We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Distributed Parameter Bioprocess Plant Identification and I-Term Control Using Decentralized Fuzzy-Neural Multi-Models

Ieroham Baruch, Rosalba Galvan-Guerra and Sergio-Miguel Hernandez M.
CINVESTAV-IPN, Mexico City, Department of Automatic Control,
Mexico

1. Introduction

In the last decade, the Computational Intelligence tools (CI), including Artificial Neural Networks (ANN) and Fuzzy Systems (FS), applying soft computing, became universal means for many applications. Because of their approximation and learning capabilities, the ANNs have been widely employed to dynamic process modeling, identification, prediction and control, (Boskovic & Narendra, 1995; Haykin, 1999; Bulsari & Palosaari, 1993; Deng & Li, 2003; Deng et al., 2005; Gonzalez-Garcia et al., 1998; Padhi & Balakrishnan, 2003; Padhi et al., 2001; Ray,1989). Many applications have been done for identification and control of biotechnological plants too, (Padhi et al., 2001). Among several possible neural network architectures the ones most widely used are the Feedforward NN (FFNN) and the Recurrent NN (RNN), (Haykin, 1999). The main NN 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. Also, a great boost has been made in the applied NN-based adaptive control methodology incorporating integral plus state control action in the control law, (Baruch et al., 2004; Baruch & Garrido, 2005; Baruch et al., 2007). The FFNN and the RNN have been applied for Distributed Parameter Systems (DPS) identification and control too. In (Pietil & Koivo, 1996), a RNN is used for system identification and process prediction of a DPS dynamics - an adsorption column for wastewater treatment of water contaminated with toxic chemicals. In (Deng & Li, 2003; Deng et al., 2005) a spectral-approximation-based intelligent modeling approach, including NNs for state estimation and system identification, is proposed for the distributed thermal processing of the snap curing oven DPS that is used in semiconductor packaging industry. In (Bulsari & Palosaari, 1993), it is presented a new methodology for the identification of DPS, based on NN architectures, motivated by standard numerical discretization techniques used for the solution of Partial Differential Equation (PDE). In (Padhi & Balakrishnan, 2003), an attempt is made to use the philosophy of the NN adaptive-critic design to the optimal control of distributed parameter systems. In (Padhi et al., 2001) the concept of proper orthogonal decomposition is used for the model reduction of DPS to form a reduced order lumped parameter problem. The optimal control problem is then solved in the time domain, in a state feedback sense, following the philosophy of adaptive critic NNs. The control solution is then mapped back to the spatial domain using the same basis functions. In (Pietil & Koivo, 1996), measurement data of an industrial process are

generated by solving the PDE numerically using the finite differences method. Both centralized and decentralized NN models are introduced and constructed based on this data. The models are implemented on FFNN using Backpropagation (BP) and Levenberg-Marquardt learning algorithms.

Similarly to the static ANNs, the fuzzy models could approximate static nonlinear plants where structural plant information is needed to extract the fuzzy rules, (Baruch et al., 2008a, Baruch et al., 2008b, Baruch et al., 2008c; Baruch & Galvan-Guerra, 2008; Baruch & Galvan-Guerra, 2009). The difference between them is that the ANN models are global models where training is performed on the entire pattern range and the FS models perform a fuzzy blending of local models space based on the partition of the input space. So, the aim of the neuro-fuzzy (fuzzy-neural) models is to merge both ANN and FS approaches so to obtain fast adaptive models possessing learning, (Baruch et al., 2008a). The fuzzy-neural networks are capable of incorporating both numerical data (quantitative information) and expert's knowledge (qualitative information), and describe them in the form of linguistic IF-THEN rules. During the last decade considerable research has been devoted towards developing recurrent neuro-fuzzy models, summarized in (Baruch et al., 2008a). To reduce the number of IF-THAN rules, the hierarchical approach could be used (Baruch et al., 2008a). A promising approach of recurrent neuro-fuzzy systems with internal dynamics is the application of the Takagi-Sugeno (T-S) fuzzy rules with a static premise and a dynamic function consequent part, (Baruch et al., 2008a). The paper of (Baruch et al., 2008a) proposed as a dynamic function in the consequent part of the T-S rules to use a Recurrent Neural Network Model (RNNM).

Some results of this RNNM approach for centralized and decentralized identification of dynamic plants with distributed parameters are given in (Baruch et al., 2008a; Baruch et al., 2008b; Baruch et al., 2008c; Baruch & Galvan-Guerra, 2008; Baruch & Galvan-Guerra, 2009). The difference between the used in the other papers fuzzy neural model and the approach used in (Baruch et al., 2008a) is that the other one used the Frasconi, Gori and Soda RNN model, which is sequential one, and in (Baruch et al., 2008a), it is used the RTNN model, which is completely parallel one. But it is not still enough because the neural nonlinear dynamic function ought to be learned, and the Backpropagation learning algorithm is not introduced in the T-S fuzzy rule. For this reason in (Baruch et al., 2008a) the RTNN BP learning algorithm (Baruch et al., 2008d) has been introduced in the antecedent part of the IF-THAN rule so to complete the learning procedure and a second hierarchical defuzzyfication BP learning level has been formed so to improve the adaptation and approximation ability of the fuzzy-neural system, (Baruch et al., 2008a). This system has been successfully applied for identification and control of complex nonlinear plants, (Baruch et al., 2008a).

The aim of this chapter is to describe the results obtained by this system for decentralized identification and control of wastewater treatment anaerobic digestion bioprocess representing a Distributed Parameter System (DPS), extending the used control laws with an integral term, so to form an integral plus state control action, capable to speed up the reaction of the control system and to augment its resistance to process and measurement noises. The analytical anaerobic bioprocess plant model (Aguilar-Garnica et al., 2006), used as an input/output plant data generator, is described by PDE/ODE, and simplified using the orthogonal collocation technique, (Bialecki & Fairwether, 2001), in four collocation points and a recirculation tank. This measurement points are used as centres of the membership functions of the fuzzyfied space variables of the plant.

2. Description of the direct decentralized fuzzy-neural control with I-term

The block-diagrams of the complete direct Fuzzy-Neural Multi-Model (FNMM) control system and its identification and control parts are schematically depicted in Fig.1, Fig. 2 and Fig. 3. The structure of the entire control system, (Baruch et al., 2008a; Baruch et al., 2008b; Baruch et al., 2008c) contained Fuzzyfier, Fuzzy Rule-Based Inference System (FRBIS), and defuzzyfier. The FRBIS contained five identification, five feedback control, five feedforward control, five I-term control, five total control T-S fuzzy rules (see Fig. 1, 2, 3 for more details).
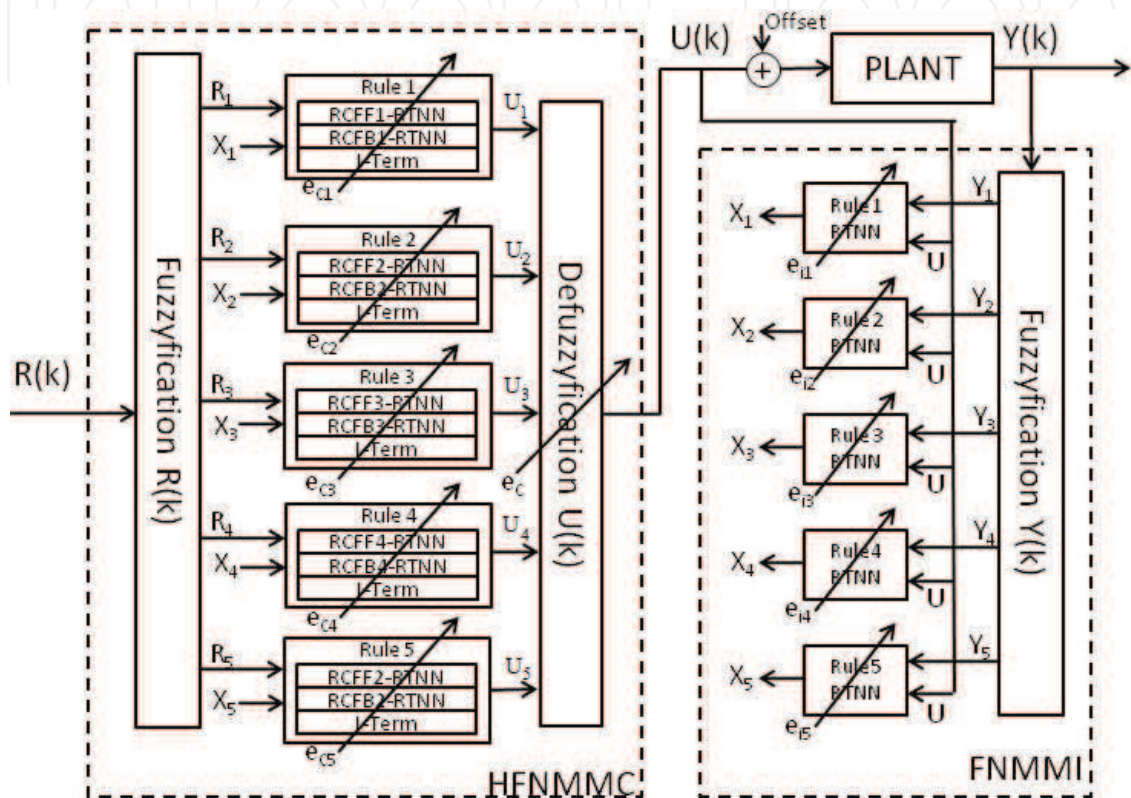

Fig. 1. Block-Diagram of the FNMM Control System

The plant output variables and its correspondent reference variables depended on space and time. They are fuzzyfied on space and represented by five membership functions which centers are the five collocation points of the plant (four points for the fixed bed and one point for the recirculation tank). The main objective of the Fuzzy-Neural Multi-Model Identifier (FNMMI), containing five rules, is to issue states and parameters for the direct adaptive Fuzzy-Neural Multi-Model Feedback Controller (FNMMFBC) when the FNMMI outputs follows the outputs of the plant in the five measurement (collocation) points with minimum error of approximation. The control part of the system is a direct adaptive Fuzzy-Neural Multi-Model Controller (FNMMC). The objective of the direct adaptive FNMM controller, containing five Feedback (FB), five Feedforward (FF) T-S control rules, five I-term control rules, and five total control rules is to speed up the reaction of the control system, and to augment the resistance of the control system to process and measurement noises, reducing the error of control, so that the plant outputs in the five measurement points tracked the corresponding reference variables with minimum error of tracking.

The upper hierarchical level of the FNMM control system is one- layer- perceptron which represented the defuzzyfier, (Baruch et al., 2008a). The hierarchical FNMM controller has

two levels – Lower Level of Control (LLC), and Upper Level of Control (ULC). It is composed of three parts (see Fig. 3): 1) Fuzzyfication, where the normalized reference vector signal contained reference components of five measurement points; 2) Lower Level Inference Engine, which contained twenty five T-S fuzzy rules (five rules for identification and twenty rules for control- five in the feedback part, five in the feedforward part, five in the I-term part, and five total control rules), operating in the corresponding measurement points; 3) Upper Hierarchical Level of neural defuzzification.

The detailed block-diagram of the FNMMI (see Fig. 2), contained a space plant output fuzzyfier and five identification T-S fuzzy rules, labeled as RI_i, which consequent parts are RTNN learning procedures, (Baruch et al, 2008 a). The identification T-S fuzzy rules have the form:

$$RI_i\text{: } \textbf{\textit{If}}\ x(k)\ \textbf{\textit{is}}\ A_i\ \textbf{\textit{and}}\ u(k)\ \textbf{\textit{is}}\ B_i\ \textbf{\textit{then}}\ Y_i = \Pi_i\ (L,M,N_i,Y_{di},U,X_i,A_i,B_i,C_i,E_i),\ i=1\text{-}5. \qquad (1)$$
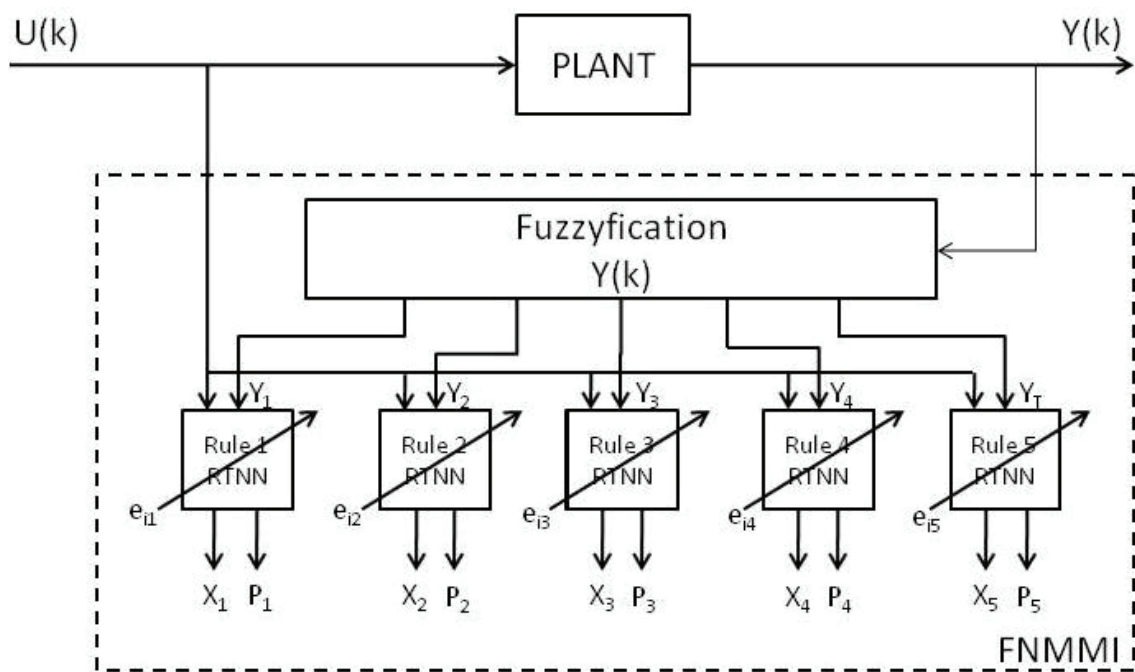

