DECENTRALIZED AND CENTRALIZED INDIRECT ADAPTIVE I-TERM NEURAL CONTROL OF A DISTRIBUTED PARAMETER BIOPROCESS PLANT

Ieroham S. Baruch, Sergio-Miguel Hernandez and Eloy Echeverria Saldierna
Department of Automatic Control, CINVESTAV-IPN
Mexico City, Mexico
e-mails: {baruch;shernandez}@ctrl.cinvestav.mx; eloyecheve@hotmail.com

Abstract

The paper proposed to use decentralized and centralized I-term neural sliding mode control of distributed parameter wastewater anaerobic digestion bioprocess plant. The bioprocess plant is described by partial differential equations simplified by means of the orthogonal collocation method and used as an input/output data generator. The obtained process data are identified by means of decentralized and centralized neural identifiers and the issued parameter and state information is used to build up decentralized and centralized sliding mode control with I-term for each collocation point output variables. For sake of comparison some control results are given for centralized optimal control of DPS. The comparative graphical and numerical simulation results of the digestion wastewater treatment system identification and control, obtained via learning, exhibited a good convergence and precise reference tracking.

1 Introduction

In the last decade, the computational intelligence and its soft computing tools, like Artificial Neural Networks (ANN), Fuzzy Systems (FS) and its hybrid neuro-fuzzy and fuzzy-neural systems, became universal means for many applications in identification, prediction and control. Because of their approximation and learning capabilities, the ANNs have been widely employed to dynamic process modeling and control, including biotechnological plants, [1]–[10]. Among several possible neural network architectures the ones most widely used are the Feedforward NN (FFNN) and the Recurrent NN (RNN), [2]. 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, [11], [12]. The FFNN and the RNN have been applied for Distributed Parameter Systems (DPS) identification and control too [3]–[9]. In [3], a RNN is used for system identification and process prediction of a DPS dynamics - an adsorption column for wastewater treatment of water contami- nated with toxic chemicals. Similarly to the static ANNs, the fuzzy models could approximate static nonlinear plants where structural plants information is needed to extract the fuzzy rules, [13]–[16]. The aim of the fuzzy-neural models is to merge both ANN and FS approaches so to obtain fast adaptive models possessing learning, [13]. The fuzzy-neural networks are capable to incorporate 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 (fuzzy-
neural) models, summarized in [13]. To reduce the number of IF-THAN rules, the hierarchical approach could be used [13]. A promising approach of recurrent fuzzy-neural systems with internal dynamics is the application of the Takagi-Sugeno (T-S) fuzzy rules with a static premise and a dynamic functional consequent part, [13]. The paper of [13] proposed as a dynamic function in the consequent part of the T-S rules to use a Recurrent Neural Network Model (RNNM). Together with the Recurrent Trainable Neural Network (RTNN) topology, the Backpropagation (BP) learning algorithm [17], [18] is incorporated in the learning procedure taking part in the IF-THAN T-S rule antecedent. To complete the fuzzy-neural system learning, a second hierarchical defuzzification BP learning level has been formed so to improve the adaptation ability of the system, [13]. For sake of comparison a centralized sliding mode RTNN control and a centralized optimal control are considered too. The aim of this paper is to describe the results obtained by this system for decentralized identification and control of wastewater treatment anaerobic digestion bioprocess [19], representing a DPS, extending the used indirect sliding mode control law 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, [11], [12]. The analytical anaerobic bioprocess plant model [19], used as an input/output plant data generator, is described by Partial Differential Equations (PDE)/Ordinary Differential Equations (ODE) and simplified using the Orthogonal Collocation Method (OCM), [20], in four points (0.2H, 0.4H, 0.6H, 0.8H) obtaining the following system of OD equations:

\[ S_{\text{in}}(0) \]

\[ Q_{\text{in}} \]

\[ S_1 \]

\[ S_2 \]

\[ X_1 \]

\[ X_2 \]

\[ S_{\text{KL}} \]

The reduced plant model (7)–(16) could be used as unknown plant model which generate input/output process data for decentralized and centralized adaptive control system design, based on the concepts, given in [13]–[16]. 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 triangular/trapezoidal membership functions (centers of the intervals of \( z \)) corresponded to the collocation points of the simplified plant model which are in fact the four measurement points of the fixed bed plus one point for the recirculation tank. So the proposed fuzzy-neural control methodology will be applied for this DPS bioprocess plant.

2 The Anaerobic Bioprocess Plant Description

The anaerobic digestion systems block diagram is depicted on Figure 1. It consists of fixed bed bioreactor and a recirculation tank.

The physical meaning of all variables and constants of the process model are summarized on Table 1.

