

Identification and Adaptive Neural Control of Time-Delayed Multivariable Plant

Ieroham Baruch
Carlos-Roman Mariaca G.

Department of Automatic Control,
CINVESTAV-IPN, Av. IPN, No. 2508
México, DF.
MÉXICO.

Tel. 5061 3800 ext. 4229, Fax: 5747 7089,

email: baruch@ctrl.cinvestav.mx

Recibido el 15 de febrero de 2005; aceptado el 19 de enero de 2006.

1. Abstract

A direct adaptive neural control scheme with single and double I-term is proposed to be applied for multivariable plant. The control scheme contains two Recurrent Trainable Neural Network (RTNN) models. The first RTNN is a plants parameter identifier and state estimator. The second RTNN is a feed-back/feed-forward controller with I-terms. The good performance of the adaptive neural control with I-terms is confirmed by closed-loop systems analysis, and by simulation results, obtained with simple effect evaporator multivariable plant, corrupted by noise and affected by small unknown input time delay.

Key words: Recurrent Neural Networks, Back-propagation Learning, Systems Identification, State Estimation, Adaptive Control, Integral Terms, Simple Effect Evaporator, Discrete-Time System, Time-Delay System.

2. Resumen (Identificación y control neuronal adaptable de plantas multivariables con retardo)

Un esquema de control neuronal directo adaptable con uno o dos términos integrales son propuestos para ser aplicados con plantas multivariables. El esquema de control contiene dos modelos de Redes Neuronales Recurrentes Entrenables (RNRE). La primera RNRE es un identificador de los

parámetros y estimador de los estados de la planta. La segunda RNRE es un controlador «feed-back/feed-forward» con término integral. El buen desempeño del control neuronal adaptable con termino integral es confirmado con un análisis del comportamiento del sistema en laso cerrado y con resultados de simulación, obtenidos usando un modelo multivariable de un evaporador de simple efecto, perturbado por ruidos y afectado por pequeños retardos en sus entradas.

Palabras clave: redes neuronales recurrentes, aprendizaje *back-propagation*, identificación de sistemas, estimación de estados, control adaptable, término integrales, evaporador de efecto simple, sistemas con retraso por tiempo.

3. Introduction

Intelligent control using Neural Networks (NN) has been applied to various control problems [1]. It is known to be effective in many situations, especially when the controlled plant exhibits nonlinearity, and the plant parameters are unknown and time-varying, especially for mechanical systems. On the other hand, the unavoidable effects of identification and control errors due to model uncertainties, together with a slow load variation caused a steady-state offset that needs to be removed. In this case an integral action added to the control compensates the plant uncertainties and load effects, and help the system to track the reference signal. Within the context of the servomechanism problem, integral action is a fundamental technique in the control repertoire and the I-PD (or PID) controllers have been the most used controllers in the industry, because of their simple structure and robust performance in wide range of operating conditions [2]. Here the PD mode is used to speed up response, whereas the PI mode is applied to eliminate the steady state offset. The PID controller parameters are tuned taking into account the plant parameters like gain, time constant, and time delay, where they could fined a good values when the relationship between the time constant and the time delay is less than one. In last years, the classical PID scheme has been completed by auto-tuning devices like Neural Networks [3], and Fuzzy Systems [4], to adjust on-line its parameters. To resolve some specific control problems in mechanical systems, some extensions to the classical PID scheme have been added. So, for regulator tasks on mechanical systems that exhibit friction the PID-controller is combined with mass and friction feed-forward [4]. The state PD-controller

plus gravity compensation terms is widely used in robot manipulators control. However, this linear state feedback controllers could not compensate inertial and Coriolis forces and cannot render asymptotic stability for path tracking tasks. To overcome this [5], a nonlinear PID controller is proposed. The major disadvantage of these controllers is that they could be applied only for Single-Input-Single-Output and not for Multi-Input Multi-Output (MIMO) systems. Also in the case of high order systems, the PD action is not sufficient to assure systems stability. In all applications the PID control needs adjustment which could be done automatically using self tuning facilities. The use of RNN for systems identification and control could overcome all these problems. In [6], [7], Baruch *et al.* proposed a new RNN and a Back-propagation (BP) like algorithm of its learning, applied for identification and control of petrochemical and biotechnological plants. The applied direct adaptive neural control system [6] contains one RNN for identification and state estimation and two adaptive neural controllers (feed-back and feed-forward) which offer a good performance and flexibility. In [8] the stability of the proposed RNN and its learning is proved and this RNN has been applied for model reference adaptive control of a DC motor. In [9] the direct adaptive neural control scheme has been extended with one or two I-terms and applied for control of multivariable plant. In [10] a rather complicated prediction-based control of unstable plant with small delay time and perturbation terms has been designed. The aim of the proposed paper is to apply a RNN control scheme with one or two I-terms for direct adaptive neural control of an industrial multivariable plant model, taken from [11], like the simple effect evaporator is, and to study its behavior in some unknown unmodeled dynamics conditions represented by an offset and a small time-delay in the plants input.