Fig. 2. Detailed block-diagram of the FNMM identifier

The detailed block-diagram of the FNMMC, given on Fig. 3, contained a spaced plant reference fuzzyfier and twenty control T-S fuzzy rules (five FB, five FF, five I-term, and five-total control), which consequent FB, and FF parts are also RTNN learning procedures, (Baruch et al., 2008a), using the state information, issued by the corresponding identification rules. The consequent part of each feedforward control rule (the consequent learning procedure) has the M, L, N_i RTNN model dimensions, R_i, Y_{di}, E_{ci} inputs and U_{ffi}, outputs used by the total control rule. The T-S fuzzy rule has the form:

$$RCFF_i\text{: } \textbf{\textit{If}}\ R(k)\ \textbf{\textit{is}}\ B_i\ \textbf{\textit{then}}\ Uff_i = \Pi_i\ (M,\ L,\ N_i,\ R_i,\ Y_{di},\ X_i,\ J_i,\ B_i,\ C_i,\ E_{ci}),\ i=1\text{-}5. \qquad (2)$$

The consequent part of each feedback control rule (the consequent learning procedure) has the M, L, N_i RTNN model dimensions, Y_{di}, X_i, E_{ci} inputs and U_{fbi}, outputs used by the total control rule. The T-S fuzzy rule has the form:

$$RCFB_i\text{: } \textbf{\textit{If}}\ Y_{di}\ \textbf{\textit{is}}\ A_i\ \textbf{\textit{then}}\ Ufb_i = \Pi_i\ (M,\ L,\ N_i,\ Y_{di},\ X_i,\ X_{ci},\ J_i,\ B_i,\ C_i,\ E_{ci}),\ i=1\text{-}5. \qquad (3)$$
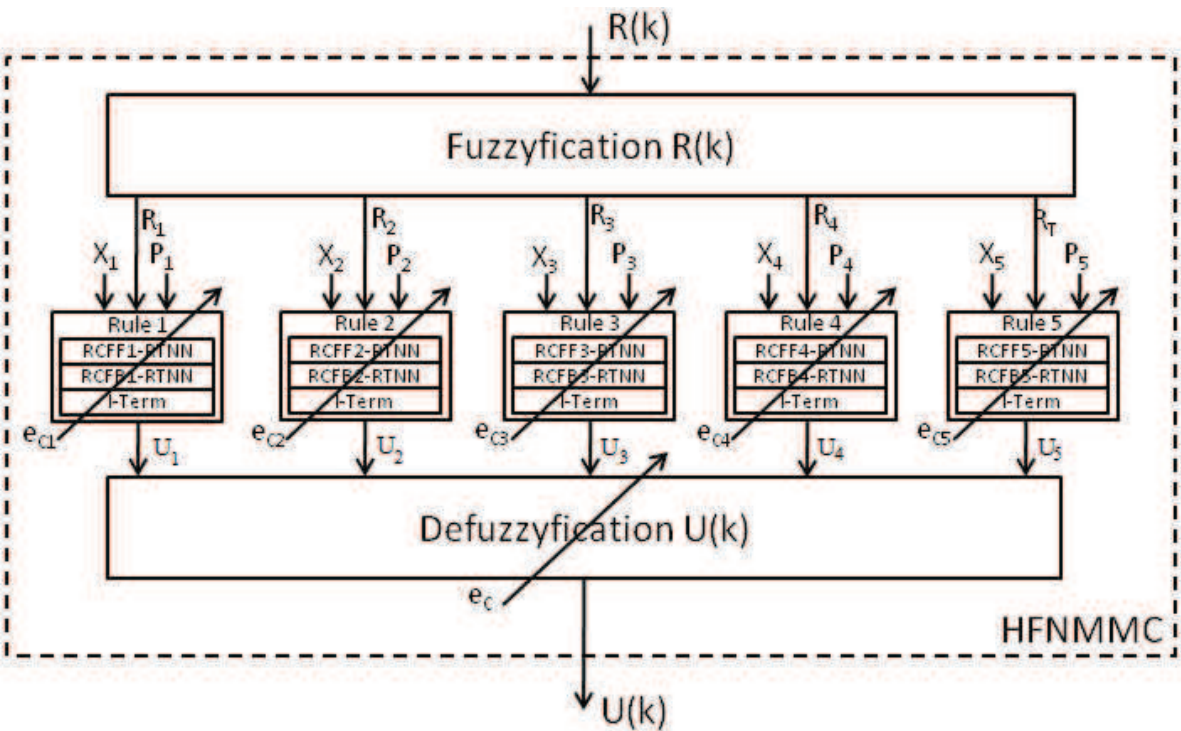
Fig. 3. Detailed block-diagram of the HFNMM controller

The I-term control algorithm is as follows:

$$UIt_i (k+1) = UIt_i (k) + To\, K_i (k)\, E_{ci} (k),\ i=1\text{-}5; \tag{4}$$

where To is the period of discretization and K_i is the I-term gain. An appropriate choice for the I-term gain K_i is a proportion of the inverse input/output plant gain, i.e.:

$$K_i (k) = \eta\, (C_i B_i)+. \tag{5}$$

The product of the pseudoinverse $(C_i B_i)+$ by the output error $E_{ci} (k)$ transormed the output error in input error which equates the dimensions in the equation of the I-term control. The T-S rule, generating the I-term part of the control executed both equations (4), (5), representing a computational procedure, given by:

$$RCIt_i\text{: } \textbf{\textit{If}}\ Y_{di}\ \textbf{\textit{is}}\ A_i\ \textbf{\textit{then}}\ UIt_i = \Pi_i (M, L, B_i, C_i, E_{ci}, To, \eta),\ i=1\text{-}5. \tag{6}$$

The total control corresponding to each of the five measurement points is a sum of its corresponding feedforward, feedback, and I-term parts, as:

$$U_i (k) = -U_{ffi} (k) + U_{fbi} (k) + U_{Iti} (k),\ i=1\text{-}5. \tag{7}$$

The total control is generated by the procedure (7) incorporated in the T-S rule:

$$RC_i\text{: } \textbf{\textit{If}}\ Y_{di}\ \textbf{\textit{is}}\ A_i\ \textbf{\textit{then}}\ U_i = \Pi_i (M, U_{ffi}, U_{fbi}, U_{Iti}),\ i=1\text{-}5. \tag{8}$$

The defuzzyfication learning procedure, which correspond to the single layer perceptron learning is described by:

$$U = \Pi (M, L, N, Y_d, Uo, X, A, B, C, E). \tag{9}$$

The T-S rule and the defuzzification of the plant output of the fixed bed with respect to the space variable z ($\lambda_{i,z}$ is the correspondent membership function), are given by:

$$RO_i: \textbf{\textit{If}} \ Y_{i,t} \ \textbf{\textit{is}} \ A_i \ \textbf{\textit{then}} \ Y_{i,t} = a_i{}^T Y_t + b_i, \ i=1,2,3,4; \tag{10}$$

$$Y_z = [\Sigma_i \ \gamma_{i,z} \ a_i{}^T] \ Y_t + \Sigma_i \ \gamma_{i,z} \ b_i \ ; \ \gamma_{i,z} = \lambda_{i,z} \ / \ (\Sigma_j \ \lambda_{j,z}). \tag{11}$$

The direct adaptive neural control algorithm, which appeared in the consequent part of the local fuzzy control rule RCFBi, (3) is a feedback control, using the states issued by the correspondent identification local fuzzy rule RIi (1).

3. Description of the indirect (sliding mode) decentralized fuzzy-neural control with I-term

The block-diagram of the FNMM control system is given on Fig.4. The structure of the entire control system, (Baruch et al., 2008a; Baruch et al., 2008b; Baruch et al., 2008c), contained Fuzzyfier, Fuzzy Rule-Based Inference System, containing twenty T-S fuzzy rules (five identification, five sliding mode control, five I-term control, five total control rules), and a defuzzyfier. Due to the learning abilities of the defuzzifier, the exact form of the control membership functions is not need to be known. The plant output variable and its correspondent reference variable depended on space and time, and they are fuzzyfied on space. The membership functions of the fixed-bed output variables are triangular or trapezoidal ones and that - belonging to the output variables of the recirculation tank are singletons. Centers of the membership functions are the respective collocation points of the plant. The main objective of the FNMM Identifier (FNMMI) (see Fig. 2), containing five T-S rules, is to issue states and parameters for the indirect adaptive FNMM Controller (FNMMC) when the FNMMI outputs follows the outputs of the plant in the five measurement (collocation) points with minimum MSE of approximation.

The objective of the indirect adaptive FNMM controller, containing five Sliding Mode Control (SMC) rules, five I-term rules, and five total control rules is to reduce the error of control, so that the plant outputs of the four measurement points tracked the corresponding reference variables with minimum MSE%. The hierarchical FNMM controller (see Fig. 5) has two levels – Lower Level of Control (LLC), and Upper Level of Control (ULC). It is composed of three parts: 1) Fuzzyfication, where the normalized reference vector signal contained reference components of five measurement points; 2) Lower Level Inference Engine, which contained twenty T-S fuzzy rules (five rules for identification, five rules for SM control, five rules for I-term control, and five rules for total control), operating in the corresponding measurement points; 3) Upper Hierarchical Level of neural defuzzification, represented by one layer perceptron, (Baruch et al., 2008a) . The detailed block-diagram of the FNMMI, given on Fig. 2, contained a space plant output fuzzyfier and five identification T-S fuzzy rules, labeled as RI$_i$, which consequent parts are learning procedures, (Baruch et al., 2008a), given by (1). The block-diagram of the FNMMC, given on Fig. 5, contained a spaced plant reference fuzzyfier, five SMC, five I-term control, and five total control T-S fuzzy rules. The consequent parts of the SMC T-S fuzzy rules are SMC procedures, (Baruch et al, 2008 a), using the state, and parameter information, issued by the corresponding identification rules. The SMC T-S fuzzy rules have the form:

$$RC_i: \textbf{\textit{If}} \ R(k) \ \textbf{\textit{is}} \ C_i \ \textbf{\textit{then}} \ U_i = \Pi_i \ (M, L, N_i, R_i, Y_{di}, X_i, A_i, B_i, C_i, E_{ci}), \ i=1-5. \tag{12}$$
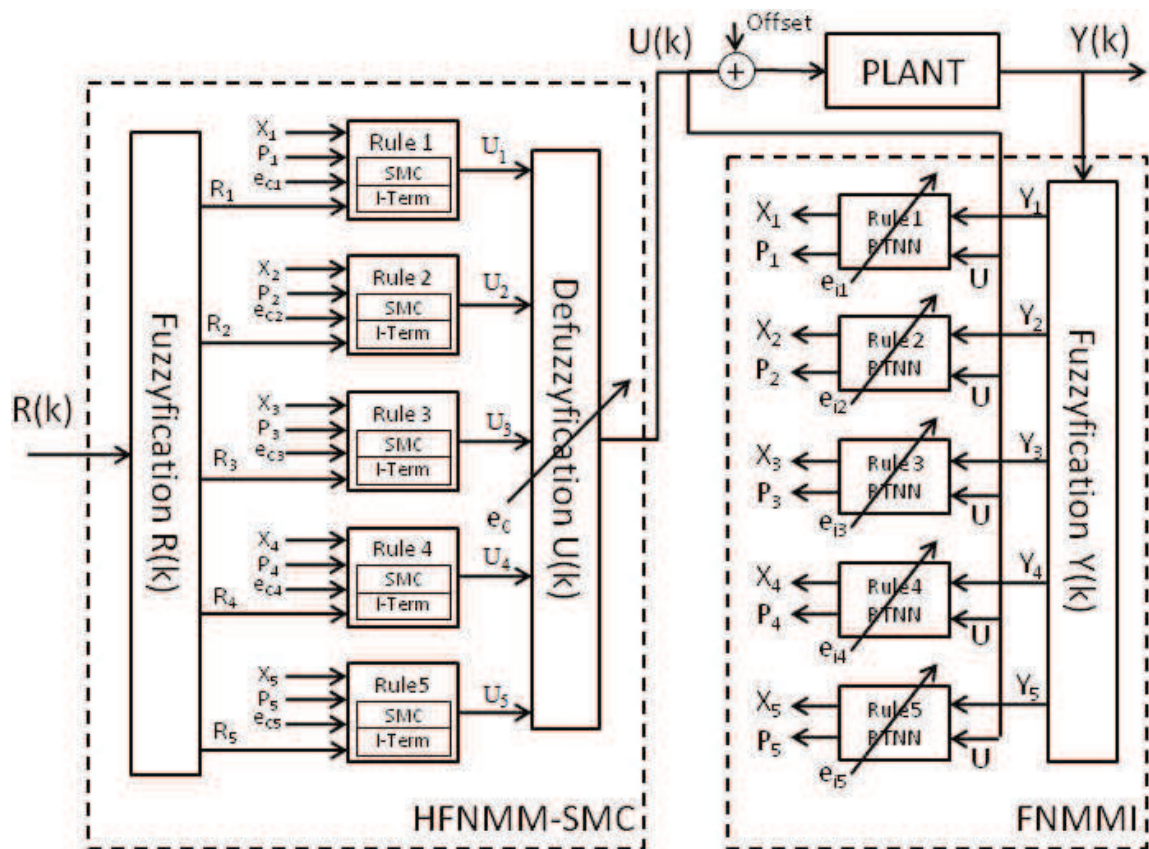
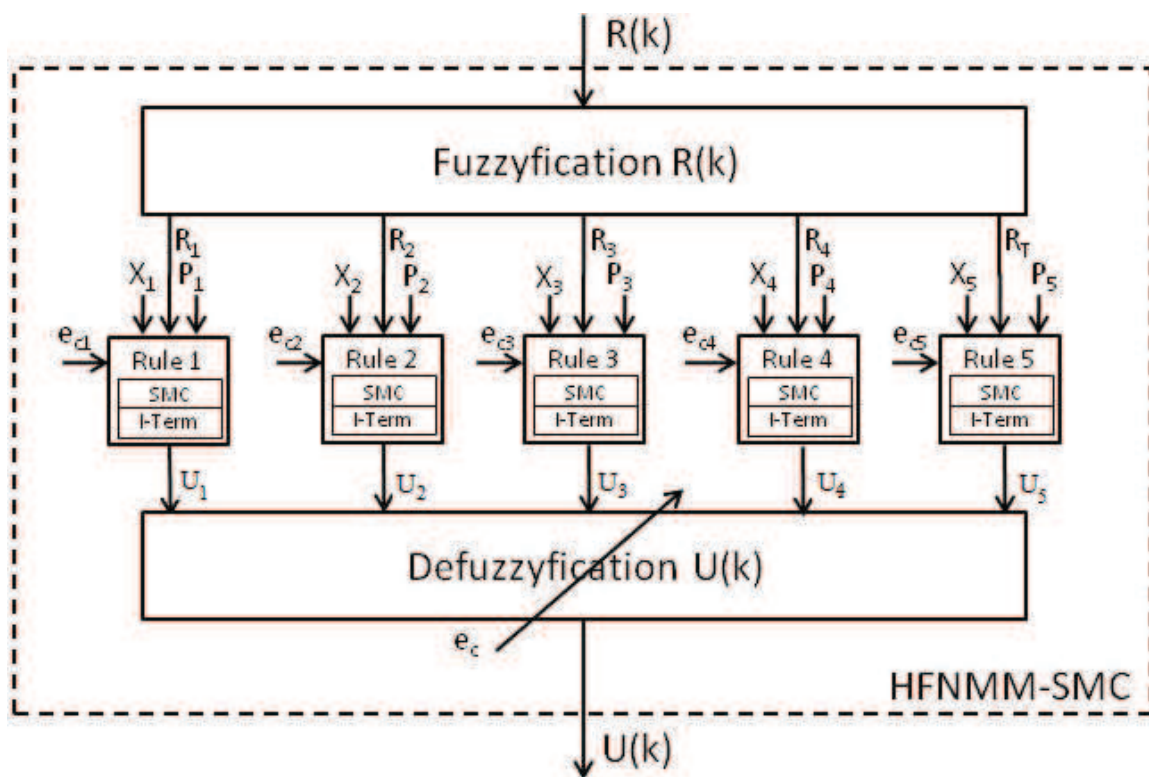Fig. 4. Block-diagram of the FNMM control system

Fig. 5. Detailed block-diagram of the HFNMM controller

The I-term control algorithm and its corresponding T-S fuzzy rule are given by (4), (5), (6). The total control corresponding to each of the five measurement points is a sum of its corresponding SMC and I-term parts, as:

$$U_i(k) = U_{smci}(k) + U_{Iti}(k), \quad i=1-5. \tag{13}$$

The total control is generated by the procedure (13) incorporated in the T-S rule:

$$RC_i: \text{If } Y_{di} \text{ is } A_i \text{ then } U_i = \Pi_i(M, U_{smci}, U_{Iti}), \quad i=1-5. \tag{14}$$

The defuzzyfication learning procedure, which correspond to the single layer perceptron learning is described by (9), (10), (11).
Next the indirect SMC procedure will be briefly described.