The complete analytical model of wastewater treatment anaerobic bioprocess, taken from [19], [14]–[16], could be described by the following system of PDE and ODE (also for the recirculation tank): For practical purpose, the full PDE anaerobic digestion process model, [19], could be reduced to an ODE system using an early lumping technique and the Orthogonal Collocation Method (OCM), [20], in four points (0.2H, 0.4H, 0.6H, 0.8H) obtaining the following system of OD equations:
<table>
<thead>
<tr>
<th><strong>Variable</strong></th>
<th><strong>Units</strong></th>
<th><strong>Name</strong></th>
<th><strong>Values</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$z$</td>
<td>$z \in [0,1]$</td>
<td>Space variable</td>
<td></td>
</tr>
<tr>
<td>$t$</td>
<td>D</td>
<td>Time variable</td>
<td></td>
</tr>
<tr>
<td>$E_z$</td>
<td>m$^2$/d</td>
<td>Axial dispersion coefficient</td>
<td>1</td>
</tr>
<tr>
<td>$D$</td>
<td>1/d</td>
<td>Dilution rate</td>
<td>0.55</td>
</tr>
<tr>
<td>$H$</td>
<td>m</td>
<td>Fixed bed length</td>
<td>3.5</td>
</tr>
<tr>
<td>$X_1$</td>
<td>g/L</td>
<td>Concentration of acidogenic bacteria</td>
<td></td>
</tr>
<tr>
<td>$X_2$</td>
<td>g/L</td>
<td>Concentration of methanogenic bacteria</td>
<td></td>
</tr>
<tr>
<td>$S_1$</td>
<td>g/L</td>
<td>Chemical Oxygen Demand</td>
<td></td>
</tr>
<tr>
<td>$S_2$</td>
<td>mmol/L</td>
<td>Volatile Fatty Acids</td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td></td>
<td>Bacteria fraction in the liquid phase</td>
<td>0.5</td>
</tr>
<tr>
<td>$k_1$</td>
<td>g/g</td>
<td>Yield coefficients</td>
<td>42.14</td>
</tr>
<tr>
<td>$k_2$</td>
<td>mmol/g</td>
<td>Yield coefficients</td>
<td>250</td>
</tr>
<tr>
<td>$k_3$</td>
<td>mmol/g</td>
<td>Yield coefficients</td>
<td>134</td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>1/d</td>
<td>Acidogenesis growth rate</td>
<td></td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>1/d</td>
<td>Methanogenesis growth rate</td>
<td></td>
</tr>
<tr>
<td>$\mu_{1\text{max}}$</td>
<td>1/d</td>
<td>Maximum acidogenesis growth rate</td>
<td>1.2</td>
</tr>
<tr>
<td>$\mu_{2\text{max}}$</td>
<td>1/d</td>
<td>Maximum methanogenesis growth rate</td>
<td>0.74</td>
</tr>
<tr>
<td>$K_1$</td>
<td>g/g</td>
<td>Kinetic parameter</td>
<td>50.5</td>
</tr>
<tr>
<td>$K_2$</td>
<td>mmol/g</td>
<td>Kinetic parameter</td>
<td>16.6</td>
</tr>
<tr>
<td>$K_3$</td>
<td>mmol/g</td>
<td>Kinetic parameter</td>
<td>256</td>
</tr>
<tr>
<td>$Q_T$</td>
<td>m$^3$/d</td>
<td>Recycle flow rate</td>
<td>0.24</td>
</tr>
<tr>
<td>$V_T$</td>
<td>m$^3$</td>
<td>Volume of the recirculation tank</td>
<td>0.2</td>
</tr>
<tr>
<td>$S_{1T}$</td>
<td>g/L</td>
<td>Concentration of Chemical Oxygen Demand in the recirculation tank</td>
<td></td>
</tr>
<tr>
<td>$S_{2T}$</td>
<td>mmol/L</td>
<td>Concentration of Volatile Fatty Acids in the recirculation tank</td>
<td></td>
</tr>
<tr>
<td>$Q_{in}$</td>
<td>m$^3$/d</td>
<td>Inlet flow rate</td>
<td>0.31</td>
</tr>
<tr>
<td>$V_B$</td>
<td>m$^3$</td>
<td>Volume of the fixed bed</td>
<td>1</td>
</tr>
<tr>
<td>$V_{eff}$</td>
<td>m$^3$</td>
<td>Effective volume tank</td>
<td>0.95</td>
</tr>
<tr>
<td>$S_{1,in}$</td>
<td>g/L</td>
<td>Inlet substrate Concentration</td>
<td></td>
</tr>
<tr>
<td>$S_{2,in}$</td>
<td>mmol/L</td>
<td>Inlet substrate Concentration</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.** Summary of the variables in the plant model
\[
\frac{dS_1}{dt} = (\mu_1 - \varepsilon D) X_1 , \quad \mu_1 = \mu_1 \max \frac{S_1}{K_i^c X_1 + S_i} , \quad (1)
\]
\[
\frac{dS_2}{dt} = (\mu_2 - \varepsilon D) X_2 , \quad \mu_2 = \mu_2 \max \frac{S_2}{K_i^c X_2 + S_i + \frac{S_2}{\Gamma_1}} , \quad (2)
\]
\[
\frac{dS_i}{dt} = \frac{E_i}{\Gamma^i} \frac{d^2 S_i}{d\tau^2} - D \frac{dS_i}{d\tau} - k_1 \mu_1 X_1 , \quad (3)
\]
\[
\frac{dS_i}{dt} = \frac{E_i}{\Gamma^i} \frac{d^2 S_i}{d\tau^2} - D \frac{dS_i}{d\tau} + k_2 \mu_1 X_1 , \quad (4)
\]
\[
S_1(0, t) = \frac{S_{1,in}(t) + RS_{in}}{R+1} , \quad S_2(0, t) = \frac{S_{2,in}(t) + RS_{in}}{R+1} , \quad R = \frac{Q_T}{D_{eff}} , \quad (5)
\]
\[
\frac{dS_1}{d\tau} (1, t) = 0 , \quad \frac{dS_2}{d\tau} (1, t) = 0 . \quad (6)
\]
\[
\frac{dX_1}{dt} = (\mu_{1,i} - \varepsilon D) X_{1,i} , \quad \frac{dX_2}{dt} = (\mu_{2,i} - \varepsilon D) X_{2,i} , \quad (7)
\]
\[
\frac{dS_{1,i}}{d\tau} = \frac{E_i}{\Gamma^i} \sum_{j=1}^{N+2} B_{i,j} S_{1,j} - D \sum_{j=1}^{N+2} A_{i,j} S_{1,j} - k_1 \mu_{1,i} X_{1,i} , \quad (8)
\]
\[
\frac{dS_{r,i}}{d\tau} = \frac{Q_T}{\Gamma^r} \left( S_{1,1} - S_{1,T} \right) , \quad \frac{dS_{r,i}}{d\tau} = \frac{Q_T}{\Gamma^r} \left( S_{2,1} - S_{2,T} \right) . \quad (9)
\]
\[
\frac{dS_{1,i}}{d\tau} = \frac{E_i}{\Gamma^i} \sum_{j=1}^{N+2} B_{i,j} S_{1,j} - D \sum_{j=1}^{N+2} A_{i,j} S_{2,j} + k_2 \mu_{1,i} X_{2,i} - k_3 \mu_{2,i} X_{2,i} , \quad (10)
\]
\[
\frac{dS_{r,i}}{d\tau} = \frac{Q_T}{\Gamma^r} \left( S_{1,N+2} - S_{1,T} \right) , \quad \frac{dS_{r,i}}{d\tau} = \frac{Q_T}{\Gamma^r} \left( S_{2,N+2} - S_{2,T} \right) , \quad (11)
\]
\[
S_{k,1} = \frac{1}{R+1} S_{k,in}(t) + \frac{R}{R+1} S_{k,T} , \quad S_{k,N+2} = \frac{K_i}{R+1} S_{k,in}(t) + \frac{K_i R}{R+1} S_{k,T} + \sum_{l=2}^{N+1} K_{k,l} \quad (12)
\]
\[
K_1 = -\frac{A_{N+1} S_{N+2} N+2}{} , \quad K_{k} = -\frac{A_{N+1} S_{N+2} N+2}{} , \quad (13)
\]
\[
A = \Lambda \phi^{-1} , \quad \Lambda = [\phi_{m,l}] , \quad \phi_{m,l} = (l-1) \phi_{m-1} , \quad (14)
\]
\[
B = \Gamma \phi^{-1} , \quad \Gamma = [\tau_{m,l}] , \quad \tau_{m,l} = (l-1) (l-2) \phi_{m-3} , \quad \phi_{m,l} = \phi_{m-1} \quad (15)
\]
\[
i = 2,...,N+2 , \quad m,l = 1,...,N+2 . \quad (16)
\]
3 Description of the Decentralized Indirect (Sliding Mode) Fuzzy-Neural Multi-Model Control System With I-Term

