Decentralized Direct and Indirect I-Term Adaptive Fuzzy-Neural Control of a Bioprocess Plant

Ieraham S. Baruch, Sergio M. Hernandez*, Eloy Echeverria S.*
Department of Automatic Control, CINVESTAV-IPN, 07360 Mexico City, MEXICO
e-mail: {baruch;shenandez;eecheve}@ctrl.cinvestav.mx

Abstract—The paper proposed to use a recurrent neural network model, and a real-time Levenberg-Marquardt algorithm of its learning for decentralized fuzzy-neural data-based modeling, identification and control of an anaerobic digestion bioprocess, carried out in a fixed bed and a recirculation tank of a wastewater treatment system. The analytical model of the digestion bioprocess, used as process data generator, represented a distributed parameter system, which is reduced to a lumped system using the orthogonal collocation method, applied in four collocation points plus one-in the recirculation tank. The paper proposed to use direct adaptive integral plus states fuzzy-neural control, and indirect adaptive I-term sliding mode fuzzy-neural control. The comparative graphical simulation results of the digestion wastewater treatment system control, exhibited a good convergence and precise reference tracking, giving slight priority to the direct control with respect to the indirect control applied.

Keywords-decentralized direct adaptive fuzzy-neural control with I-term; decentralized indirect fuzzy-neural sliding mode control; hierarchical fuzzy-neural multi model identifier; anaerobic digestion wastewater bioprocess plant; distributed parameter system; Levenberg-Marquardt learning; Takagi-Sugeno fuzzy rules with recurrent neural network antecedent parts.

I. INTRODUCTION

In the last decade, the Computational Intelligence tools (CI), like Artificial Neural Networks (ANN), Fuzzy Systems (FS), and its hybrid neuro-fuzzy, [1], [2], and fuzzy-neural, [3], systems, became universal means for many applications in identification, and control. Because of their approximation and learning capabilities, the ANNs have been widely employed to dynamic process modeling and control, including biotechnological plants, [4]-[13]. Among several possible neural network architectures the ones most widely used are the Feedforward NN (FFNN) and the Recurrent NN (RNN), [5]. 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 [5]. 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, [14], [15]. The FFNN and the RNN have been applied for Distributed Parameter Systems (DPS) identification and control too [6]-[13]. In [6], 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. Similarly to the static ANNs, the fuzzy models could approximate static nonlinear plants where structural plant’s information is needed to extract the fuzzy rules, [16]-[18]. The aim of the fuzzy-neural models is to merge both ANN and FS approaches so to obtain fast adaptive models possessing learning, [16]. The fuzzy-neural model is 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 [16]. To reduce the number of IF-THAN rules, the hierarchical approach could be used [16]. 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, [16]. The paper of [16] 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 RNN topology, the Backpropagation (BP) learning algorithm [16] 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, [16]. In [17], [18] it is proposed to control the continuous DPS wastewater bioprocess plant, described in [19], [20], using recurrent neural networks. The applied Recurrent Trainable Neural Network (RTNN) topology and Backpropagation (BP) learning are described in [7], [14]. The aim of this paper is to extend the obtained in [17], [18] results of decentralized DPS bioprocess control using the Levenberg-Marquardt learning algorithm, [21], and incorporating an I-term, [14], [15], in the direct and indirect fuzzy-neural control to augment the system resistance to imperfections and noise.

*The Ph.D. students Sergio-Miguel Hernandez M., and Eloy Echeverria S., are thankful to CONACYT, Mexico for the fellowships received during their studies in DCA, CINVESTAV-IPN, Mexico.
II. DESCRIPTION OF THE DIRECT FUZZY-NEURAL MULTI-MODEL CONTROL SYSTEM WITH I-TERM

The block-diagrams of the complete control system and its identification and control parts are depicted in Figs. 1, 2 and 3.