3.1 Sliding mode control system design

Here the indirect adaptive neural control algorithm, which appeared in the consequent part of the local fuzzy control rule RCi (12) is viewed as a Sliding Mode Control (SMC), (Baruch et al., 2008a; Baruch et al., 2008d), designed using the parameters and states issued by the correspondent identification local fuzzy rule RIi (1), approximating the plant in the corresponding collocation point.
Let us suppose that the studied local nonlinear plant model possess the following structure:

$$X_P(k+1)=F[X_P(k),-U_P(k)]; \ Y_P(k)=G[X_P(k)], \tag{15}$$

where: $X_P(k)$, $Y_P(k)$, $U(k)$ are plant state, output and input vector variables with dimensions N_p, L and M, where L>M (rectangular system) is supposed; F and G are smooth, odd, bounded nonlinear functions. The linearization of the activation functions of the local learned identification RTNN model, which approximates the plant leads to the following linear local plant model:

$$X(k+1)=AX(k)+BU(k); \ Y(k)=CX(k); \tag{16}$$

where L > M (rectangular system), is supposed. Let us define the following sliding surface with respect to the output tracking error:

$$S(k+1) = E(k+1) + \sum_{i=1}^{P} \gamma_i E(k-i+1); \quad |\gamma_i| < 1; \tag{17}$$

where: S(·) is the sliding surface error function; E(·) is the systems local output tracking error; γ_i are parameters of the local desired error function; P is the order of the error function. The additional inequality in (17) is a stability condition, required for the sliding surface error function. The local tracking error is defined as:

$$E(k) = R(k) - Y(k); \tag{18}$$

where R(k) is a L-dimensional local reference vector and Y(k) is an local output vector with the same dimension. The objective of the sliding mode control systems design is to find a control action which maintains the systems error on the sliding surface assuring that the output tracking error reached zero in P steps, where P<N, which is fulfilled if:

$$S(k+1) = 0. \tag{19}$$

As the local approximation plant model (16), is controllable, observable and stable, (Baruch et al., 2004; Baruch et al., 2008d), the matrix A is block-diagonal, and L>M (rectangular system is supposed), the matrix product (CB) is nonsingular with rank M, and the plant states X(k) are smooth non- increasing functions. Now, from (16)-(19), it is easy to obtain the equivalent control capable to lead the system to the sliding surface which yields:

$$U_{eq}(k) = (CB)^+ \left[-CAX(k) + R(k+1) + \sum_{i=1}^{P} \gamma_i E(k-i+1) \right] + Of, \tag{20}$$

$$(CB)^+ = \left[(CB)^T (CB) \right]^{-1} (CB)^T. \tag{21}$$

Here the added offset Of is a learnable M-dimensional constant vector which is learnt using a simple delta rule (see Haykin, 1999, for more details), where the error of the plant input is obtained backpropagating the output error through the adjoint RTNN model. An easy way for learning the offset is using the following delta rule where the input error is obtaned from the output error multiplying it by the same pseudoinverse matrix, as it is:

$$Of(k+1) = Of(k+1) = Of(k) + \eta(CB)^+ E(k). \tag{22}$$

If we compare the I-term expression (4), (5) with the Offset learning (22) we could see that they are equal which signifyed that the I-term generate a compensation offset capable to eliminate steady state errors caused by constant perturbations and discrepances in the reference tracking caused by non equal input/output variable dimensions (rectangular case systems). So introducing an I-term control it is not necessary to use an compensation offset in the SM control law (20).

The SMC avoiding chattering is taken using a saturation function inside a bounded control level Uo, taking into account plant uncertainties. So the SMC has the form:

$$U(k) = \begin{cases} U_{eq}(k), & \text{if } \|U_{eq}(k)\| < U_0 \\ \dfrac{-U_0 U_{eq}(k)}{\|U_{eq}(k)\|}, & \text{if } \|U_{eq}(k)\| \geq U_0. \end{cases} \tag{23}$$

The proposed SMC cope with the characteristics of the wide class of plant model reduction neural control with reference model, and represents an indirect adaptive neural control, given by (Baruch et al., 2004). Next we will give description of the used RTNN topology and learning.

4. Description of the RTNN topology and learning

4.1 RTNN topology and recursive BP learning

The block-diagrams of the RTNN topology and its adjoint, are given on Fig. 6, and Fig. 7. Following Fig. 6, and Fig. 7, we could derive the dynamic BP algorithm of its learning based

on the RTNN topology using the diagrammatic method of (Wan & Beaufays, 1996). The RTNN topology and learning are described in vector-matrix form as:

$$X(k+1) = AX(k) + BU(k); \; B = [B_1 \; ; B_0]; \; U^T = [U_1 \; ; U_2]; \tag{24}$$

$$Z_1(k) = G[X(k)]; \tag{25}$$

$$V(k) = CZ(k); \; C = [C_1 \; ; C_0]; \; Z^T = [Z_1 \; ; Z_2]; \tag{26}$$

$$Y(k) = F[V(k)]; \tag{27}$$

$$A = \text{block-diag} \, (Ai), \; | \, Ai \, | < 1; \tag{28}$$

$$W(k+1) = W(k) + \eta \, \Delta W(k) + \alpha \, \Delta W_{ij}(k-1); \tag{29}$$

$$E(k) = T(k) - Y(k); \tag{30}$$
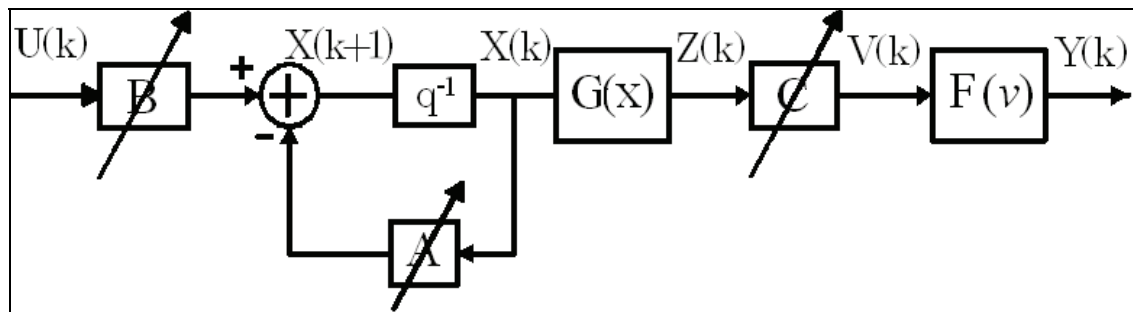
Fig. 6. Block diagram of the RTNN model
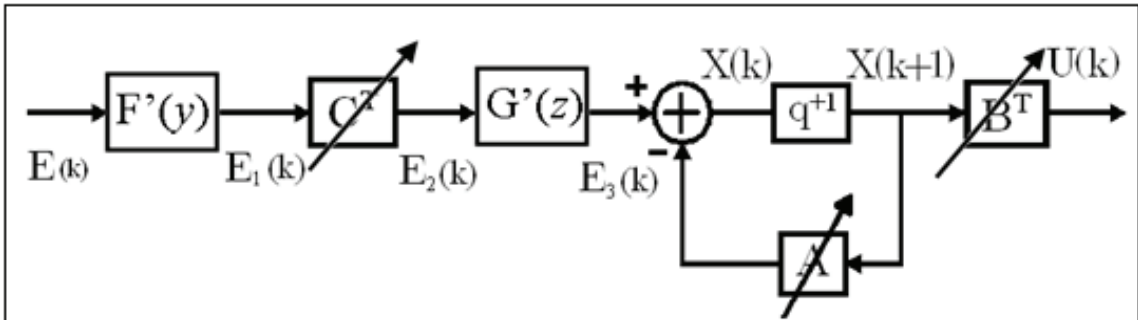
Fig. 7. Block diagram of the adjoint RTNN model

$$E_1(k) = F'[Y(k)] \, E(k); \; F'[Y(k)] = [1 - Y^2(k)]; \tag{31}$$

$$\Delta C(k) = E_1(k) \, Z^T(k); \tag{32}$$

$$E_3(k) = G'[Z(k)] \, E_2(k); \; E_2(k) = C^T(k) \, E_1(k); \; G'[Z(k)] = [1 - Z^2(k)]; \tag{33}$$

$$\Delta B(k) = E_3(k) \, U^T(k); \tag{34}$$

$$\Delta A(k) = E_3(k) \, X^T(k); \tag{35}$$

$$\text{Vec}(\Delta A(k)) = E_3(k) \ast X(k); \tag{36}$$

where: X, Y, U are state, augmented output, and input vectors with dimensions N, (L+1), (M+1), respectively, 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; T is the (Lx1) plant output vector, considered as a RNN reference; 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(·)-activation functions with corresponding dimensions; F′[·], G′[·] are the derivatives of these 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; η, α are learning rate parameters; ΔC is an weight correction of the learned matrix C; ΔB is an weight correction of the learned matrix B; ΔA is an weight correction of the learned matrix A; the diagonal of the matrix A is denoted by Vec(·) and equation (34) represents its learning as an element-by-element vector products; E, E_1, E_2, E_3, are error vectors with appropriate dimensions, predicted by the adjoint RTNN model, given on Fig.7. The stability of the RTNN model is assured by the activation functions (-1, 1) bounds and by the local stability weight bound condition, given by (28). Below a theorem of RTNN stability which represented an extended version of Nava's theorem, (Baruch et al., 2008d) is given.

Theorem of stability of the BP RTNN used as system identifier (Baruch et al., 2008d). Let the RTNN with Jordan Canonical Structure is given by equations (24)-(28) (see Fig.6) and the nonlinear plant model, is as follows:

$$X_{p.}(k+1) = G[\, X_p(k),\, U(k)\,],$$

$$Y_p(k) = F[\, X_p(k)\,];$$

where: $\{Y_p(\cdot),\, X_p(\cdot),\, U(\cdot)\}$ are output, state and input variables with dimensions L, N_p, 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 (29)-(36), 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:

$$L(k) = L_1(k) + L_2(k);$$

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

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

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

where: $\widetilde{W}_A(k) = \hat{A}(k) - A^*$, $\widetilde{W}_B(k) = \hat{B}(k) - B^*$, $\widetilde{W}_C(k) = \hat{C}(k) - C^*$, are vectors of the estimation error and (A^*, B^*, C^*), $(\hat{A}(k), \hat{B}(k), \hat{C}(k))$ 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.:

$$L(k+1) = L_1(k+1) + L_2(k+1) < 0,$$
$$\Delta L(k+1) = L(k+1) - (k);$$

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

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

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 $\eta_{\max}$ changes adaptively during the RTNN learning 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. The Rate of Convergence Lemma used, , is given below. The complete proof of that Theorem of stability is given in (Baruch et al., 2008d).

Rate of Convergence Lemma (Baruch et al., 2008a). Let ΔL_k is defined. Then, applying the limit's definition, the identification error bound condition is obtained as:

$$\overline{\lim_{k \to \infty}} \frac{1}{k} \sum_{t=1}^{k} \left(|E(t)|^2 + |E(t-1)|^2\right) \le d.$$

Proof. Starting from the final result of the theorem of RTNN stability:

$$\Delta L(k) \le -\eta(k)|E(k)|^2 - \alpha(k)|E(k-1)|^2 + d$$

and iterating from k=0, we get:

$$L(k+1) - L(0) \le -\sum_{t=1}^{k}|E(t)|^2 - \sum_{t=1}^{k}|E(t-1)|^2 + dk,$$

$$\sum_{t=1}^{k}\left(|E(t)|^2 + |E(t-1)|^2\right) \le dk - L(k+1) + L(0) \le dk + L(0).$$

From here, we could see that d must be bounded by weight matrices and learning parameters, in order to obtain: $\Delta L(k) \in \mathbb{L}(\infty)$.

As a consequence:. $A(k) \in \mathbb{L}(\infty)$, $B(k) \in \mathbb{L}(\infty)$, $C(k) \in \mathbb{L}(\infty)$,

The stability of the HFNMMI could be proved via linearization of the activation functions of the RTNN models and application of the methodology using LMI.