The block-diagram of the FNMM control system is given on Figure 2.

Figure 2. Block diagram of the indirect (SM) decentralized HFNMM control system with I-term

The structure of the entire control system, [13]-[16], 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 defuzzifier. 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 Figure 3), 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.

Figure 3. Detailed block-diagram of the FNMM identifier

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 Figure 4) has two levels – Lower Level of Control (LLC) and Upper Level of Control (ULC).

Figure 4. Detailed block-diagram of the indirect (SM) decentralized HFNMM controller with I-term

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, [13]. The block-diagram of the FNMMI, (see Figure 3), contained a space plant output fuzzyfier and
five identification T-S fuzzy rules $R_l$, which consequent parts are learning procedures, [13], given by.

$$\text{If } x(k) \text{ is } A_l \text{ and } u(k) \text{ is } B_l, \text{ then}$$

$$Y_l = \prod_l (L, M, N, Y_{dl}, U, X_l, A_l, B_l, C_l, E_l), \quad (17)$$

$$l=1-5.$$ 

The recursive I-term control algorithm is as follows:

$$U_{It}(k+1) = U_{It}(k) + T_0 K_i(k) E_{ci}(k), \quad (19)$$

$$i=1-5.$$

Where: $T_0$ is the period of discretization and $K_i(k)$ 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.:

$$K_i(k) = \eta (C_iB_i)^+. \quad (20)$$

The total control (22) is generated by the procedure incorporated in the T-S rule:

$$\text{If } Y_{di} \text{ is } A_i, \text{ then}$$

$$U_i = \prod_i (M, U_{ff_i}, U_{fb_i}, U_{Iti}), \quad (23)$$

$$i=1-5.$$ 

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

$$U = \prod (M, L, N, Y_d, Uo, X, A, B, C, E) \quad (24)$$

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:

$$Y_{zt} = \sum_{i} \gamma_{i,z} a_i T Y_t + b_i, \quad (25)$$

$$i=1,2,3,4; \quad Y_z = \sum_{i} \gamma_{i,z} a_i T Y_t + \sum_{i} \gamma_{i,z} b_i; \quad (26)$$

Here the indirect adaptive neural control algorithm, which appeared in the consequent part of the local fuzzy control rule $R_C i$ (18) is viewed as a Sliding Mode Control (SMC), [13]–[16], designed using the parameters and states issued by the corresponding identification local fuzzy rule $R_{l}$ (17), approximating the plant in the corresponding collocation point. The SMC design is considered as follows.

## 4 Description of the Centralized Indirect (Sliding Mode) Neural Control System With I-Term

Here the DPS nonlinear plant, described by ODE, is considered to have excessive high-dimensional measurements which means that the plant output dimension is greater than the plant control input one (rectangular system), requiring special reference choice, representing a data fusion technique. Furthermore the used control laws are extended 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 noise. The block-diagram of the control system is given on Figure 5. It contained...
a recurrent neural identifier RTNN 1 and a Sliding Mode (SM) controller with entries – the reference signal R, the output error Ec and the states X and parameters A, B, C, estimated by the neural identifier RTNN-1. The total control is a sum of the SM control and the I-term control, computed using the equations (22), (19), (20)

Let us suppose that the studied local nonlinear plant model possess the following structure:

\[ X_p(k+1) = F[X_p(k), U(k)]; \quad Y_p(k) = G[X_p(k)] \quad (27) \]

