RNN Identification and Control of Nonlinear Plants Using the Recursive Marquardt Algorithm

Ieroham Baruch, Saul Escalante M., and Carlos R. Mariaca-Gaspar
CINVESTAV-IPN, Ave. IPN No 2508,
A.P. 14-470 Mexico D.F., C.P. 07360 MEXICO
E-mail: {baruch, sescalante, cmariaca}@ctrl.cinvestav.mx
(Paper received on August 19, 2006, accepted on September 27, 2006)

Abstract. The paper proposed a new Recurrent Neural Network (RNN) model for systems identification and states estimation of nonlinear mechanical plants. The RNN model is learned by the second order recursive learning algoritm of Levenberg-Marquardt (L-M). The estimated states of the recurrent neural network model are used for direct adaptive trajectory tracking control systems design. The system contains also a noise rejection output filter. The applicability of the proposed neural control system is confirmed by simulation results with MIMO mechanical plant and compared with the results obtained by the Backpropagation learning algorithm. The results show a good convergence of both algorithms with priority of the L-M algorithm.

1. Introduction

The recent advances in understanding of the working principles of artificial neural networks (ANN) and the rapid growth of available computational resources led to the development of a wide number of ANN-based modeling, identification, prediction and control applications, [1]-[6], especially in the field of mechanical engineering and robotics. The ability of ANN to approximate complex non-linear relationships without prior knowledge of the model structure makes them a very attractive alternative to classical modeling and control techniques. Many of the applications currently reported are based on the classical Nonlinear Autoregressive Moving Average (NARMA) model, where a Feedforward Neural Network (FFNN) is implemented, [4], [5], [6]. However, the FFNN has in general a static structure, therefore it is adequate to approximate mainly static (nonlinear) relationships and their real-time applications for dynamical systems require the introduction of external time-delayed feedbacks, [5]. The application of the FFNN for modeling, identification and control of nonlinear dynamic plants caused some problems which could be summarized as follows: 1. The dynamic systems modeling usually is based on the NARMA model which need some information of input/output model orders, and input and output tap-delays ought to be used, [5], [6]; 2. The FFNN application for Multi-Input Multi-Output (MIMO) systems identification needs some relative order structural information; 3. The ANN model structure ought to correspond to the structure of the identified plant where four different input/output plant models are used, [5]; 4. The lack of universality in ANN architectures caused some difficulties in its learning and a Backpropagation through time

© H. Sossa and R. Barrón (Eds.)
Special Issue in Neural Networks and Associative Memories
Research in Computing Science 21, 2006, pp. 151-160

learning algorithm needs to be used. [7]; 5. Most of NARMA-based ANN models are sequential in nature and introduced a relative plant-dependent time-delay; 6. Most of the ANN-based models are nonparametric ones. [5]. and so, not applicable for an indirect adaptive control systems design; 7. All this ANN-based models does not perform state and parameter estimation in the same time, [4]; 8. All this models are appropriate only for identification of nonlinear plants with smooth, single, odd, nonsingular nonlinearities.

Recurrent Neural Networks (RNN) possesses its internal time-delayed feedbacks, so they are promising alternative for system identification and control, particularly when the task is to model dynamical systems [2], [3], [4], [7]. Their main advantage is the reduced complexity of the network structure. However, the analysis of state of the art in the area of classical RNN-based modeling and control has also shown some of their inherent limitations as follows: 1. The RNN input vector consists of a number of past system inputs and outputs and there is not a systematic way to define the optimal number of past values [4] and usually, the method of trials and errors is performed; 2. The RNN model is naturally formulated as a discrete model with fixed sampling period, therefore, if the sampling period is changed, the network has to be trained again; 3. It is assumed that the plant order is known, which represents a quite strong modeling assumption in general, [5]. Driven by these limitations, a new Recurrent Trainable Neural Network (RTNN) topology and the recursive Backpropagation (BP) type learning algorithm in vector-matrix form was derived, [8], [9], [10], and its convergence was studied, [9]. But the recursive BP algorithm, applied for RTNN learning, is a gradient descent first order learning algorithm which not permits to augment the precision and to accelerate the learning. So, the aim of the paper is to apply for RTNN learning a second order algorithm like the Levenberg-Marquardt (L-M) algorithm, [11], [12], [13], it is. The RTNN and the L-M algorithm of its learning will be applied for identification and control of a mechanical MIMO plant, taken from [6].