Theorem of stability of the BP RTNN used as a direct system controller. Let the RTNN with Jordan Canonical Structure is given by equations (24)-(28) and the nonlinear plant model, is given above. Under the assumption of RTNN identifiability made, the application of the BP learning algorithm for A(·), B(·), C(·), in general matricial form, described by equations (29)-(36) without momentum term, and the learning rate η (k) (here it is considered as time-dependent and normalized with respect to the error) are derived using the following Lyapunov function:

$$L(k) = L_1(k) + L_2(k);$$

where: L1 (k) and L2 (k) are given by:

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

$$L_2(k) = tr\left(\widetilde{W}_{A_{(k)}} \widetilde{W}_{A_{(k)}}^T\right) + tr\left(\widetilde{W}_{B_{(k)}} \widetilde{W}_{B_{(k)}}^T\right) + tr\left(\widetilde{W}_{C_{(k)}} \widetilde{W}_{C_{(k)}}^T\right);$$

where: $\widetilde{W}_{A_{(k)}} = \widehat{A}(k) - A^*$, $\widetilde{W}_{B_{(k)}} = \widehat{B}(k) - B^*$, $\widetilde{W}_{C_{(k)}} = \widehat{C}(k) - C^*$, are vectors of the estimation error and $\left(A^*, B^*, C^*\right)$ and $\left(\widehat{A}(k), \widehat{B}(k), \widehat{C}(k)\right)$ denoted the ideal neural weight and the estimate of the neural weight at the k-th step, respectively, for each case.

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

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

$$L(k+1) = L1(k+1) + L2(k+1) < 0,$$

$$\Delta L(k+1) = L(k+1) - L(k);$$

where the condition for L1 (k+1) < 0 fulfillment is that the maximum rate of learning is inside the limits:

$$0 < \eta_{max} < \frac{2}{\vartheta_{max}^2 \psi_{max}^2},$$

and for L2(k+1) < 0, we have:

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

Note that η_max changes adaptively during the learning process of the network, where:

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

Here all: the unmodelled dynamics, the approximation errors and the perturbations, are represented by the β-term, and the complete proof of that theorem and the rate of convergence lemma, are given in (Baruch et al., 2008d).

4.2 Recursive Levenberg-Marquardt RTNN learning

The general recursive L-M algorithm of learning, (Baruch & Mariaca-Gaspar, 2009) is given by the following equations:

$$W(k+1) = W(k) + P(k)\nabla Y[W(k)]E[W(k)], \tag{37}$$

$$Y[W(k)] = g[W(k), U(k)], \tag{38}$$

$$E^2[W(k)] = \{Y_p(k) - g[W(k), U(k)]\}^2, \tag{39}$$

$$DY[W(k)] = \frac{\partial g[W(k), U(k)]}{\partial W}\bigg|_{W=W(k)}; \tag{40}$$

where W is a general weight matrix (A, B, C) under modification; P is the covariance matrix of the estimated weights updated; DY[·] is an nw-dimensional gradient vector; Y is the RTNN output vector which depends of the updated weights and the input; E is an error vector; Yp is the plant output vector, which is in fact the target vector. Using the same RTNN adjoint block diagram (see Fig.7), it was possible to obtain the values of the gradients DY[·] for each updated weight, propagating the value D(k) = I through it. Applying equation (40) for each element of the weight matrices (A, B, C) in order to be updated, the corresponding gradient components are as follows:

$$DY[C_{ij}(k)] = D_{1,i}(k)Z_j(k), \tag{41}$$

$$D_{1,i}(k) = F_j'[Y_i(k)], \tag{42}$$

$$DY[A_{ij}(k)] = D_{2,i}(k)X_j(k), \tag{43}$$

$$DY[B_{ij}(k)] = D_{2,i}(k)U_j(k), \tag{44}$$

$$D_{2,i}(k) = G_i'[Z_j(k)]C_iD_{1,i}(k). \tag{45}$$

Therefore the Jacobean matrix could be formed as:

$$DY[W(k)] = [DY(C_{ij}(k)), DY(A_{ij}(k)), DY(B_{ij}(k))] \tag{46}$$

The P(k) matrix was computed recursively by the equation:

$$P(k) = \alpha^{-1}(k)\left\{P(k-1) - P(k-1)\Omega[W(k)]S^{-1}[W(k)]\Omega^T[W(k)]P(k-1)\right\} ; \qquad (47)$$

where the S($\cdot$), and $\Omega(\cdot)$ matrices were given as follows:

$$S[W(k)] = \alpha(k)\Lambda(k) + \Omega^T[W(k)]P(k-1)\Omega[W(k)] \qquad (48)$$

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

$$\Lambda(k)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix}; \quad 10^{-4} \le \rho \le 10^{-6}; \qquad (49)$$

$$0.97 \le \alpha(k) \le 1; \quad 10^3 \le P(0) \le 10^6.$$

The matrix $\Omega(\cdot)$ had a dimension (nwx2), whereas the second row had only one unity element (the others were zero). The position of that element was computed by:

$$i = k \bmod(nw) + 1; \quad k > nw \qquad (50)$$

After this, the given up topology and learning are applied for an anaerobic wastewater distributed parameter decentralized system identification.

5. Analytical model of the anaerobic digestion bioprocess plant

The anaerobic digestion systems block diagram is depicted on Fig.8. It consists of a fixed bed reactor and a recirculation tank. The physical meaning of all variables and constants (also its values), are summarized in Table 1. The complete analytical model of wastewater treatment anaerobic bioprocess, taken from (Aguilar-Garnica et al., 2006), could be described by the following system of PDE:

$$\frac{\partial X_1}{\partial t} = (\mu_1 - \varepsilon D)X_1, \quad \mu_1 = \mu_{1\max}\frac{S_1}{K_{S1}X_1 + S_1} , \qquad (51)$$

$$\frac{\partial X_2}{\partial t} = (\mu_2 - \varepsilon D)X_2, \quad \mu_2 = \mu_{2s}\frac{S_1}{K_{S2}X_2 + \dfrac{S_2^2}{K_{I2}}} , \qquad (52)$$

$$\frac{\partial S_1}{\partial t} = \frac{E_z}{H^2}\frac{\partial^2 S_1}{\partial z^2} - D\frac{\partial S_1}{\partial t} - k_1\mu_1 X_1 , \qquad (53)$$

$$\frac{\partial S_2}{\partial t} = \frac{E_z}{H^2}\frac{\partial^2 S_2}{\partial z^2} - D\frac{\partial S_2}{\partial t} - k_2\mu_1 X_1 , \qquad (54)$$

$$S_1(0,t) = \frac{S_{1,in}(t) + RS_{1T}}{R+1}, \quad S_2(0,t) = \frac{S_{2,in}(t) + RS_{2T}}{R+1}, \quad R = \frac{Q_T}{DV_{eff}} , \qquad (55)$$

Variable	Units	Name	Value
z	$z \in [0,1]$	*Space variable*	
t	D	*Time variable*	
E_z	m²/d	*Axial dispersion coefficient*	1
D	1/d	*Dilution rate*	0.55
H	m	*Fixed bed length*	3.5
X_1	g/L	*Concentration of acidogenic bacteria*	
X_2	g/L	*Concentration of methanogenic bacteria*	
S_1	g/L	*Chemical Oxygen Demand*	
S_2	mmol/L	*Volatile Fatty Acids*	
ε		*Bacteria fraction in the liquid phase*	0.5
k_1	g/g	*Yield coefficients*	42.14
k_2	mmol/g	*Yield coefficients*	250
k_3	mmol/g	*Yield coefficients*	134
μ_1	1/d	*Acidogenesis growth rate*	
μ_2	1/d	*Methanogenesis growth rate*	
μ_{1max}	1/d	*Maximum acidogenesis growth rate*	1.2
μ_{2s}	1/d	*Maximum methanogenesis growth rate*	0.74
K_{1s}'	g/g	*Kinetic parameter*	50.5
K_{2s}'	mmol/g	*Kinetic parameter*	16.6
K_{I2}'	mmol/g	*Kinetic parameter*	256
Q_T	m³/d	*Recycle flow rate*	0.24
V_T	m³	*Volume of the recirculation tank*	0.2
S_{1T}	g/L	*Concentration of Chemical Oxygen Demand in the recirculation tank*	
S_{2T}	mmol/L	*Concentration of Volatile Fatty Acids in the recirculation tank*	
Q_{in}	m³/d	*Inlet flow rate*	0.31
V_B	m³	*Volume of the fixed bed*	1
V_{eff}	m³	*Effective volume tank*	0.95
$S_{1,in}$	g/L	*Inlet substr. Concentration*	
$S_{2,in}$	mmol/L	*Inlet substr. Concentration*	

Table 1. Summary of the variables in the plant model

$$\frac{\partial S_1}{\partial z}(1,t) = 0, \quad \frac{\partial S_2}{\partial z}(1,t) = 0 , \tag{56}$$

$$\frac{dS_{1T}}{dt} = \frac{Q_T}{V_T}\left(S_1(1,t) - S_{1T}\right), \quad \frac{dS_{2T}}{dt} = \frac{Q_T}{V_T}\left(S_2(1,t) - S_{2T}\right). \tag{57}$$

For practical purpose, the full PDE anaerobic digestion process model (51)-(57), taken from (Aguilar-Garnica et al., 2006), could be reduced to an ODE system using an early lumping technique and the Orthogonal Collocation Method (OCM), (Bialecki & Fairwether, 2001), in four points (0.2H, 0.4H, 0.6H, 0.8H) obtaining the following system of OD equations:
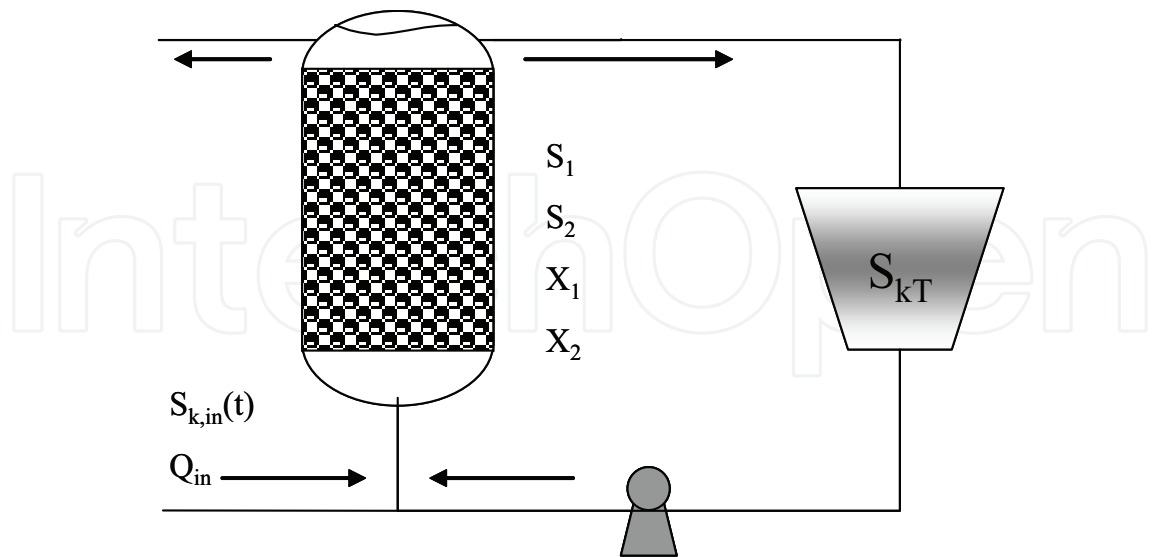
Fig. 8. Block-diagram of anaerobic digestion bioreactor

$$\frac{dX_{1,i}}{dt} = \left(\mu_{1,i} - \varepsilon D\right)X_{1,i}, \quad \frac{dX_{2,i}}{dt} = \left(\mu_{2,i} - \varepsilon D\right)X_{2,i}, \tag{58}$$

$$\frac{dS_{1,i}}{dt} = \frac{E_z}{H^2}\sum_{j=1}^{N+2} B_{i,j}S_{1,j} - D\sum_{j=1}^{N+2} A_{i,j}S_{1,j} - k_1\mu_{1,i}X_{1,i}, \tag{59}$$

$$\frac{dS_{2,i}}{dt} = \frac{E_z}{H^2}\sum_{j=1}^{N+2} B_{i,j}S_{1,j} - D\sum_{j=1}^{N+2} A_{i,j}S_{2,j} - k_2\mu_{1,i}X_{2,i} - k_3\mu_{2,i}X_{2,i}, \tag{60}$$

$$\frac{dS_{1T}}{dt} = \frac{Q_T}{V_T}\left(S_{1,N+2} - S_{1T}\right), \quad \frac{dS_{2T}}{dt} = \frac{Q_T}{V_T}\left(S_{2,N+2} - S_{2T}\right), \tag{61}$$

$$S_{k,1} = \frac{1}{R+1}S_{k,in}\left(t\right) + \frac{R}{R+1}S_{kT}, \quad S_{k,N+2} = \frac{K_1}{R+1}S_{k,in}\left(t\right) + \frac{K_1 R}{R+1}S_{kT} + \sum_{i=1}^{N+1} K_i S_{k,i}, \tag{62}$$

$$K_1 = \frac{A_{N+2,1}}{A_{N+2,N+2}}, \quad K_i = \frac{A_{N+2,i}}{A_{N+2,N+2}}, \tag{63}$$

$$A = \Lambda\phi^{-1}, \quad \Lambda = \left[\omega_{m,l}\right] = \left(l-1\right)z_m^{l-2}, \tag{64}$$

$$B = \Gamma\phi^{-1}, \quad \Gamma = \left[\tau_{m,l}\right], \quad \tau_{m,l} = \left(l-1\right)\left(l-2\right)z_m^{l-3}, \quad \phi_{m,l} = z_m^{l-1}, \tag{65}$$

$$i = 2,\ldots N+2, \quad m,l = 1,\ldots N+2. \tag{66}$$

The reduced plant model (58)-(66), could be used as unknown plant model which generate input/output process data for decentralized adaptive FNMM control system design, based on the concepts, given in (Baruch et al., 2008a; Baruch et al., 2008b; Baruch et al., 2008c; Baruch et al., 2008d). The mentioned concepts could be applied for this DPS fuzzyfying the space variable z, which represented the height of the fixed bed. Here the centers of the membership functions with respect to z corresponded to the collocation points of the simplified plant model which are in fact the four measurement points of the fixed bed, adding one more point for the recirculation tank.

6. Simulation results

In this paragraph, graphical and numerical simulation results of system identification, direct and indirect control, with and without I-term, will be given. For lack of space we will give graphical results only for the X1 variable. Furthermore the graphical results for the other variables possessed similar behavior.