Figure 1. Block diagram of the FNMM control system

Figure 2. Detailed block-diagram of the FNMM identifier

Figure 3. Detailed block-diagram of the direct HFNMM controller

The structure of the entire control system [16]-[18] 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 Figures 1, 2, 3 for more details). 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, [17]. 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 Figures 1, 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 Figure 2), contained a space plant output fuzzyfier and five identification T-S fuzzy rules, labeled as RIi, which consequent parts are L-M RTNN learning procedures, [17], [18], [21]. The identification T-S fuzzy rule is given by:

\[
\text{RI}_i: \text{If } x(k) \text{ is } A_i \text{ and } u(k) \text{ is } B_i \text{ then } Y_i = \prod_i (L,M,N_i,Y_{di},U,X_i,A_i,B_i,C_i,E_i), \text{ } i=1-5
\]  

The detailed block-diagram of the FNMMC, given on Figure 3, contained a spaced plant output fuzzyfier and five 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, [16]-[18], 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_{fbi}, outputs used by the total control rule. The T-S fuzzy rule has the form:

\[
\text{RCFF}_i: \text{If } R(k) \text{ is } B_i \text{ then } U_{ffi} = \prod_i (M, L, N_i, R_i, Y_{di}, X_i, J_i, B_i, C_i, E_{ci}), \text{ } i=1-5
\]  

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:
If $Y_{d,t}$ is $A_i$, then

$$Uf_{bi} = \Pi_i (M, L, N, Y_{d,t}, X, J_i, B_i, C_i, E_{c,i}), i=1-5.$$  \hspace{1cm} (3)

The I-term control algorithm is as given by:

$$U_{li} (k+1) = U_{li} (k) + To K_i (k) E_{ci} (k), i=1-5; \hspace{1cm} (4)$$

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.:

$$K_i (k) = \frac{\eta}{(C_i B_i)^+}. \hspace{1cm} (5)$$

The product of the pseudo-inverse $(C_i B_i)^+$ by the output error $E_{ci} (k)$ transformed 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:

If $Y_{d,t}$ is $A_i$, then

$$U_{Iti} = \Pi_i (M, L, B_i, C_i, E_{ci}, To, \eta), i=1-5. \hspace{1cm} (6)$$

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

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

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

If $Y_{d,t}$ is $A_i$, then

$$U_i = \Pi_i (M, U_{ffi}, U_{fbi}, U_{Iti}), i=1-5. \hspace{1cm} (8)$$

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

$$U = \Pi (M, L, N, Y_{d,t}, U_0, X, A, B, C, E) \hspace{1cm} (9)$$

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:

If $Y_{x,t}$ is $A_i$, then

$$Y_{x,t} = a_i^T Y_t + b_i, i=1,2,3,4; \hspace{1cm} (10)$$

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

III. DESCRIPTION OF THE INDIRECT (SLIDING MODE) FUZZY-NEURAL MULTI-MODEL CONTROL SYSTEM WITH I-TERM

The block-diagram of the FNMM control system is given on Figure 4. The structure of the entire control system, [16]-[18], 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 fuzzified 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 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 indirect FNMM controller (see Figures 4, 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, [17]. The block-diagram of the FNMMI, (see Figure 2), contained a space plant output fuzzyfier and five identification T-S fuzzy rules RI, which consequent parts are L-M learning procedures, [18], [21], given by (1).

The block-diagram of the FNMMC, given on Figure 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, [17], [18], using the state, and parameter information, issued by the corresponding identification rules. The SMC T-S fuzzy rules have the form:

\[ RC_i: \text{If } R(k) \text{ is } C_i \text{ then} \]
\[ U_{ismc}(k) = \prod_{i=1}^{M} (M, L, N_i, R_i, Y_{di}, X_i, A_i, B_i, C_i, E_{ci}), i=1-5 \]

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

\[ U_i(k) = U_{ismc}(k) + U_{iti}(k), i=1-5 \]

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

\[ RCT_i: \text{If } Y_{di} \text{ is } A_i \text{ then} \]
\[ U_i = \prod_{i=1}^{M} (M, U_{ismc}, U_{iti}), i=1-5 \]

The defuzzification L-M learning procedure, [21], which corresponds to the single layer perceptron learning, is given by (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 (10), (11). The indirect (SMC) adaptive neural control algorithm, which appeared in the consequent part of the local fuzzy control rule \( RCl \), (12) is a feedback control, using the parameters and states issued by the identification local fuzzy rule \( Rli \) (1).