2. Topology and Learning of the RTNN

RTNN Topology: A *Recurrent Trainable Neural Network* model and its learning algorithm of dynamic *Backpropagation-type*, together with the explanatory figures and stability proofs, are described in [9]. The RTNN topology, given in vector-matrix form is described by the following equations:

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

$$Z(k) = S[X(k)] \tag{2}$$

$$Y(k) = S[CZ(k)] \tag{3}$$

$$A = block - diag(A_{ii}); | A_{ii} | < 1 \tag{4}$$

Where: Y, X, and U are, respectively, output, state and input vectors with dimensions l, n, m; A is a (nxn)- state block-diagonal weight matrix; A_{ii} is an i-th diagonal block of A with (lxl) dimension. Equation (4) represents the local stability conditions, imposed on all blocks of A; B and C are (nxm) and (lxn)- input and output weight matrices; S is vector-valued sigmoid or hyperbolic tangent-activation function, [9]; k is a discrete-time variable. The stability of the RTNN model is assured by the activation functions and by the local stability condition (4).

BP Learning of the RTNN: The commonly used BP updating rule, [9], is given by:

$$W_{ij}(k+1) = W_{ij}(k) + \eta\Delta W_{ij}(k) + \alpha\Delta W_{ij}(k-1) \quad (5)$$

Where: W_{ij} is a general weight, denoting each weight matrix element (C_{ij}, A_{ij}, B_{ij}) in the RTNN model, to be updated; ΔW_{ij} (ΔC_{ij}, ΔJ_{ij}, ΔB_{ij}), is the weight correction of W_{ij}; while; η and α are learning rate parameters. The weight updates are computed by the following equations:

$$\Delta C_{ij}(k) = [T_j(k)\text{-}Y_j(k)]S_j'[Y_j(k)]Z_i(k) \quad (6)$$

$$\Delta A_{ij}(k) = RX_i(k-1) \quad (7)$$

$$R = C_i(k)[T(k)\text{-}Y(k)]S_j'[Z_i(k)] \quad (8)$$

$$\Delta B_{ij}(k) = RU_i(k) \quad (9)$$

Where: ΔA_{ij}, ΔB_{ij}, ΔC_{ij} are weight corrections of the weights A_{ij}, B_{ij}, C_{ij}, respectively; (T-Y) is an error vector of the output RTNN layer, where T is a desired target vector and Y is a RTNN output vector, both with dimensions l; X_i is an i-th element of the state vector; R is an auxiliary variable; S_j' is derivative of the activation function. Stability proof of this learning algorithm is given in [9]. The described above RTNN is applied for identification and adaptive control of a nonlinear MIMO plant.

Recursive Levenberg-Marquardt Algorithm for RTNN Learning: The recursive L-M algorithm of learning, [11], [12], [13], is given by the following equations:

$$W(k+1) = W(k) + P(k)\nabla Y[W(k)]e[W(k)] \quad (10)$$

$$Y[W(k)] = g[W(k), U(k)] \quad (11)$$

$$E[W(k)] = e^2[W(k)] = \{g[W(k), U(k)] - Y_p(k)\}^2 \quad (12)$$

$$\nabla Y[W(k)] = \frac{\partial}{\partial W} g[W, U(k)]\Big|_{W=W(k)} \quad (13)$$

The Jacobean matrix elements for the RTNN topology are as follows:

$$\nabla Y_C[C_{ij}(k)] = S_j'(Y_j(k))Z_i(k) \quad (14)$$

$$\nabla Y_A[A_{ij}(k)] = R_1 X_i(k-1) \quad (15)$$

$$\nabla Y_B[B_{ij}(k)] = R_1 U_i(k) \quad (16)$$

$$R_1 = C_i(k)S_j'(Z_i(k)) \quad (17)$$

So the Jacobean matrix could be formed as:

$$\nabla Y[W(k)] = [\nabla Y_C(C_{ij}(k)), \nabla Y_A(A_{ij}(k)), \nabla Y_B(B_{ij}(k))] \tag{18}$$

The P(k), S(.), and Ω(.) matrices are given as follows:

$$P(k) = \frac{1}{\alpha(k)}\{P(k-1) - P(k-1)\Omega[W(k)]S^{-1}[W(k)]\Omega^T[W(k)]P(k-1)\} \tag{19}$$

$$S[W(k)] = \alpha(k)\Lambda(k) + \Omega^T[W(k)]P(k-1)\Omega[W(k)] \tag{20}$$

$$\Omega^T[W(k)] = \begin{bmatrix} & & \nabla Y^T[W(k)] & & \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}; \Lambda(k)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix}; \tag{21}$$

$$0.97 \leq \alpha(k) \leq 1$$

The next part compares the simulation results obtained with both BP and L-M learning algorithms for MIMO nonlinear plant.