6.1 Simulation results of the system identification using L-M RTNN learning

The decentralized FNMM identifier used a set of five T-S fuzzy rules containing in its consequent part RTNN learning procedures (1). The RTNN topology is given by the equations (24)-(28), the BP RTNN learning is given by (29)-(36), and the L-M RTNN learning is given by (37)-(50). The topology of the first four RTNNs is (2-6-4) (2 inputs, 6 neurons in the hidden layer, 4 outputs) and the last one has topology (2-4-2), corresponding to the fixed bed plant behavior in each collocation point and the recirculation tank. The RTNNs identified the following fixed bed variables: X_1 (acidogenic bacteria), X_2 (methanogenic bacteria), S_1 (chemical oxygen demand) and S_2 (volatile fatty acids), in the following four collocation points, z=0.2H, z=0.4H, z=0.6H, z=0.8H, and the following variables in the recirculation tank: S_{1T} (chemical oxygen demand) and S_{2T} (volatile fatty acids). The graphical simulation results of RTNNs L-M learning are obtained on-line during 600 iteration with a step of 0.1 sec. The learning rate parameters of RTNN have small values which are different for the different measurement point variables ($\rho=0.1$ and $\alpha=0$). The Figs. 9-11 showed graphical simulation results of open loop decentralized plant identification. The MSE of the decentralized FNMM approximation of plant variables in the collocation points, using the L-M and BP RTNN learning are shown in Tables 2 and 3. The input signals applied are:

$$S_{1,in} = 0.55 + 0.15\cos\left(\frac{3\pi}{80}t\right) + 0.3\sin\left(\frac{\pi}{80}t\right),$$ (67)

$$S_{2,in} = 0.55 + 0.05\cos\left(\frac{3\pi}{40}t\right) + 0.3\sin\left(\frac{\pi}{40}t\right).$$ (68)

The graphical y numerical results of decentralized FNMM identification (see Fig. 9-11, and Tables 2, 3) showed a good HFNMMI convergence and precise plant output tracking (MSE 0.0083 for the L-M, and 0.0253 for the BP RTNN learning in the worse case).

Collocation point	X_1	X_2	S_1/S_{1T}	S_2/S_{2T}
z=0.2	0.0013	0.0012	0.0049	0.0058
z=0.4	0.0013	0.0013	0.0058	0.0049
z=0.6	0.0013	0.0013	0.0071	0.0055
z=0.8	0.0014	0.0013	0.0083	0.0070
Recirculation tank			0.0080	0.0058

Table 2. MSE of the decentralized FNMM approximation of the bioprocess output variables in the collocation points, using the L-M RTNN learning
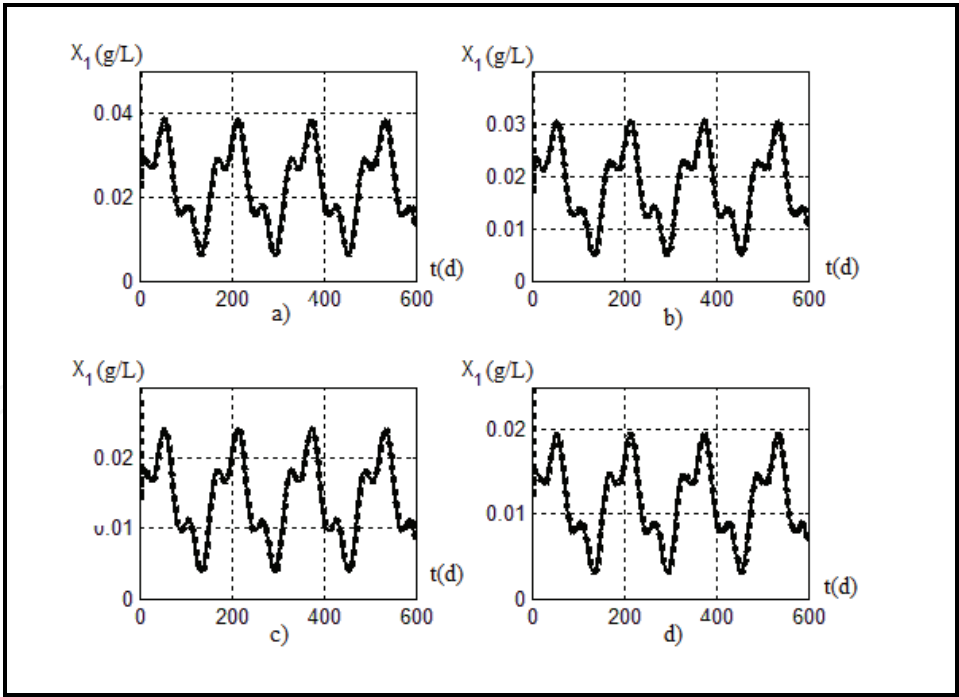
Collocation point	X_1	X_2	S_1/S_{1T}	S_2/S_{2T}
z=0.2	0.0015	0.0023	0.0145	0.0192
z=0.4	0.0015	0.0044	0.0098	0.0164
z=0.6	0.0030	0.0009	0.0092	0.0133
z=0.8	0.0046	0.0048	0.0045	0.0086
Recirculation tank			0.0168	0.0253

Table 3. MSE of the decentralized FNMM approximation of the bioprocess output variables in the collocation points, using the BP RTNN learning

Fig. 9. Graphical simulation results of the FNMM identification of X1 in a) Z=0.2H; b) 0.4H; c) 0.6H; d) 0.8H (acidogenic bacteria in the corresponding fixed bed points) by four fuzzy rules RTNNs (dotted line-RTNN output, continuous line-plant output) for 600 iteration of L-M RTNN learning
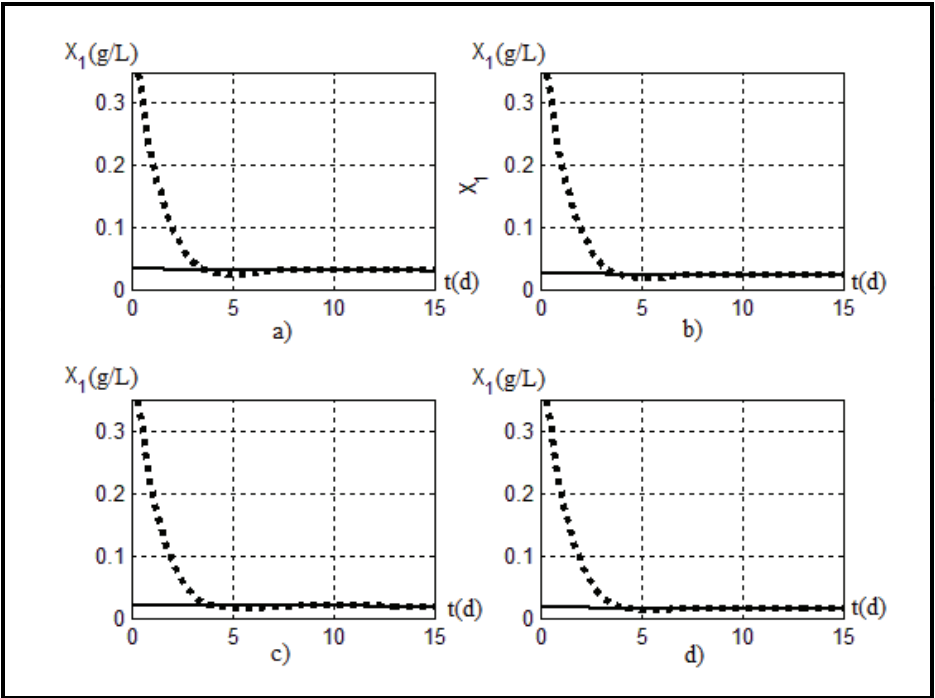
Fig. 10. Detailed graphical simulation results of the FNMM identification of X1 in a) Z=0.2H; b) 0.4H; c) 0.6H; d) 0.8H (acidogenic bacteria in the corresponding fixed bed points) by four fuzzy rules RTNNs (dotted line-RTNN output, continuous line-plant output) for the first 15 iterations of the L-M RTNN learning
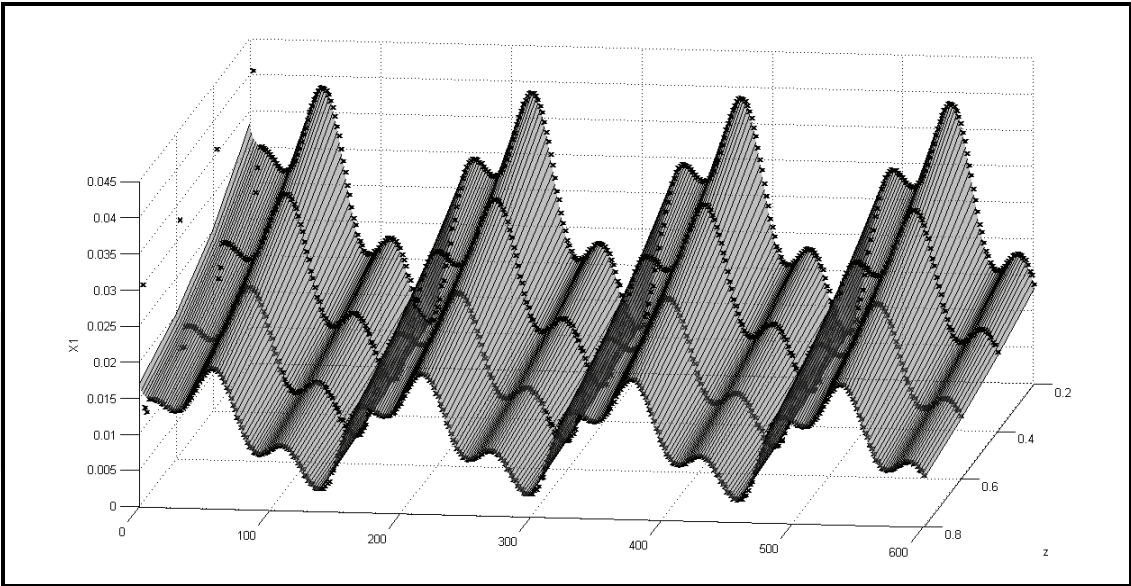
Fig. 11. Graphics of the 3d view of X1 space/time approximation during its L-M RTNN learning in four points

6.2 Simulation results of the direct HFNMM control with I-term and L-M RTNN learning

The topology of the first four RTNNs is (12-14-2) for the variables in the collocation points x=0.2H, z=0.4H, z=0.6H, z=0.8H and for the recirculation tank is (8-10-2). The graphical simulation results of RTNNs L-M learning are obtained on-line during 600 iterations (2.4

hours) with a step of 0.1 sec. The learning parameters of RTNN are $\rho=0.2$ and $\alpha=1$; while the parameter of the I-term are $\eta=0.01$ and $\alpha=1e-8$. Finally the topology of the defuzzifier neural network is (10-2) with parameters $\eta=0.0035$ and $\alpha=0.00001$.

The Figs. 12-17 showed graphical simulation results of the direct decentralized HFNMM control with and without I-term, where the outputs of the plant are compared with the reference signals. The reference signals are train of pulses with uniform duration and random amplitude. The MSE of control for each output signal and each measurement point are given on Table 4 For sake of comparison the MSE of direct decentralized HFNMM proportional control (without I-term) for each output signal and each measurement point are given on Table 5.
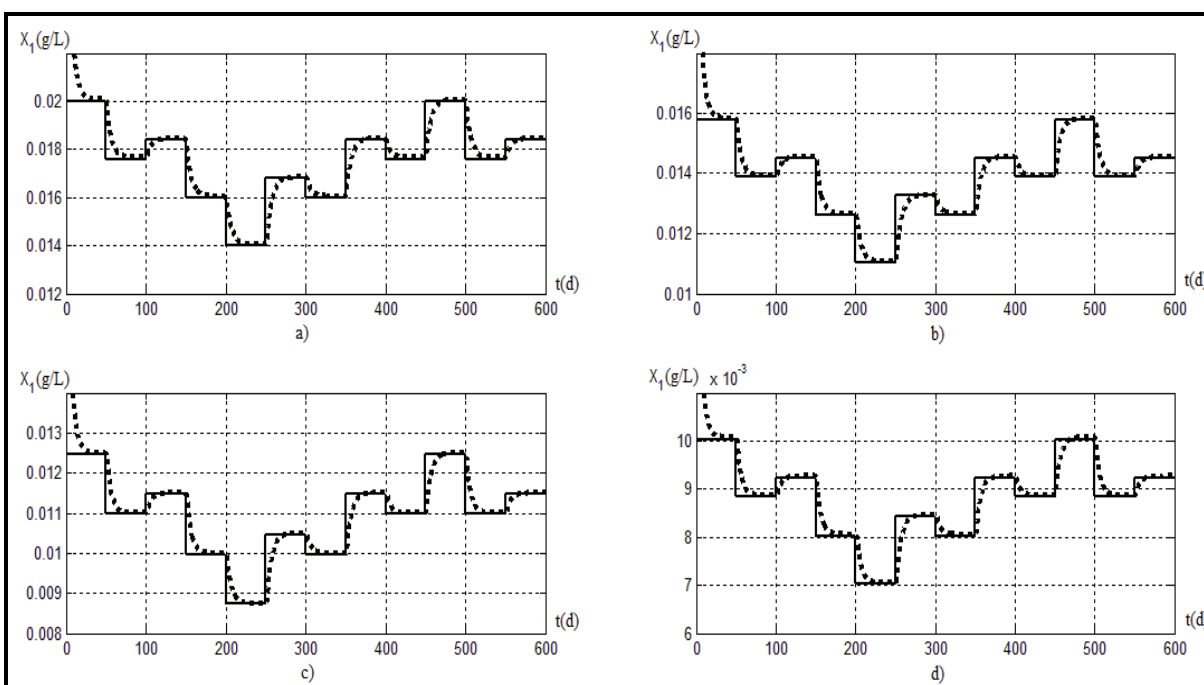
Fig. 12. Results of the direct decentralized HFNMM I-term control of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-reference) in four collocation points (a) 0.2H,b) 0.4H, c) 0.6H, d) 0.8H) for 600 iterations