4. Development

4.1 Recurrent Neural Model Description

In [6], [7], [8] a discrete-time model of RTNN, and the dynamic Back-propagation weight updating rule, are given. The RTNN model is described by the following equations:

$$X(k+1)=JX(k)+BU(k), S(k)=\theta [X(k)]; \quad (1)$$

$$Y(k)=\theta [C S(k)] \quad (2)$$

$$J=\text{block-diag}(J_i), |J_i| < 1 \quad (3)$$

Where: $X(k)$ is a N -state vector; $U(k)$ is a M -input vector; $Y(k)$ is a L - output vector; $S(k)$ is a N output vector of the hidden layer; $\theta(\cdot)$ is a vector-valued activation function (saturation, sigmoid or hyperbolic tangent) with appropriate dimension; J is a weight state-block-diagonal matrix, where the stability condition (3) is imposed on its blocks; B and C are weight

input and output matrices with appropriate dimensions and block structure, corresponding to the block structure of J . As it can be seen, the given RTNN model is a completely parallel parametric one, so it is useful for identification and control purposes. Parameters of that model are the weight matrices J , B , C and the state vector $X(k)$. The general BP learning algorithm is given in the form:

$$W_{ij}(k+1)=W_{ij}(k) + \eta \Delta W_{ij}(k) + \alpha \Delta W_{ij}(k-1) \quad (4)$$

Where: W_{ij} (C , J , B) is the ij -th weight element of each weight matrix (given in parenthesis) of the RTNN model to be updated; ΔW_{ij} is the weight correction of W_{ij} ; η , α are learning rate parameters. The updates ΔC_{ij} , ΔJ_{ij} , ΔB_{ij} of the model weights C_{ij} , J_{ij} , B_{ij} are given by:

$$\Delta C_{ij}(k) = [T_j(k)-Y_j(k)] \theta' [Y_j(k)] S_i(k) \quad (5)$$

$$\Delta B_{ij}(k) = R U_i(k) \quad (6)$$

$$\Delta J_{ij}(k) = R X_i(k-1) \quad (7)$$

$$R = C_i(k) [T(k)-Y(k)] \theta' [S_j(k)] \quad (8)$$

Where: $T(k)$ is a target vector with dimension L and $[T(k)-Y(k)]$ is an output error vector also with the same dimension; R is an auxiliary variable; $\theta'(\cdot)$ is the derivative of the activation function, which for the hyperbolic tangent one is given by: $\theta'(x)=1-x^2$.

In [9] the following Theorem of stability is proved:

Theorem of stability: Let the RTNN with Jordan Canonical Structure is given by equations (1)-(3) and the stable Bounded-Input-Bounded-Output (BIBO) nonlinear plant model is supposed to be:

$$X(k+1) = f[X(k), U(k)] \quad (9)$$

$$Y(k) = h[X(k)] \quad (10)$$

Where: $X(k)$, $U(k)$, and $Y(k)$ are plant state, input, and output vector variables with dimensions N_p , M , L , respectively (here $L=M$ is accepted); $h(\cdot)$, $f(\cdot)$ are smooth bounded nonlinear functions. Under the assumptions made, the application of the BP learning algorithm for $J(k)$, $B(k)$, $C(k)$ given in general matrix form described by (4) and the learning rates $\eta(k)$, $\alpha(k)$ normalized with respect to the error, and derived using the following Lyapunov function:

$$L(k) = \|J(k)\|^2 + \|B(k)\|^2 + \|C(k)\|^2 \quad (11)$$

Then the identification error is bounded, and:

$$\Delta L(k) \leq -\eta(k) |E(k)|^2 - \alpha(k) |E(k-1)|^2 - d; \quad (12)$$

$$E(k) = Y_p(k) - Y(k)$$

Where: $\Delta L(k) = L(k) - L(k-1)$; the learning parameters $\eta(k)$, $\alpha(k)$ are normalized and depends on the error with the following bounds: $0 \leq \|\eta(k)\| \leq 1$, $0 \leq \|\alpha(k)\| \leq 1$; d is bounded error perturbation term.

As it could be seen from equations (1), (2), the proposed canonical RTNN architecture is a two-layer hybrid one with one feed-forward output layer and one recurrent hidden layer. The main advantage of the proposed two-layer Jordan Canonical Form (JCF) RTNN architecture is that it has a minimum number of weights to learn. The application of this RNN model for systems identification of MIMO nonlinear plant does not require additional structural information about the relative systems order. The RTNN architecture is described in state-space vector-matrix form and could serve as a nonlinear dynamic identifier and one-step ahead state predictor/estimator.

