15th Triennial World Congress, Barcelona, Spain

AN ADAPTIVE NEURAL CONTROL SYSTEM OF A DC MOTOR DRIVE.

Ieroham Baruch, Ignacio Ramon Ramirez Palacios, Jose Martin Flores and Ruben Garrido.

Automatic Control Department
CINVESTAV-IPN
P.O. Box 14-740. 07300 Mexico, D.F.
e-mail: baruch, iramirez, garrido@ctrl.cinvestav.mx

Abstract: An improved parallel recurrent neural network with canonical architecture, named Recurrent Trainable Neural Network (RTNN), and an error based back-propagation through time learning algorithm, are applied to a D.C. motor drive identification and control. The unknown nonlinear dynamics of the motor together with the load are identified by the RTNN. The trained RTNN identifier is combined with a desired reference model and a RTNN controller in a direct adaptive control scheme, so in order to achive a desired trajectory tracking of the motor speed and position. Finally, the applicability of the RTNN topology and learning is illustrated by experimental results. *Copyright © 2002 IFAC.*

Keywords: Neural networks, real-time systems, identification, adaptive control, reference signals, motor control.

1. INTRODUCTION

Recent advances in understanding of the working principles of artificial neural networks has given a tremendous boost to the application of these modeling tools for control of nonlinear systems, (Hunt et al., 1992). Most of the current applications rely on the classical NARMA approach; here a feedforward network is used to synthesise the nonlinear map (Chen and Billings, 1992). This approach is powerful in itself, but has some disadvantages: the network inputs are a number of past system inputs and outputs, so to find out the optimum number of past values a trial and error must be carried on; the model is naturally formulated in discrete time with fixed sampling period, so if the sampling period is changed the network must be trained again; the problem associated with the stability, convergence, and the rate of convergence of these networks are not clearly understood; the problem of discrete time nonlinear control is not fully understood and there is not a framework available for analysis; the necesary condition of the plant's order to be known.

Besides, to avoid these dificulties come's shine out the recurrent neural networks, but they still have some disadvantages: some Recurrent Neural Networks are not trainable in the whole, others are not in the feedback loop, (Pham and Yildrin, 1995); some of them are applied only to the SISO case, but not to the MIMO case, (Yip and Pao, 1994); the problem of stability is not considered on almost the whole of them, especially in the training period.

Along with the developments in the area of neural networks, tremendous strides have been made in the area of nonlinear control analysis using diferent techniques like: algebraic and diferential-geometric ones (Isidori, 1989). These tools provided a natural basis for the analysis of dynamic neural controllers. It is here that the recurrent neural networks come into their own.

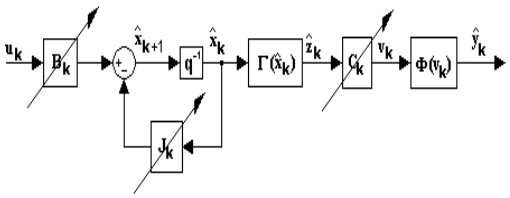

Fig. 1. Block diagram of RTNN.

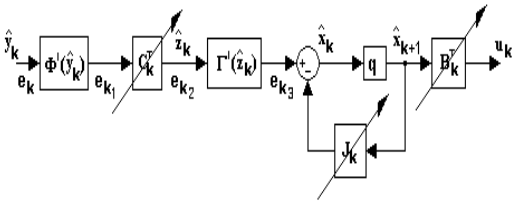

Fig. 2. Block diagram of RTNN adjoint network.

The control strategies that can be evolved using differential geometric techniques, involve the plant linearisation, using what is known as linearising state feedback. In this approach, the key principle is the cancellation of nonlinear terms, yielding a linear plant, which can then be controlled using standard techniques and that's why it's called globally linearising control, (Kravaris and Chung, 1987). The objective of this paper is to propose a recurrent neural network, to derive a vector matricial error based backpropagation-like learning algorithm and to show it's applicability in real time identification and adaptive tracking control of a D.C. motor drive's velocity and position.

