4

Recurrent Neural Network Identification and Adaptive Neural Control of Hydrocarbon Biodegradation Processes

Ieroham Baruch[1], Carlos Mariaca-Gaspar[1], and Josefina Barrera-Cortes[2]
CINVESTAV-IPN, Mexico City, [1]Department of Automatic Control, [2]Department of Biotechnology & Bioengineering, Mexico

1. Introduction

The Recent advances in understanding of the working principles of artificial neural networks has given a tremendous boost to identification, prediction and control tools of nonlinear systems, (Narendra & Parthasarathy, 1990; Chen & Billings, 1992; Hunt et al., 1992; Miller III et al., 1992; Pao et al., 1992; Su et al., 1992; Narendra & Mukhopadhyay, 1994; Boskovic & Narendra, 1995; Ku & Lee, 1995; Baruch et al., 1996; Jin & Gupta, 1999, Haykin, 1999; Mastorocostas & Theocharis, 2006; Kazemy et all., 2007). The main network property namely the ability to approximate complex non-linear relationships without prior knowledge of the model structure makes them a very attractive alternative to the classical modeling and control techniques. This property has been proved by the universal approximation theorem, (Haykin, 1999). Among several possible network architectures the ones most widely used are the feedforward and the recurrent neural networks. In a feed-forward neural network 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 neural dynamic plant model. A recurrent neural network has local feedback connections to some of the previous layers. Such a structure is suitable alternative to the first one when the task is to model dynamic systems, and the universal approximation theorem has been proved for the recurrent neural networks too. The preferences given to recurrent neural network identification with respect to the classical methods of process identification are clearly demonstrated in the solution of the "bias-variance dilemma", (Haykin, 1999). Furthermore, the derivation of an analytical plant model, the parameterization of that model and the Least Square solution for the unknown parameters have the following disadvantages: (a) the analytical model did not include all factors having influence to the process behavior; (b) the analytical model is derived taking into account some simplifying suppositions which not ever match; (c) the analytical model did not described all plant nonlinearities, time lags and time delays belonging to the process in hand; (d) the analytical model did not include all process and measurement noises which are sensor and actuator dependent. In (Sage, 1968) the method of invariant imbedding has been described. This method seemed to be a universal tool for simultaneous state and

Source: Recurrent Neural Networks, Book edited by: Xiaolin Hu and P. Balasubramaniam, ISBN 978-953-7619-08-4, pp. 400, September 2008, I-Tech, Vienna, Austria

parameter estimation of nonlinear plants but it suffer for the same drawbacks because a complete nonlinear plant model description is needed.
So, the unknown nonlinear technological processes needed a new tool for modeling and identification capable to correlate experimental data and to estimate parameters and states in the same time, processing noisy measurements. Such efficient tool is the recurrent neural Kalman filter, where the estimated parameters and states could be used directly for control.

2. Description of the recurrent neural Kalman filter

2.1 Topology and learning of the recurrent trainable neural network

The Recurrent Trainable Neural Network (RTNN) topology, given on Fig. 1, is a hybrid one. It has one recurrent hidden layer and one feedforward output layer. This topology is inspired from the Jordan canonical form of the state-space representation of linear dynamic systems (Baruch et al., 1996) adding activation functions to the state and the output equations so to convert it to recurrent neural network named Recurrent Trainable Neural Network described by the equations:

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

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

$$A = \text{block-diag}(Ai), \; |Ai| < 1 \tag{3}$$

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

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

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

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

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.
The RTNN topology has been derived independently of the hybrid Diagonal Recurrent Neural Network (DRNN) (see Ku & Lee, 1995) with which it have the following differences: (a) the state equation (1) of the RTNN is linear and the state equation of the DRNN is nonlinear (the activation functions are in the closed loop). This made the RTNN completely controllable and the state, and parameter information X, A, B of RTNN directly applicable for state-space control purposes. On the other side, the controllability of the DRNN depends on the type of the activation functions (see Sontag & Sussmann, 1997); (b) the state weight matrix A of the RTNN is defined as block- diagonal (3) and some stability bounds have been imposed to it which preserved the RTNN stability during the learning. The DRNN was defined as block-diagonal later (Mastorocostas & Theocharis, 2006; Kazemy et al., 2007) and

some algorithmic measures have been taken to maintained the stability of DRNN during the learning. For the RTNN the learning of the Jordan blocks is resolved in universal manner, defining diagonal and full-matrix Backpropagation (BP) learning options; (c) the RTNN include thresholds in the inputs of both layers which facilitated the nonlinear systems identification, especially in lack of a-priory information about the approximated nonlinear plant. The DRNN did not apply thresholds; (d) the output layer of the DRNN is linear, and that of the RTNN is nonlinear, which permitted it to perform better approximation of nonlinear plants. Furthermore, depending on the plant structure, the topology of the RTNN could be extended with additional feedforward output or input layers which augmented the approximation ability of the RTNN. The observability of the DRNN has been proved by (Albertini & Sontag, 1994). The observability of the RTNN is assumed and it is fulfilled when the reference signal entered in the limits of the activation functions. The dynamic BP algorithm of RTNN learning is derived using the adjoint RTNN topology, predicting the output error (see Fig. 2). The adjoint RTNN is built applying the Separation theorem (Sage, 1968) and the diagrammatic method of (Wan & Beaufays, 1996). The BP algorithm is:

$$W(k+1) = W(k) + \eta\, \Delta W(k) + \alpha\, \Delta W(k-1);\ |W_{ij}| < W_0 \tag{8}$$

$$E(k) = Y_d(k) - Y(k);\ E_1(k) = F'[Y(k)]\, E(k) \tag{9}$$

$$F'[Y(k)] = [1-Y^2(k)] \tag{10}$$

$$\Delta C(k) = E_1(k)\, Z^T(k) \tag{11}$$

$$E_3(k) = G'[Z(k)]\, E_2(k);\ E_2(k) = C^T(k)\, E_1(k) \tag{12}$$

$$G'[Z(k)] = [1-Z^2(k)] \tag{13}$$

$$\Delta B(k) = E_3(k)\, U^T(k) \tag{14}$$

$$\Delta A(k) = E_3(k)\, X^T(k) \tag{15}$$

$$Vec(\Delta A(k)) = E_3(k) \square X(k) \tag{16}$$

Where: F′[.], G′[.] are derivatives of the tanh(.) functions; W is a general weight, denoting each weight matrix (C, A, B) in the RTNN model, to be updated; ΔW (ΔC, ΔA, ΔB), is the weight correction of W; Y_d is an L-dimensional output of the approximated plant taken as a reference for RTNN learning; η, α are learning rate parameters; ΔC is a weight correction of C; ΔB is a weight correction of B; ΔA is a weight correction of A; the diagonal of the matrix A is denoted by Vec (A(k)) where (16) represents its learning as an element-by-element vector product; E, E_1, E_2, E_3, are error vectors (see Fig. 2).

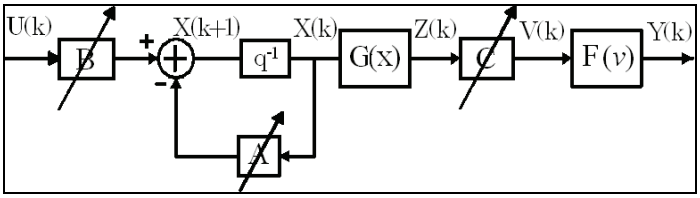


Fig. 1. Block diagram of the RTNN model

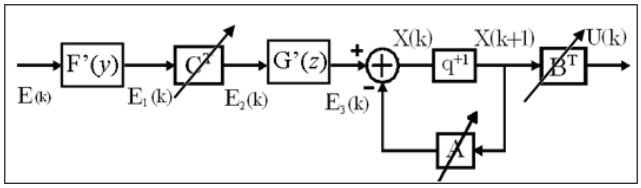


Fig. 2. Block diagram of the adjoint RTNN model

The dimension of the state vector X of the RTNN is chosen using the simple rule of thumb which is: N=L+M. 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 is assured by the activation functions [-1, 1] bounds and by the local stability weight bound conditions given by (3), (8). The stability of the RTNN movement around the optimal weight point will be proven extending the proof of (Nava et al., 2004), as it is stated below.

Theorem of stability of the RTNN. Let the RTNN with Jordan Canonical Structure is given by equations (1)-(7) (see Fig.1) and the nonlinear plant model, is as follows:

$$X_{d.}(k+1) = G[\, X_d(k), U(k)\,] \tag{17}$$

$$Y_d(k) = F[\, X_d(k)\,] \tag{18}$$

Where: $\{Y_d(.), X_d(.), U(.)\}$ are output, state and input variables with dimensions L, N_d, M, respectively; F(.), G(.) are vector valued nonlinear functions with respective dimensions. Under the assumption of RTNN identifiability made, the application of the BP learning algorithm for A(.), B(.), C(.), in general matricial form, described by equation (8)-(16), and the learning rates $\eta(k)$, $\alpha(k)$ (here they are considered as time-dependent and normalized with respect to the error) are derived using the following Lyapunov function:

$$L(k) = L_1(k) + L_2(k) \tag{19}$$

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

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

$$L_2(k) = \mathrm{tr}\left(\tilde{W}_A(k)\tilde{W}_A^T(k)\right) + \mathrm{tr}\left(\tilde{W}_B(k)\tilde{W}_B^T(k)\right) + \mathrm{tr}\left(\tilde{W}_C(k)\tilde{W}_C^T(k)\right)$$

Where:

$$\tilde{W}_A(k) = \hat{A}(k) - A^*, \tilde{W}_B(k) = \hat{B}(k) - B^*, \tilde{W}_C(k) = \hat{C}(k) - C^*$$

are vectors of the parameter 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 the neural weight at the k-th step, respectively, for each case.

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

$$\begin{aligned} &L(k+1) = L_1(k+1) + L_2(k+1) < 0 \\ &\Delta L(k+1) = L(k+1) - L(k) \end{aligned} \tag{20}$$

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

$$\frac{\left(1-\frac{1}{\sqrt{2}}\right)}{\psi_{max}} < \eta_{max} < \frac{\left(1+\frac{1}{\sqrt{2}}\right)}{\psi_{max}}$$

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

$$\Delta L_2(k+1) < -\eta_{max}|e(k+1)|^2 - \alpha_{max}|e(k)|^2 + d(k+1)$$

Note that η_{max} changes adaptively during the RTNN learnig and:

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

Where all: the unmodelled dynamics, the approximation errors and the perturbations, are represented by the d-term, and the complete proof of that theorem, is given in Apendix A. The Rate of Convergence Lemma, used, is given in (Nava et al., 2004).

2.2 Topology and learning of the Kalman filter recurrent neural network

Let us consider the linearized plant model (17), (18), represented in a state-space form:

$$X_{d.}(k+1) = A_d(k)\, X_d(k) + B_d(k)\, U(k) + \Theta_1(k) \quad (21)$$

$$Y_d(k) = C_d(k)\, X_d(k) + \Theta_2(k) \quad (22)$$

Where: E [.] means mathematical expectation; the process and measurement noises $\Theta_1(.)$, $\Theta_2(.)$ are white, with $\Theta_1(k)$, $\Theta_2(s)$ and the initial state $X_d(k_0)$ independent and zero mean for all k, s, with known variances $E[X_d(k)\, X_d^T(k)] = P_0$, $E[\Theta_1(k)\, \Theta_1^T(k)] = Q(k)\, \delta(k-\tau)$, $E[\Theta_2(k)\, \Theta_2^T(k)] = R(k)\, \delta(k-\tau)$, where $\delta(k-\tau)=1$ if $k=\tau$, and 0 otherwize. The optimal Kalman filter theory is completely described in (Sage, 1968) and we would not repeated it here.

