the BP and the L-M learning are given in Fig.2 and Fig.3. A 10% white noise is added to the plant inputs and outputs and the behavior of the plant identification has been studied accumulating some statistics of the final MSE% (ξ_{av}) for BP and L-M learning, which results are given in Tables 3 and 4 for 20 runs respectively.

The mean average cost for all runs (ε) of KFRNN plant identification and the standard deviation (σ) with respect to the mean value and the deviation (Δ) are presented in Table 2 for BP and L-M algorithm. The computations of these values are done by the following formulas:

$$\varepsilon = \frac{1}{n}\sum_{k=1}^{n}\xi_{av_k}, \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Delta_i^2}\ ,\ \Delta = \xi_{av} - \varepsilon \tag{37}$$

The proposed direct adaptive neural control system was applied for the given up CSTR plant. The Fig. 4 compared the reference and the plant output signals applying the L-M algorithm of learning for the KFRNN and both RTNNs. The obtained graphical (Fig. 2 and Fig. 3) and numerical (Table 2) identification results showed that the convergence of the L-M learning algorithm outperformed the BP one. The given control results (Fig. 4) showed a fast reaction and convergence of the L-M learned RNNs.

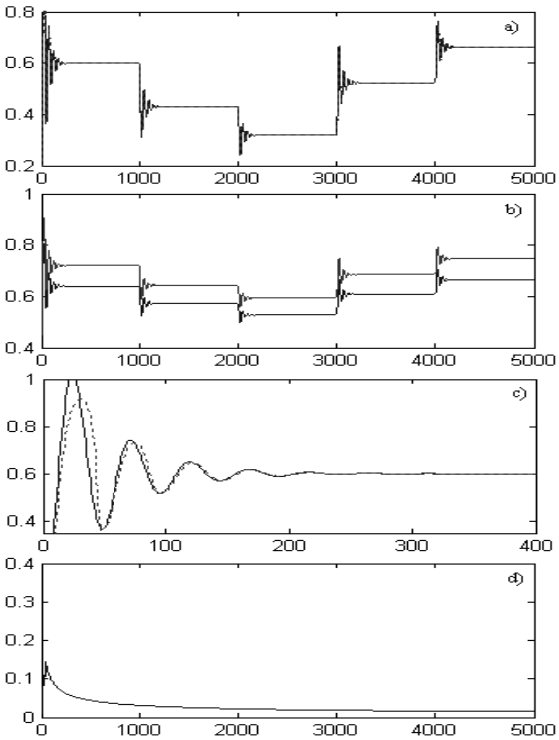

Fig. 3. Graphical results of identification using L-M KFRNN learning. a) Comparison of the plant output (continuous line) and KFRNN output (pointed line); b) state variables; c) comparison of the plant output (continuous line) and KFRNN output (pointed line) in the first instants; d) MSE% of identification.

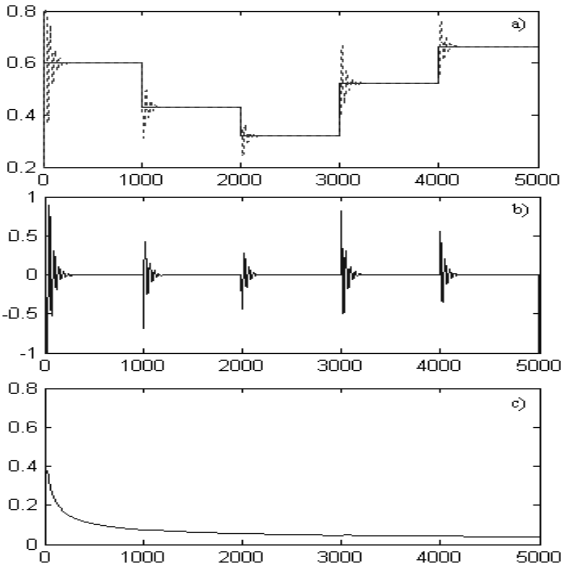

Fig. 4. Detailed graphical simulation results of CSTR plant direct neural control using LM RTNN learning a) comparison between the plant output and the reference signal; b) control signal; c) MSE% of control

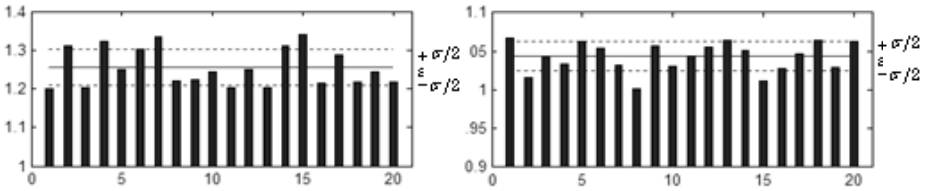

Fig. 5. Comparison between the final MSE% for 20 runs of Identification program: a) using BP algorithm of learning, b) using L-M algorithm of learning

Table 2. Standard deviations and mean average values of identification validation using the BP and L-M algorithms of KF RNN learning

BP algorithm	L-M algorithm
ε = 1.2552	ε = 1.04374
σ = 0.0476	σ = 0.0193

Table 3. MSE% of 20 runs of the identification program using the KFRNN BP algorithm

No	1	2	3	4	5
MSE%	1.2010	1.3104	1.2025	1.3229	1.2494
No	6	7	8	9	10
MSE%	1.3011	1.3356	1.2218	1.2228	1.2439
No	11	12	13	14	15
MSE%	1.2041	1.2501	1.2038	1.3111	1.3399
No	16	17	18	19	20
MSE%	1.2154	1.2872	1.2186	1.2434	1.2189