3. Direct Adaptive Neural Control System Structure

The block-diagram of the control system is given on Figure 2. It contains a recurrent neural identifier RTNN-1, two neural controllers (feedback and feedforward) RTNN-2, RTNN-3, and a low pass noise rejection filter. In the direct adaptive neural control, the weight parameters of the feedback and feedforward controllers are learned so to minimize the cost function which is the reference tracking quadratic instantaneous error of the plant output.

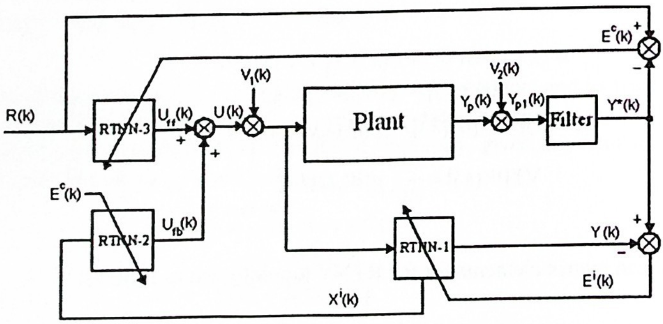

Fig. 1. Block - diagram containing neural identifier, feedback and feedforward neural controllers, and plant output noise rejection filter.

The structure of the closed loop system contains a neural identifier issuing an estimated state vector to the feedback neural plant dynamics compensator. The control feedback signal is added to the signal of the feedforward neural controller. The feedforward controller represented in fact an inverse model of the feedback closed loop system and depends on the reference signal. The system is completed by a low-pass filter which is aimed to reject the plant and measurement noises. So, the RTNN-1 identified the combined dynamics of the plant and the filter, and estimates the stated of this complex dynamics. The plant and the filter (this also includes the input and output noises) are described by the following state-space nonlinear models:

$$X_p(k+1) = F[X_p(k), U(K), V_1(k)] \tag{22}$$

$$Y_p(k) = \varphi(X_p(k)) \tag{23}$$

$$Y_{p1}(k) = Y_p(k) + V_2 \tag{24}$$

$$Y^*(k+1) = A^*Y^*(k) + B^*Y_{p1}(k) = A^*Y^*(k) + B^*Y_p(k) + B^*V_2 \tag{25}$$

Here the input, output and state dimensions of the plant are m, l, n_p. The filter dynamics is completely decupled, so the state matrix is diagonal (lxl) one. The plant equations (22) and (23) could be linearized and written in the same state-space form as:

$$X_p(k+1) = A_pX_p(k) + B_pU(k) + B_pV_1(k) \tag{26}$$

$$Y_p(k) = C_pX_p(k) \tag{27}$$

The linearized identification RTNN-1 could be also described by a state space model:

$$X^i(k+1) = A^iX^i(k) + B^iU(k) \tag{28}$$

$$Y^i(k) = C^iX^i(k) \tag{29}$$

Here the input, output and state dimensions of the RTNN-1 are m, l, n_i. The feedback neural RTNN-2 controller has a similar linearized state space representation, which input is the estimated systems state, issued by the RTNN-1:

$$X^c{}_{fb}(k+1) = A^c{}_{fb}X^c{}_{fb} + B^c{}_{fb}X^i(k) \tag{30}$$

$$U_{fb}(k) = -C^c{}_{fb}X^c{}_{fb}(k) \tag{31}$$

Here the input, output and state dimensions of the RTNN-2 are n_i, m, n_{fb}. The feedforward neural RTNN-3 controller could be described in the same manner as:

$$X^c{}_{ff}(k+1) = A^c{}_{ff}X^c{}_{ff} + B^c{}_{ff}R(k) \tag{32}$$

$$U_{ff}(k) = C^c{}_{ff}X^c{}_{ff}(k) \tag{33}$$

Here the input, output and state dimensions of the RTNN-3 are l, m, n_{ff}.