For us the Kalman Filter (KF) is a full rank optimal state estimator capable to estimate the systems state, to filter the process and measurement noises taking in hand all plant information available like: input/output plant data, all parameters of the plant model (21), (22), and the given up noise and initial state statistics (mean and variance). The basic Kalman filter equations for the estimated state and output variables are given by:

$$X_{e.}(k+1) = A_e(k)\, X_e(k) + K_e(k)\, Y_d(k) + B_d(k)\, U(k) \quad (23)$$

$$A_e(k) = A_d(k) - K_e(k)\, C_d(k) \quad (24)$$

$$Y_e(k) = C_d(k)\, X_e(k) \quad (25)$$

Where: $X_e(k)$ is the estimated state vector with dimension N_e; $A_e(k)$ is a (N_e x N_e) closed-loop KF state matrix; $Y_e(k)$ is the estimated plant output vector variable with dimension L; $K_e(k)$ is the optimal Kalman filter gain matrix with dimension (N_e x L). This gain matrix is computed applying the optimal Kalman filtering methodology given in (Sage, 1968). So, the KF performed noise filtration by means of an optimal closed-loop feedback which has the drawback that the feedback amplified the noise components of the error, especialy when the

feedback gain is high. The second draback is that the KF design needs a complete plant parameter and noise information, which means that if the plant data are incomplete the process noise level is augmented. To overcome this we need to take special measures like to augment the filtering capabilities of the KF.
So, the Kalman filter could not estimate parameters and states in the same time processing noisy measuremets with unknown noise statistics, and it will be our task. To resolve this task we need to derive the topology and the BP learning algorithm of a new recurrent Kalman filter neural network. First of all we could rewrite the equation (23) defining a new extended input vector, containing all available input/output information issued by the plant, and second – we could modify the output equation (25) so to convert it to an output noise filter. After that we obtain:

$$X_e(k+1) = A_d(k)\, X_e(k) - K_e(k)\, Y_e(k) + B_2(k)\, U_e(k) \tag{26}$$

$$B_2 = [B_d\ ;\ K_e];\ U_e^T = [U\ ;\ Y_d] \tag{27}$$

$$Z(k) = C_d(k)\, X_e(k) \tag{28}$$

$$Y_e(k+1) = A_2\, Y_e(k) + Z(k) \tag{29}$$

Now comparing the equations (26)-(29) with the RTNN topology (1)-(7) we decided to extend the RTNN topology adding local and global feedbacks so that to fulfil KF requirements. The obtained new Kalman Filter Recurrent Neural Network (KF RNN) topology is given on Fig. 3, where the first layer represented the plant model, the second layer represented the output noise filtering model, and it has also a global output feedback. The KF RNN topology is described by the equations:

$$X(k+1) = A_1 X(k) + BU(k) - DY(k) \tag{30}$$

$$B = [B_1\ ;\ B_0];\ U^T = [U_1\ ;\ U_2] \tag{31}$$

$$A_1 = \text{block-diag}\,(A_{1,i}),\ |\ A_{1,i}\ | < 1 \tag{32}$$

$$Z_1(k) = G[X(k)] \tag{33}$$

$$C = [C_1\ ;\ C_0];\ Z^T = [Z_1\ ;\ Z_2] \tag{34}$$

$$V_1(k) = CZ(k) \tag{35}$$

$$V(k+1) = V_1(k) + A_2 V(k) \tag{36}$$

$$A_2 = \text{block-diag}\,(A_{2,i}),\ |A_{2,i}\ | < 1 \tag{37}$$

$$Y(k) = F[V(k)] \tag{38}$$

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_1, A_2 are (NxN) and (LxL)

block-diagonal weight matrices; 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. Here the input vector U and the input matrix B of the KF RNN are augmented so to fulfill the specifications (27) and the matrix D corresponded to the feedback gain matrix of the KF. So the KF RNN topology corresponded functionally to the KF definition (26)-(29) and ought to be learnt applying the BP learning algorithm which is in fact an unrestricted optimization procedure, derived using the adjoint KF RNN (see Fig.4) for KF RNN topology, applying the Separation theorem (Sage, 1968) and the diagrammatic method (Wan & Beaufays, 1996). The BP learning algorithm, expressed in vector-matricial form is:

$$W(k+1) = W(k) + \eta\, \Delta W(k) + \alpha\, \Delta W(k-1);\ |W_{ij}| < W_0 \tag{39}$$

$$E(k) = Y_d(k) - Y(k);\ E_1(k) = F'[Y(k)]\, E(k) \tag{40}$$

$$F'[Y(k)] = [1 - Y^2(k)] \tag{41}$$

$$\Delta C(k) = E_1(k)\, Z^T(k) \tag{42}$$

$$\Delta A_2(k) = E_1(k)\, V^T(k) \tag{43}$$

$$Vec(\Delta A_2(k)) = E_1(k) \otimes X(k) \tag{44}$$

$$E_3(k) = G'[Z(k)]\, E_2(k);\ E_2(k) = C^T(k)\, E_1(k) \tag{45}$$

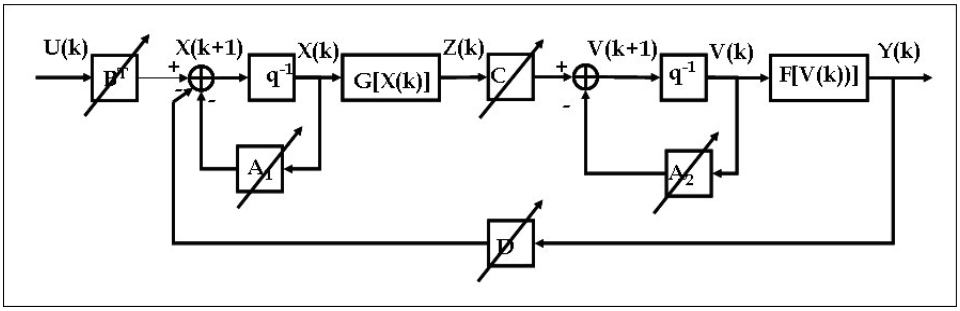


Fig. 3. Block diagram of the KF RNN model

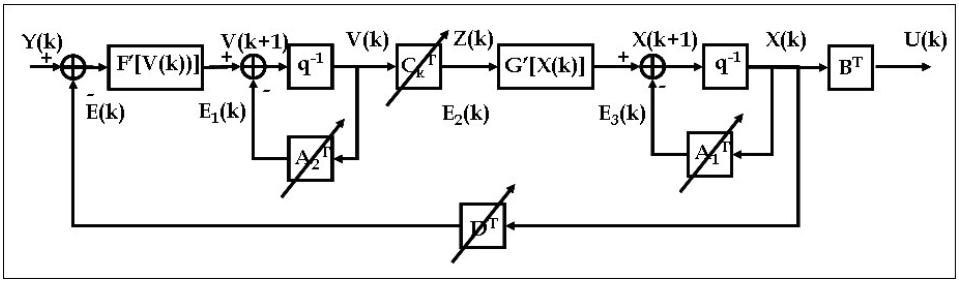


Fig. 4. Block diagram of the adjoint KF RNN model

$$G'[Z(k)] = [1-Z^2(k)] \tag{46}$$

$$\Delta B(k) = E_3(k)\, U^T(k) \tag{47}$$

$$\Delta D(k) = E_3(k)\, Y^T(k) \tag{48}$$

$$\Delta A_1(k) = E_3(k)\, X^T(k) \tag{49}$$

$$Vec(\Delta A_1(k)) = E_3(k) \square X(k) \tag{50}$$

Where: F'[.], G'[.] are derivatives of the tanh(.) functions; W is a general weight, denoting each weight matrix (C, A_1, A_2, B, D) in the KF RNN model, to be updated; ΔW (ΔC, ΔA_1, ΔA_2, ΔB, ΔD), is a weight correction of W; Y_d is an L-dimensional output of the approximated plant taken as a reference for KF RNN learning; η, α are learning rate parameters; ΔC is a weight correction of C; ΔB is a weight correction of B; ΔD is a weight correction of D, ΔA_1 is a weight correction of A_1 , ΔA_2 is a weight correction of A_2; the diagonals of the matrices A_1, A_2 are denoted by Vec ($A_1(k)$), Vec ($A_2(k)$), respectively, where (44), (50) represented their learning as an element-by-element vector products; E, E_1, E_2, E_3, are error vectors (see Fig. 4), predicted by the adjoint KF RNN model.

So, the KF RNN is capable to issue parameter and state estimations for control purposes, thanks to the optimization capabilities of the BP learning algorithm, applying the "correction for error" delta rule of learning (see Haykin, 1999). The stability of the KF RNN model is assured by the activation functions [-1, 1] bounds and by the local stability weight bound conditions given by (32), (37). The stability of the KF RNN movement around the optimal weight point has been proved by one theorem and the Rate of Convergence Lemma (see Nava et al., 2004), following the same way as for the RTNN. It is stated below.

Theorem of stability of the KF RNN. Let the KF RNN is given by equations (30)-(38) (see Fig.3) and the nonlinear plant model, is given by equations (17), (18). Under the assumption of KF RNN identifiability made, the application of the BP learning algorithm for C, A_1, A_2, B, D, in general matricial form, described by equation (39)-(50), and the learning rates η (k), α (k) (here they are considered as time-dependent and normalized with respect to the error) are derived using the following Lyapunov function:

$$L(k) = L_1(k) + L_2(k) \tag{51}$$

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

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

$$L_2(k) = tr\left(\tilde{W}_{A1}(k)\tilde{W}_{A_1}^T(k)\right) + tr\left(\tilde{W}_{A2}(k)\tilde{W}_{A_2}^T(k)\right) + tr\left(\tilde{W}_B(k)\tilde{W}_B^T(k)\right) + tr\left(\tilde{W}_C(k)\tilde{W}_C^T(k)\right) + tr\left(\tilde{W}_D(k)\tilde{W}_D^T(k)\right)$$

Where:

$\tilde{W}_{A1}(k) = \hat{A}_1(k) - A_1^*, \tilde{W}_{A2}(k) = \hat{A}_2(k) - A_2^*, \tilde{W}_B(k) = \hat{B}(k) - B^*, \tilde{W}_C(k) = \hat{C}(k) - C^*, \tilde{W}_D(k) = \hat{D}(k) - D^*$

are vectors of the estimation error and $(A_1^*, A_2^*, B^*, C^*, D^*)$ and $(\hat{A}_1(k), \hat{A}_2(k), \hat{B}(k), \hat{C}(k), \hat{D}(k))$ denote the ideal optimal neural weight and the estimate of the neural weight at the k-th step, respectively, for each case.

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

$$\begin{aligned} L(k+1) &= L_1(k+1)+L_2(k+1)<0 \\ \Delta L(k+1) &= L(k+1) - L(k) \end{aligned} \tag{52}$$

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

$$\frac{\left(1-\frac{1}{\sqrt{2}}\right)}{\psi_{max}}<\eta_{max}<\frac{\left(1+\frac{1}{\sqrt{2}}\right)}{\psi_{max}}$$

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

$$\Delta L_2(k+1)<-\eta_{max}|e(k+1)|^2-\alpha_{max}|e(k)|^2+d(k+1)$$

Note that η_{max} changes adaptively during learnig process of the network and

$$\eta_{max}=\max_{i=1}^{5}\{\eta_i\}$$

Where all: the unmodeled dynamics, the approximation errors and the perturbations, are represented by the d-term, and the complete proof of that theorem can be easily obtained following the same procedure detailed in Appendix A derived for the RTNN.