The paper contents is as follows: in the first part we have the complete description of the Recurrent Trainable Neural Network (RTNN), and its training algorithm; the second part is based on the identification and control schemes description of a D.C. Motor drive, and finally - experimental results, are given.

2. RTNN TOPOLOGY AND LEARNING

The RTNN topology is given by the following equations, (Baruch, et al., 1996):

$$\hat{x}_{k+1} = J_k \hat{x}_k + B_k u_k \tag{1}$$

$$\hat{z}_k = \Gamma(\hat{x}_k) \tag{2}$$

$$\hat{y}_k = \Phi(C_k \hat{z}_k) \tag{3}$$

Where: $(\hat{y}_k, \hat{x}_k, u_k)$ are output, state and input variables with dimensions l, n, m, respectively; J_k is a block diagonal weight matrix of the hidden layer feedback; B_k and C_k are input and output weight matrices with dimensions (nxm) and (lxm), respectively; $\Gamma(.), \Phi(.)$ are vector-valued activation functions, with functional elements like: saturation, sigmoid or hyperbolic tangent. The eigenvalues of the RTNN model ought to be placed in the unit circle, so some restrictions on the weight matrix J_k =*block-diag*(J_{ki}), are imposed during the learning $(|J_{kii}| < 1)$.

The RTNN topology is given on Figure 1. Its adjoint network is given on Figure 2. The proposed RTNN has a linear time varying structure properties like: controllability, observability, reachability, which are proved in (Sontag and Albertini, 1994), (Sontag and Susmann, 1997). These properties of the RTNN structure signify that starting from the block-diagonal matrix structure of J_k, we can find a correspondence in the block structure of the matrices B_k and C_k, that's show us how to find out the ability of RTNN's learning. The main advantage of this discrete RTNN (which is really a Jordan Canonical RNN model), is of being an universal hybrid neural network model with one or two feedforward layers, and one recurrent hidden layer, where the weigth matrix J_k is a block-diagonal one. So, the RTNN posses a minimal number of learning parameters, and the performance of the RTNN is fully parallel. The described RTNN arquitecture could be used as an one step ahead state predictor/estimator and systems identificator. Another property of the RTNN model is that this model is globally nonlinear, but locally linear. That is why the matrices J_k, B_k, C_k, generated by learning , could be used to design a controller law. Furthermore, the RTNN model is robust, due to the dynamic weigth adaptation law, based on the sensitivity model of the RTNN, (Wan and Beaufays, 1996).

The general RTNN Backpropagation Throug Time (BPTT) learning algorithm, written in a matricial form, is given by the following equation:

$$W_{k+1} = W_k + \eta \Delta W_k + \alpha \Delta W_{k-1} \tag{4}$$

where: W_k is the weigth matrix, being modified (J_k, B_k, C_k); ΔW_k is the weigth matrix correction $(\Delta J_k, \Delta B_k, \Delta C_k)$; η is a learning rate parameter $(|\eta| < 1)$, and α is a momentum term learning parameter $(|\alpha| < 1)$. The momentum term of this learning algorithm is used when some error oscillations occured.

The output layer's weigth matrix elements update, for the discrete time model of the RTNN, is realized by the following equation, (Baruch, et al., 1996):

$$\Delta C_{k_{ij}} = e_{k_i} \Phi'_{k_i}(\hat{y}_{k_i})\ \hat{z}_{k_j} \tag{5}$$

where: $\Delta C_{k_{ij}}$ is the weight update of an ij-th element of the learned matrix C_k; e_{k_i} is the i-th element of the output error vector; $\Phi'_{k_i}(\hat{y}_i)$ is the i-th element of the first derivative of the activation function vector, which is expressed as a function of the output vector $\hat{y}_k$; $\hat{z}_{k_j}$ is the j-th element of the output vector of the hidden layer. The vector matricial form of this equation

could be obtained following the block-diagram of the RTNN adjoint network and introducing some other auxiliar matrices and vectors, Fig. 2.