### A. Sliding Mode Control Systems Design

Here the indirect adaptive control algorithm, which appeared in the consequent part of the local fuzzy control rule \( RCl \) (12) is viewed as a Sliding Mode Control (SMC), [17], [18], [21], designed using the parameters and states issued by the correspondent identification local fuzzy rule \( Rli \) (1), approximating the plant in the corresponding collocation point. Let us suppose that the studied local nonlinear plant model possess the following structure:

\[ X_p(k+1) = F[X_p(k), -U_p(k)]; Y_p(k) = G[X_p(k)] \]

Where: \( X_p(k), Y_p(k), U(k) \) are plant state, output and input vector variables with dimensions \( Np, L \) 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:

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

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

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

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 (17) is a stability condition, required for the sliding surface error function. The local tracking error is defined as:

\[ E(k) = R(k) - Y(k); \]

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 (16), is controllable, observable and stable, [17], [18], [21], 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)-(18), it is easy to obtain the equivalent control capable to lead the system to the sliding surface which yields:

\[ U_{eq}(k) = (CB)^T \left[ -CA(k) + R(k) + \sum_{i=1}^{P} \gamma_i E(k-i+1) \right] \]

\[ (CB)^T = \left[ (CB)^T (CB) \right]^{-1} (CB)^T. \]
IV. DESCRIPTION OF THE RTNN TOPOLOGY AND LEVENBERG-MARQUARDT LEARNING ALGORITHM

The block-diagram of the RTNN topology is given on Figure 6. Following Figure 6, we could derive the dynamic BP algorithm of its learning based on the RTNN topology and its adjoint obtained using the diagrammatic method, [21].

![Figure 6: Block diagram of the RTNN model](image)

The Levenberg–Marquardt (L-M) recursive algorithm of learning, [21], could be considered as a continuation of the BP algorithm and it will be used here. The RTNN topology could be described in vector-matrix form as it is (see Figure 6):

\[ X(k+1) = AX(k) + BU(k); \]
\[ Z(k) = G[X(k)]; \]
\[ V(k) = CZ(k); \]
\[ Y(k) = F[V(k)]; \]
\[ A = \text{block-diag}(A_i), |A_i| < 1; \]
\[ W = \text{general weight matrix (A, B, C)} \]

The general recursive L-M algorithm of learning, [21], is given by the following equations:

\[ W(k+1) = W(k) + P(k)Y^2(k)W(k)E^2(k) \]
\[ Y(k) = g[W(k), U(k)] \]
\[ E(k) = Y(k) - g[W(k), U(k)] \]
\[ D(k) = \frac{\partial g}{\partial W}[W(k), U(k)] \]

Where: \( W \) is a general weight matrix (A, B, C) under modification; \( P \) is the covariance matrix of the estimated weights, updated; \( D(k) \) is the Nw-dimensional gradient vector; \( Y \) is the RTNN output vector which depends on the updated weights and the input; \( E \) is an error vector; \( Y_p \) is the plant output vector, which is in fact the target vector. Using the RTNN adjoint block diagram, [21], obtained by the RTNN topology (see Figure 6) it was possible to obtain the values of the gradients \( D(k) \) for each updated weight, propagating the value \( D(k) = 1 \) through it. Applying equation (29) for each element of the weight matrices (A, B, C) in order to be updated, the corresponding gradient components are as follows:

\[ D(k) = \frac{\partial g}{\partial W}[W(k), U(k)] \]

Therefore the Jacobean matrix could be formed as:

\[ D(k) = \text{computation of } D(k) \]

![Image](image)

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

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

where the \( S(k) \) and \( \Omega(k) \) matrices were given as follows:

\[ S(k) = \alpha(k)\Lambda(k) + \Omega^2(k)W(k)P(k-1)W(k), \]
\[ \Lambda(k) = \left[ \begin{array}{cc} 1 & 0 \\ 0 & \rho \end{array} \right]; \]
\[ 0.97 \leq \alpha(k) \leq 1; \]
\[ 10^{-3} \leq \rho \leq 10^{-5}. \]

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

V. SIMULATION RESULTS

A. Simulation Results of Systems Identification

In this paragraph, graphical and numerical simulation results of system identification, direct and indirect control, with and without I-term, obtained by the given in [17]-[19] bioprocess plant, 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. 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 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), [19]. The graphical simulation results of RTNNs L-M learning, [21], are obtained on-line during 600 iterations of L-M learning with a step of 0.1 sec. The L-M learning rate parameters of RTNN have small values which are different for the different measurement point variables (\( \rho=0.1 \) and \( \alpha=0 \)). The Figures 7-9 showed graphical simulation results of open loop decentralized plant identification.