4.2. Direct I-term Adaptive Neural Control of Time-Delayed Plant

Two control schemes are considered in this paper - with one and with two integrals in the control part. The block diagram of the first control scheme, containing one integral block is shown on Fig.1. This control scheme contains RNN-1 identifier, one feed-back / feed-forward RNN-2 controller, one I-term, an unknown time delay τ and a con-stant offset $O(k)$ in the input of the plant.

Let us suppose that the studied stable continuous-time nonlinear plant is linearized around an operation point and normalized. Let us also add some unmodeled dynamics represented by a small unknown time delay and a constant offset $O(\cdot)$ to its input. So the continuous state space plant model is given by:

$$X(t) = AX(t) + BU(t) + BO(t - \tau) \quad (13)$$

$$Y(t) = CX(t) \quad (14)$$

Let us to discretize the plants equations (13), (14), as it is:

$$\begin{aligned} X_p(k+1) &= A_p X_p(k) + B_p U(k-t_o) + B_p O(k-t_o) \\ Y_p(k) &= C_p X_p(k) \end{aligned}$$

Where: $A_p \in \mathfrak{R}^{(N_p \times N_p)}$; is a state matrix; $B_p \in \mathfrak{R}^{(N_p \times M)}$ is an input matrix; $C_p \in \mathfrak{R}^{(N_p \times L)}$ is an output matrix, t_o is a discrete time delay variable, and $L=M$ is supposed. This stable linearized plant is identified by a RTNN with topology, given by equations (1) to (3) which is learned by the stable BP learning algorithm, given by equations (4) to (8), where the identification error $E_i(k) = Y_p(k) - Y^i(k)$ tends to zero ($E_i \rightarrow 0, k \rightarrow \infty$). This

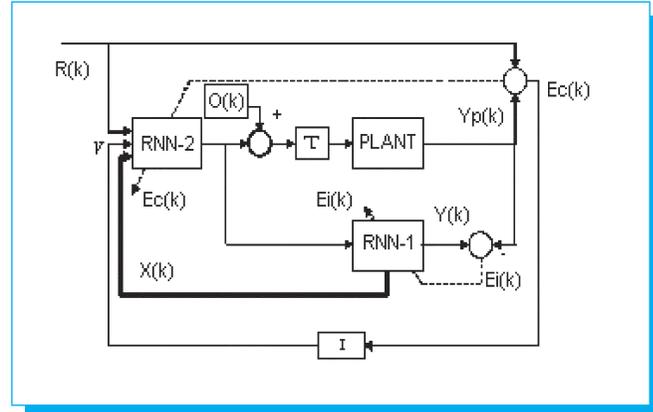


Fig. 1. Block-diagram of the direct adaptive neural control system containing one I-term

identification error could be considered acceptable if it reached a value less than 2%. Following the same way as in [8] we should linearize the RNN-1 model activation functions (see equations. (1) to (3)). When the learning error of identification reached small value, the following linear state-space RNN-1 model, could be obtained:

$$X^i(k+1) = J^i X^i(k) + B^i U(k) \quad (17)$$

$$Y^i(k) = C^i X^i(k) \quad (18)$$

Where: $X^i(k)$ is a N_i -dimensional state vector; $Y^i(k)$ is a L -dimensional output vector; $J^i \in \mathfrak{R}^{(N_i \times N_i)}$, $B^i \in \mathfrak{R}^{(N_i \times M)}$ and $C^i \in \mathfrak{R}^{(N_i \times L)}$ are constant matrices. Here we suppose $N_i \geq N_p$ and $L = M$ (equal input-output dimensions). It is proved that the identification RTNN learning BP algorithm is convergent, [9], and it is proved that the RTNN model is stable, controllable and observable, [9], [12], [13], so the identification error $E^i(k) = Y_p(k) - Y^i(k)$ tends to zero, and the identification RTNN output tends to the plant output ($Y^i(k) \rightarrow Y_p(k)$). Following the block scheme of the system, given on Fig.1, we could define the controller RNN-2 to have the same topology and learning as the identification RNN-1, and its neural model also could be linearized as:

$$X^c(k+1) = J^c X^c(k) - B_1^c V(k) - B_2^c X^i(k) + B_3^c R(k) \quad (19)$$

$$U(k) = C^c X^c(k) \quad (20)$$

Where: $X^c(k)$ is a N_c dimensional state vector ($N_c \leq L + M + N_i$ is supposed); $J^c \in \mathfrak{R}^{(N_c \times N_c)}$; $B_1^c \in \mathfrak{R}^{(N_c \times L)}$; $B_2^c \in \mathfrak{R}^{(N_c \times N_i)}$; $B_3^c \in \mathfrak{R}^{(N_c \times L)}$; $C^c \in \mathfrak{R}^{(M \times N_p)}$ are constant matrices ($L=M$ is supposed); the I-term $V(k) \in \mathfrak{R}^{(L \times 1)}$ output is defined as:

$$V(k+1) = V(k) + T_o Y_p(k) \quad (21)$$

To derive the dynamics of the closed-loop system we need to define the following statements and z -transfer functions, derived from its corresponding state space representations:

$$W^p(z) = W_1^p(z) D(z); D(z) = \text{diag} (z^{-n_i}), i=1, \dots, M \quad (22)$$

$$W_1^p(z) = C^p (zI - A^p)^{-1} B^p \quad (23)$$

$$P^i(z) = (zI - J^i)^{-1} B^i; X^i(z) = P^i(z) U(z) \quad (24)$$

$$I(z) = (zI - I)^{-1} T_o; V(z) = I(z) Y^p(z) \quad (25)$$

$$Q_1(z) = C^c (zI - J^c)^{-1} B^c_1 \quad (26)$$

$$Q_2(z) = C^c (zI - J^c)^{-1} B^c_2 \quad (27)$$

$$Q_3(z) = C^c (zI - J^c)^{-1} B^c_3 \quad (28)$$

The RTNN learning BP algorithm, given by the equations (4) to (8) is proved to be convergent (see the Theorem of stability, given above), and the RTNN model is proved to be stable, controllable and observable (see [9], [12], [13]), so the identification and control errors $E^i(k) = Y_p(k) - \hat{Y}_p(k)$, and $E^c(k) = R(k) - Y_p(k)$ tends to zero. The plant is also supposed to be BIBO stable. So the transfer functions (22) to (28) are stable with minimum phase. Expressing (15), (16) and (19), (20) in z -operator form, taking into account (22), (23), and using (24) to (28) yields:

$$Y_p(z) = W_1^p(z) D(z) U(z) + W_1^p(z) D(z) O(z) \quad (29)$$

$$U(z) = -Q_1(z) I(z) Y_p(z) - Q_2(z) P_i(z) U(z) + Q_3(z) R(z) \quad (30)$$

$$U(z) = [I + Q_2(z) P_i(z)]^{-1} [-Q_1(z) I(z) Y_p(z) + Q_3(z) R(z)] \quad (31)$$

Substituting (31) in (29) after some manipulations yields:

$$\begin{aligned} \{(z-1)I + W_1^p(z) D(z) [I + Q_2(z) P_i(z)]^{-1} Q_1(z) T_o\} Y_p(z) = \\ W_1^p(z) D(z) [I + Q_2(z) P_i(z)]^{-1} Q_3(z) (z-1) R(z) + \\ + W_1^p(z) D(z) (z-1) O(z) \end{aligned} \quad (32)$$

The equation (32) shows that the closed loop system remains stable if the time delays of the plant are not greater than the corresponding time constants of the continuous plant model, which is also condition for a normal PID regulator with a SISO plant. The proposed control method does not require information about the relative order of the plant like [14] does. It is seen also that the I-term reduces the steady-state systems error, which tend to zero when k tends to infinity. The block diagram of the second control scheme, containing two integral blocks is shown on Fig.2. The difference with respect to that, given on Fig.1 is that it contains one more I-term. In this case the equation (19) changed and obtained the form:

$$X^c(k+1) = J^c X^c(k) - B^c_4 Z(k) - B^c_1 V(k) - B^c_2 X^i(k) + B^c_3 R(k) \quad (33)$$

Where: $B^c_4 \in \mathfrak{R}^{(N_c \times L)}$ is a constant matrix, and the second I-term output vector is $Z(k) \in \mathfrak{R}^{(L \times 1)}$. The corresponding additional stable (or neutral) minimum phase transfer functions are defined as:

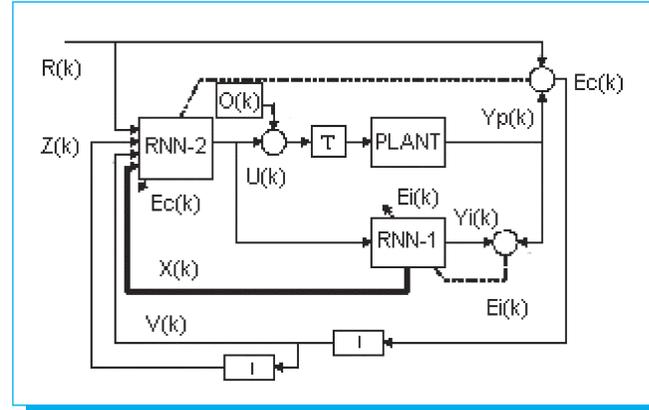


Fig. 2. Block-diagram of the direct adaptive neural control system containing two I-terms.