Let us define the (lxl) diagonal Jacobean matrix: $\Phi'_k(\hat{y}_k) = diag[\Phi'_{k_i}(y_{k_i})]$; $i = 1, ..., l$. So, the $(lx1)$ transformed output error vector is given by:

$$e_{k_1} = \Phi'_k(\hat{y}_k)e_k \tag{6}$$

and the output weight update matrix will be the following:

$$\Delta C_k = e_{k_1}\ \hat{z}_k^T \tag{7}$$

The weight update of the hidden layer's weigth matrix elements, for the discrete time model of the RTNN, is realized by the following equation, (Baruch, et al., 1996):

$$\Delta J_{k_{ij}} = C_{k_i} e_{k_1} \Gamma'_{k_i}(\hat{z}_{k_i})\ \hat{x}_{k_j} \tag{8}$$

$$\Delta B_{k_{ij}} = C_{k_i} e_{k_1} \Gamma'_{k_i}(\hat{z}_{k_i})\ u_{k_j} \tag{9}$$

where: $\Delta J_{k_{ij}}$ is the weight update of an ij-th element of the general weight matrix J_k under learning (here for sake of generality, it is assumed that J_k is a full (nxn) weight matrix); C_{k_i} is the i-th row vector, taken from the transposed matrix C^T; e_k is the output error vector; $\Gamma'_{k_i}(z_{k_i})$ is the i-th element of the first derivative of the activation function vector, which is expressed as a function of the output vector of the hidden layer z_k; x_{k_j} is the j-th element of the state vector x_k; $\Delta B_{k_{ij}}$ is the weight update of an ij-th element of the learned matrix B_k; u_{k_j} is the j-th element of the input vector u_k. The vector matricial form of this two equations could be obtained introducing some other auxiliar matrices and vectors.

Let us define the $(nx1)$ error vector which appear in the output of the hidden layer as:

$$e_{k_2} = C^T{}_k\ e_{k_1} \tag{10}$$

Let us also define the (nxn) diagonal Jacobean matrix: $\Gamma'_k(\hat{z}_k) = diag[\Gamma'_{k_i}(\hat{z}_{k_i})]$; $i = 1, ..., n$. So, the transformed hidden layer output error vector is given by:

$$e_{k_3} = \Gamma'_k(\hat{z}_k)e_{k_2} \tag{11}$$

and the weight matrix correction of the general weight matrix J_k will be the following:

$$\Delta J_k = e_{k_3}\ \hat{x}_k^T \tag{12}$$

If the matrix J_k is a diagonal matrix , $J_k = diag(J_{k_i})$; $i = 1, ..., n$; $|J_{k_{ii}}| < 1$ (stability condition imposed) , and the diagonal of this matrix is a $(nx1)$ vector, denoted by vJ_k, then the weight update of this vector is the following:

$$\Delta vJ_k = e_{k_3} \bigotimes x_k \tag{13}$$

The update of the (nxm) weight input matrix B, is the following:

$$\Delta B_k = e_{k_3}\ u_k^T \tag{14}$$

The next paragraph gives an identification and control scheme, applied for a model reference adaptive control of a D.C. motor drive velocity and position.

3. REAL-TIME RTNN IDENTIFICATION AND CONTROL OF A DC MOTOR DRIVE. EXPERIMENTAL RESULTS

In this section the effectiveness of the Adaptive Neural Control scheme is illustrated by a real time DC motor system identification and control (in two control experiments, separatelly for speed and position control), using three RTNNs, one for the identification and the others two for the control. The first problem here is to identify the discrete-time nonlinear plant by means of a RTNN. Let us assume that the equations of the DC motor drive dynamics, separatelly for systems speed and position, are described by the following equations, (Weerasooriya and El-Sharkawi, 1991):

$$\omega_{k+1} = f[\omega_k, ..., \omega_{k-m}] + V_k \tag{15}$$

$$\theta_{k+1} = g[\theta_k, ..., \theta_{k-m}] + V_k \tag{16}$$

where: m is the order of the system; ω_k, θ_k are the DC motor angular velocity and position in time instant k; V_k is the terminal voltage.