3. Description of the adaptive control schemes

3.1 Indirect adaptive control scheme (sliding mode control)

The indirect adaptive control using the RTNN as plant identifier has been described in (Baruch et al., 2001a; Baruch et al., 2001b; Baruch et al., 2005). Later the proposed indirect control has been derived as a Sliding Mode Control (SMC) and some preliminary results of SMC of unknown hydrocarbon biodegradation processes have been reported (see Baruch et al., 2007a; Baruch et al., 2007b). The block diagram of the indirect adaptive control scheme is shown on Fig. 5. It contained identification and state estimation KF RNN and a sliding mode controller. The stable nonlinear plant is identified by a KF RNN model with topology, given by equations (30)-(38) learned by the stable BP-learning algorithm, given by equations (39)-(50), where the identification error tends to zero. The simplification and linearization of the neural identifier equations (30)-(33), omitting the DY(.) and $K_eY_d(.)$, (27) parts, leads to the next local linear plant model, extracted from the complete KF RNN model:

$$X(k+1) = A_1X(k) + BU(k) \tag{53}$$

$$Z(k) = H\,X(k);\ H = C\,G'(Z) \tag{54}$$

Where G'(.) is the derivative of the activation function and L = M, is supposed.
In (Young et al., 1999), the sliding surface is defind with respect to the state variables and the SMC objective is to move the states form an arbitrary space position to the sliding surface in finite time. In (Levent, 2003), the sliding surface is also defined with respect to the states but the states of a SISO systems are obtained from the plant outputs by differentiation. In (Eduards et al., 2003), the sliding surface definition and the control objectives are the same. The equivalent control systems design is done with respect to the plant output, but

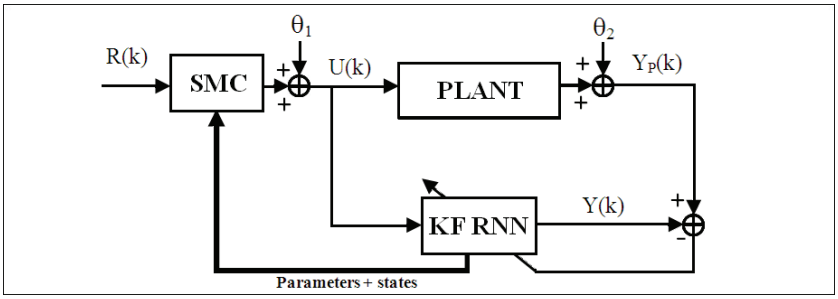


Fig. 5. Block diagram of the closed-loop system containing KF RNN identifier and a SMC

the reachability of the stable output control depended on the plant structure. In (Baruch et al., 2007a; Baruch et al., 2007b), the sliding surface is derived directly with respect to the plant outputs which facilitated the equivalent SMC systems design.
Let us define the following Sliding Surface (SS) as an output tracking error function:

$$S(k+1)=E(k+1)+\sum_{i=1}^{P}\gamma_i\, E(k-i+1);\quad |\gamma_i| < 1 \tag{55}$$

Where: S(.) is the Sliding Surface Error Function (SSEF) defined with respect to the plant output; E(.) is the systems output tracking error; γ_i are parameters of the desired stable SSEF; P is the order of the SSEF. The tracking error and its iterate are defined as:

$$E(k) = R(k) - Z(k);\ E(k+1) = R(k+1) - Z(k+1) \tag{56}$$

Where R(k), Z(k) are L-dimensional reference and output vectors of the local linear plant model. The objective of the sliding mode control systems design is to find a control action which maintains the systems error on the sliding surface which assure that the output tracking error reaches zero in P steps, where P < N. So, the control objective is fulfilled if:

$$S(k+1) = 0 \tag{57}$$

Now, let us to iterate (54) and to substitute (53) in it so to obtain the input/output local plant model, which yields:

$$Z(k+1) = H\,X(k+1) = H\,[AX(k) + BU(k)] \tag{58}$$

From (55), (56), and (57) it is easy to obtain:

$$R(k+1) - Z(k+1) + \sum_{i=1}^{P}\gamma_i\, E(k-i+1) = 0 \tag{59}$$

The substitution of (58) in (59) gives:

$$R(k+1) - HAX(k) - HBU(k) + \sum_{i=1}^{P}\gamma_i\, E(k-i+1) = 0 \tag{60}$$

As the local approximation plant model (53), (54), is controllable, observable and stable, (see the proofs of the preceeding paragraph), the matrix A_1 is diagonal, and L = M, then the matrix product (HB), representing the plant model static gain, is nonsingular, and the plant

states X(k) are smooth non-increasing functions. Now, from (60) it is easy to obtain the equivalent control capable to lead the system to the sliding surface which yields:

$$U_{eq}(k) = (HB)^{-1} [- HAX(k) + R(k+1) + \sum_{i=1}^{P} \gamma_i \, E(k-i+1)] \tag{61}$$

Following (Young et al., 1999), the SMC avoiding chattering is taken using a saturation function instead of sign one. Here the saturation level Uo is chosen with respect to the load level perturbation. So the SMC takes the form:

$$U^{*}(k) = \begin{cases} U_{eq}(k), & \text{if } \left\| U_{eq}(k) \right\| < U_0 \\ -U_0 U_{eq}(k) / \left\| U_{eq}(k) \right\|, & \text{if } \left\| U_{eq}(k) \right\| \geq U_0 \end{cases} \tag{62}$$

It is easy to see that the substitution of the equivalent control (61) in the linear plant model (53), (54) show an exact complete plant dynamics compensation which avoided oscillations, so that the chattaring effect is not observed. Furthermore, the designed plant output sliding mode equivalent control substituted the multi-input multi-output coupled high order dynamics of the linearized plant with desired decoupled low order one.

3.2 Direct adaptive neural control scheme

The Direct Adaptive Neural Control (DANC) using the RTNN as plant identifier and plant controller has been described in (Baruch et al., 2001b; Flores et al., 2001; Baruch et al., 2004; Baruch et al., 2005). The block-diagram of the control system is given on Fig. 6. It contains a recurrent neural identifier, and two recurrent neural controllers (feedback and feedforward). Let us to write the following *z*-transfer- function representations of the plant, the state estimation part of the KF RNN, the feedback and the feedforward controllers:

$$W_p(z) = C_p \, (zI - A_p)^{-1} \, B_p \tag{63}$$

$$P_i(z) = (zI - A_i)^{-1} \, B_i \tag{64}$$

$$Q_1(z) = C_{cfb} \, (zI - A_{cfb})^{-1} \, B_{cfb} \tag{65}$$

$$Q_2(z) = C_{cff} \, (zI - A_{cff})^{-1} \, B_{cff} \tag{66}$$

The control systems *z*-transfer functions (63)-(66) are connected by the following equation, derived from the Fig. 6, and given in *z*-operational form:

$$Y_p(z) = W_p(z) \, [I + Q_1(z) \, P_i(z)]^{-1} \, Q_2(z) \, R(z) + \theta(z) \tag{67}$$

$$\theta(z) = W_p(z) \, \theta_1(z) + \theta_2(z) \tag{68}$$

Where $\theta(z)$ is a noise term. The RTNN and the KF RNN topologies are controllable and observable. The BP algorithm of learning is convergent (Baruch et al., 2002; Nava et al., 2004). Then the identification and control errors tend to zero.

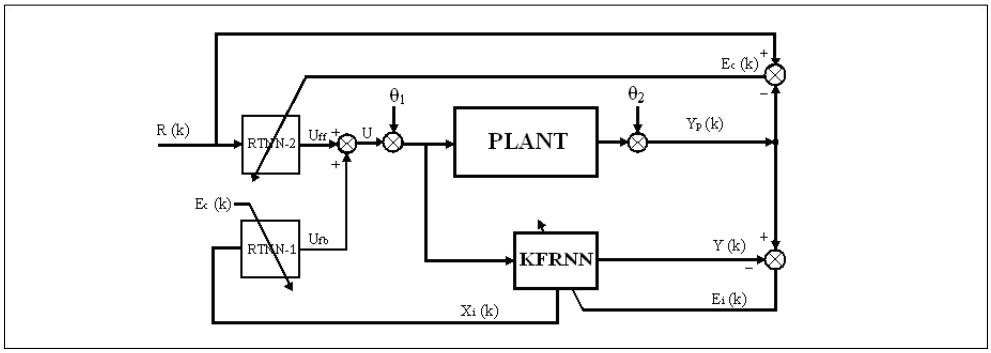


Fig. 6. Block - diagram of the control system containing neural identifier and two adaptive neural controllers.

$$E_i(k) = Y_p(k) - Y(k) \rightarrow 0;\ k \rightarrow \infty \tag{69}$$

$$E_c(k) = R(k) - Y_p(k) \rightarrow 0;\ k \rightarrow \infty \tag{70}$$

This means that each transfer function given by equations (63)-(66) 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.

4. Experimental and simulation results

A time ago the KF RNN has been applied for prediction of various bioprocesses like the Fed-Batch fermentation kinetics of Bacillus Thuringiensis (Valdes-Castro et al., 2003), the osmotic dehydration process (Baruch et al., 2005), and the hydrocarbon degradation profiles in a biopile system (De la Torre-Sanchez et al., 2006). Some preliminary results of application of the KF RNN used as systems identifier in a sliding mode controlled bioremediation processes have been presented in various scientific conferences like (Baruch et al., 2007 a; Baruch et al., 2007b). In this part those results would be described with more details. The bioremediation process at hand is considered as completely unknown and represented by input/output records of normalized noisy data.

4.1 Experimental and simulation results obtained for the biopile system

Description of the Process and the Experiment. Biological treatment is attractive as a potentially low-cost technology, which converts toxic organic contaminants into CO_2 and biomass. Since the 70's, this technology has been applied for the hydrocarbon degradation, and today, it is considered as the best alternative to cleanup polluted soils. Bioremediation in biopile system is an *ex-situ* Solid Substrate Fermentation (SSF) technology, based on the ability of micro-organisms to degrade pollutant hydrocarbon compounds (Alexander, 1994). The often used bio-stimulation technique consists on the activation of the native soil micro-organisms by addition of nutrients, water, oxygen (for aerobic process) and a bulking agent that let it to improve the oxygen supplied to the microorganisms. The Solid Substrate

Fermentation takes place in the absence of free water, so it offers the advantage of reducing the place and cost requirements. The SSF disadvantage consists of the complexity and heterogeneity of the solid matrix, which makes quite difficult the measurement and control of process variables. The interest of the biopile technology is an inherent temperature increase inside the biopile - from the centre to the surface, which favors the sequential development of a microbial population growth associated to the temperature profile and residual pollution. Temperature increase can reach 60^{o}C, so it is frequently controlled by an air flux supplied to the biopile columns. Besides, controlling the temperature, the air flux is a source of fresh oxygen to the microorganisms. The next environmental conditions are recommended for an adequate hydrocarbon biodegradation in biopile system: pH ≈ 7; humidity at 50-60% of the water holding capacity of soil; average temperature of 30^{o}C. It is important to supply an adequate air flux, since a low one could not be enough for satisfying the microbial requirements, but a high air one could dry the solid matrix. In this study, it is used a crumb-limose soil from a site polluted near a refinery in México. The pollution of 165000 ppm, consist on different residues of crude oil process and refining. The soil was dried and blended with ocorn used as a bulking agent 10:1 (% v/v), which was milled and sterilized. The moisture was adjusted at 60% of water retention capacity, and C:N:P ratio at 100:10:1 according to analyses done. Tergitol 1% (p/p) was used as surfactant to enhance contaminant desorption from soil. The equipment used is shown on Fig. 7a, and the Input/Output full KF RNN learning pattern in shown on Fig. 7b. The biopile system consists of twenty one columns (1.0 m height x 3.81 cm i.d.), constructed to allow the monitoring through 28 days, almost each other day. Each column has sample ports located at the sites every 25 cm, and was fitted with water vessels to humidify the air entering the columns. The columns were housed in a chamber provided with temperature control, and the air was supplied at a constant pressure via a manifold. The experiment consists of seven sets of fermentation data taken for different air flux (180, 360, 450 and 540 ml/min) and different temperature (20 and 40^{o}C). The duration of the bioremediation process depends on the volume of the soil under treatment and the type and concentration of the contaminants in it. In our case 28 days are sufficient to degrade 60% of the contaminants which is considered sufficient for our experiment. The evolution of the hydrocarbon removal was evaluated from solid samples periodically extracted from the biopile for analysis of pH (potentiometer), humidity (gravimetric method), oxygen consumption and carbon dioxide production - by gas chromatography, Total Petroeum Hydrocarbons (TPH) - by infrared spectroscopy, following soxhlet extraction with dichloromethane (EPA Method 3540C).