Also, for sake of comparison, graphical results of direct decentralized HFNMM proportional control (without I-term) only for the X1 variable will be presented. The results show that the proportional control could not eliminate the static error due to inexact approximation and constant process or measurement disturbances. The graphical and numerical results of direct decentralized HFNMM I-term control (see Fig. 12-13, and Tables 4, 5) showed a good reference tracking (MSE is of 0.0097 for the I-term control and 0.0119 for the control without I-term in the worse case). The results showed that the I-term control eliminated constant disturbances and approximation errors and the proportional control could not.
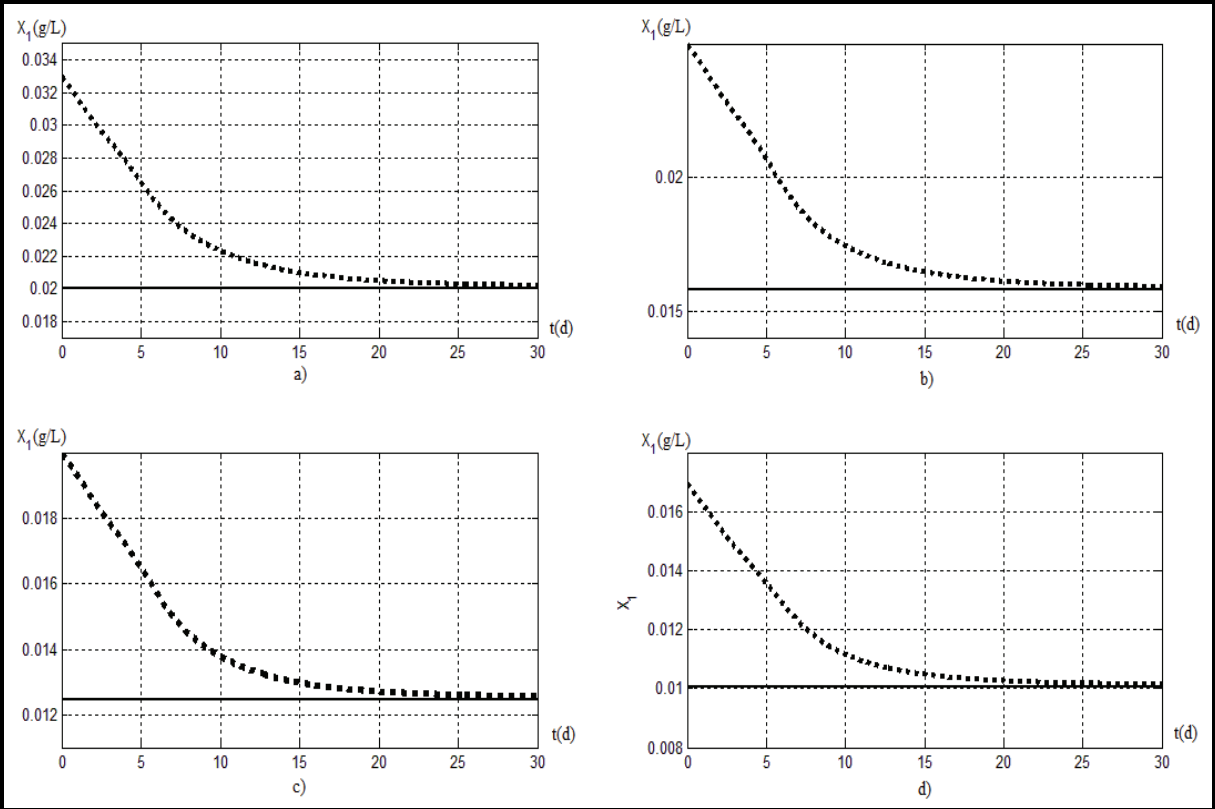
Fig. 13. Detailed graphical results of the direct decentralized HFNMM I-term control of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for the first 30 iterations
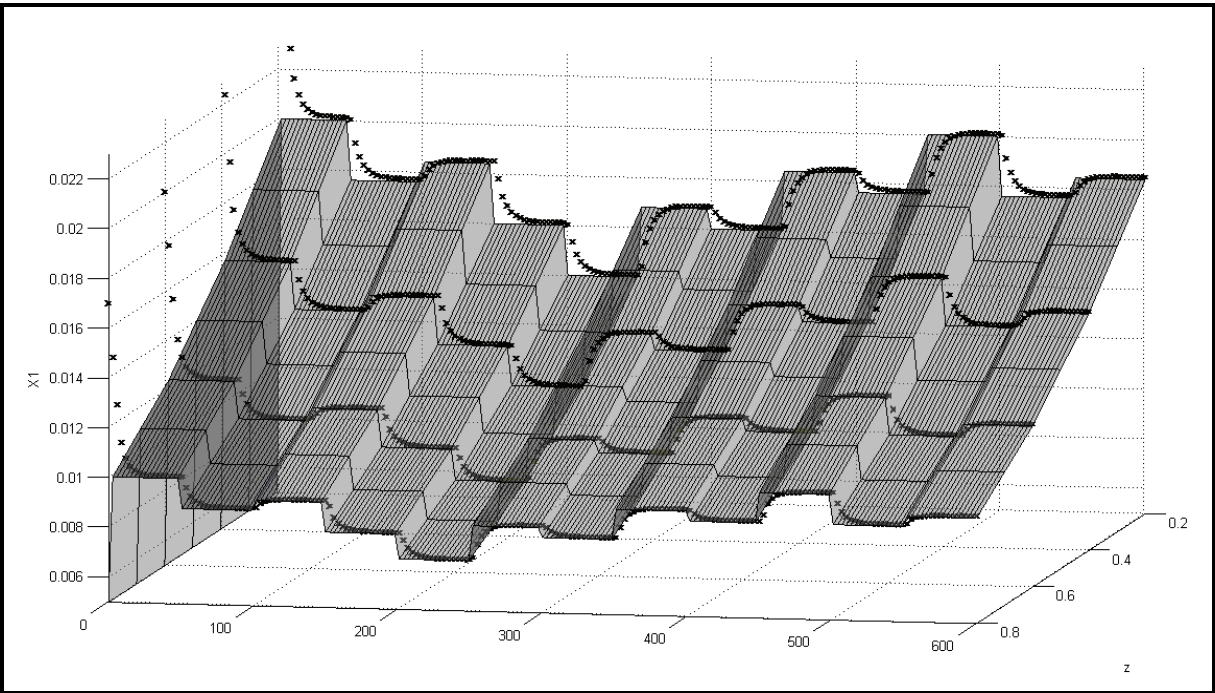

Fig. 14. Graphics of the 3d view of X1 space/time approximation and direct decentralized HFNMM I-term control in four collocation points of the fixed bed
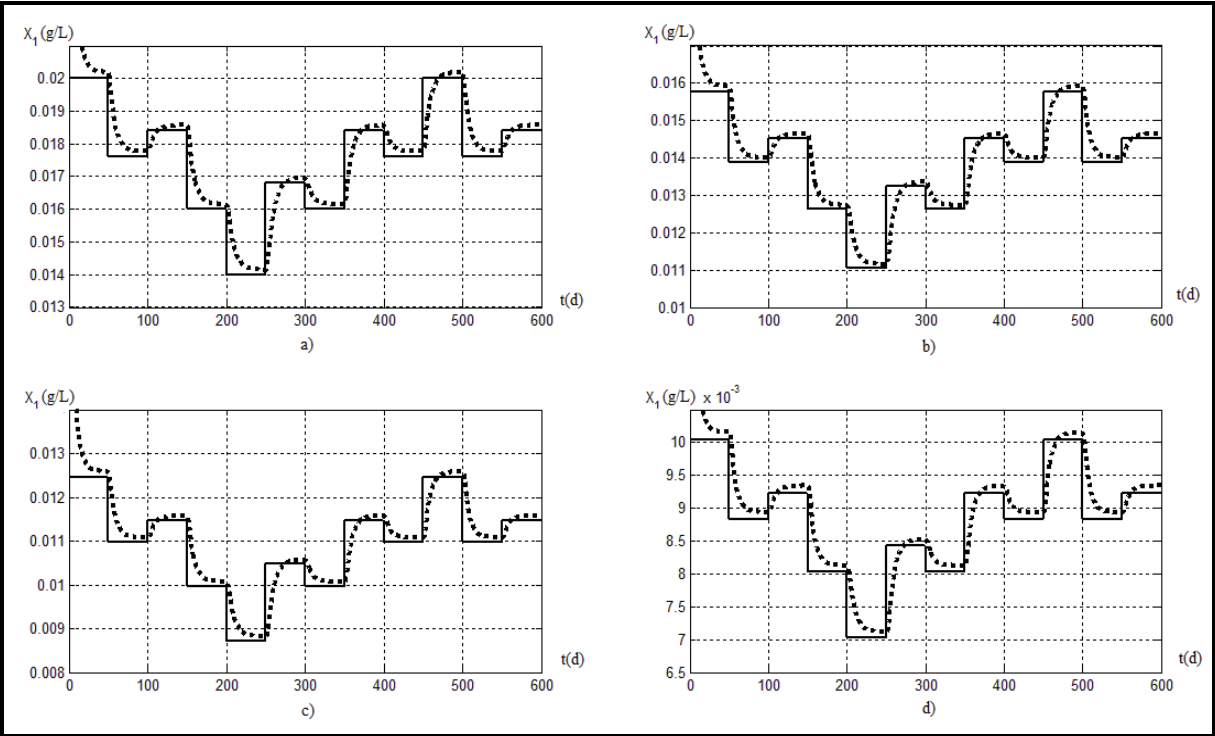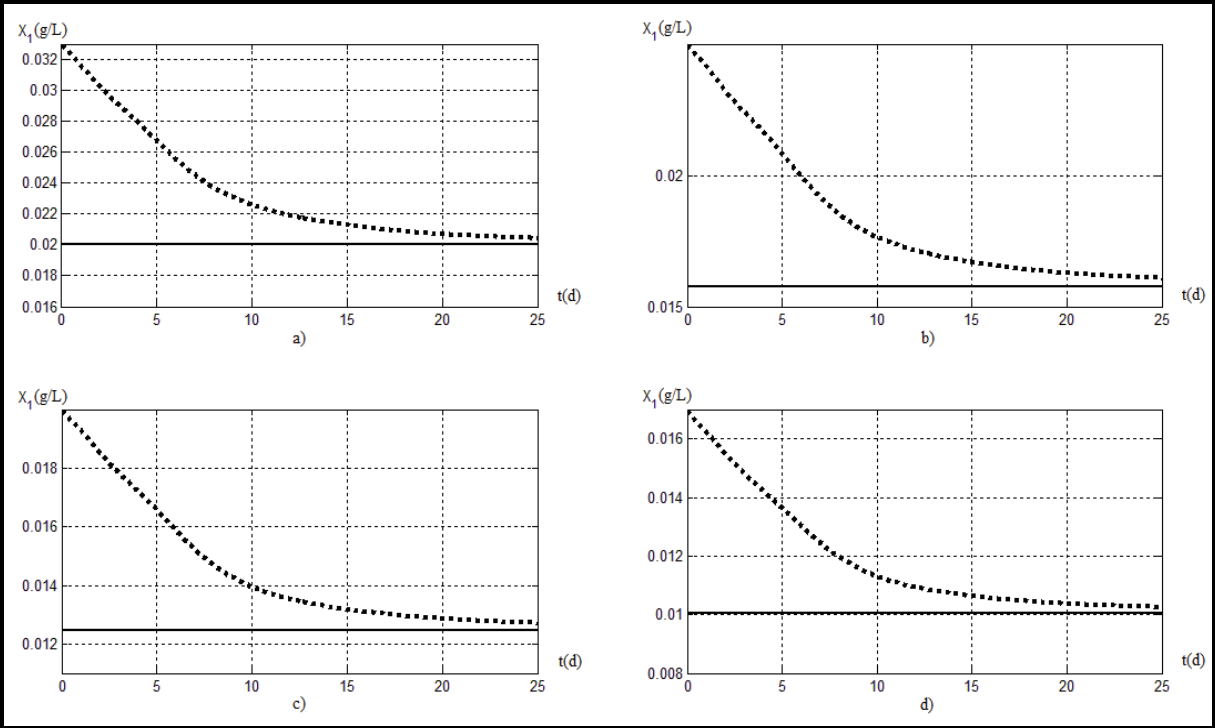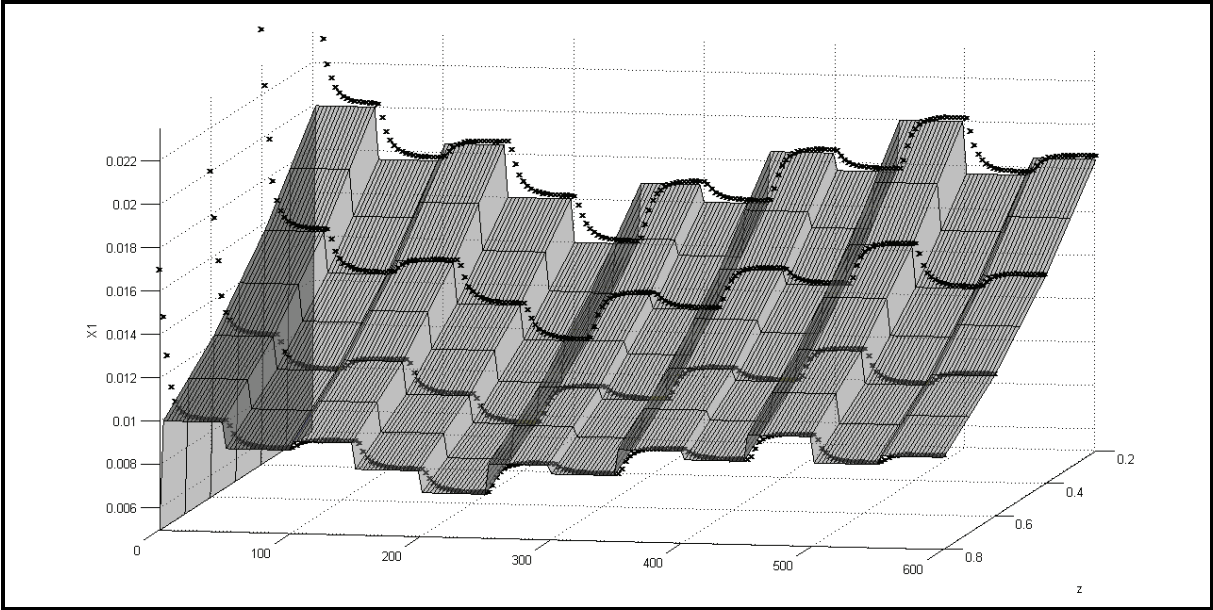
Fig. 15. Results of the direct decentralized HFNMM control without I-term of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for 600 iterations

Fig. 16. Detailed graphical results of the direct decentralized HFNMM control without I-term of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for the first 30 iterations

Fig. 17. Graphics of the 3d view of X1 space/time approximation and direct decentralized HFNMM proportional control (without I-term) in four collocation points of the fixed bed

Collocation point	X_1	X_2	S_1/S_{1T}	S_2/S_{2T}
z=0.2	0.0011	0.0013	0.0065	0.0097
z=0.4	0.0009	0.0011	0.0051	0.0090
z=0.6	0.0008	0.0011	0.0042	0.0074
z=0.8	0.0006	0.0010	0.0037	0.0063
Recirculation tank			0.0060	0.0086

Table 4. MSE of the direct decentralized HFNMM I-term control of the bioprocess plant

Collocation point	X_1	X_2	S_1/S_{1T}	S_2/S_{2T}
z=0.2	0.0012	0.0016	0.0084	0.0119
z=0.4	0.0009	0.0014	0.0068	0.0107
z=0.6	0.0007	0.0012	0.0055	0.0089
z=0.8	0.0006	0.0010	0.0045	0.0073
Recirculation tank			0.0068	0.0092

Table 5. MSE of the direct decentralized HFNMM proportional control (without I-term) of the bioprocess plant

6.3 Simulation results of the indirect HFNMM I-term SMC and L-M RTNN learning

The neural network used as defuzifier in the control with BP learning rule has the toplogy (10-2) with learning parameters η=0.005 and α=0.00006. For the simuulation with the L-M RTNN learning we use a saturation U_0=1 with γ=0.8. In the integral term we used the parameters for the offset (Of), η=0.01 and α=1e-8. The Figs. 18-23 showed graphical simulation results of the indirect (sliding mode) decentralized HFNMM with and without I-term control. The MSE of control for each output signal and each measurement point are given on Table 6. The reference signals are train of pulses with uniform duration and random amplitude and the outputs of the plant are compared with the reference signals.