Let us to write the following z-transfer functional representations of the given up state-space equations for the plant, filter, feedback and feedforward controllers:

$$W^p(z) = C^p(zI - A_p)^{-1} B_p; W^c(k) = C^c(zI - A^c)^{-1} B^c; P^i(z) = (zI - A^i)^{-1} B^i \tag{34}$$
$$Q_1(z) = C^c{}_{fb}(zI - A^c{}_{fb})^{-1} B_{fb}; Q_2(z) = C^c{}_{ff}(zI - A^c{}_{ff})^{-1} B_{ff} \tag{35}$$

The control systems z-transfer functions (34), (35) are connected by the following equations, given in z-operational form:

$$Y^c(z) = W^c(z)Y_p + W^c(z)V_2; X^i(z) = P^i(z)U(z) \tag{36}$$
$$U_{fb}(z) = -Q_1(z)X^i(z); U_{ff}(z) = Q_2(z)R(z) \tag{37}$$
$$Y_p(z) = W_p(z)U(z) + W_pV_1(z); U(z) = U_{ff}(z) + U_{fb}(z) \tag{38}$$

Effectuating some substitutions and mathematical manipulations we could obtain the following statement for the systems control variable:

$$U(z) = [I + Q_1(z)P^i(z)]^{-1} Q_2(z)R(z) \tag{39}$$

The substitution of the control into the plant equation yields:

$$Y_p(z) = W_p(z)[I + Q_1(z)P^i(z)]^{-1} Q_2(z)R(z) + W_p(z)V_1(z) \tag{40}$$

The substitution of the plant equation into the systems output equation finally gives:

$$Y^c(z) = W^c(z)W_p(z)[I + Q_1(z)P^i(z)]^{-1}Q_2(z)R(z) + V_3(z) \tag{41}$$

Where $V_3(.)$ is a generalized noise term, given as:

$$V_3(z) = W^c(z)[W_p(z)V_1(z) + V_2(z)] \tag{42}$$

The RTNN topology is controllable and observable, [9], and the L-M algorithm of learning are convergent, so the identification and control errors tend to zero ($E^i(k) = Y^c(k) - Y^i(k) \to 0; \quad k \to \infty; E^c(k) = R(k) - Y^c(k) \to 0; \quad k \to \infty$) which means that each transfer functions given by equations (34), (35) is stable with minimum phase. From (41) it is seen that the dynamics of the stable low pass filter is independent from the dynamics of the plant and it does not affects the stability of the closed-loop system. The closed-loop system is stable and the RTNN-2 controller compensates the combined "plant plus filter" dynamics. The RTNN-3 feedforward controller dynamics is an inverse dynamics of the closed-loop systems one, which assure a precise reference tracking in spite of the presence of process and measurement noises.

4. Simulation Results

Let us consider the MIMO mechanical plant governed by the following state-space discrete-time nonlinear dynamics equations, taken from [6]:

$$X_1(k+1)=0.9X_1(k)\sin[X_2(k)]+\left[2+1.5\frac{X_1(k)U_1(k)}{1+X^2{}_1U^2{}_1(k)}\right]U_1+\left[X_1(k)+\frac{2X_1(k)}{1+X^2{}_1(k)}\right]U_2 \quad (43)$$

$$X_2(k+1)=X_3(k)\{1+\sin[2X_3(k)]\}+\frac{X_3(k)}{1+X^2{}_3(k)} \quad (44)$$

$$X_3(k+1)=\{3+\sin[2X_1(k)]\}U_2(k) \quad (45)$$

$$Y_1(k)=X_1(k);Y_2(k)=X_2(k) \quad (46)$$

The input, output and state dimensions of the plant are 2, 2, and 3. The reference signals R_1, R_2 of the control system are chosen as:

$$R_1(k)=0.007\sin(\pi k/10)+0.0075\sin(\pi k/25)+0.006\sin(\pi k/50) \quad (47)$$

$$R_2(k)=0.0075\sin(\pi k/40)+0.007\sin(\pi k/60)+0.006\sin(\pi k/80) \quad (48)$$

The control systems structure was applied to the given up plant and the results of its functioning are given on Fig. 2, Fig.3, Fig. 4 for BP and L-M learning applied for systems with 10% process and measurement noises, with and without noise filter. The results obtained for control MSE% are summarized in Table 1.

Table 1. The MSE% of control given for BP and L-M algorithms of learning for control systems with and without noise filter

Learning / noise filtering	MSE1 (%)	MSE2 (%)
BP with filter	0.91	0.72
BP without filter	1.28	2.19
L-M with filter	0.66	0.64
L-M without filter	1.03	1.41

The results show the good convergence of both learning algorithms. The noise terms augmented the MSE% of control in both cases for systems without noise filters. The L-M algorithm of learning is more precise but more complex that the BP one.