Process Identification. The graphical results of the experimental neural biodegradation process identification are given on Fig. 8 a – for KF RNN learning, and on Fig. 8 b – for KF RNN generalization. The Input Learning Pattern (ILP) proposed is conformed by the: ILP(AF, TEMP, pH, HU, O2, CO2, TPH). The Output Learning Pattern (OLP) includes: OLP(pH, HU, O2, CO2, TPH). The KF RNN used for modeling and identification of the hydrocarbon degradation process in biopile system has seven inputs, twelve neurons in the hidden layer and five outputs. The number of neurons (twelve) in the hidden layer was determined in an experimental way, applying the rule of thumb and according to the Mean Square Error (MSE%) of learning. The learning algorithm is a version of the dynamic BP one, specially designed for this KF RNN topology. The described above learning algorithm is applied simultaneously to 7 degradation kinetic data sets (patterns), realized below different conditions of air flow and temperature in the ranges 180-540 mi/min and 25-50^{o}C,

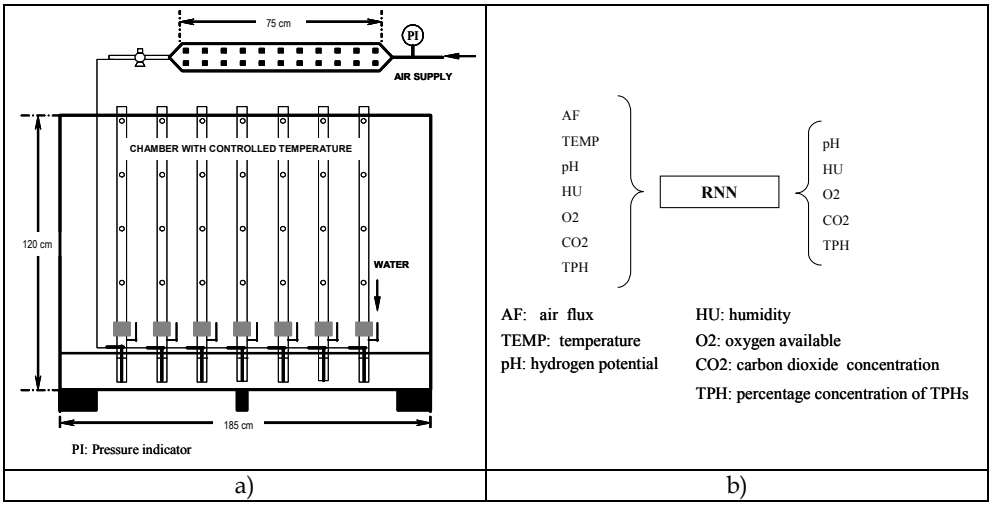


Fig. 7. a) Sketch of the biopile system; b) Learning pattern of the full KF RNN model

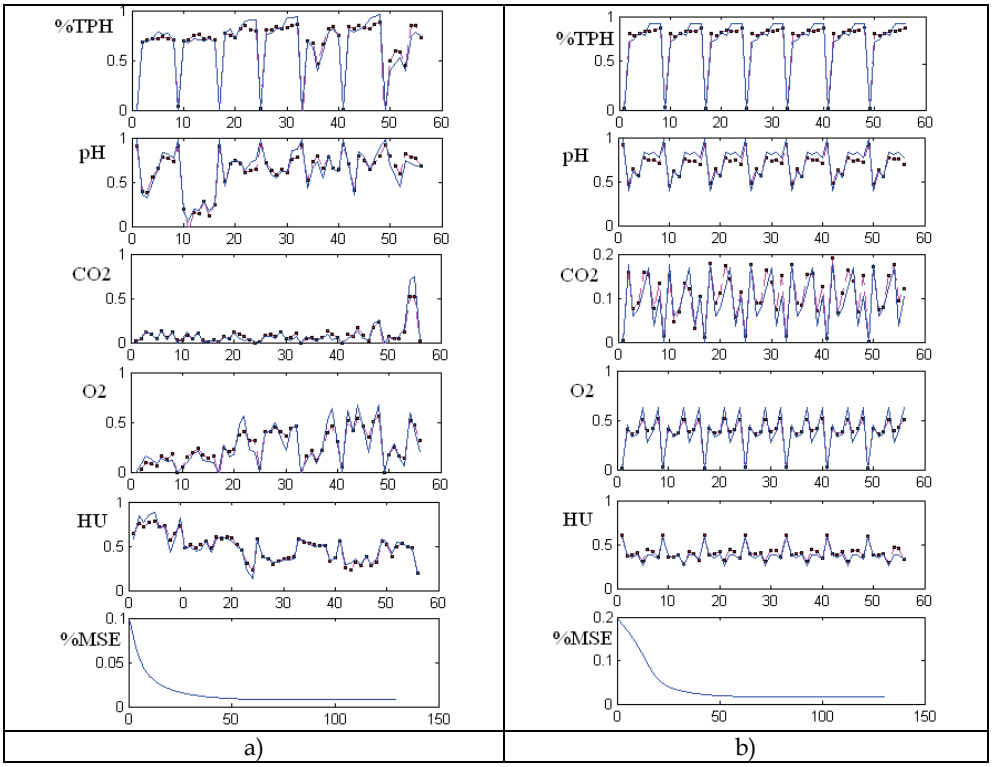


Fig. 8. Graphical results of experimental biodegradation process identification; a) graphical results of KF RNN learning (%TPH, pH, CO2, O2, HU, and MSE%); b) graphical results of KF RNN generalization (%TPH, pH, CO2, O2, HU, and MSE%)

and containing 8 points each one. The experimental data were normalized in the range 0-1 due to the great difference in magnitude between them. The 7 data sets are considered as an epoch of learning, containing 56 points. After each epoch of learning, the 7 pattern sets are interchanged in an arbitrary manner from one epoch to another. An unknown kinetic data set, repeated 7 times, is used as a generalization data set. The learning is stopped when the MSE% of learning and generalization reached values below 2%, and the relationship $|\Delta W_{ij}(k)|/|W_{ij}(k)|*100\%$ reached values below or equal of 2% for all updated parameters**.** This error was attained after 131 epochs of learning. The graphical results shown on Fig. 8 a. compared the experimental data for the 7 degradation kinetics with the outputs of the KF RNN during the last epoch of learning. The variables compared and plotted subsequently for the last epoch of learning are % degradation in TPH, pH, carbon dioxide (CO2), oxygen available (O2), % of humidity (HU) and the mean square error (MSE%) given for 131 epochs of learning. The learning rate is 0.9, the momentum rate is 0.8, the epoch size contains 56 points, the convergence is obtained after 131 epochs of learning. The final MSE% of learning is below 2%. The generalization of the KF RNN was carried out reproducing a degradation kinetics which is not included in the training process. This degradation process was carried out at AF = 360 ml/min and temperature of 20ºC. The operational conditions of this degradation process are in the range of operational conditions studied. The generalization results shown on Fig. 8 b. compare the experimental data for the one unknown degradation kinetics (repeated 7 times so to maintain the epoch size) with the output of the KF RNN. The same experimental data %TPH, pH, CO2, O2, HU, MSE% (continuous line) are compared with the KF RNN outputs (pointed line) and are plotted subsequently for the last epoch of generalization. The final MSE% of KF RNN generalization is below 2%.

Simulation Results Obtained with the Sliding Mode Control and the Direct Adaptive Neural Control. The graphical simulation results of the controlled system with both controls are given on Fig. 9a,b and the MSE% of control is given in Table 1, Table 2 for 20 runs of the control program (SMC and DANC) with data mixed with 10% measurement Gaussian noise with different variance for each run. A simplified RTNN process model, extracted from the complete KF RNN model has been used to design both control systems and to issue the state vector. The RTNN particular model used as a feedforward controller has 2 inputs or references (%TPH, CO2), two outputs as control signals (AF, HU) and 9 states. The RTNN feedback controller has the topology (12, 9, 2). The RTNN particular plant model has 2 inputs (AF, HU), two outputs (%TPH, CO2) and 12 states. In that reduced model, depending on the available measurements, the input and output patterns are chosen as: ILP(AF, HU, CO2, TPH); OLP(CO2, TPH). For both control schemes, the two system set points (continuous line) are compared with the two plant outputs (%TPH, CO2) (pointed line) and are plotted subsequently for seven sets of set point data. The control variables shown are: AF, HU. However the lost of water is pretended to be compensated by the wet saturated air flux with controlled humidity introduced, which could accelerate the bioremediation process in the biopile system. The obtained MSE% of control in the end of the process for both control schemes is below 1%. The behaviour of the control system in the presence of 10% white Gaussian noise (with different SEED parameter at each run) added to the plant outputs could be studied acumulating some statistics of the final MSE% (ξ_{av}) for multiple run of the control program (see Table 1 for SMC and Table 2 for DANC). The mean average cost for all runs (ε) of control, the standard deviation (σ) with respect to the mean value and the deviation (Δ) are computed by means of the following formulas:

No	1	2	3	4	5
MSE%	1.106	1.0035	1.001	1.0951	0.93454
No	6	7	8	9	10
MSE%	1.146	1.3214	1.225	1.4721	1.1206
No	11	12	13	14	15
MSE%	1.3185	1.1544	1.1821	1.0316	1.1267
No	16	17	18	19	20
MSE%	1.1295	1.3268	1.1842	1.2858	1.1993

Table 1. Final MSE (%) of control (ξ_{av}) for 20 runs of the SMC control program

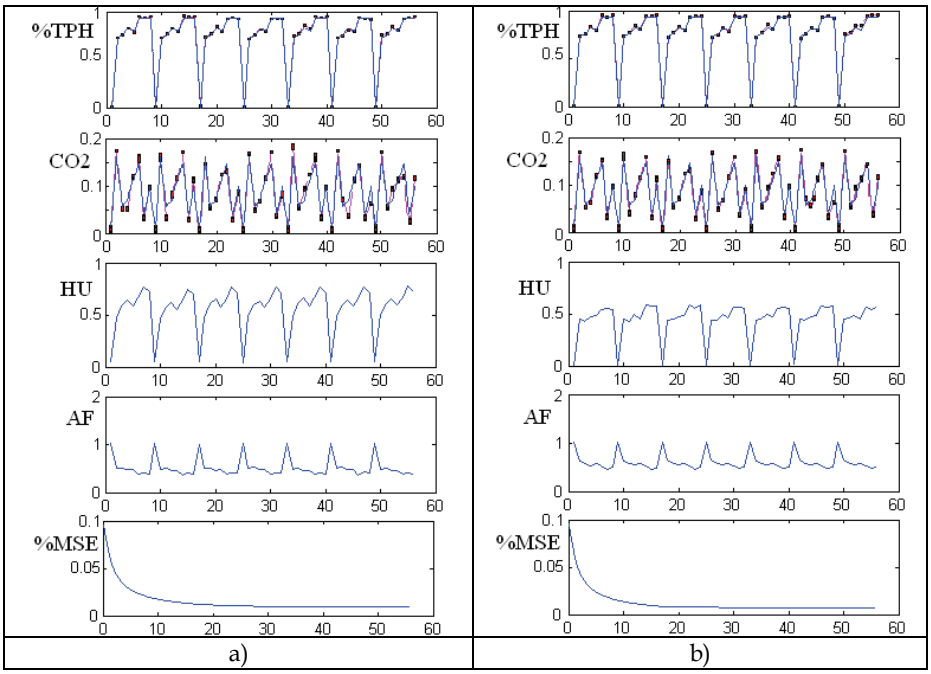


Fig. 9. a) Graphical results of the biodegradation process SMC; b) Graphical results of the biodegradation process DANC; for both schemes the variables shown are (%TPH, CO2, AF, HU, MSE%)