Collocation point	X_1	X_2	S_1/S_{1T}	S_2/S_{2T}
z=0.2	0.0010	0.0011	0.0052	0.0089
z=0.4	0.0007	0.0009	0.0040	0.0084
z=0.6	0.0006	0.0009	0.0037	0.0063
z=0.8	0.0006	0.0008	0.0034	0.0061
Recirculation tank			0.0051	0.0074

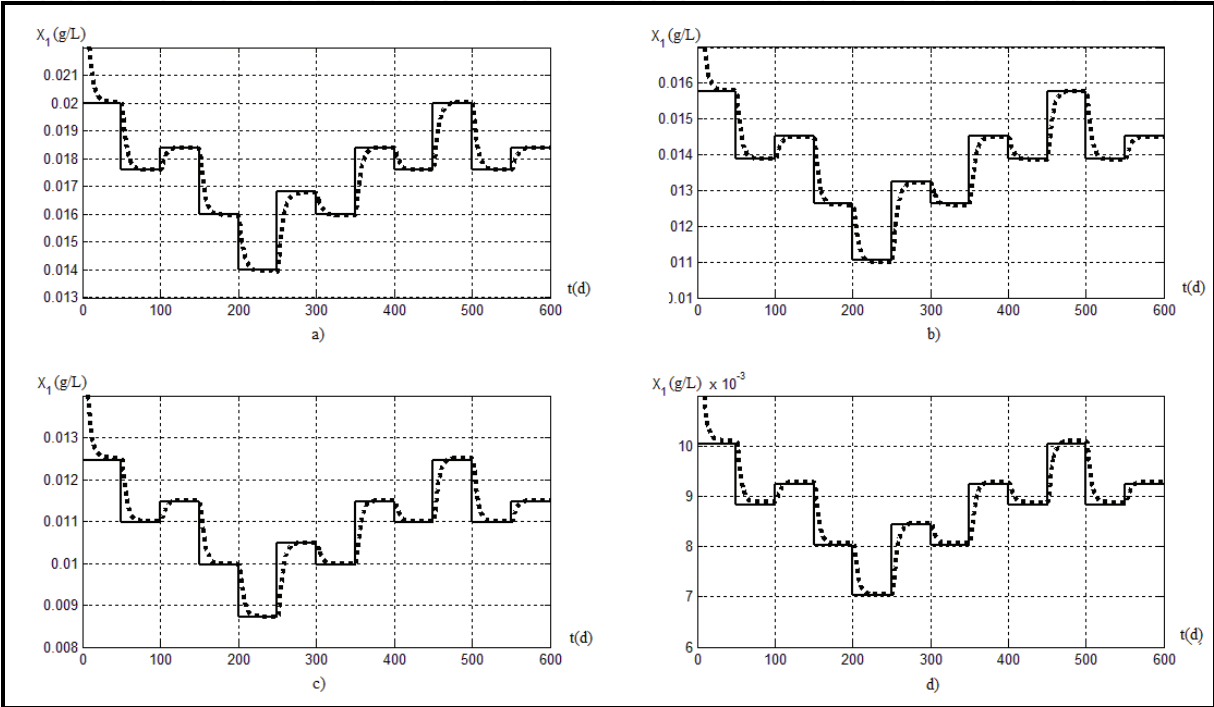Table 6. MSE of the indirect decentralized HFNMM I-term control of the bioprocess plant

Fig. 18. Results of the indirect (SMC) decentralized HFNMM I-term control of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for 600 iterations

Collocation point	X_1	X_2	S_1/S_{1T}	S_2/S_{2T}
z=0.2	0.0013	0.0018	0.0101	0.0139
z=0.4	0.0010	0.0016	0.0083	0.0125
z=0.6	0.0008	0.0014	0.0068	0.0104
z=0.8	0.0007	0.0012	0.0057	0.0085
Recirculation tank			0.0070	0.0095

Table 7. MSE of the indirect decentralized HFNMM proportional control (without I-term) of the bioprocess plant

The graphical y numerical results (see Fig. 18-23, and Tables 6, 7) of the indirect (sliding mode) decentralized control showed a good identification and precise reference tracking (MSE is about 0.0089 in the worse case). The comparison of the indirect and direct decentralized control showed a good results for both control methods (see Table 3 and Table 4) with slight priority for the indirect control (9.8315e-5 vs. 1.184e-4) due to its better plant dynamics compensation ability and adaptation.
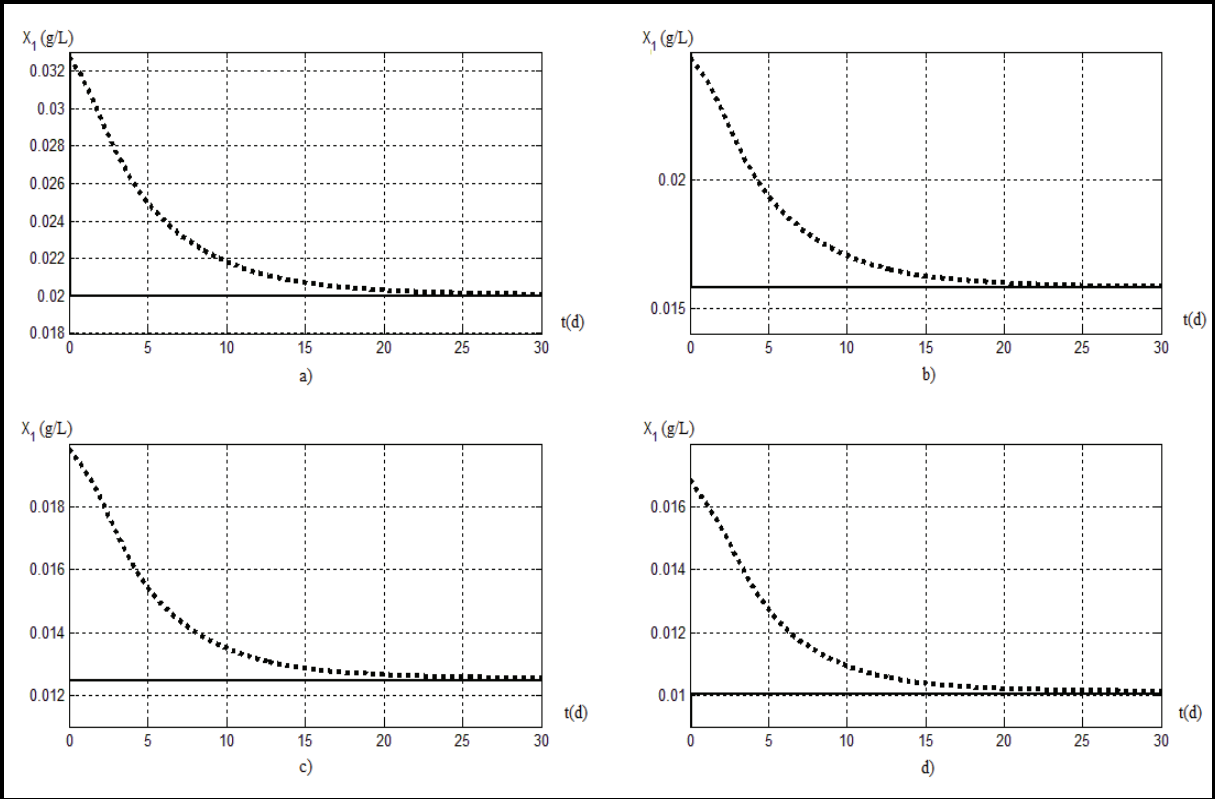
Fig. 19. Detailed graphical results of the indirect (SMC) decentralized HFNMM I-term control of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for the first 30 iterations
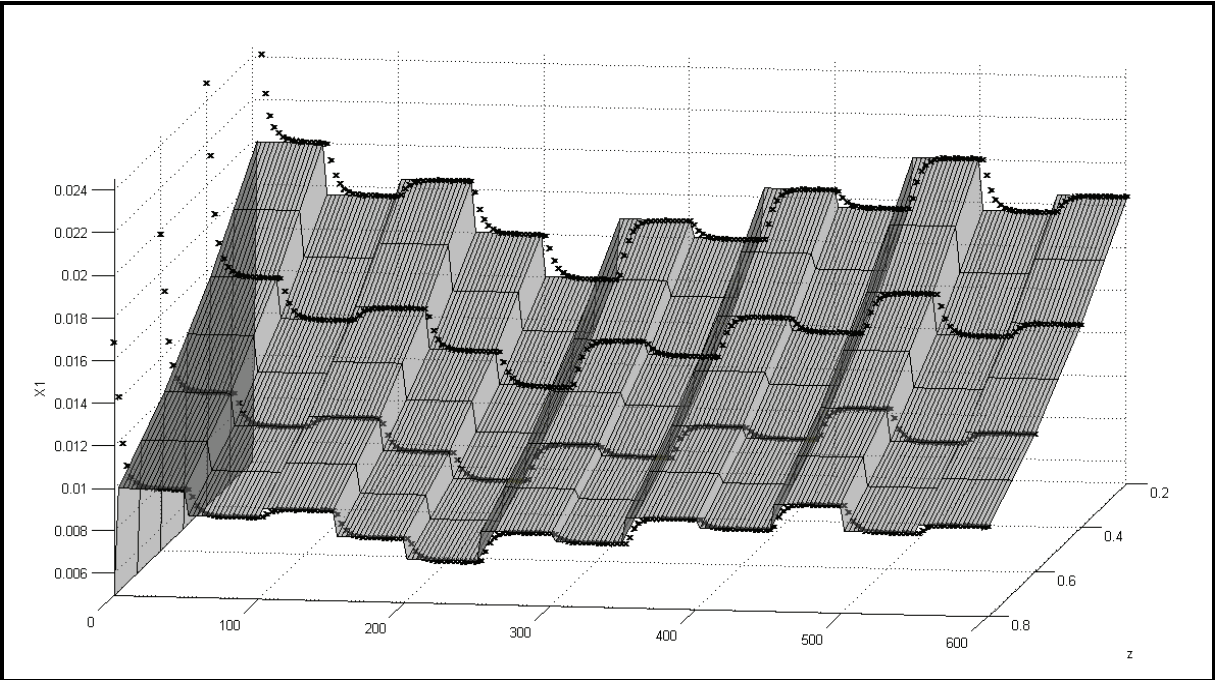

Fig. 20. Graphics of the 3d view of X1 space/time approximation and indirect decentralized HFNMM I-term control in four collocation points of the fixed bed
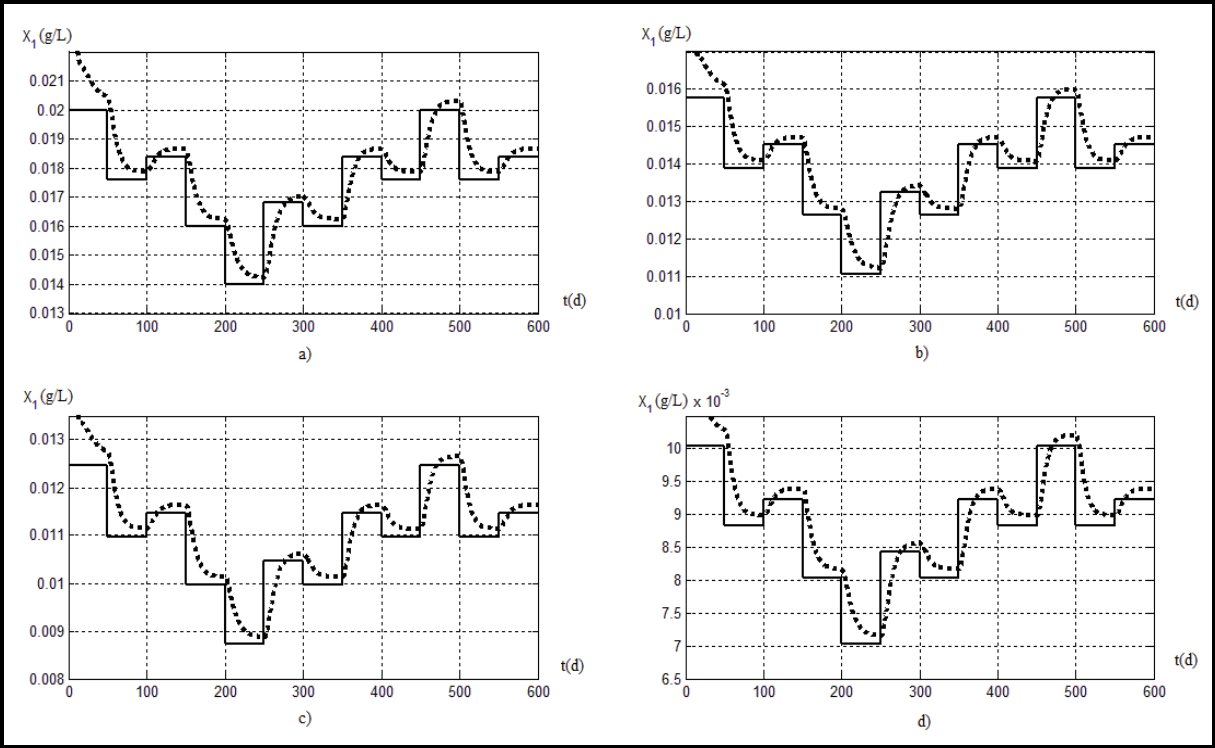
Fig. 21. Results of the indirect (SMC) decentralized HFNMM proportional control (without I-term) of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for 600 iterations
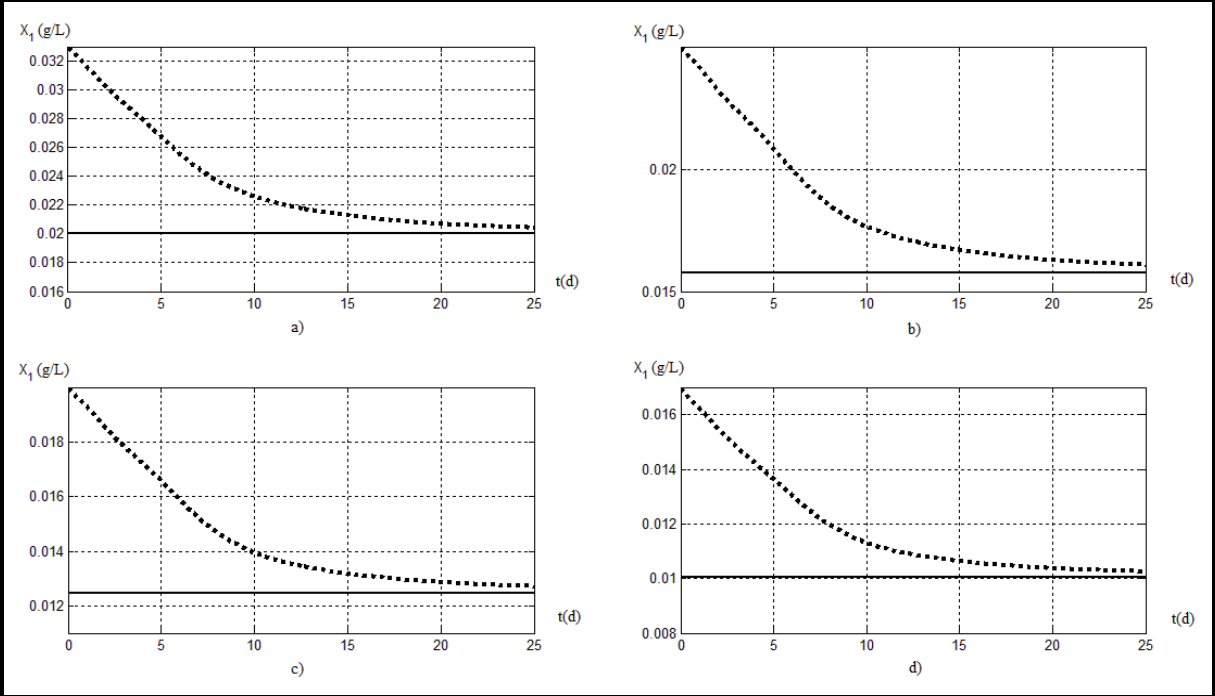