The MSE of the decentralized FNMM approximation of all plant variables in all collocation points, using the L-M learning are shown in Table 1. The input signals applied for system identification are:
The graphical and numerical results of decentralized FNMM identification (see Figures 7-9, and Table 1) showed a good convergence and precise plant output tracking (MSE is 0.0083 for the L-M RTNN learning in the worse case).

### B. Simulation Results of the Decentralized Direct FNMMC

In the case of direct decentralized control, the topology of the first four control RTNNs is (12-14-2) for the variables in the fixed bed collocation points $z=0.2H$, $z=0.4H$, $z=0.6H$, $z=0.8H$ and for the recirculation tank it 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 L-M learning parameters of RTNN are $\rho=0.2$ and $\alpha=1$, while the parameter of the I-term is $\eta=0.01$. Finally, the topology of the defuzzyfier neural network is (10-2) with learning parameters $\eta=0.0035$ and $\alpha=0.00001$ of the BP learning. The Figures 10-12 showed graphical simulation results of the direct decentralized HFNMM control with 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. For sake of comparison, the Figure 13 show the graphical results of direct decentralized HFNMM proportional control (without I-term) for $X_1$ output in four measurement points. The final MSE of DANC with L-M learning for each output signal and each measurement point are given on Table 2.
The comparison between the process graphics on Figure 10 and Figures 13 showed a great notable difference. The results showed 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 control (see Figures 10-12, and Tables 2) showed a good reference tracking (MSE is of 0.0097 for the I-term DANC 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.

C. Simulation Results of the Decentralized Indirect FNMMC

The neural network used as defuzzifier in the control with BP learning rule has the topology (10-2) with learning parameters \( \eta = 0.005 \) and \( \alpha = 0.00006 \). For the simulation with the HFNMM I-term indirect control we use a saturation level \( U_0 = 1 \) and parameter \( \gamma = 0.8 \) for the SMC. In the integral term, \( (4) \), we used the parameter \( \eta = 0.01 \), (see \( (5) \)). The Figures 14, 15 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 Tables 3.

TABLE II. MSE OF THE DIRECT DECENTRALIZED HFNMM I-TERM CONTROL OF THE BIOPROCESS PLANT

<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>0.0011</td>
<td>0.0013</td>
<td>0.0065</td>
<td>0.0097</td>
</tr>
<tr>
<td>( z = 0.4 )</td>
<td>0.0009</td>
<td>0.0011</td>
<td>0.0051</td>
<td>0.0090</td>
</tr>
<tr>
<td>( z = 0.6 )</td>
<td>0.0008</td>
<td>0.0011</td>
<td>0.0042</td>
<td>0.0074</td>
</tr>
<tr>
<td>( z = 0.8 )</td>
<td>0.0006</td>
<td>0.0010</td>
<td>0.0037</td>
<td>0.0063</td>
</tr>
<tr>
<td>Recirc. tank</td>
<td></td>
<td></td>
<td>0.0060</td>
<td>0.0086</td>
</tr>
</tbody>
</table>

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 graphical and numerical results of indirect decentralized HFNMM control (see Figures 14, 15, and Tables 3) showed a good reference tracking (MSE is of 0.0089 for I-term IANC in the worse case).

TABLE III. MSE OF THE INDIRECT DECENTRALIZED HFNMM I-TERM CONTROL OF THE BIOPROCESS PLANT

<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>0.0010</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>
The results of Figures 14, 15 showed that the I-term control eliminated constant disturbances and approximation errors, and the proportional control (without I-term) could not. The comparison of the final worse case MSE for the direct and indirect control give slight priority of the indirect control (see Tables 2, 3).

![Figure 15: 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-system reference) in four collocation points for 600 iterations: a) 0.2H, b) 0.4H, c) 0.6H, d) 0.8H)](image)

VI Conclusions

The paper proposed to use a 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 OCM in four collocation points (plus one of the recirculation tank) represented centers of membership functions of the space fuzzyfied output variables. The applied L-M algorithm of T-S rules RTNN learning has a fast convergence and great precision (MSE is 0.0083 for the worse case L-M RTNN learning). The obtained from the FNMMI convergence and great precision (MSE is 0.0083 for the worse case L-M RTNN learning). The obtained from the FNMMI convergence and great precision (MSE is 0.0083 for the worse case L-M RTNN learning).

REFERENCES