No	1	2	3	4	5
MSE%	0.9805	0.8207	1.0421	0.8148	0.8813
No	6	7	8	9	10
MSE%	0.8227	1.0959	0.8990	0.8100	1.0881
No	11	12	13	14	15
MSE%	1.0551	0.9569	0.8227	1.0619	1.0891
No	16	17	18	19	20
MSE%	1.0518	1.1173	0.8045	1.0454	1.1012

Table 2. Final MSE (%) of control (ξ_{av}) for 20 runs of the DANC control program

$$\varepsilon = \frac{1}{n}\sum_{k=1}^{n}\xi_{av_k}\ ;\ \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Delta_i^2}\ ;\ \Delta = \xi_{av} - \varepsilon \tag{71}$$

Where *k* is the run number and n is equal to 20. The mean and standard deviation values of process error, obtained for the SMC, are respectively: ε = 1.1682%; σ= 0.1276 %. The mean and standard deviation values of process control, obtained for DANC, are respectively: ε = 0.9680 %; σ= 0.0583 % which is a little bit smaller that the results, obtained for the SMC due to the nonlinearity and adaptivity of the DANC, which contained two RTNN controllers.

4.2 Simulation results obtained for the rotating drum

Description of the Process and the Experiment. For the bioremediation process one challenge is to provide enough O_2 and nutrients to enable rapid conversion of contaminants by either indigenous microorganisms or inoculated species (Alexander, 1994). Another challenge is to achieve efficient contact between the active micro-organisms and the contaminants, which may represented a problem with *in-situ* treatment. An attractive alternative to overcome this problem is to apply a biological treatment in slurry phase using Horizontal Rotating Drum (HRD) (see the schematic diagram given on Fig. 10a). The HRD can effectively mix heterogeneous blends of a wide range of particle sizes and high solid concentration (more than 60 %), (Alexander, 1994). The HRD operated with oxygen supply or aeration. Independently of the type of HRD operation (open or close), the insufficiency of water decreased the efficiency of hydrocarbon degradation in HRD favouring the formation of hydrocarbon balls (Alexander, 1999). So one objective of the process control is to maintain the humidity at 60%, which is the maximal solid concentration determined as the best for hydrocarbon removal from polluted soils treated in open rotating slurry bioreactors. Nowadays, semi empirical models, based on the Monod equation, have been developed to predict micro-organism growth as a function of available contaminants concentration. However, as the application of such models requires experimental work for calculating the kinetics parameters involved, so an alternative modelling technique is required. The KF RNN model offers many advantages as the possibility to approximate complex non linear high order multivariable processes, as the biodegradation process is. The bioremediation of polluted soils selected for modelling purpose was carried out by bio-stimulation in slurry phase using an open HRD. A silt loam (sand 36.5%w/w, silt 62.5% w/w and clay 1% w/w) soil of an average diameter of 15 μm, particle diameter in the range 2 - 75 μm, was used. The soil was contaminated with 50000 ppm of crude oil collected from a contaminated zone located near from a petroleum refinery. The slurry was prepared with 40% weight of soil (715 g) and 60% weight of a mineral solution (formula in $kg \cdot m^{-3}$: $(NH_4)_2SO_4$, 19; KH_2PO_4, 1.7; $MgSO_4$, 1; $CaCl_2 \cdot 2H_2O$, 0.005; $FeCl_3 \cdot 6H_2O$, 0.0025; yeast extract, 0.59; tergitol - 0.5%). The slurry was added to a HRD of 4 litters (13 cm diameter by 30 cm long), which was opened, on the flat faces, for a natural air supply (see Fig. 10a). The drum was operated during 19 days at a fix turning in the interval 3.5-20 RPM. During this time, the reactor was daily weighted in order to replace the water lost, so to maintain constant the water concentration. Samples were removed each day for analysis of residual hydrocarbons, pH, water concentration and slurry viscosity. The hydrocarbon concentration was determined by an infrared spectrometer; the pH was measured with a Beckman Φ potentiometer; water concentration was calculated by difference of two sequence data of the drum weight; finally, slurry viscosity was measured with an AND Vibro-viscometer SV-10 (MED BY A&D LTD).

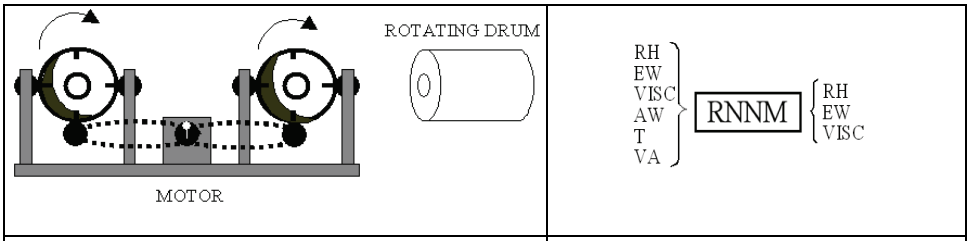


Fig. 10. a) Schematic diagram of a rotating drum system. b) Learning pattern

The biodegradation process was repeated at a different turning value (3.5, 5, 7.5, 10, 15, 20 RPM) in order to vary the oxygen available into the HRD. The learning pattern (see Fig. 10b) used for KF RNN training is composed by six input variables and three output variables. In order to avoid saturation problems in the RNNM training, the variables of the learning pattern are normalized in the interval 0-1. The measured variables are: Residual Hydrocarbon Concentration (RH), Evaporated Water (EW); Soil Viscosity (VISC), Added Water (AW); Temperature (T); Velocity of Agitation (VA). The RNNM outputs are: OUT (RH, EW, VISC). Depending on the available measurements and the control objectives, this model could be simplified, where the input- output pattern is chosen as: ILP (RH, EW, AW, VA); OLP (RH, EW). This reduced model will be used for SMC and DANC system design.

Process Identification. The described above learning algorithm is applied simultaneously to four fermentation kinetic data, represented by its input/output learning data patterns, and containing 19 points each (one per day). The total time of learning is 200 epochs, where the epoch size, corresponding to the number of data, is 76 iterations. After each epoch of training, the 4 sets are interchanged in an arbitrary manner from one epoch to another. The learning is stopped when the MSE% of learning reached values below 1.5%, the MSE% of generalization reached valued below 2%, and the relationship $|\Delta W_{ij}(k)|/|W_{ij}(k)|*100\%$ reached values below or equal of 2% for all updated parameters. Graphical results of RNNM training are given in Fig. 11a for the last epoch of learning. In the graphics, the output variables of the KF RNN are compared with the experimental data. The Fig. 11a compared the 4 kinetics experimental data with those, issued by the KF RNN. The output process data of 76 points are the hydrocarbon residual, the water requirements and the soil viscosity (EW, RH, VISC). The last figure in Fig. 11a represents the evolution of the mean squared error of approximation (MSE%) for whole learning process of 200 epochs. An unknown set of kinetic data, containing 19 points and repeated 4 times, so to maintain the same 76-points epoch size, is used as a validation (generalization) set, and these results are given on Fig. 11b. The obtained graphical results of KF RNN training and generalization shows a good convergence with an MSE% below 1.5% for the training and 2% for the generalization.

Simulation Results Obtained with the Sliding Mode Control and the Direct Adaptive Neural Control. A simplified RTNN process model extracted from the KFRNN complete identified model has been used to design SMC and DANC systems. The RTNN particular model has two inputs (AW, VA), two outputs (EW, RH) and nine states. The SMC SSEF is chosen as a first order one (P=1) with parameters Uo=1, γ=0.07, L=M=2. The DANC RTNN particular model used as a feedforward controller has two reference inputs (EW, RH), two outputs as control signals (VA, AW) and six states. The feedback RTNN controller has topology (12, 6, 2). The graphical simulation results of the controlled system outputs (EW,

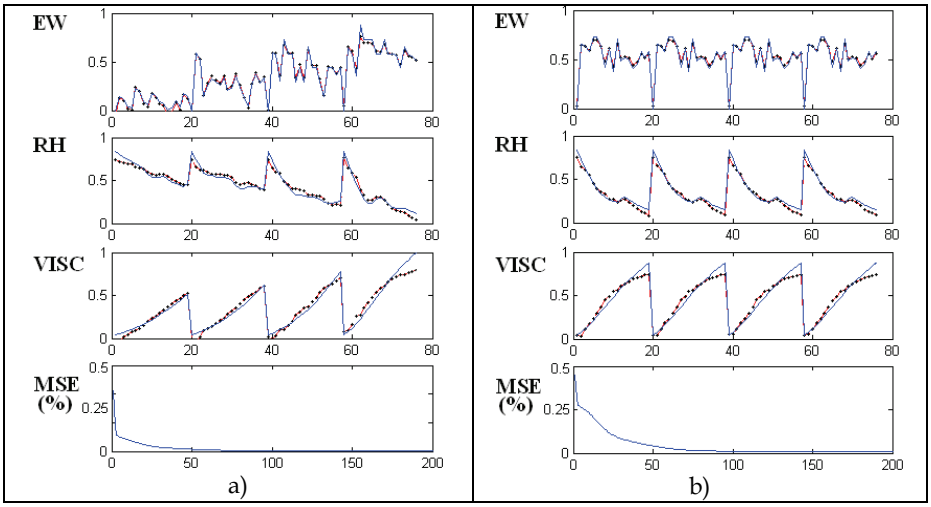


Fig. 11. Graphical results of experimental biodegradation process identification; a) graphical results of KF RNN learning (EW, RH, VISC, and MSE%.); b) graphical results of KF RNN generalization (EW, RH, VISC, and MSE%.)

RH), and the control variables (AW, VA) for both control schemes are given on Fig. 12 a,b for 76 points (one epoch of learning). For both control schemes, the two system set points (continuous line) are compared with the plant outputs (EW, RH) (data point line) and are plotted subsequently for four sets of set point data. The MSE% of control is given also in Fig. 12 a,b for all 200 epochs of learning. For both control schemes, the obtained MSE% of control at the end of the process is below 1%. The behaviour of the control system in the presence of 5% white Gaussian noise (with different SEED parameter at each run) added to the plant output has been studied accumulating some statistics of the final MSE% (ξ_{av}) for multiple run of the control program (SMC and DANC), which results are given on Table 3, Table 4 for 20 runs. The mean average cost for all runs (ε) of control, the standard deviation (σ) with respect to the mean value and the deviation (Δ) are computed using the formulas (71). The mean and standard deviation values of process error, obtained for the SMC are respectively: ε = 0.6663 %; σ= 0.0593 %. The mean and standard deviation values of process control, obtained for the DANC are respectively: ε = 0.5456 %; σ= 0.0124 %, which is slightly smaller with respect to the SMC, due to the nonlinearity and the adaptivity of the DANC, which contained two RTNN controllers.