$$Z(k+1) = Z(k) + T_o V(k) \quad (34)$$

$$I_1(z) = (zI - I)^{-2} T_o^2; Z(z) = I_1(z) Y^p(z) \quad (35)$$

$$Q_4(z) = C^c (zI - J^c)^{-1} B^c_4 \quad (36)$$

Performing some substitutions in a similar way as above, we could obtain the following closed-loop systems equation for the case of double I-term:

$$\begin{aligned} \{(z-1)^2 I + W_1^p(z) D(z) [I + Q_3(z) P_i(z)]^{-1} [Q_1(z) T_o^2 + \\ + Q_2(z) (z-1) T_o]\} Y_p(z) = \{W_1^p(z) D(z) [I + Q_3(z) P_i(z)]^{-1} \\ Q_4(z) (z-1)^2 R(z) + W_1^p(z) D(z) (z-1)^2 O(z)\} \end{aligned} \quad (37)$$

The equation (37) shows that the closed loop system remains also stable for small delays, and it is obvious that both I-terms reduces the steady-state systems error, which tend to zero when $k \rightarrow \infty$. In the next part, the quality of the derived above two adaptive neural control schemes with I-terms will be illustrated by a simple effect evaporator control.

4.3. Simulation Results

Sketch of the simple effect evaporator is shown on Fig.3.

List of symbols used.

- M_1, C_1 Input volumetric flow and solution concentration;
- M_2, C_2 Output volumetric flow and solution concentration;
- M_s Input steam mass flow;
- P_1, P_2 Superior and inferior chamber pressures;
- M_v Steam mass flow produced by the evaporator;
- M_e Output steam mass flow;
- h Equivalent level of the solution in tubes;
- F_a Condenser input water flow;

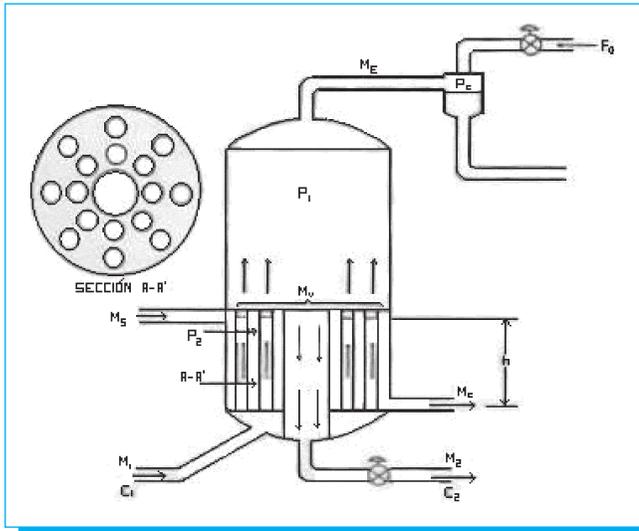


Fig. 3. Sketch of the simple effect evaporator.

- γ_1, γ_2 Input and output solution densities;
 G_l Evaporator's solution mass;
 G_x Equipment solids mass;
 G_s Steam mass in the lower chamber;
 G_v Steam mass in the upper chamber;
 V_0 Space volume under the tubes;
 S_t Transfer section of tubes;
 S_0 Valve position;
 d_1, d_2 Constants;
 e_1, e_2 Constants;
 μ, P_1 Constant; constant pressure of liquid discharge;
 L, K Evaporation heat, constant of heat transfer;
 θ_s, θ_l Steam temp.; boiling temp. of the liquid;
 V_c, γ_s Vol. of the heating chamber; steam density;
 M_c, L_1 Condense steam flow; evaporation heat;
 V_v, γ_v Vol. of the upper chamber; steam density;
 β Constant;
 K_1, K_2 Constants;
 F_a Water flow.

The model is based on the derivation of the mass balance equations of the evaporator (see [11]), which are