The RTNN I is applied as a systems identifier, (Weerasooriya and El-Sharkawi, 1991) so to estimate, separatelly in two control experiments, the values of the unknown nonlinear functions $f[.], g[.]$.

The motor speed and position are estimated separatelly by the RTNN I, following the equations:

$$\hat{\omega}_k = \Phi[\hat{x}_k];\quad \hat{x}_{k+1} = \Gamma[\hat{x}_k, V_k] \tag{17}$$

$$\hat{\theta}_k = \Phi[\hat{x}_k];\quad \hat{x}_{k+1} = \Gamma[\hat{x}_k, V_k] \tag{18}$$

where: $\Gamma[.], \Phi[.]$ denotes the nonlinear RTNN functions. The main objetive of the systems identification, using RTNN is to obtain the plant states (separatelly for speed and position in two control experiments), based on the weight update BPTT learning algorithm, minimizing the mean squared output error between the RTNN I output and the output of the nonlinear plant. The neural control input V_k to the motor at the k^{th} time step can be designed as:

$$V_k = N_1[\hat{x}_k] + N_2[r_k] \tag{19}$$

where $N_1[.]$, $N_2[.]$ are generated by RTNN feedback and feedforward controllers. The adaptive

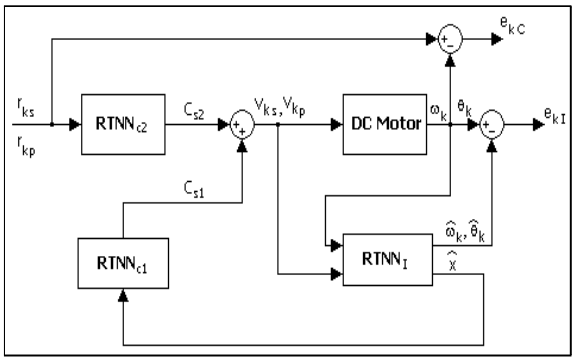

Fig. 3. Block- diagram of the adaptive control system.

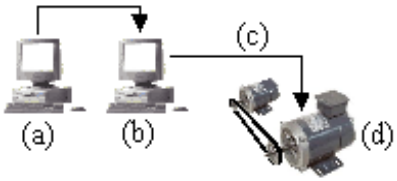

Fig. 4. Real-time system configuration for DC-motor system identification and control. (a) MatLab-Simulink and WinCon Server. (b) WinCon Client and MultiQ. (c) Encoder. (d) DC-motor.

control scheme is shown on the Fig 3. Here the system controller is split in a feedback neural controller $N_1[.]$, which uses the estimated states as inputs, so to create a classic stabilization feedback, and a feedforward neural controller $N_2[.]$, which is an adaptive system's inverse plant model, (Hunt et al., 1992).The identification error e_{kI} is used to train the $RTNN_I$, and the control error e_{kC} is used to train the $RTNN_{c1}$ and the $RTNN_{c2}$.

The configuration of the experimental DC motor platform containing the control and the measurement components is shown on the Fig. 4. A 24 volts, 8 amperes DC-motor, driven by a power amplifier and connected by a data acquisition control board (Multi-Q®), with the PC, has been used. The RTNNs are programmed in MatLab®-Simulink® and WinCon® , which is a real-time Windows 95® application that runs Simulink® generated code using the real-time Workshop® to achieve digital real-time control on a PC.