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:

\[ X(k+1) = AX(k) + BU(k); \quad Y(k) = CX(k) \quad (28) \]

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

\[ S(k+1) = E(k+1) + \sum_{i=1}^{P} \gamma_i E(k - i + 1); \quad |\gamma_i| < 1; \quad (29) \]

Where: \( S(.) \) is the sliding surface error function; \( E(.) \) is the systems local output tracking error; \( \gamma_i \) are parameters of the local desired error function; \( P \) is the order of the error function. The additional inequality in (29) is a stability condition, required for the sliding surface error function. The local tracking error is defined as \( E(k) = R(k) - Y(k) \), 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 \( S(k+1) = 0 \). As the local approximation plant model (28), is controllable, observable and stable, \([10]–[13]\), 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 (28), (29), taking into account the mentioned above observations, it is easy to obtain the equivalent control capable to lead the system to the sliding surface which yields:

\[ (30), (31) \]

Here the added offset \( \text{Of} \) is a learnable \( M \)-dimensional constant vector which is learnt using a simple delta rule (see, [2] 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 obtained from the output error multiplying it by the same pseudoinverse matrix, as it is (32).

If we compare the I-term expression (19), (20) with the Offset learning (32) we could see that they are equal which signified that the I-term generate a compensation offset capable to eliminate steady state errors caused by constant perturbations and discrepancies 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 a compensation offset in the SM control law (30). The SMC avoiding chattering is taken using a saturation function inside a bounded control level \( U_0 \), taking into account plant uncertainties. 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, \([10]–[12]\).
\[ U_{eq}(k) = (CB)^+ \left[ -CAX(k) + R(k+1) + \sum_{i=1}^{p} \gamma_i E(k-i+1) \right] + Of \]  
(30)

\[ (CB)^+ = \left[ (CB)^T (CB) \right]^{-1} (CB)^T. \]  
(31)

\[ Of(k+1) = Of(k+1) = Of(k) + \eta (CB)^+ E(k). \]  
(32)

\[ X_e(k+1) = A_e X_e(k) + B_e U(k) \]  
(33)

\[ U(k) = -[B_e^T P_e(k) B_e + R]^{-1} [B_e^T P_e(k) B_e] X_e(k) \]  
(34)

\[ P_e(k+1) = A_e^T [P_e(k) - P_e(k) B_e (B_e^T P_e(k) B_e + R)^{-1} B_e^T P_e(k)] A_e + Q \]  
(34A)

5 Description of the Centralized Optimal Neural Control System With I-Term

The block-diagram of the centralized optimal control system is given on Figure 6.

\[ V(k+1) = V(k) + (CB)^+ Ec(k+1) \]

Where the \((CB)^+\) transformed the output error \(Ec\) in input error \(Eu\). The state space equation of the extended system is given by the following equation (33): Where: \(X_e = [X|V]^T\) is a state vector with dimension (L+N) and:

\[ A_e = \begin{bmatrix} A & 0 \\ -(CB)^{-1}CA & I \end{bmatrix} ; B_e = \begin{bmatrix} B \\ -I \end{bmatrix} \]

The optimal I-term control is given by: (34).

Where the \(P_e\) is solution of the discrete Riccati equation:(34A).

The given up optimal control is more complicated than the SMC and here it is used for purpose of comparison.

6 Description of the RTNN Topology and Backpropagation Learning

Block-diagrams of the RTNN topology and its adjoint, are given on Figure 7 and Figure 8.

Following Figure 7 and Figure 8, we could...
\[ X(k+1) = AX(k) + BU(k); B = [B_1 ; B_0]; U^T = [U_1 ; U_2]; \]  
\[ Z_1(k) = G[X(k)]; \]  
\[ V(k) = CZ(k); C = [C_1 ; C_0]; Z^T = [Z_1 ; Z_2]; \]  
\[ Y(k) = F[V(k)]; \]  
\[ A = \text{block-diag}(A_i), |A_i| < 1; \]  
\[ W(k+1) = W(k) + \eta \Delta W(k) + \alpha \Delta W_{ij}(k-1); \]  
\[ E(k) = T(k)-Y(k); \]  
\[ E_1(k) = F'[Y(k)] E(k); F'[Y(k)] = [1-Y^2(k)]; \]  
\[ \Delta C(k) = E_1(k) Z^T(k); \]  
\[ E_2(k) = G'[Z(k)] E_2(k); E_2(k) = C^T(k) E_1(k); G'[Z(k)] = [1-Z^2(k)]; \]  
\[ \Delta B(k) = E_3(k) U^T(k); \]  
\[ \Delta A(k) = E_3(k) X^T(k); \]  
\[ \text{Vec}(\Delta A(k)) = E_3(k) X(k); \]  
\[ W(k + 1) = W(k) + P(k) \nabla Y[W(k)] E[W(k)] \]  
\[ Y[W(k)] = g[W(k) , U(k)] \]  
\[ E^2[W(k)] = \{Y_p(k) - g[W(k) , U(k)]\}^2 \]  
\[ DY[W(k)] = \frac{\partial g[W(k) , U(k)]}{\partial W} \bigg|_{W=W(k)} \]
derive the dynamic BP algorithm of its learning based on the RTNN topology using the diagrammatic method [10].

The RTNN topology and learning are described in vector-matrix form as:

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 the derivatives of these 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; ∆W (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 (47) 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 Figure 8. 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 (39). Below a theorem of RTNN stability [17], [18], is given.