Table 4. MSE% of 20 runs of the identification program using the KFRNN L-M algorithm

No	1	2	3	4	5
MSE%	1.0665	1.0162	1.0425	1.0340	1.0624
No	6	7	8	9	10
MSE%	1.0533	1.0320	1.0013	1.0575	1.0311
No	11	12	13	14	15
MSE%	1.0431	1.0554	1.0645	1.0517	1.0123
No	16	17	18	19	20
MSE%	1.0284	1.0467	1.0642	1.0287	1.0626

9 Conclusion

The paper proposed a new KFRNN model for systems identification and states estimation of nonlinear plants. The KFRNN is learned by the BP and by the second order recursive learning algorithm of Levenberg-Marquardt. The estimated states of the recurrent neural network model are used for direct adaptive trajectory tracking control systems design. The applicability of the proposed neural control system is confirmed by simulation results with a CSTR plant. The results showed good convergence of both L-M and BP learning algorithms. It was seen that the L-M algorithm of learning is more precise (see Table 2) but more complex than the BP one. The authors thank the support of the Mexican Government (CONACYT, SNI, SIP 20120746, COFAA and PIFI IPN).

References

1. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamic Systems Using Neural Networks. IEEE Trans. Neural Networks 1(1), 4–27 (1990)
2. Hunt, K.J., Sbarbaro, D., Zbikowski, R., Gawthrop, P.J.: Neural Networks for Control Systems - A Survey. Automatica 28(6), 1083–1112 (1992)
3. Pao, S.A., Phillips, S.M., Sobajic, D.J.: Neural Net Computing and Intelligent Control Systems. International Journal of Control 56, 263–289 (1992)

4. Baruch, I.S., Escalante, S., Mariaca-Gaspar, C.R.: Identification, Filtering and Control of Nonlinear Plants by Recurrent Neural Networks Using First and Second Order Algorithms of Learning. International Journal of Dynamics of Continuous, Discrete and Impulsive Systems, Series A: Mathematical Analysis, Special Issue on Advances in Neural Networks-Theory and Applications 14 (S!), Part 2, 512–521 (2007) ISSN 1201-3390
5. Baruch, I.S., Mariaca-Gaspar, C.R., Barrera-Cortes, J.: Recurrent Neural Network Identification and Adaptive Neural Control of Hydrocarbon Biodegradation Processes. In: Hu, X., Balasubramaniam, P. (eds.) Recurrent Neural Networks, pp. 61–88. I-Tech/ARS Press, Croatia (2008) ISBN 978-3-902613-28-8
6. Asirvadam, V.S., McLoone, S.F., Irwin, G.W.: Parallel and Separable Recursive Levenberg-Marquardt Training Algorithm. In: Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 129–138 (2002)
7. Ngia, L.S., Sjöberg, J., Viberg, M.: Adaptive Neural Nets Filter Using a Recursive Levenberg-Marquardt Search Direction. IEEE Signals, Systems and Computer 1, 697–701 (1998)
8. Ngia, L.S., Sjöberg, J.: Efficient Training of Neural Nets for Nonlinear Adaptive Filtering Using a Recursive Levenberg Marquardt Algorithm. IEEE Trans. on Signal Processing 48, 1915–1927 (2000)
9. Schmidt, L.D.: The Engineering of Chemical Reactions. Oxford University Press, New York (1998) ISBN 0-19-510588-5
10. Liu, S.-R., Yu, J.-S.: Robust Control Based on Neuro-Fuzzy Systems for a Continuous Stirred Tank Reactor. In: Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, pp. 1483–1488 (2002)
11. Zhang, T., Guay, M.: Adaptive Nonlinear Control of Continuously Stirred Tank Reactor Systems. In: Proceedings of the American Control Conference, Arlington, pp. 1274–1279 (2001)
12. Wan, E., Beaufays, F.: Diagrammatic Method for Deriving and Relating Temporal Neural Networks Algorithms. Neural Computations 8, 182–201 (1996)
13. Mariaca-Gaspar, C.R.: Topologies, Learning and Stability of Hybrid Neural Networks, Applied for Nonlinear Biotechnological Processes, Ph. D. Thesis, Dept. Automatic Control, CINVESTAV-IPN. Mexico (2009)
14. Nava, F.R., Baruch, I.S., Poznyak, A., Nenkova, B.: Stability Proofs of Adavanced Recurrent Neural Networks Topology and Learning, Comptes Rendus. Proceedings of the Bulgarian Academy of Sciences 57(1), 27–32 (2004)
15. Baruch, I.S., Barrera-Cortés, J., Hernández, L.A.: A Fed-Batch Fermentation Process Identification and Direct Adaptive Neural Control with Integral Term. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 764–773. Springer, Heidelberg (2004)
16. Baruch, I.S., Georgieva, P., Barrera-Cortes, J., Feyo de Azevedo, S.: Adaptive Recurrent Neural Network Control of Biological Wastewater Treatment. International Journal of Intelligent Systems, Special issue on Soft Computing for Modelling, Simulation and Control of Nonlinear Dynamical Systems 20(2), 173–194 (2005) ISSN 0884-8173
17. Baruch, I.S., Mariaca-Gaspar, C.R.: A Levenberg-Marquardt Learning Applied for Recurrent Neural Identification and Control of a Wastewater Treatment Bioprocess. International Journal of Intelligent Systems 24, 1094–1114 (2009)
18. Lightbody, G., Irwin, G.W.: Nonlinear Control Structures Based on Embedded Neural Systems Models. IEEE Trans. Neural Networks 8, 553–557 (1997)