3.1 *First experiment. Adaptive control of a DC motor speed*

In the first experiment, the objective of the designed adaptive neural control system is to drive the DC-motor, so that its speed ω_k follows a prescribed trajectory r_k. This is done by letting the DC motor to follow the selected signal reference r_k, given by the following equation:

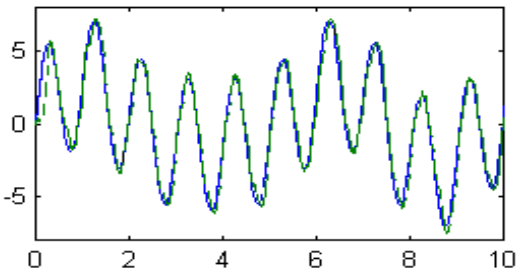

Fig. 5. Real-time adaptive neural control of a DC-motor speed. Comparison between the reference signal (—) and the DC-motor output signal (- - -) in the first ten seconds of the control experiment.

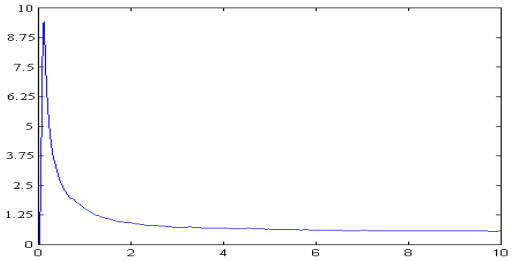

Fig. 6. The mean square error of speed control (MSE%).

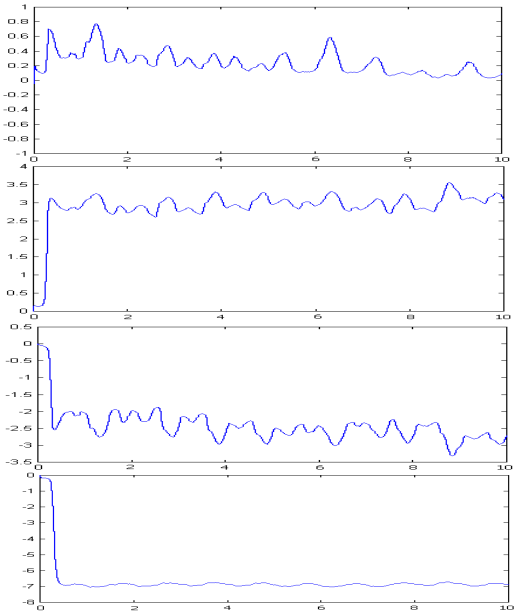

Fig. 7. The states of the plant, estimated by the identification RTNN. a) first state; b) second state; c) third state; d) fourth state.

$$r_k = 2.25sin(2\pi k/5) + 0.75sin(2\pi k/3) + 4.5sin(2\pi k) \quad (20)$$

The results of the on-line DC motor drive speed control, are given on Fig. 5, 6, 7 and Fig. 8.

For this experiment, the identification RTNN I has the topology 1,4,1 and the NN controllers, RTNN$_{C1}$and RTNN$_{C2}$, have the topologies 4,4,1 and 1,2,1, respectively. The learning parameters

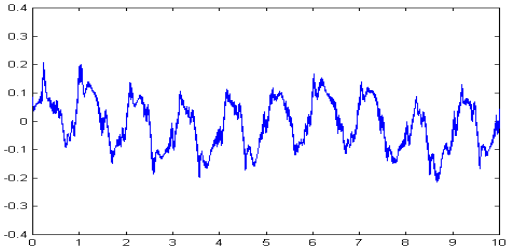

Fig. 8. The control signal .

for the speed control experiment are given on Table 1.

Table 1: Learning parameters in speed control

NN	η	α
RTNN$_I$	0.01	0.01
RTNN$_{C1}$	0.01	0.01
RTNN$_{C2}$	0.01	0.01

The sampling period for all signals of this experiment is: **To=0.01.**The figures 5, 6, 7, 8, show the performance of the adaptive neural control speed system to track the reference set point signal. It can be seen that the motor speed follows the reference signal trajectory, and the MSE% of control rapidly decreases and finally reach the value less than 1.25%.