5. Conclusion

The chapter proposes a new Kalman filter closed loop topology of recurrent neural network for identification and modeling of an unknown hydrocarbon degradation process carried out in a biopile system and a rotating drum. The proposed KF RNN contained a recurrent neural plant model, a recurrent neural output plant filter and posses global and local feedbacks. The learning algorithm is a modified version of the dynamic Backpropagation one derived using the adjoint KF RNN topology by means of the diagramatic method. The obtained KF RNN model issued parameters and states information appropriate for control systems design purposes.

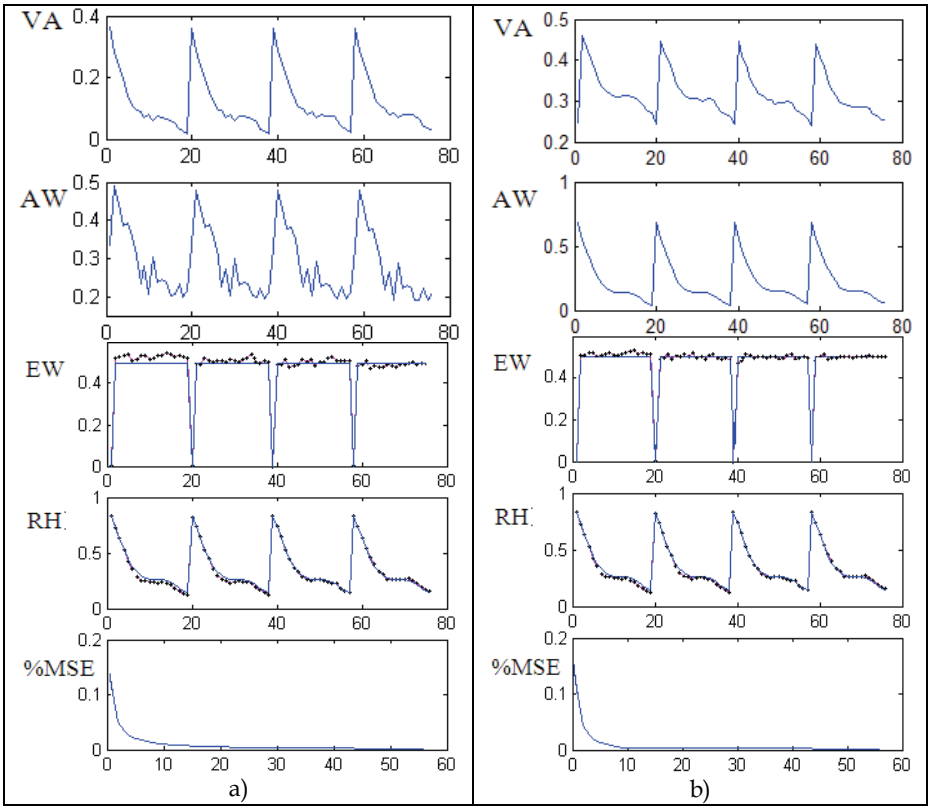


Fig. 12. a) Graphical results of the biodegradation process SMC; b) Graphical results of the biodegradation process DCD; for both schemes the variables shown are (VA, AW, EW, RH, MSE%)

No	1	2	3	4	5
MSE%	0.6434	0.6577	0.7669	0.6805	0.6662
No	6	7	8	9	10
MSE%	0.5757	0.5835	0.7043	0.7040	0.6350
No	11	12	13	14	15
MSE%	0.6602	0.7759	0.7732	0.6566	0.6408
No	16	17	18	19	20
MSE%	0.6481	0.6061	0.7240	0.6514	0.5725

Table 3. Final MSE (%) of control (ξ_{av}) for 20 runs of the control program

The obtained complete KF RNN model is simplified and used to design an indirect sliding mode control and a direct recurrent feedback-feedforward neural control. The simulation results obtained with the recurrent neural model learning and control exhibited a good convergence and precise reference tracking. The MSE% of the KF RNN learning and generalization is below 2% and the MSE% of the indirect and direct control is below 1%.

No	1	2	3	4	5
MSE%	0.5187	0.5449	0.5788	0.5738	0.5496
No	6	7	8	9	10
MSE%	0.5208	0.5732	0.5418	0.5672	0.5576
No	11	12	13	14	15
MSE%	0.5619	0.5040	0.5468	0.5471	0.5029
No	16	17	18	19	20
MSE%	0.5752	0.5744	0.5228	0.5065	0.5440

Table 4. Final MSE (%) of control (ξ_{av}) for 20 runs of the control program

Some statistical results of multiple run of the control program with noisy data, obtained with both control schemes are also given. The results show a slight priority of the DANC with respect to the SMC due to the better adaptation abilities to the first one.

6. References

Albertini, F. & Sontag, E. (1994). State observability in recurrent neural networks. *System and Control Letters,* Vol. 22, No 4, (April 1994) page numbers (235-244), ISSN 0167-6911

Alexander, M. (1999). *Biodegradation and Bioremediation*. Academic Press, ISBN 0-12-049861, New York

Baruch, I.S.; Stoyanov, I.P. & Gortcheva, E. (1996). Topology and learning of a class RNN, *ELEKTRIK (Turkish Journal of Electrical Engineering and Computer Sciences),* Vol. 4 Supplement, No 1, (January 1996) page numbers (35-42), ISSN: 1300-0632

Baruch, I.S.; Gortcheva, E. & Garrido, R. (1999). Redes neuronales recurrentes para la identificacion de objetos no lineales (in spanish). *Cientifica, (The Mexican Journal of Electromechanical Engineering, ESIME-IPN)*, Vol. 3, No 14, (March-April 1999) page numbers (39-45), ISSN 1665-0654

Baruch, I.S.; Flores, J.M.; Thomas, F. & Garrido, R. (2001). Adaptive neural control of nonlinear systems. In: *Artificial Neural Networks-ICANN 2001, LNCS 2130*, Dorffner, G., Bischof, H., Hornik, K. (Eds.), page numbers (930-936), Springer, ISBN 3-540-42486-5, Berlin

Baruch, I.S.; Flores, J.M. & Nenkova, B. (2001). Design of indirect adaptive neural control systems. *Cybernetics and Information Technologies (Bulgarian Academy of Sciences),* Vol. 1, No 1, (January 2001) page numbers (81- 94), ISSN 1311-9702

Baruch, I.S.; Flores, J.M.; Nava, F.; Ramirez, I.R. & Nenkova, B. (2002). An Advanced neural network topology and learning applied for identification and control of a D.C. motor, *Proceedings of the First International IEEE Symposium on Intelligent Systems*, pp. 289-295, ISBN 0-7803-7601-3, Varna Bulgaria, September 2002, IEEE Inc., New York

Baruch, I.S.; Barrera-Cortes, J. & Hernandez, L.A. (2004). A fed-batch fermentation process identification and direct adaptive neural control with integral term. In: *MICAI 2004: Advances in Artificial Intelligence, LNAI 2972,* Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (Eds.), page numbers (764-773), Springer-Verlag, ISBN 3-540-21459-3, Berlin Heidelberg New York

Baruch, I.S.; Georgieva, P.; Barrera-Cortes, J. & Feyo de Azevedo, S. (2005). 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, (O.Castillo, and P.Melin - guest editors), Vol. 20, No 2, (February 2005) page numbers (173-194), ISSN 0884-8173

Baruch, I.S.; Genina-Soto, P. & Barrera-Cortes, J. (2005). Predictive neural model of an osmotic dehydration process. *Journal of Intelligent Systems, Special Issue on Hybrid Intelligent Systems for Time Series Prediction*, *(O.Castillo, and P.Melin - guest editors),* Vol. 14, No 2-3, (February-March 2005) page numbers (143-155), ISSN 0334-1860

Baruch, I.S.; Mariaca-Gaspar, C.R. & Barrera-Cortes, J. (2007). Neural modelling and sliding mode control of bio-degradation process in a rotating bioreactor, In: *Preprints of the 8-th IFAC International Symposium on Dynamics and Control of Process Systems, DYCOPS*, vol. 2, Foss, B., Alvarez, J. (Eds.), pp. 261-266, Cancun Mexico, June 6-8, 2007, IFAC

Baruch, I.S.; Mariaca-Gaspar, C.R.; Cruz-Vega, I. & Barrera-Cortes, J. (2007). Sliding mode control of a hydrocarbon degradation in biopile system using recurrent neural network model, In: *MICAI 2007: Advances in Artificial Intelligence, LNAI 4827*, Gelbukh, A., Kuri-Morales, A.F. (Eds.), page numbers (1184-1194), Springer, ISBN-10 3-540-76630-8, Berlin Heidelberg New York

Boskovic, J.D. & Narendra, K. S. (1995). Comparison of linear, nonlinear and neural-network-based adaptive controllers for a class of fed-batch fermentation processes. *Automatica*, Vol. 31, No 6, (June 1995) page numbers (817-840), ISSN 0005-1098

Chen, S. & Billings, S.A. (1992). Neural networks for nonlinear dynamics system modeling and identification. *International Journal of Control,* Vol. 56, No 2, (August 1992) page numbers (319-346), ISSN 0020-7179

De la Torre-Sanchez, R.; Baruch, I.S. & Barrera –Cortes, J. (2006). Neural prediction of hydrocarbon degradation profiles developed in a biopile. *Expert Systems with Applications*, Vol. 31, No 3 (April 2006) page numbers (283-389), ISSN 0957-4174

Eduards, C.; Spurgeon, S.K. & Hebden, R.G. (2003). On the design of sliding mode output feedback controllers. *International Journal of Control, Special Issue Dedicated to Vadim Utkin on the Occasion of his 65th Birthday (Guest editor: Leonid M. Fridman),* Vol. 76, No 9/10, (15 June/10 July 2003) page numbers (893-905), ISSN 0020-7179

Flores, J. M.; Baruch, I. S. & Garrido, R. (2001). Red neuronal recurrente para identificación y control de sistemas no lineales (in spanish). *Científica, (The Mexican Journal of Electromechanical Engineering, ESIME-IPN)*, Vol. 5, No 1, (January-March 2001) page numbers (11-20), ISSN 1665-0654

Haykin, S. (1999). *Neural Networks, a Comprehensive Foundation. Second Edition.* Section 2.13, pp. 84-89; Section 4.13, pp. 208-213. Prentice-Hall, ISBN 0-13-273350-1, Upper Saddle River New Jersey

Hunt, K. J.; Sbarbaro, D.; Zbikowski, R., & Gawthrop, P. J. (1992). Neural network for control systems - a survey. *Automatica*, Vol. 28, No 6 (December 1992) page numbers (1083-1112), ISSN 0095-0963

Jin, L. & Gupta, M. (1999). Stable dynamic backpropagation learning in recurrent neural networks. *IEEE Transactions on Neural Networks*, Vol. 10, No 6, (November 1999) page numbers (1321-1334), ISSN 1045-9227

Kazemy, A.; Hosseini, S.A. & Farrokhi, M. (2007) Second order diagonal recurrent neural network, *Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE,* pp. 251-256, ISBN 978-1-4244-0755-2, 4-7 June 2007, Vigo, Spain, IEEE Inc., New York

Ku, C.C. & Lee, K.Y. (1995). Diagonal recurrent neural networks for dynamic systems control. *IEEE Transactions on Neural Networks*, Vol. 6, No 1, (January 1995) page numbers (144-156), ISSN 1045-9227

Levent, A. (2003). Higher order sliding modes, differentiation and output feedback control. *International Journal of Control, Special Issue Dedicated to Vadim Utkin on the Occasion of his 65th Birthday (Guest editor: Leonid M. Fridman),* Vol. 76, No 9/10, (15 June/10 July 2003) page numbers (924-941), ISSN 0020-7179