Fig. 22. Detailed graphical results of the indirect (SMC) decentralized HFNMM proportional control (without I-term) of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for the first 25 iterations

For sake of comparison, graphical results of indirect decentralized HFNMM proportional control (without I-term) only for the X1 variable are presented. The results show that the proportional control could not eliminate the static error due to inexact approximation and constant process or measurement disturbances.
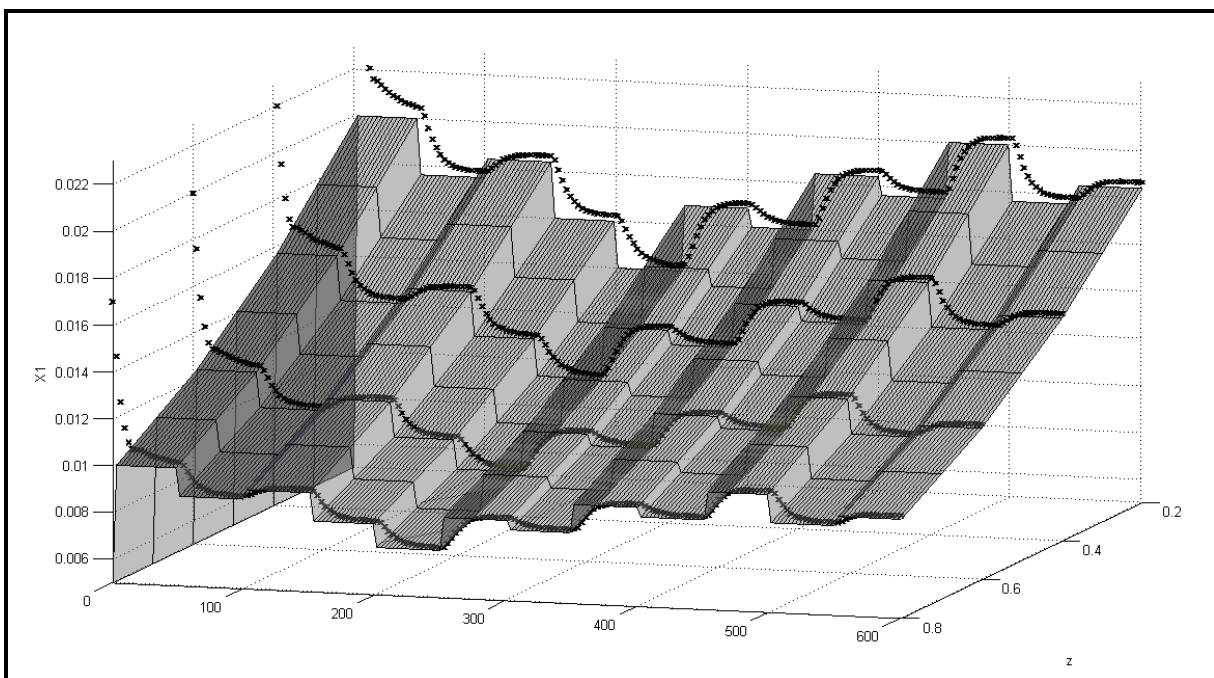

Fig. 23. Graphics of the 3d view of X1 space/time approximation and indirect decentralized HFNMM proportional control (without I-term) in four collocation points of the fixed bed

7. Conclusion

The chapter proposed decentralized recurrent fuzzy-neural identification, direct and indirect I-term control of an anaerobic digestion wastewater treatment bioprocess, composed by a fixed bed and a recirculation tank, represented a DPS. The simplification of the PDE process model by ODE is realized using the orthogonal collocation method in four collocation points (plus the recirculation tank) represented centers of membership functions of the space fuzzyfied output variables. The obtained from the FNMMI state and parameter information is used by a HFNMM direct and indirect (sliding mode) control with or without I-term. The applied fuzzy-neural approach to that DPS decentralized direct and indirect identification and I-term control exhibited a good convergence and precise reference tracking eliminating static errors, which could be observed in the MSE% numerical results given on Tables 4 and 6 (2.107e-5 vs. 1.184e-4 vs. 9.8315e-5 in the worse case).

8. References

Aguilar-Garnica F., Alcaraz-Gonzalez V. & V. Gonzalez-Alvarez (2006), Interval Observer Design for an Anaerobic Digestion Process Described by a Distributed Parameter Model, *Proceedings of the Second International Meeting on Environmental Biotechnology and Engineering: 2IMEBE*, pp. 1-16, ISBN 970-95106-0-6, Mexico City, Mexico, September 2006

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.; Beltran-Lopez, R.; Olivares-Guzman, J.L. & Flores, J.M. (2008a). A fuzzy-neural multi-model for nonlinear systems identification and control. *Fuzzy Sets and Systems*, Elsevier, Vol. 159, No 20, (October 2008) page numbers (2650-2667) ISSN 0165-0114

Baruch, I.S & Galvan-Guerra, R. (2008). Decentralized Direct Adaptive Fuzzy-Neural Control of an Anaerobic Digestion Bioprocess Plant, In: Yager, R.R., Sgurev, V.S., Jotsov, V. (eds), *Fourth International IEEE Conference on Intelligent Systems*, Sept. 6-8, Varna, Bulgaria, ISBN: 978-1-4244-1740-7, vol. I, Session 9, Neuro-Fuzzy Systems, IEEE (2008): pp. 9-2 to 9-7.

Baruch, I.S. & Galvan-Guerra, R. (2009). Decentralized Adaptive Fuzzy-Neural Control of an Anaerobic Digestion Bioprocess Plant. In: *Proc. of the 2009 International Fuzzy Systems Association World Congress, 2009 European Society for Fuzzy Logic and Technology Conference, IFSA/ EUSFLAT 2009*, 20.07-24.07.09, Lisbon, Portugal, ISBN 978-989-95079-6-8 (2009): pp. 460-465

Baruch, I.S.; Galvan-Guerra, R.; Mariaca-Gaspar, C.R. & Castillo, O. (2008b). Fuzzy-Neural Control of a Distributed Parameter Bioprocess Plant. *Proceedings of theIEEE International Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence, WCCI 2008*, June 1-6, 2008, Hong Kong, ISBN: 978-1-4244-1819-0, ISSN 1098-7584, IEEE (2008), pp. 2208-2215

Baruch, I.S.; Galvan-Guerra, R.; Mariaca-Gaspar, C.R. & Melin, P. (2008c). Decentralized Indirect Adaptive Fuzzy-Neural Multi-Model Control of a Distributed Parameter Bioprocess Plant. *Proceedings of International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence, WCCI 2008*, June 1-6, 2008, Hong Kong, ISBN: 978-1-4244-1821-3, ISSN: 1098-7576, IEEE (2008) pp. 1658-1665

Baruch, I.S & Garrido, R. (2005). A direct adaptive neural control scheme with integral terms. *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 (213-224), ISSN 0884-8173

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.; Hernandez, L.A.; Mariaca-Gaspar, C.R. & Nenkova, B. (2007). An adaptive sliding mode control with I-term using recurrent neural identifier. *Cybernetics and Information Technologies, BAS, Sofia, Bulgaria*, Vol. 7, No 1, (January 2007) page numbers 21-32, ISSN 1311-9702

Baruch, I.S. & Mariaca-Gaspar, C.R. (2009). A Levenberg-Marquardt learning algorithm applied for recurrent neural identification and control of a wastewater treatment bioprocess. *International Journal of Intelligent Systems,* Vol. 24, No 11, (November 2009) page numbers (1094-1114), ISSN 0884-8173

Baruch, I. S., Mariaca-Gaspar, C.R., and Barrera-Cortes, J. (2008d). Recurrent Neural Network Identification and Adaptive Neural Control of Hydrocarbon Biodegradation Processes. In: Hu, Xiaolin, Balasubramaniam, P. (eds.): *Recurrent Neural Networks*, I-Tech Education and Publishing KG, Vienna, Austria, ISBN 978-953-7619-08-4, (2008) Chapter 4: pp. 61-88

Bialecki B. & Fairweather G. (2001), Orthogonal Spline Collocation Methods for Partial Differential Equations, *Journal of Computational and Applied Mathematics*, vol. 128, No 1-2, (March 2001) page numbers (55-82), ISSN 0377-0427

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, (June1995), page numbers (817-840), ISSN 0005-1098

Bulsari, A. & Palosaari, S. (1993). Application of neural networks for system identification of an adsorption column. *Neural Computing and Applications*, Vol. 1, No 2, (Jun2 1993) page numbers (160-165), ISSN 0941-0643

Deng H. & Li H.X. (2003). Hybrid Intelligence Based Modeling for Nonlinear Distributed Parameter Process with Applications to the Curing Process. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 4, No. 4, (October 2003) page numbers (3506 - 3511), ISSN 1062-922X

Deng H., Li H.X. & Chen G. (2005). Spectral-Approximation-Based Intelligent Modeling for Distributed Thermal Processes. *IEEE Transactions on Control Systems Technology*, Vol. 13, No. 5, (September 2005) page numbers (686-700), ISSN 1063-6536

Gonzalez-Garcia R, Rico-Martinez R. & Kevrekidis I. (1998). Identification of Distributed Parameter Systems: A Neural Net Based Approach, *Computers and Chemical Engineering*, Vol. 22, No 1, (March 2003) page numbers (S965-S968), ISSN 0098-1354

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 07458

Padhi R. & Balakrishnan S. (2003). Proper Orthogonal Decomposition Based Optimal Neurocontrol Synthesis of a Chemical Reactor Process Using Approximate Dynamic Programming, *Neural Networks*, vol. 16, No 5-6, (June 2003) page numbers (719 - 728), ISSN 0893-6080

Padhi R., Balakrishnan S. & Randolph T. (2001). Adaptive Critic based Optimal Neuro Control Synthesis for Distributed Parameter Systems, *Automatica*, Vol. 37, No 8, (August 2001) page numbers (1223-1234), ISSN 0005-1098

Pietil S. & Koivo H.N. (1996). Centralized and Decentralized Neural Network Models for Distributed Parameter Systems, *Proceedings of CESA'96 IMACS Multiconference on Computational Engineering in Systems Applications*, pp. 1043-1048, ISBN 2-9502908-9-2, Lille, France, July 1996, Gerf EC Lille, Villeneuve d'Ascq, FRANCE

Ray, W.H. (1989). *Advanced Process Control*, pp.133-242, Butterworths Publishers, ISBN 0-409-90231-4, Boston, London, Singapore, Sydney, Toronto, Wellington

Wan E. & Beaufays F. (1996). Diagrammatic Method for Deriving and Relating Temporal Neural Networks Algorithms, *Neural Computations*, Vol. 8, No.1, (January 1996),) page numbers (182–201), ISSN 0899-7667
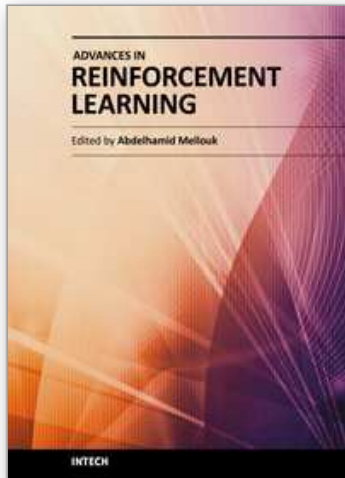
Advances in Reinforcement Learning

Edited by Prof. Abdelhamid Mellouk

Reinforcement Learning (RL) is a very dynamic area in terms of theory and application. This book brings together many different aspects of the current research on several fields associated to RL which has been growing rapidly, producing a wide variety of learning algorithms for different applications. Based on 24 Chapters, it covers a very broad variety of topics in RL and their application in autonomous systems. A set of chapters in this book provide a general overview of RL while other chapters focus mostly on the applications of RL paradigms: Game Theory, Multi-Agent Theory, Robotic, Networking Technologies, Vehicular Navigation, Medicine and Industrial Logistic.

How to reference

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

Ieroham Baruch, Rosalba Galvan-Guerra and Sergio-Miguel Hernandez M. (2011). Distributed Parameter Bioprocess Plant Identification and I-Term Control Using Decentralized Fuzzy-Neural Multi-Models, Advances in Reinforcement Learning, Prof. Abdelhamid Mellouk (Ed.), ISBN: 978-953-307-369-9, InTech, Available from: http://www.intechopen.com/books/advances-in-reinforcement-learning/distributed-parameter-bioprocess-plant-identification-and-i-term-control-using-decentralized-fuzzy-n

INTECH

open science | open minds