3.2 *Second experiment. Adaptive control of a DC motor position*

In the second experiment, the objective of the adaptive neural control system designed, is to drive the DC-motor, so that its position, θ_k, follows a desired position, r_k. This is done by letting the DC motor to follow a selected reference signal position r_k, computed by the following equation, (Spong and Vidyasagar, 1989):

$$r_k = \begin{cases} q_0 + 3(q_1 - q_0)k^2/tf^2 - 2(q_1 - q_0)k^3/tf^3 & k < tf \\ q_1 & tf \le k \le 2tf \\ q_0 - 3(q_0 - q_1)k^2/tf^2 + 2(q_0 - q_1)k^3/tf^3 & 2tf < k < 3tf \end{cases} \quad (21)$$

Where the parameters q_0, q_1, t_f are: $q_0 = -\pi/2$; $q_1 = \pi/2$; and $t_f = 4$. This signal has been selected, because we want to test the motor with a smooth reference signal.

The results of the on-line DC motor drive position control, are given in Fig 9, 10, 11 and 12.

For this experiment, the identification RTNN I has the topology 1,4,1 and the NN controllers, RTNN$_{C1}$and RTNN$_{C2}$, have the topologies 4,4,1 and 1,2,1, respectively.

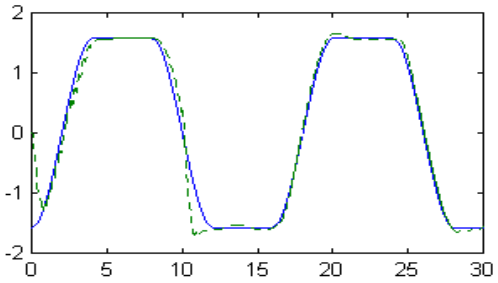

Fig. 9. Real-time adaptive neural control of a DC-motor position. Comparison between the reference signal (——) and the DC-motor output signal (- - -) in the first thirty seconds of the control experiment.

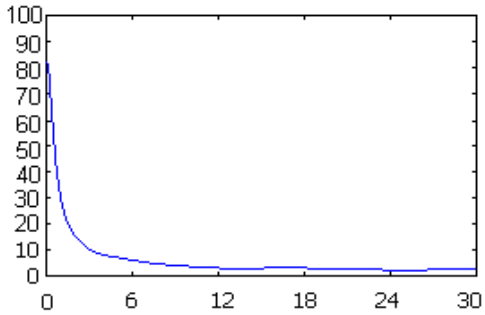

Fig. 10. The mean square error of position control (MSE%).

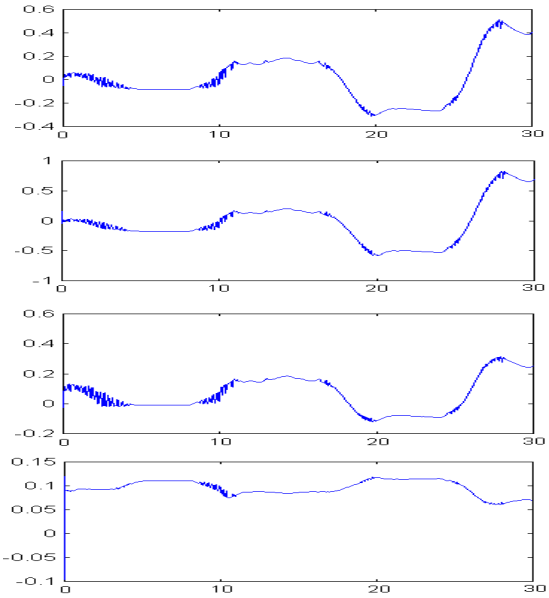

Fig. 11. The states of the plant, estimated by the identification RTNN. a) first state; b) second state; c) third state; d) fourth state.

The learning parameters for the position experiment are given in Table 2

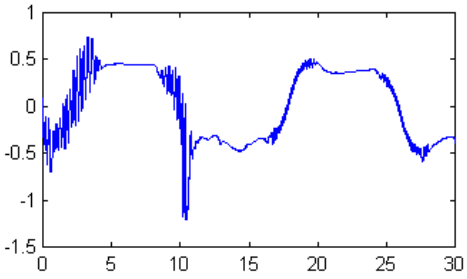

Fig. 12. The control signal.