6.1 Theorem of Stability of the RTNN

Let the RTNN with Jordan Canonical Structure is given by equations (35)–(39) (see Figure 7) and the nonlinear plant model, is as follows:

\[ X_d(k+1) = G[X_d(k), U(k)] \]
\[ Y_d(k) = F[X_d(k)] \]

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(\cdot), B(\cdot), C(\cdot), in general matricial form, described by equation (40)–(47) 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:

\[ L_1(k) = \frac{1}{2} \epsilon^2(k) \]
\[ L_2(k) = \text{tr}(\tilde{W}_A(k)\tilde{W}_A^*(k)) + \text{tr}(\tilde{W}_B(k)\tilde{W}_B^*(k)) + \text{tr}(\tilde{W}_C(k)\tilde{W}_C^*(k)) \]

Where:

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

vectors of the estimation error and (\hat{A}^*, \hat{B}^*, \hat{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.:

\[ L(k + 1) = L_1(k + 1) + L_2(k + 1) < 0 \]
\[ ∆L(k + 1) = L(k + 1) - L(k) \]

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

\[ \frac{(1 - \frac{1}{\gamma})}{\psi_{\text{max}}} < \eta_{\text{max}} < \frac{(1 + \frac{1}{\gamma})}{\psi_{\text{max}}} \]

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

\[ ∆L_2(k + 1) \leq -\eta_{\text{max}} |e(k + 1)|^2 - \alpha_{\text{max}} |e(k)|^2 + d(k + 1) \]

Note that η_{\text{max}} changes adaptively during the RTNN learning and:

\[ η_{\text{max}} = \max_{i=1}^3 \{ η_i \} \]

Where all: the unmodelled dynamics, the approximation errors and the perturbations, are represented by the d-term. Applying the Lemma of RTNN convergence for the given above result for L_2(k + 1) < 0 we could conclude that: the d-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 obtained: $$A(k) \in L_\infty$$, $$B(k) \in L_\infty$$, $$C(k) \in L_\infty$$, $$D(k) = I$$ through it. Applying equation (51) for each updated weight, propagating the value $$D(k) = I$$ through it. Applying equation (51) for each element of the weight matrices (A, B, C) in order to be updated, the corresponding gradient components are obtained as follows: (52)–(60).

### 6.2 Lemma of RTNN Convergence

Applying the limit’s definition, the identification error bound condition is obtained as:

$$\lim_{t \to \infty} \sum_{i=1}^{k} \left( |e_i|^2 + |e_{i-1}|^2 \right) \leq d$$

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

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

And iterating from $$k=0$$, we get:

$$L_2(k+1) - V(0) \leq - \sum_{t=1}^{k} |e_t|^2 - \sum_{t=1}^{k} |e_{t-1}|^2 + dk$$

$$\sum_{k=1}^{k} \left( |e_t|^2 + |e_{t-1}|^2 \right) \leq dk - L_2(k+1) + L_2(0)$$

$$dk - L_2(k+1) + L_2(0) \leq dk + L_2(0)$$

After this, let us divide by k and applying the limit’s definition, the identification error bound condition is obtained in the final form:

$$\lim_{t \to \infty} \sum_{i=1}^{k} \left( |e_i|^2 + |e_{i-1}|^2 \right) \leq d$$

From here we can see that the term d must be bounded by weight matrices and the learning parameter, in order to obtain:

$$\Delta L_2(k) \in L_\infty$$

### 7 Description of the RTNN Levenberg-Marquardt Learning

The Levenberg-Marquardt (L-M) recursive algorithm of learning, [18], could be considered as a continuation of the BP algorithm and it will be used here. Following Figure 7, the RTNN topology could be described in vector-matrix form as it is given by the equations (35)–(39). The general recursive L-M algorithm of learning, [18], is given by the following equations:(48)–(51)

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; X 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 Figure 8), it was possible to obtain the values of the gradients DY[,] for each updated weight, propagating the value D(k) = 1 through it. Applying equation (51) for each element of the weight matrices (A, B, C) in order to be updated, the corresponding gradient components are obtained as follows: (52)–(60).

The matrix $$\Omega(.)$$ had a dimension (nw×2), whereas the second row had only one unity element (the others were zero). The position of that element was computed by:

$$i = kmod(nw) + 1; \quad k > nw \quad (61)$$

After this, the given topology and learning are incorporated in the decentralized T-S identification and control rules or applied directly as RTNN identifier as part of an indirect centralized control scheme. Then it is applied for an anaerobic wastewater distributed parameter decentralized or centralized system identification and control in each collocation point.

### 8 Simulation Results

#### 8.1 Simulation Results of Decentralized and Centralized System Identification

**Results of decentralized system identification:** In this paragraph, graphical and numerical simulation results of decentralized system identification are 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. The Figs. 9–11 showed graphical simulation results of open loop decentralized plant identification of X1 using L-M learning. For sake of comparison, some results for
\[
DY[C_{ij}(k)] = D_{1,i}(k)Z_j(k), \quad D_{1,i}(k) = F'_{j}[Y_i(k)] \tag{52}
\]
\[
DY[A_{ij}(k)] = D_{2,i}(k)X_j(k), \tag{53}
\]
\[
DY[B_{ij}(k)] = D_{2,i}(k)U_j(k), \tag{54}
\]
\[
D_{2,i}(k) = G'_{i}[Z_j(k)]C_iD_{1,i}(k). \tag{55}
\]

Therefore the Jacobean matrix could be formed as:

\[
DY[W(k)] = [DY(C_{ij}(k)), DY(A_{ij}(k)), DY(B_{ij}(k))]. \tag{56}
\]

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

\[
P(k) = \alpha^{-1}(k) \{P(k-1) - P(k-1)\Omega[W(k)]S^{-1}[W(k)]\Omega^T[W(k)]P(k-1)\} \tag{57}
\]