5. Conclusions

The paper proposed a new RTNN model for systems identification and states estimation of nonlinear mechanical plants. The RTNN is learned 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 system contains also a noise rejection output filter, which dynamics is separated from the dynamics of the control system. The applicability of the proposed neural control system is confirmed by simulation results with a MIMO mechanical plant and compared with the results obtained by the BP learning algorithm. The results summarized in Table 1 show good convergence of both L-M and BP learning algorithms. The presence of noise terms augmented the MSE% of control in both cases of systems without noise filters. The L-M algorithm of learning is more precise but more complex then the BP one.

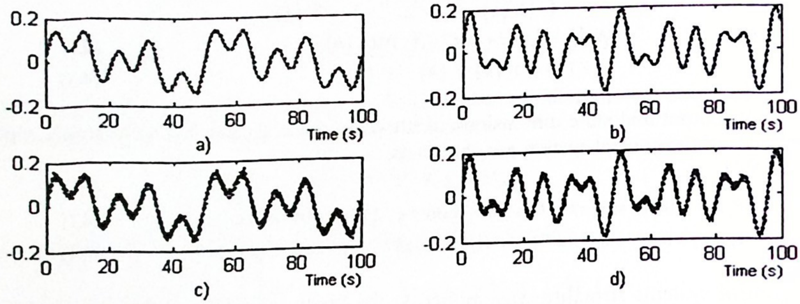

Fig. 2. Comparative results of a direct adaptive trajectory tracking control with BP learning, with (a,b) and without (c,d) noise filtering; a) Comparison between the first plant output Yp1(doted line, d-l) and the reference signal R1 (continuous line, c-l) for the case of system with noise filtering; b) Comparison between the second plant output Yp2 (d-l) and the reference signal R2 (c-l) for the case of system with noise filtering; c) Comparison between the first plant output Yp1(d-l) and the reference signal R1 (c-l) for the case of system without noise filtering; d) Comparison between the second plant output Yp2 (d-l) and the reference signal R2 (c-l) for the case of system without noise filtering.

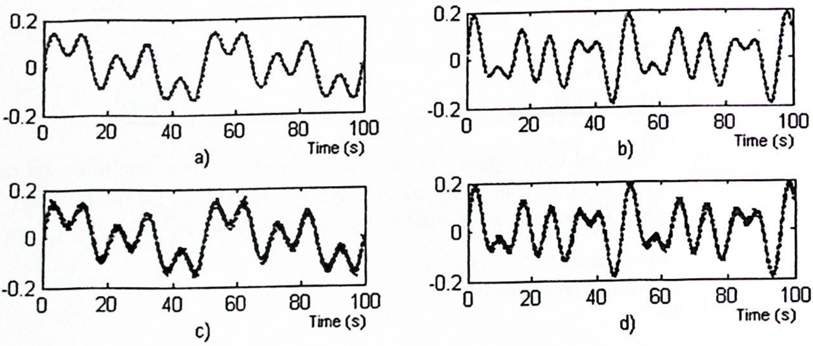

Fig. 3. Comparative results of a direct adaptive trajectory tracking control with L-M learning with (a,b) and without (c,d) noise filtering; a) Comparison between the first plant output Yp1(d-l) and the reference signal R1(c-l) for the case of system with noise filtering; b) Comparison

between the second plant output Yp2(d-l) and the reference signal R2(c-l) for the case of system with noise filtering; c) Comparison between the first plant output Yp1(d-l) and the reference signal R1(c-l) for the case of system without noise filtering; d) Comparison between the second plant output Yp2(d-l) and the reference signal R2(c-l) for the case of system without noise filtering.