Mastorocostas, P.A. & Theocharis, J.B. (2006). A stable learning algorithm for block-diagonal recurrent neural networks: application to the analysis of lung sounds. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics,* Vol. 36, No 2 (April 2006) page numbers (242-254), ISSN 1083-4419

Miller, W.T., III; Sutton, R.S. & Werbos, P.J. (1990). *Neural Networks for Control*, A Bradford Book, MIT Press, ISBN 0-262-13261-3, Cambridge, Massachusetts, London, England

Narendra, K.S. & Parthasarathy, K. (1990). Identification and control of dynamic systems using neural networks. *IEEE Transaction on Neural Networks,* Vol. 1, No. 1, (January 1990) page numbers (4-27), ISSN 1045-9227

Narendra, K.S. & Mukhopadhyay, S. (1994). Adaptive control of nonlinear multivariable systems using neural networks. *Neural Networks*, Vol. 7, No 5, (May 1994) page numbers (737-752), ISSN 0893-6080

Nava, F.; Baruch, I.S.; Poznyak, A. & Nenkova, B. (2004). Stability proofs of advanced recurrent neural networks topology and learning. *Comptes Rendus (Proceedings of the Bulgarian Academy of Sciences)*, Vol. 57, No 1, (January 2004) page numbers (27-32), ISSN 0861-1459

Pao, S.A.; Phillips, S.M. & Sobajic, D. J. (1992). Neural net computing and intelligent control systems. *International Journal of Control*, *Special Issue on Intelligent Control,* Vol. 56, No 3, (August 1992) page numbers (263-289), ISSN 0020-7179

Sage, A.P. (1968) *Optimum Systems Control*. Prentice-Hall Inc., Library of Congress Catalog Number 68-20862, Englewood Cliffs, New Jersey

Sontag, E. & Sussmann, H. (1997). Complete controlability of continuos time recurrent neural network. *System and Control Letters*, Vol. 30, No 4, (May 1997) page numbers (177-183), ISSN 0167-6911

Su, Hong-Te; McAvoy, Th. J. & Werbos, P. (1992). Long-term predictions of chemical processes using recurrent neural networks: a parallel training approach. *Industrial Engineering Chemical Research*, Vol. 31, No 5, (May 1992) page numbers (1338-1352), ISSN 0888-5885

Valdez-Castro, L.; Baruch, I.S. & Barrera-Cortes, J. (2003). Neural networks applied to the prediction of fed-batch fermentation kinetics of Bacillus Thuringiensis. *Bioprocess and Biosystems Engineering*, Vol. 25, No 4, (January 2003) page numbers (229-233), ISSN: 1615-7591

Wan, E. & Beaufays, F. (1996). Diagrammatic method for deriving and relating temporal neural networks algorithms. *Neural Computation,* Vol. 8, No 2, (February 1996) page numbers (182-201), ISSN 0899-7667

Young, K.D.; Utkin, V.I. & Ozguner, U. (1999). A control engineer's guide to sliding mode control. *IEEE Trans. on Control Systems Technology*, Vol. 7, No 3, (May 1999) page numbers (328-342), ISSN 1063-6536

7. Appendix: Proof of the Theorem of RTNN Stability

Let the Recurrent Trainable Neural Network with Jordan Canonical Structure given by (1), (2), (3), (4), (5), (6), (7) and the nonlinear plant model as follows:

$$x(k+1)=g[x(k),u(k)] \tag{A.1}$$

$$y(k)=f[x(k)] \tag{A.2}$$

and the plant and activation functions fulfill the following assumptions:

Assumption 1: The plant dynamics is locally Lipchitz, so the functions $g(\cdot)$, $f(\cdot)$ are as:

$$f:=\{f \mid f=\sigma+\Delta f, \|\Delta f\| \le f_0+f_1\|x(k)\|\}$$

$$g:=\{g \mid g=\sigma+\Delta g, \|\Delta g\| \le g_0+g_1\|x(k)\|\}$$

and Δg , Δf are modeling errors, which reflex the effect of unmodeled dynamics.

Assumption 2: The activation function has the following Taylor approximation:

$$\sigma\left(\bar{\theta}\right)=\sigma(\theta)+\frac{\partial\sigma\left(\bar{\theta}\right)}{\partial\bar{\theta}}\left(\bar{\theta}-\theta\right)+\varsigma$$

with the approximation error bound given by:

$$\|\varsigma\|^2 \le \frac{L}{2}\left\|\bar{\theta}-\theta\right\|^2$$

and the signal error defined by:

$$e(k)=\hat{y}(k)-y(k)$$

$$e(k+1)=\hat{y}(k+1)-y(k+1)=F[C(k)\hat{x}(k)]-F\left[C^*\hat{x}(k)\right]-\Delta f[x(k)]$$

Now, let us define the state estimation error, add and subtract the RTNN to the last equation and apply the Assumption 2, then:

$$\Delta(k)=\hat{x}(k)-x(k)$$

$$\Delta(k+1)=\hat{x}(k+1)-x(k+1)=G[A(k)\hat{x}(k)+B(k)u(k)]-G\left[A^*\hat{x}(k)+B^*u(k)\right]-\Delta g[x(k),u(k)]$$

Let us now define the output identification error and put it in terms of the state estimation error as:

$$e(k+1)=F'(k)\delta C(k)[G'(k)(\delta A(k)\hat{x}(k)+\delta B(k)u(k))+\Theta_3]+F'(k)\delta C^*\left[G'(k)\left(A^*\Delta(k)-B^*O_F\right)+\Theta_4\right]$$
$$+\Theta_1+\Theta_2-\Delta f(x(k),\bar{u}(k))$$

Where: the term $\bar{u}(k)=u(k)+O_F$; the $\Theta_{1,2,3,4}$ are the higher order terms in the Taylor series approximation; $\Delta f(x(k),\bar{u}(k))$ is the unmodeled dynamics; O_F is an offset.

If Assumptions 1 and 2 fulfil, the learning algorithm for the RTNN is given by (8) and the learning parameters η_k, α_k are normalized and depended on the output error structure.

Then, the approximation error is bounded.

Consider a Lyapunov candidate function as

$$L(k)=L_1(k)+L_2(k) \tag{A.3}$$

In which $L_1(k)$ and $L_2(k)$ are given by:

$$L_1(k)=\frac{1}{2}e^2(k) \tag{A.4}$$

$$L_2(k)=\mathrm{tr}\left(\tilde{W}_A(k)\tilde{W}_A^T(k)\right)+\mathrm{tr}\left(\tilde{W}_B(k)\tilde{W}_B^T(k)\right)+\mathrm{tr}\left(\tilde{W}_C(k)\tilde{W}_C^T(k)\right) \tag{A.5}$$

Where:

$$\tilde{W}_{A1}(k)=\hat{A}(k)-A^*,\tilde{W}_B(k)=\hat{B}(k)-B^*,\tilde{W}_C(k)=\hat{C}(k)-C^*$$

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 consider the equation (A.4). The change of the Lyapunov function in two consecutive samples due of the training process is obtained by:

$$\Delta L_1(k)=L_1(k+1)-L_1(k)=\left[e(k+1)-e(k)\right]\left[e(k)+\tfrac{1}{2}e(k+1)-\tfrac{1}{2}e(k)\right] \tag{A.6}$$

Then, defining $\Delta e(k)$ as the difference between two consecutive error samples, the equation (A.6) becomes:

$$\Delta L_1(k)=\Delta e(k)\left[e(k)+\tfrac{1}{2}\Delta e(k)\right] \tag{A.7}$$

Where: $\Delta e(k)$ can be defined as:

$$\Delta e(k)=\left[\frac{\partial e(k)}{\partial W}\right]\Delta\vec{W} \tag{A.8}$$

Putting all weights into one vector as

$$\vec{W}=\left[\left[\vec{A}\right]^T\left[\vec{B}\right]^T\left[\vec{C}\right]^T\right]^T \tag{A.9}$$

Where:

$$\vec{A}=\left[\left[\vec{A}_1\right]^T\left[\vec{A}_2\right]^T L\left[\vec{A}_n\right]^T\right]^T,\vec{B}=\left[\left[\vec{B}_1\right]^T\left[\vec{B}_2\right]^T L\left[\vec{B}_m\right]^T\right]^T,\vec{C}=\left[\left[\vec{C}_1\right]^T\left[\vec{C}_2\right]^T L\left[\vec{C}_n\right]^T\right]^T$$

which represents the weight vectors constructed by their columns. Also let:

$$\eta=\begin{bmatrix}\eta^A & & \\ & \eta^B & \\ & & \eta^C\end{bmatrix},\quad \alpha=\begin{bmatrix}\alpha^A & & \\ & \alpha^B & \\ & & \alpha^C\end{bmatrix} \tag{A.10}$$

Where: (η^A,η^B,η^C) and $(\alpha^A,\alpha^B,\alpha^C)$ represented the learning rate matrix, the momentum rate matrix corresponding to (A,B,C), respectively, and $\eta^A=\eta_1 I_A$, $\eta^B=\eta_2 I_B$, $\eta^C=\eta_3 I_C$, $\alpha^A=\alpha_1 I_A$, $\alpha^B=\alpha_2 I_B$, $\alpha^C=\alpha_3 I_C$. Moreover, $\eta_i(i=1,...,3)$ and $\alpha_i(i=1,...,3)$ are two positive constants, and I_Z is an identity matrix with Z representing A,B,C, respectively. Now, we could define $\underline{\Delta}W$ and derive an expression for $\Delta L_1(k)$:

$$\underline{\Delta}W=\eta\Delta W(k)+\alpha\Delta W(k-1) \tag{A.11}$$

$$\begin{aligned}\underline{\Delta}W&=-\eta\frac{\partial L_1(k)}{\partial W}-\alpha\frac{\partial L_1(k-1)}{\partial W}=-e(k)\eta\frac{\partial e(k-1)}{\partial W}-e(k-1)\alpha\frac{\partial e(k-1)}{\partial W}\\&=-e(k)\begin{bmatrix}\eta^A & & \\ & \eta^B & \\ & & \eta^C\end{bmatrix}\times\left[\left[\frac{\partial e(k)}{\partial A}\right]^T\left[\frac{\partial e(k)}{\partial B}\right]^T\left[\frac{\partial e(k)}{\partial C}\right]^T\right]^T\\&\quad-e(k-1)\begin{bmatrix}\alpha^A & & \\ & \alpha^B & \\ & & \alpha^C\end{bmatrix}\times\left[\left[\frac{\partial e(k-1)}{\partial A}\right]^T\left[\frac{\partial e(k-1)}{\partial B}\right]^T\left[\frac{\partial e(k-1)}{\partial C}\right]^T\right]^T\end{aligned} \tag{A.12}$$

$$\begin{aligned}\Delta e(k)&=\left[\frac{\partial e(k)}{\partial W}\right]^T\underline{\Delta}W\\&=-e(k)\times\left(\eta_1\left\|\frac{\partial e(k)}{\partial A}\right\|^2+\eta_2\left\|\frac{\partial e(k)}{\partial B}\right\|^2+\eta_3\left\|\frac{\partial e(k)}{\partial C}\right\|^2\right)\\&\quad-e(k\text{-}1)\times\left(\alpha_1\left\|\frac{\partial e(k-1)}{\partial A}\right\|^2+\alpha_2\left\|\frac{\partial e(k-1)}{\partial B}\right\|^2+\alpha_3\left\|\frac{\partial e(k-1)}{\partial C}\right\|^2\right)\end{aligned} \tag{A.13}$$

$$\begin{aligned}\gamma&=\eta_1\left\|\frac{\partial e(k)}{\partial A}\right\|^2+\eta_2\left\|\frac{\partial e(k)}{\partial B}\right\|^2+\eta_3\left\|\frac{\partial e(k)}{\partial C}\right\|^2\\\lambda&=\alpha_1\left\|\frac{\partial e(k-1)}{\partial A}\right\|^2+\alpha_2\left\|\frac{\partial e(k-1)}{\partial B}\right\|^2+\alpha_3\left\|\frac{\partial e(k-1)}{\partial C}\right\|^2\end{aligned} \tag{A.14}$$

$$\Delta e(k+1) = -\gamma e(k+1) - \lambda e(k) \tag{A.15}$$

$$\begin{aligned}\Delta L_1(k+1) &= \Delta e(k+1)\left[e(k+1)+\tfrac{1}{2}\Delta e(k+1)\right] \\ &= -\tfrac{1}{2}e^2(k+1)\left[2\gamma-\gamma^2\right]+e(k+1)e(k)[\gamma-1]\lambda+\tfrac{1}{2}\lambda^2 e^2(k)\end{aligned} \tag{A.16}$$