Where the \( S(\cdot) \) and \( \Omega(\cdot) \) matrices were given as follows:

\[
S[W(k)] = \alpha(k)\Lambda(k) + \Omega^T[W(k)]P(k-1)\Omega[W(k)], \tag{58}
\]
\[
\Omega^T[W(k)] = \begin{bmatrix}
\nabla^TY^T[W(k)] - \frac{1}{\alpha(k)} & \cdots & 0
\end{bmatrix}; \tag{59}
\]
\[
\Lambda(k)^{-1} = \begin{bmatrix}
1 & 0 \\
0 & \rho
\end{bmatrix}; \quad 10^{-4} \leq \rho \leq 10^{-6}; \quad 0.97 \leq \alpha(k) \leq 1; \quad 10^3 \leq P(0) \leq 10^6. \tag{60}
\]
Figure 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 T-S fuzzy rules with RTNNs (dotted line-RTNN output, continuous line-plant output) for 600 iteration of L-M RTNN learning.

Figure 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 T-S fuzzy rules with RTNNs (dotted line-RTNN output, continuous line-plant output) for the first 10 iterations of the L-M RTNN learning.
initial short time of X1 BP learning are given on Figure 12.

**Figure 11.** Graphics of the 3d view of X1 space/time approximation during its L-M RTNN learning in four points

The input signals applied for system identification are:

The MSE of the decentralized FNMM approximation of all output plant variables in all collocation points, using the L-M and BP learning are shown in Tables 2, 3.

The topology of the first four identification 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, incorporated in T-S identification rules, 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 two plant inputs are $S_{1in}$ (concentration of acidogenic bacteria in the substrate) and $S_{2in}$ (concentration of methanogenic bacteria in the substrate). For lack of space we shall show some graphical results only for the X1 variable. The topology of the RTNN-1 is (2, 20, 18), the activation functions are tanh(.) for both layers. The learning rate parameters for the BP algorithm of learning are $\alpha=0$, $\eta=0.4$ and for the L-M learning - the forgetting factor is $\alpha=1$, the regularization constant is $\rho=0.001$ and the initial value of the P matrix is an identity matrix with dimension 420x420. The simulation results of RTNN-1 system identification are obtained on-line during 400 days with a step of 0.5 day. The identification inputs used are the same given above as (62), (63). The Figure 13 showed the initial short time L-M identification of the variable $X_1$ in four collocation points. The Figure 14 showed the initial short time BP identification of the same variable $X_1$ in the same four collocation points.

From Figure 13 and Figure 14 we could see that both algorithms reached fast the optimum point but the L-M algorithm is more precise and more complicated than the BP learning algorithm.

**Some results of centralized system identification:** The centralized RTNN identified 18 output plant variables, which are 4 variables for each collocation point $z=0.2H$, $z=0.4H$, $z=0.6H$, $z=0.8H$ of the fixed bed as: $X_1$ (acidogenic bacteria), $X_2$ (methanogenic bacteria), $S_1$ (chemical oxygen demand) and $S_2$ (volatile fatty acids) and the next variables in the recirculation tank: $S_{1T}$ (chemical oxygen demand) and $S_{2T}$ (volatile fatty acids). The two plant inputs are $S_{1in}$ (concentration of acidogenic bacteria in the substrate) and $S_{2in}$ (concentration of methanogenic bacteria in the substrate). For lack of space we shall show some graphical results only for the X1 variable. The topology of the RTNN-1 is (2, 20, 18), the activation functions are tanh(.) for both layers. The learning rate parameters for the BP algorithm of learning are $\alpha=0$, $\eta=0.4$ and for the L-M learning - the forgetting factor is $\alpha=1$, the regularization constant is $\rho=0.001$ and the initial value of the P matrix is an identity matrix with dimension 420x420. The simulation results of RTNN-1 system identification are obtained on-line during 400 days with a step of 0.5 day. The identification inputs used are the same given above as (62), (63). The Figure 13 showed the initial short time L-M identification of the variable $X_1$ in four collocation points. The Figure 14 showed the initial short time BP identification of the same variable $X_1$ in the same four collocation points.

From Figure 13 and Figure 14 we could see that both algorithms reached fast the optimum point but the L-M algorithm is more precise and more complicated than the BP learning algorithm.

8.2 Simulation Results of Decentralized and Centralized Indirect (Sliding Mode) System Control with I-term

**Results of decentralized indirect (SM) system control with I-term:** The Figs. 15–20 showed graphical simulation results of the indirect (sliding mode) decentralized HFNMM with and without I-term control.
Figure 12. Detailed graphical simulation results of the FNMM identification of $X_1$ in a) $Z=0.2H$; b) $0.4H$; c) $0.6H$; d) $0.8H$ (acidogenic bacteria in the corresponding fixed bed points) by four T-S fuzzy rules with RTNNs (dotted line-RTNN output, continuous line-plant output) for the first 10 iterations of the BP RTNN learning

\[ S_{1,in} = 0.55 + 0.15 \cos \left( \frac{3\pi}{30} t \right) + 0.3 \sin \left( \frac{\pi}{30} t \right), \quad (62) \]

\[ S_{2,in} = 0.55 + 0.05 \cos \left( \frac{3\pi}{40} t \right) + 0.3 \sin \left( \frac{\pi}{40} t \right). \quad (63) \]