Table 2: Learning parameters in position control

NN	η	α
RTNN$_I$	0.01	0.01
RTNN$_{C1}$	0.00001	0.0001
RTNN$_{C2}$	0.04	0.005

The sampling period for all signals of this experiment is: **To=0.01.** The experimental results, given on figures 9, 10, 11 and 12, show the performance of the adaptive neural control position system to track the reference set point signal. It can be seen that the motor position follows the reference signal trajectory, and the MSE% of control rapidly decreases and finally reach the value less than 3.5%.

4. CONCLUSIONS

In this paper, an improved parallel recurrent neural network, named Recurrent Trainable NN, with a Jordan canonical structure, and a new matricial learning algorithm, are proposed. The RTNN is robust with respect to the unmodeled dynamics and uncertainties because of its feedback topology and its Backpropagation Through Time learning algorithm, based on an adjoint sensitivity NN model. This property turns it as a powerfull representative of nonlinear maps. In the same way, this paper confirms the RTNN identification and control abilities by series of real-time experiments, performed with a DC-motor drive mechanical system. A real-time adaptive DC-motor velocity and position tracking control schemes, containing three RTNNs has been successfuly experimented. The experimental results show that the unknown nonlinear dynamics of a DC motor drive and its unknown load, has been sucessfully captured by the identification RTNN. The control experiment exhibit a good performance of the real time adaptive neural control system and fast convergence of all RTNNs.

5. REFERENCES

Baruch,I., I. Stoyanov and E. Gortcheva, (1996). Topology and Learning of a Class RNN. *ELEKTRIK*, **Vol. 4 Supplement**, 35-42.

Chen, S. and S.A. Billings, (1992). Neural Networks for Nonlinear Dynamics System Modelling and Identification. *International Journal of Control*. **No. 56**, 263-289.

Hunt, K.J., D. Sbarbaro, R. Zbikowski, and P.J. Gawthrop, (1992). Neural Networks for Control Systems - A Survey. *Automatica*, **vol. 28, No. 6**, 1083-1112.

Isidori, A. (1989). *Nonlinear Control Systems*. 3th Edition, 549, Springer-Verlag, Great Britain.

Kravaris, C. and C.B. Chung, (1987). Nonlinear State Feedback synthesis by global input/output Linearization. *AIChE Journal*, **No. 33**, 592-603.

Pham, D.T. and S. Yildirim, (1995). Robot Control using Jordan Neural Networks, *Proc. of the Internat. Conf. on Recent Advances in Mechatronics, Istanbul, Turkey, Aug. 14-16, 1995, (O. Kaynak, M. Ozkan, N. Bekiroglu, I. Tunay, Eds., Bogaziçi University Printhouse, Istanbul, Turkey, 1995)*, **Vol. II**, 888-893.

Sontag, E. and F. Albertini, (1994). State Observability in Recurrent Neural Networks, *System and Control Letters*, **No. 22**, 235-244.

Sontag, E.and H. Sussmann, (1997). Complete Controlability of Continuos Time Recurrent Neural Network, *System and Control Letters*, **No. 30**, 177-183.

Spong, W. M. and M.Vidyasagar, (1989). *Robot Dynamics and Control*, 336, John Wiley & Sons, United States.

Wan, E. and F. Beaufays, (1996). Diagrammatic method for deriving and Relating temporal Neural Networks Algorithms. *Neural Computations*. **Vol 8. No. 4**, 182-201.

Weerasooriya, S. and M.A. El-Sharkawi, (1991). Identification and Control of DC Motor Using Back-Propagation Neural Networks, *IEEE Transactions on Energy Conversion*, **Vol. 6, No. 4**, 663-669.

Yip, P.P.C and Y.H. Pao, (1994). A Recurrent Neural Net Approach to One-Step Ahead Control Problems. *IEEE Transactions on SMC*, **Vol.24, No.4**, 678-683.