Proposing: $\lambda=\gamma-1$, then:

$$\Delta L_1(k+1) = -\tfrac{1}{2}e^2(k+1)\left[-2\gamma^2+4\gamma-1\right]-\tfrac{1}{2}\lambda^2[\Delta e(k)]^2 \tag{A.17}$$

According to the Lyapunov's stability theory, the convergence could be be guaranteed, if $\Delta L(k+1)<0$, thus $-2\gamma^2+4\gamma-1>0$, and

$$\left(1-\frac{1}{\sqrt{2}}\right)<\gamma<\left(1+\frac{1}{\sqrt{2}}\right) \tag{A.18}$$

That is :

$$\left(1-\frac{1}{\sqrt{2}}\right)<\eta_1\left\|\frac{\partial e(k)}{\partial A}\right\|^2+\eta_2\left\|\frac{\partial e(k)}{\partial B}\right\|^2+\eta_3\left\|\frac{\partial e(k)}{\partial C}\right\|^2<\left(1+\frac{1}{\sqrt{2}}\right) \tag{A.19}$$

Let $\eta_{max}=\max_{i=1}^{3}\{\eta_i\}$; thus, as long as:

$$\frac{\left(1-\frac{1}{\sqrt{2}}\right)}{\left\|\frac{\partial e(k)}{\partial A}\right\|^2+\left\|\frac{\partial e(k)}{\partial B}\right\|^2+\left\|\frac{\partial e(k)}{\partial C}\right\|^2}<\eta_{max}<\frac{\left(1+\frac{1}{\sqrt{2}}\right)}{\left\|\frac{\partial e(k)}{\partial A}\right\|^2+\left\|\frac{\partial e(k)}{\partial B}\right\|^2+\left\|\frac{\partial e(k)}{\partial C}\right\|^2} \tag{A.20}$$

Note that $\|\cdot\|$ is the Euclidean norm, therefore:

$$\left\|\frac{\partial e(k)}{\partial A}\right\|^2+\left\|\frac{\partial e(k)}{\partial B}\right\|^2+\left\|\frac{\partial e(k)}{\partial C}\right\|^2=\left\|\frac{\partial e(k)}{\partial W}\right\|^2 \tag{A.21}$$

Now let : $\psi(k)=\frac{\partial e(k)}{\partial W}=-\frac{\partial y(k)}{\partial W}$ and $\psi_{max}=\max_k\|\psi(k)\|$, then:

$$\frac{\left(1-\frac{1}{\sqrt{2}}\right)}{\psi_{max}}<\eta_{max}<\frac{\left(1+\frac{1}{\sqrt{2}}\right)}{\psi_{max}} \tag{A.22}$$

Now, working with equation (A.5), we have:

$$L_2(k)=\mathrm{tr}\left(\tilde{W}_A(k)\tilde{W}_A^T(k)\right)+\mathrm{tr}\left(\tilde{W}_B(k)\tilde{W}_B^T(k)\right)+\mathrm{tr}\left(\tilde{W}_C(k)\tilde{W}_C^T(k)\right) \tag{A.23}$$

Considering the change of the Lyapunov function in two consecutive samples of the training process, and substituting the quantities: $\tilde{W}_{A(k)}=\hat{A}(k)-A^*$, $\tilde{W}_B(k)=\hat{B}(k)-B^*$, $\tilde{W}_C(k)=\hat{C}(k)-C^*$, we get:

$$\Delta L_2(k) = L_2(k+1)\text{-}L_2(k) = \mathrm{tr}\begin{pmatrix} \hat{A}(k+1)\hat{A}^T(k+1) - \hat{A}(k+1)A^{*T} - A^*\hat{A}^T(k+1) + A^*A^{*T} \\ -\hat{A}(k)\hat{A}^T(k) + \hat{A}(k)A^{*T}(k) + A^*\hat{A}^T(k) - A^*A^{*T} \end{pmatrix}$$
$$+\mathrm{tr}\begin{pmatrix} \hat{B}(k+1)\hat{B}^T(k+1) - \hat{B}(k+1)B^{*T} - B^*\hat{B}^T(k+1) + B^*B^{*T} \\ -\hat{B}(k)\hat{B}^T(k) + \hat{B}(k)B^{*T} + B^*\hat{B}^T(k) - B^*B^{*T} \end{pmatrix} \quad (A.24)$$
$$+\mathrm{tr}\begin{pmatrix} \hat{C}(k+1)\hat{C}^T(k+1) - \hat{C}(k+1)C^{*T} - C^*\hat{C}^T(k+1) + C^*C^{*T} \\ -\hat{C}(k)\hat{C}^T(k) + \hat{C}(k)C^{*T} + C^*\hat{C}^T(k) - C^*C^{*T} \end{pmatrix}$$

Applying the learning law (8) and the trace properties we obtained:

$$\begin{aligned} \Delta L_2(k) = {} & \eta_{\max}^2\left[\left\|\Delta\hat{A}(k)\right\|^2 + \left\|\Delta\hat{B}(k)\right\|^2 + \left\|\Delta\hat{C}(k)\right\|^2\right] + \alpha_{\max}^2\left[\left\|\Delta\hat{A}(k\text{-}1)\right\|^2 + \left\|\Delta\hat{B}(k\text{-}1)\right\|^2 + \left\|\Delta\hat{C}(k\text{-}1)\right\|^2\right] \\ & + 2\eta_{\max}\mathrm{tr}\left(\tilde{A}(k)\Delta\hat{A}^T(k) + \tilde{B}(k)\Delta\hat{B}^T(k) + \tilde{C}(k)\Delta\hat{C}^T(k)\right) \\ & + 2\alpha_{\max}\mathrm{tr}\left(\tilde{A}(k)\Delta\hat{A}^T(k\text{-}1) + \tilde{B}(k)\Delta\hat{B}^T(k\text{-}1) + \tilde{C}(k)\Delta\hat{C}^T(k\text{-}1)\right) \\ & + 2\alpha_{\max}\eta_{\max}\mathrm{tr}\left(\Delta\hat{A}(k)\Delta\hat{A}^T(k\text{-}1) + \Delta\hat{B}(k)\Delta\hat{B}^T(k\text{-}1) + \Delta\hat{C}(k)\Delta\hat{C}^T(k\text{-}1)\right) \end{aligned} \quad (A.25)$$

Substituting the learning values and errors (9)-(15), we obtained terms like:

$$2\eta_{\max}|e(k)|^2\text{-}2\eta_{\max}\xi(k)e(k)\ ;\ 2\eta_{\max}|e(k\text{-}1)|^2\text{-}2\eta_{\max}\xi(k\text{-}1)e(k\text{-}1) \quad (A.26)$$

Applying the following inequality: $X^TY + \left(X^TY\right)^T \le X^T\Lambda X + Y^T\Lambda^{-1}Y$, which is valid for any $X,Y \in \Re^{n\times m}$, and for any positive definite matrix $0 < \Lambda = \Lambda^T \in \Re^{n\times n}$, we obtained:

$$2\eta_{\max}e(k)\xi(k) = (\eta_{\max}e(k))\xi(k) + \xi(k)(\eta_{\max}e(k)) \le \eta_{\max}^2\|e(k)\|_{\Lambda_1}^2 + \|\xi(k)\|_{\Lambda_1^{-1}}^2\ ;$$

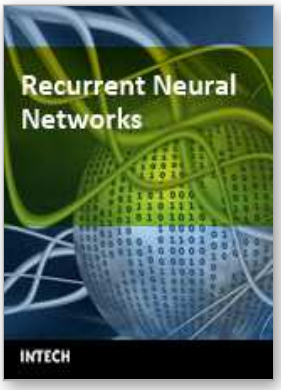
$$2\alpha_{\max}e(k\text{-}1)\xi(k\text{-}1) = (\alpha_{\max}e(k\text{-}1))\xi(k\text{-}1) + \xi(k\text{-}1)(\alpha_{\max}e(k\text{-}1)) \le \alpha_{\max}^2\|e(k\text{-}1)\|_{\Lambda_2}^2 + \|\xi(k\text{-}1)\|_{\Lambda_2^{-1}}^2 \quad (A.27)$$

Analyzing (A.27) term by term and applying the Rayleigh inequality: $\lambda_{\min}(\Lambda)\|x\|^2 \le x^T\Lambda x \le \lambda_{\max}(\Lambda)\|x\|^2$ we obtained a statement for $\Delta L_2(k)$. Making inner terms equal to one as in the unit circle condition for discrete time, at last we get the final condition:

$$\Delta L_2(k) \le -\eta_{\max}|e(k)|^2\text{-}\alpha_{\max}|e(k\text{-}1)|^2 + d(k) \quad (A.28)$$

$$d(k) = \|\xi(k)\|_{\Lambda_1^{-1}}^2 + \|\xi(k\text{-}1)\|_{\Lambda_2^{-1}}^2 \quad (A.29)$$

Where $d(k)$ represented the unmodeled dynamics and/or perturbations term. Applying the Rate of Convergence Lemma (Nava et al., 2004) for the result (A.28) we could conclude that: the $d(k)$ - term must be bounded by the weight matrices and the learning parameter in order to obtain the final result: $\Delta L_2(k) \in L_\infty$. As a consequence we get : $A_{(k)} \in L_\infty, B_{(k)} \in L_\infty, C_{(k)} \in L_\infty$. From equations (A.22) and (A.28) we easily could get the inequality (A.20). Therefore the boundedness of $L(k)$, $k \in Z_0^+$ is guaranteed.



Recurrent Neural Networks

Edited by Xiaolin Hu and P. Balasubramaniam

ISBN 978-953-7619-08-4
Hard cover, 400 pages
Publisher InTech
Published online 01, September, 2008
Published in print edition September, 2008

The concept of neural network originated from neuroscience, and one of its primitive aims is to help us understand the principle of the central nerve system and related behaviors through mathematical modeling. The first part of the book is a collection of three contributions dedicated to this aim. The second part of the book consists of seven chapters, all of which are about system identification and control. The third part of the book is composed of Chapter 11 and Chapter 12, where two interesting RNNs are discussed, respectively.The fourth part of the book comprises four chapters focusing on optimization problems. Doing optimization in a way like the central nerve systems of advanced animals including humans is promising from some viewpoints.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ieroham Baruch, Carlos Mariaca-Gaspar and Josefina Barrera-Cortes (2008). Recurrent Neural Network Identification and Adaptive Neural Control of Hydrocarbon Biodegradation Processes, Recurrent Neural Networks, Xiaolin Hu and P. Balasubramaniam (Ed.), ISBN: 978-953-7619-08-4, InTech, Available from: http://www.intechopen.com/books/recurrent_neural_networks/recurrent_neural_network_identification_and_adaptive_neural_control_of_hydrocarbon_biodegradation_pr

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821