<table>
<thead>
<tr>
<th>Collocation point</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$S_1/S_{1T}$</th>
<th>$S_2/S_{2T}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z=0.2$</td>
<td>$0.0013$</td>
<td>$0.0012$</td>
<td>$0.0049$</td>
<td>$0.0058$</td>
</tr>
<tr>
<td>$z=0.4$</td>
<td>$0.0013$</td>
<td>$0.0013$</td>
<td>$0.0058$</td>
<td>$0.0049$</td>
</tr>
<tr>
<td>$z=0.6$</td>
<td>$0.0013$</td>
<td>$0.0013$</td>
<td>$0.0071$</td>
<td>$0.0055$</td>
</tr>
<tr>
<td>$z=0.8$</td>
<td>$0.0014$</td>
<td>$0.0013$</td>
<td>$0.0083$</td>
<td>$0.0070$</td>
</tr>
<tr>
<td>Recirculation tank</td>
<td></td>
<td></td>
<td>$0.0080$</td>
<td>$0.0058$</td>
</tr>
</tbody>
</table>

Table 2. MSE of the decentralized FNMM approximation of the all bioprocess output variables in all collocation points, using the L-M RTNN learning

<table>
<thead>
<tr>
<th>Collocation point</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$S_1/S_{1T}$</th>
<th>$S_2/S_{2T}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z=0.2$</td>
<td>$0.0015$</td>
<td>$0.0023$</td>
<td>$0.0145$</td>
<td>$0.0192$</td>
</tr>
<tr>
<td>$z=0.4$</td>
<td>$0.0015$</td>
<td>$0.0044$</td>
<td>$0.0098$</td>
<td>$0.0164$</td>
</tr>
<tr>
<td>$z=0.6$</td>
<td>$0.0030$</td>
<td>$0.0009$</td>
<td>$0.0092$</td>
<td>$0.0133$</td>
</tr>
<tr>
<td>$z=0.8$</td>
<td>$0.0046$</td>
<td>$0.0048$</td>
<td>$0.0045$</td>
<td>$0.0086$</td>
</tr>
<tr>
<td>Recirculation tank</td>
<td></td>
<td></td>
<td>$0.0168$</td>
<td>$0.0253$</td>
</tr>
</tbody>
</table>

Table 3. MSE of the decentralized FNMM approximation of the all bioprocess output variables in all collocation points, using the BP RTNN learning
Figure 13. Detailed graphical simulation results of the centralized RTNN 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 RTNNs (dotted line-RTNN output, continuous line-plant output) for the first 50 iterations of the L-M RTNN learning.

Figure 14. Detailed graphical simulation results of the centralized RTNN 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 RTNNs (dotted line-RTNN output, continuous line-plant output) for the first 50 iterations of the BP RTNN learning.
Figure 15. Graphical simulation results of the control signals generated by the indirect (SM) decentralized HFNM control system with I-term.

Figure 16. Graphical simulation results of the indirect (SM) decentralized HFNMM I-term control of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-systems reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for 600 iterations.
Figure 17. Detailed graphical simulation results of the indirect (SM) 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

Figure 18. Graphics of the 3d view of X1 space/time approximation and indirect (SM) decentralized HFNMM I-term control in four collocation points of the fixed bed

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. The Figure 15 showed the control signals generated by the decentralized indirect HFNMM control system with I-term. This graphics showed that the issued control signals are smooth and without noise. The MSE of I-term decentralized control using L-M learning for each output signal and each measurement point are given on Tables 4. The MSE of proportional (without I-term) decentralized control using L-M learning for each output signal and each measurement point are given on Table 5.

The neural network used as a hierarchical control system defuzzyfier has the topology (10–2) with BP learning parameters $\eta=0.005$ and $\alpha=0.00006$. For the simulation with the HFNMM I-term indirect control we use $U_0=1$ and $\gamma=0.8$. In the integral term we used the parameters for the offset (Of) learning, $\eta=0.01$ and $\alpha=1e-8$. The graphical and numerical results of indirect decentralized HFNMM I-term control (see Figures 15–18 and Tables 4) showed a good reference tracking (MSE is of 0.0089 for the I-term control in the worse case). The graphical and numerical results of the same control system without I-term (see Figures 19, 20 and Table 5) showed a bad reference tracking and static error due to constant perturbations (the MSE is 0.0139 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 (without I-term) could not (compare Figures 16, 17, with Figures 19, 20 to see the difference).

Some results of centralized indirect (SM) system control with I-term: In the case of centralized indirect adaptive I-term control, the control signal is sum of the I-term control signal and the SM control signal, computed using the state and parameter information issued from the RTNN-1 neural identifier (see eqns. (30), (31)). Some X1 centralized indirect
Figure 19. Graphical simulation results of the indirect (SM) decentralized HFNMM proportional control (without I-term) of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-systems reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for 600 iterations

Figure 20. Detailed graphical simulation results of the indirect (SM) 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 15 iterations
control simulation results with I-term are given on Figures 21, 22. The given on Figures 21, 22 graphical results of I-term SMC showed smooth exponential behavior, fast convergence and the removal of the constant noise terms and uncertainties. The Figures 23 illustrated the behavior of the SMC system without I-term perturbed by a constant noise. The Figure 23 showed that the constant perturbation in the input of the plant caused a deviation of the plant output X1 with respect of the set point R1 and this occurred for all other plant output signals and measurement points.

Some results of centralized optimal control with I-term using neural identifier and L-M learning: The I-term extended the identified local linear plant model so it is part of the indirect optimal control algorithm. The Figures 24, 25 illustrated the centralized optimal control with I-term. The given on Figures 24, 25 graphical results of I-term optimal control showed smooth exponential behaviour, fast convergence and the removal of the constant noise terms when the I-term is used. The results showed that the I-term control eliminated constant disturbances and approximation errors and the proportional control could not.