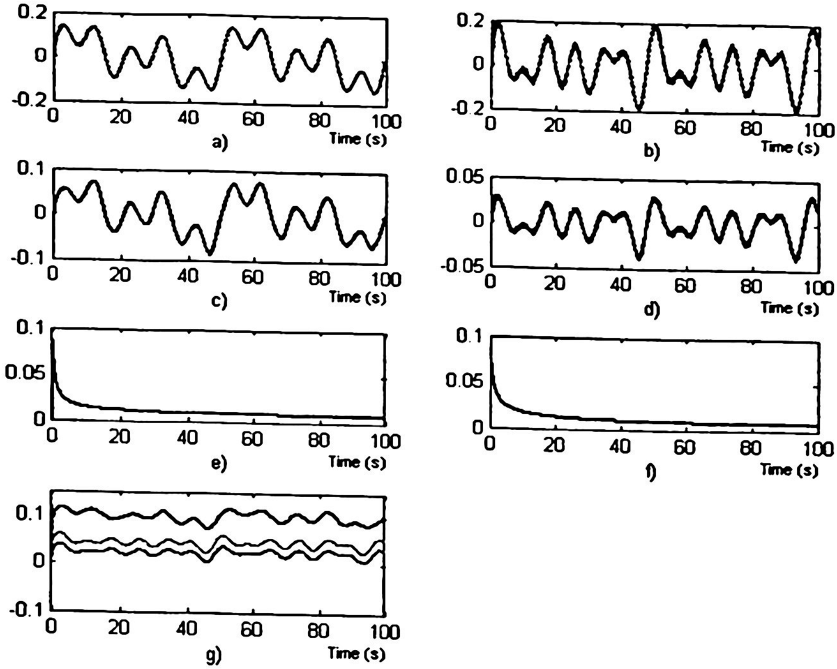

Fig. 4. Some additional results of a direct adaptive trajectory tracking control with L-M learning with noise filtering; a) Comparison between the first plant output Yp1 (c-l) and the first output Yi1 (d-l) of the identification RTNN-1; b) Comparison between the second plant output Yp2 (c-l) and the second output Yi2(d-l) of the identification RTNN-1; c) First control signal U1; d) Second control signal U2; e) Mean squared error of control (MSE1%) for the first controlled output; f) Mean squared error of control (MSE2%) for the second controlled output; g) Systems state variables, estimated by RTNN-1.

Acknowledgements

We want to thank CONACYT-MEXICO for the scholarship given to Saul Fernando Escalante Magana, MS student and Carlos Roman Mariaca Gaspar, PhD student, both students at the CINVESTAV-IPN, MEXICO.

References

[1] Miller III, W.T., Sutton, R.S., Werbos P.J. (1992). Neural Networks for Control, MIT Press, London.

[2] Chen, S., Billings, S. A. (1992). Neural Networks for Nonlinear Dynamics System Modeling and Identification. International Journal of Control, **56**:319-346.

[3] Pao, S.A., Phillips, S.M., Sobajic, D. J. (1992). Neural Net Computing and Intelligent Control Systems. International Journal of Control, **56**:263-289.

[4] Hunt, K.J., Sbarbaro, D., Zbikowski, R., Gawthrop, P.J. (1992). Neural Network for Control Systems - A Survey. Automatica, **28** (6):1083-1112.

[5] Narendra, K.S., Parthasarathy, K. (1990). Identification and Control of Dynamic Systems Using Neural Networks. IEEE Transactions on Neural Networks, 1(1):4-27.

[6] Narendra K., Mukhopadhyay S. (1994). Adaptive Control of Nonlinear Multivariable Systems using Neural Networks. Neural Networks, 737-752.

[7] L. Jin, and M. Gupta (1999). Stable Dynamic Backpropagation Learning in Recurrent Neural Networks. IEEE Transactions on Neural Networks, **10**:1321-1334.

[8] Baruch I., J.M., Flores, F., Thomas and R. Garrido. (2001). Adaptive Neural Control of Nonlinear Systems. In: Dorffner, G., Bischof, H., Hornik, K. (eds.): Artificial Neural Networks-ICANN 2001, Lecture Notes in Computer Science Vol. 2130. Springer, Berlin, pp. 930-936.

[9] Baruch, I. S., Flores, J. M., Nava, F., Ramirez, I. R., Nenkova, B. (2002). An Adavanced Neural Network Topology and Learning, Applied for Identification and Control of a D.C. Motor. In: Sgurev, V.S., Jotsov, V. (eds.): Proc. of the First Int. IEEE Symposium on Intelligent Systems, Varna, Bulgaria, pp. 289-295.

[10] Flores, J. M., Baruch, I., Garrido, R. (2001). Red Neuronal Recurrente para Identificación y Control de Sistemas No Lineales. Científica-ESIME-IPN, **5** (1):11-20.

[11] Asirvadam, V. S., McLoone, S. F. and Irwin G. W. (2002). Parallel and Separable Recursive Levenberg-Marquardt Training Algorithm. In: Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 129-138.

[12] Ngia, L. S., Sjöberg J., and Viberg, M. (1998). Adaptive Neural Nets Filter Using a Recursive Levenberg Marquardt Search Direction. IEEE Signals, Systems and Computer, **1**: 697-701.

[13] Ngia, L. S., Sjöberg, J. (2000). Efficient Training of Neural Nets for Nonlinear Adaptive Filtering Using a Recursive Levenberg Marquardt Algoritm. IEEE Trans. on Signal Processing, **48**:1915-1927.