$$\frac{dG_l}{dt} = \gamma_1 M_1 - \gamma_2 M_2 - M_v \quad (38)$$

$$\frac{dG_x}{dt} = \gamma_1 M_1 C_1 - \gamma_2 M_2 C_2 \quad (39)$$

$$\frac{dG_v}{dt} = M_v - M_e \quad (40)$$

$$\frac{dG_v}{dt} = M_v - M_e \quad (41)$$

$$G_l = (V_0 + s_l h) \gamma_2 \quad (42)$$

$$G_x = G_l C_2 \quad (43)$$

$$\gamma_1 = d_1 c_1 + e_2 \quad (44)$$

$$\gamma_2 = d_2 c_2 + e_2 \quad (45)$$

$$M_2 = \mu s_0 \sqrt{2 \frac{P_1 - P_l}{\gamma_2} - 2gh} \quad (46)$$

$$M_v = 1/L [K \cdot s (\theta_s - \theta_l)] \quad (47)$$

$$\theta = \theta_l (P_v, C_2) \quad (48)$$

$$G_s = V_c \cdot \gamma_s \quad (49)$$

$$\gamma_s = d_3 P_2 + e_3 \quad (50)$$

$$\theta_s = d_4 P_2 + e_4 \quad (51)$$

$$M_c = 1/L_1 [K \cdot s (\theta_s - \theta_l)] \quad (52)$$

$$G_v = V_v \cdot \gamma_v \quad (53)$$

$$\gamma_v = d_5 P_1 + e_5 \quad (54)$$

$$M_e = \beta \sqrt{(P_1 - P_c) \gamma_s} \quad (55)$$

$$P_c = K_1 F_a + K_2 \quad (56)$$

The system of differential and algebraic equations (38) – (56) completely characterizes the dynamics of the simple effect evaporator. These equations are linearized and normalized so to obtain the state space dynamics equation (13), (14), with the following state, input and output vectors: $x^T = (h, P_1, P_2, C_2)$; $u^T = (M_s, S_0, F_a)$; $y = (h, P_1, P_2)$; and the following time constants in the respective states: $T_h = 0.250$ min; $T_{p_1} = 0.078$ sec.; $T_{p_2} = 0.300$; $T_{c_2} = 6$ min. Then the plant equations are discretized with a time period $T_o = 0.0125$ min, so to obtain the state space representation given by (15), (16) with the following state, input and output nonzero matrix elements:

$$\begin{aligned}
 a_{11} &= 0.9813, a_{12} = -0.0141, a_{13} = -0.0095, a_{14} = -0.0003, \\
 a_{22} &= 0.1646, a_{23} = 0.1123, a_{24} = -0.0007, a_{32} = -0.0100, \\
 a_{33} &= 0.0722, a_{34} = 0.0176, a_{43} = 0.0010, a_{45} = 0.9980;
 \end{aligned}$$

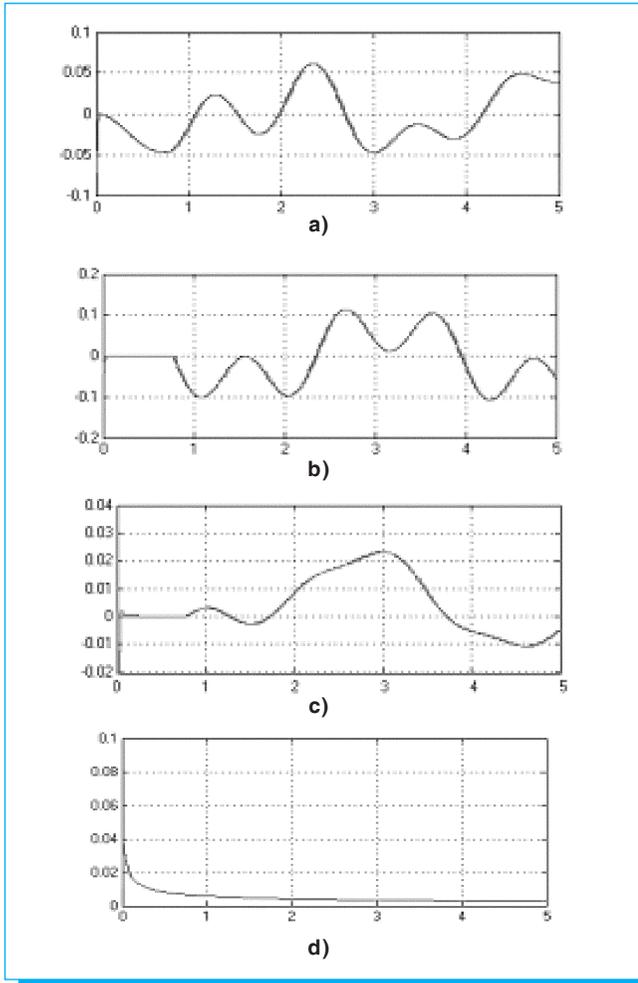


Fig. 4. Graphical results of plant identification. Comparison of the plant output and the RNN output; a) for the h -output; b) for the P_1 output; c) for the P_2 output; d) MSE% of identification.

$$\begin{aligned}
 b_{11} &= -0.0022, b_{12} = -0.0100, b_{13} = 0.0215, b_{21} = 0.0514, \\
 b_{23} &= -1.0973, a_{31} = 0.1385, b_{33} = 0.0277, b_{41} = -0.0003; \\
 c_{11} &= 1.0, c_{22} = 1.0, c_{33} = 1.0.
 \end{aligned}$$