9 Conclusions

The paper proposed to use decentralized recurrent fuzzy-neural identification and centralized recurrent neural identification of an anaerobic digestion wastewater treatment bioprocess, composed by a fixed bed and a recirculation tank, represented a nonlinear DPS. The simplification of the PDE process model by ODE is realized using the OCM in four collocation points (plus one of the recirculation tank) represented centers of membership functions of the space fuzzied output variables in the decentralized case. The centralized case could be considered as a fusion system with excessive measurements. The applied L-M algorithm of learning exhibited a fast convergence and great precision exceeding the BP algorithm in both cases. The obtained from the decentralized and centralized identifiers state and parameter information is used by decentralized and centralized indirect (Sliding Mode) controllers. For sake of comparison, some simulation results are given for the centralized indirect optimal control. The applied fuzzy-neural and neural approaches to the DPS decentralized and centralized indirect identification and I-term control exhibited a good convergence and precise reference tracking in both cases outperforming the centralized indirect optimal control. Finally, it could be said that the decentralized control posses more flexibility applying RTNN with small dimension in each

<table>
<thead>
<tr>
<th>Collocation point</th>
<th>X1</th>
<th>X2</th>
<th>S1/ST</th>
<th>S2/ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>z=0.2</td>
<td>0.0010</td>
<td>0.0011</td>
<td>0.0052</td>
<td>0.0089</td>
</tr>
<tr>
<td>z=0.4</td>
<td>0.0007</td>
<td>0.0009</td>
<td>0.0040</td>
<td>0.0084</td>
</tr>
<tr>
<td>z=0.6</td>
<td>0.0006</td>
<td>0.0009</td>
<td>0.0037</td>
<td>0.0063</td>
</tr>
<tr>
<td>z=0.8</td>
<td>0.0006</td>
<td>0.0008</td>
<td>0.0034</td>
<td>0.0061</td>
</tr>
<tr>
<td>Recirculation tank</td>
<td></td>
<td></td>
<td>0.0051</td>
<td>0.0074</td>
</tr>
</tbody>
</table>

Table 4. MSE of the indirect (SM) decentralized HFNMM I-term control of all output plant variables in all collocation points

<table>
<thead>
<tr>
<th>Collocation point</th>
<th>X1</th>
<th>X2</th>
<th>S1/ST</th>
<th>S2/ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>z=0.2</td>
<td>0.0013</td>
<td>0.0018</td>
<td>0.0101</td>
<td>0.0139</td>
</tr>
<tr>
<td>z=0.4</td>
<td>0.0010</td>
<td>0.0016</td>
<td>0.0083</td>
<td>0.0125</td>
</tr>
<tr>
<td>z=0.6</td>
<td>0.0008</td>
<td>0.0014</td>
<td>0.0068</td>
<td>0.0104</td>
</tr>
<tr>
<td>z=0.8</td>
<td>0.0007</td>
<td>0.0012</td>
<td>0.0057</td>
<td>0.0085</td>
</tr>
<tr>
<td>Recirculation tank</td>
<td></td>
<td></td>
<td>0.0070</td>
<td>0.0095</td>
</tr>
</tbody>
</table>

Table 5. MSE of the indirect (SM) decentralized HFNMM proportional control (without I-term) of all output plant variables in all collocation points
Figure 21. Graphical simulation results of the indirect (SM) centralized RTNN I-term control of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-systems reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for 1000 iterations.

Figure 22. Detailed graphical simulation results of the indirect (SM) centralized RTNN 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 25 iterations.
Figure 23. Graphical simulation results of the indirect (SM) centralized RTNN proportional control (without I-term) of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-systems reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for 1000 iterations.

Figure 24. Graphical simulation results of the centralized I-term optimal control of X1 (acidogenic bacteria in the fixed bed) (dotted line-plant output, continuous-systems reference) in four collocation points (a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H) for 1000 iterations.
DECENTRALIZED AND CENTRALIZED INDIRECT ADAPTIVE I-TERM NEURAL... 177

Figure 25. Detailed graphical simulation results of the centralized I-term optimal 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 25 iterations

collocation point in spite of the centralized control using a huge RTNN with great dimension coordinated all collocation points.

References


JOURNAL OF ARTIFICIAL INTELLIGENCE AND SOFT COMPUTING RESEARCH (JAISCR) is published quarterly by the Polish Neural Network Society and the Academy of Management (SWSPiZ) in Lodz, Poland.

PUBLISHING AND EDITORIAL OFFICE:
Academy of Management (SWSPiZ)
Information Technology Institute (ITI)
Sienkiewicza 9
90-113 Lodz
Tel.: +48 42 6646654
Fax.: +48 42 6366251
E-mail: jaiscr@kik.pcz.pl
URL: http://jaiscr.eu

Print: Drukarnia GREEN, pl. Komuny Paryskiej 4, 90-007 Lodz, tel.: +48 42 6322713

Copyright © 2011 Academy of Management (SWSPiZ), Lodz, Poland. All rights reserved.

AIMS AND SCOPE:
Journal of Artificial Intelligence and Soft Computing Research is a refereed international journal whose focus is on the latest scientific results and methods constituting soft computing. The areas of interest include, but are not limited to:

Artificial Intelligence in Modelling and Simulation
Artificial Intelligence in Scheduling and Optimization
Bioinformatics
Computer Vision
Data Mining
Distributed Intelligent Processing
Evolutionary Design
Expert Systems
Fuzzy Computing with Words
Fuzzy Control
Fuzzy Logic
Fuzzy Optimisation
Hardware Implementations
Intelligent Database Systems
Knowledge Engineering
Multi-agent Systems
Natural Language Processing
Neural Network Theory and Architectures
Robotics and Related Fields
Rough Sets Theory: Foundations and Applications
Speech Understanding
Supervised and Unsupervised Learning
Theory of Evolutionary Algorithms
Various Applications