The results of the first simulation experiment of RNN plant identification are shown on Fig. 4 a-d. The RNN topology is (3, 10, 3- three inputs, ten neurons in the hidden layer and three outputs). The learning parameters are $\eta=0.7$, $\alpha=0.05$. The first and the third inputs are delayed with $\tau_1 = 1$ min, and $\tau_3 = 0.75$ min. The three input plant signals have the form:

$$M_s = 0.1 \sin(0.05 \cdot k\pi) - 0.07 \sin(0.125 \cdot k\pi) \quad (57)$$

$$S_0 = 0.15 \sin(0.075 \cdot k\pi) + 0.1 \sin(0.25 \cdot k\pi) \quad (58)$$

$$F_a = 0.05 \sin(0.1 \cdot k\pi) + 0.05 \sin(0.3 \cdot k\pi) \quad (59)$$

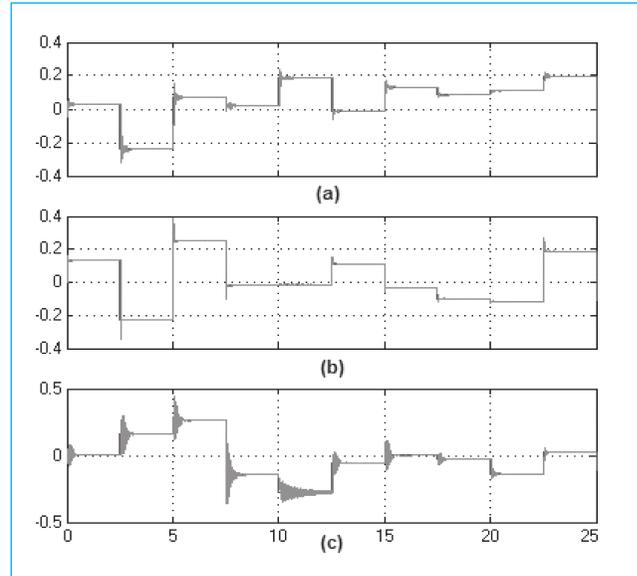


Fig. 5. Graphical simulation results obtained with a control scheme containing one I-term. Comparison of the plant output and the reference signal; a) for the h -output; b) for the P_1 output; c) for the P_2 output

The graphics on Fig. 4a-c compare the outputs of the plant with the respective outputs of the RNN in a 5 min simulation, and the Fig. 4d gives the Means Squared Error (MSE%) of identification, which reached the final value of 1.7%. The results exhibit a fast convergence and a good representation of the time delays. The simulation results with the controlled plant are obtained using a white noise (WN-function) reference signal with parameters T_m , p , s , *i.e.* sequence of pulses with different amplitude in the interval $[-0.5, 0.5]$ and frequency 2.5. The simulation results obtained with the first control scheme (see Fig.1) are given on Fig.5 a-c. The RNN-1 topology is (3,6,3)- and the RNN-2 topology is (12,10,3). The learning parameters for both RNNs are $\eta=0.6$, $\alpha=0.05$. It is added a 20% constant offset to all plant inputs, and the first and the third inputs are delayed with $\tau=0.75$ sec. The graphics compare the output of the plant and the respective reference signal. The results exhibit a good convergence (MSE% below 2.5%).

The simulation results given on Fig. 6 a-c corresponds to the control scheme given on Fig.2. The RNN-1 topology is also (3,6,3) and for the RNN-2 it is (15,10,3). The learning parameters for both RNNs are $\eta=0.5$, $\alpha=0.01$. A 40% constant offset is added to all plant inputs. The first and the third inputs are delayed with $\tau_1=2.0$ sec., and $\tau_3=0.75$ sec., respectively. The results exhibit a good convergence (MSE% below 2.5%) and a good resistance to small delays.

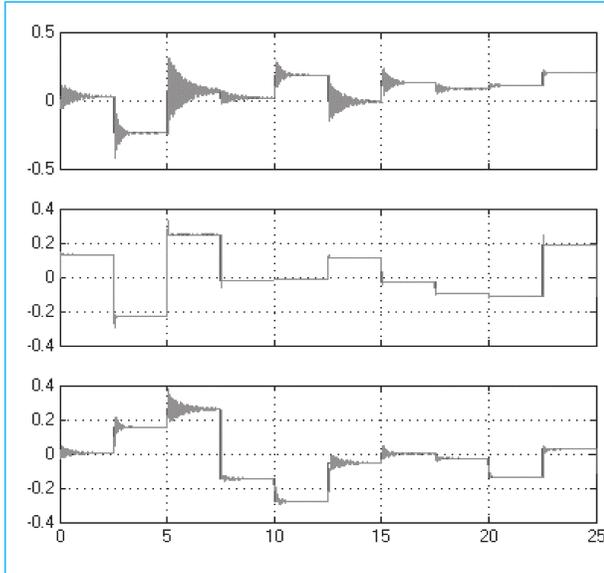


Fig. 5. Graphical simulation results obtained with a control scheme containing two I-terms; comparison of the plant output and the reference signal; a) for the h -output; b) for the P_1 output; c) for the P_2 output.

5. Conclusions

A direct adaptive neural control scheme with single and double I-term is proposed to be applied for time-delayed multivariable plant. The control scheme contains two Recurrent Trainable Neural Network (RTNN) models. The first RTNN is a plants parameter identifier and state estimator. The second RTNN is an integral plus states controller. The good performance of the adaptive neural control with I-terms is confirmed by closed-loop systems analysis, and by simulation results obtained with simple effect evaporator multivariable plant corrupted by noise, and affected by an unknown input time delay. The results show a good convergence (1.7% MSE of identification and 2.5% MSE of control) and ability of the I-term adaptive neural control to compensate a great constant offset (up to 40%).

6. References

- [1] S. Omatu, M. Khalil, and R. Yusof, *Neuro-Control and Their Applications*, Springer-Verlag, Berlin Heidelberg, 1995.
- [2] I. Cervantes, and J. Alvarez-Ramirez, «On the PID Tracking of Robot-Manipulators», *Systems Control Letters*, Vol. 42, 2001, pp. 37-46.
- [3] J.M. Lima, A.B. Azevedo, N. Duarte, C.M. Fonseca, A.E. Ruano, and P.J. Fleming, *Neuro-Genetic PID Autotuning*. Proc. of the European Control Conference, Porto, Portugal, Sept. 4-7, 2001, pp.3899-3904.
- [4] N.B. Almutairi, and M.Y. Chow, *A Modified PI Control Action with a Robust Adaptive Fuzzy Controller Applied to DC Motor*. Proc. of the IEEE Int. Joint Conf. on Neural Networks, Washington D.C., USA, July 15-19, 2001, [122.PDF].
- [5] V. Parra-Vega, and S. Arimoto, *Nonlinear PID Control with Sliding Modes for Tracking of Robot Manipulators*. Proc. of the IEEE Int. Conf. on Control Applications, Mexico City, Mexico, Sept. 5-7, 2001, pp. 351-356.
- [6] I.Baruch, J.M.Godoy, J.R.Valle R. and J.M.Flores, *Adaptive Neural Control of a Two Phase Hydrocarbon Separator*. (M.H.Hamza - Ed.) Proc. of the IASTED Int. Conference on Control and Applications, Cancun, Mexico, May 2002, ACTA Press, Anaheim, Calgary 2002, ISBN: 0-88986-330-X, pp. 161-166.
- [7] I.Baruch, J.Barrera, J.Perez M., and Luis Alberto Hernandez P., «An adaptive neural control of a fed-batch fermentation processe». Proc. of the IEEE Int. Conference on Control Applications, Istanbul, Turkey, June 23-25, 2003, pp. 808-812.
- [8] I.Baruch, A. Martinez Q. and R.Garrido, *Direct Adaptive Neural Control Scheme with Integral-Plus-State Action*. (O.Castillo - Ed.) Proc. of the IASTED Int. Conference on Neural Networks and Computational Intelligence., Cancun, Mexico, May 2003, ACTA Press, Anaheim, Calgary ISBN: 0-88986-347-4, pp. 84-89.
- [9] I. Baruch, J.M. Flores, F. Nava, I.R. Ramirez, and B. Nenkova, *An Adavanced Neural Network Topology and Learning, Applied for Identification and Control of a D.C. Motor*, (Sgurev V., Jotsov V., eds.) Proc. of the First Int. IEEE Symposium on Intelligent Systems, Varna, Bulgaria, Sept. 2002, IEEE Inc., N. Y., ISBN 0-7803-7601-3, pp. 289-295.
- [10] R.Lozano, P.Castillo, P.Garcia, and A.Dzul, «Robust Prediction-Based Control for Instable Delay Systems: Application to the Yaw Control of a Mini-Helicopter», *Automatica*, Vol. 40, 2004, pp. 603-612.
- [11] A.Aguado, J.Enriquez, and J.Pascual, *Modern Control Theory (in Spanish -Teoria Moderna del Control)*, Editorial Academia, ACC, La Habana 2, Cuba, 1980, pp. 53-86.
- [12] E. Sontag, and F. Albertini, «State Observability in Recurrent Neural Networks», *System and Control Letters*, Vol. 22, 1994, pp. 235-244.
- [13] E. Sontag, and H. Sussmann, «Complete Controllability of Continuous Time Recurrent Neural Network», *System and Control Letters*, Vol. 30, 1997, pp. 177-183.
- [14] K. Narendra, and S. Mukhopadhyay, «Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks», *Neural Networks*, Vol. 7, 1994, pp. 737-752.