GMDH ALGORITHM IMPLEMENTED IN THE INTELLIGENT IDENTIFICATION OF A BIOPROCESS

Fidel Hernández Lozano, lozano@cmw.ecasa.avianet.cu
Universidad de Camaguey. Professor Adjunto. Camaguey, Cuba

Francisco Herrera Fernández, frank00cuba@gmail.com
Universidad do Estado do Amazonas, Prof. Conv., Manaus, Brasil.

Abstract. *The concentration of biomass is one of the main variables in fermentative processes that exist in the biotechnology industry. There are several techniques for estimating it. It is very difficult and expensive to measure it continuously and on line. This paper presents an alternative for estimating biomass concentration using a neural network – based model, applied to a batch-feed type operation fermentative process. A modified GMDH algorithm (Group Method of Data Handling) was applied, using a new structure based on the combination of a genetic algorithm and the use of a negative feedback loop. The model was tested in the fermentation of yeast Pichia pastoris for the production of a recombinant. The stability and capacity of generalization is demonstrated. Using the mean square error behaviour as criterion the proposed method was compared with other neural networks - based structures.*

Keywords*: modeling, fermentation, GMDH, negative feedback, genetic algorithm*

1. INTRODUCTION

The methodology of the identification systems has been developed to determine mathematical models of processes based on known data of a system. In this area, the general guidelines for the design of classifiers and / or predictors are widely known, however there is consensus on the need for a particularized approach for each application.

The models used in the bioengineering area are complex due to the characteristics of these processes (Passoni, 2005). This concept becomes from the fact that the biological activity generates information with special features, being most notable the following:

- The obtained information from process has a non-homogeneous structure due to the complexity of the organisms alive.
- The information is emerging from the dynamics of change associated with the functional properties of the studied phenomena.

The techniques of Artificial Intelligence, AI, are being applied in this field significantly in recent decades, and among them those known as Artificial Neural Networks, ANNs, characterized by their properties of learning and generalization, (Leiva, 2006). It's often necessary to take into account their potential for induction, which can be implemented by software, (Miroslav Šnorek, 2006). There are several methods to obtain inductive models. The Group Method of Data Handling methods, GMDH, (Ivakhnenko AG, 1971) is well known.

Solla (1989) proves that the probability that a given ANN shows a certain behavior depends not only on the learning algorithm of the network, but also of its architecture. The procedure for increasing the probability of correct correspondence between inputs and outputs, it usually relapses in the learning but, as argued in (Happel and Murr, 1990), to increase the relation between neural network model and process is necessary to impose restrictions on its topology, which should be in correspondence, as much as possible, with the structure of the process. Thus, for complex processes with internal feedback loops, models of neural network with similar features are required. These models have better efficiency in the long-term prediction (Yuan and Vanrolleghem, 1999).

Feedforward neural networks, FFNN, have several design-related limitations that can be improved with a change in its architecture. On the other hand, recurrent neural networks, RNN, have several characteristics that make them superior. First, RNN can converge quickly to a target value for a given performance with a certain margin of error, meaning less iterations. Another advantage, RNN presents a better non-linear behavior during the learning. (Lau, 1992) However, these advantages are a significant cost: the processing time is increased. This feature becomes a principal difficulty in applications involving a large number of neurons. For this reason, the use of this architecture should be a compromise between performance and speed.

Another tool of AI that is being expanded nowadays is the genetic algorithm, GA, based on the theory of biological evolution. This it is applied in the optimization of mathematical models of systems (Holland, 1975) (Ferreira, 2001, 2004).

In this paper an identification of fermentation process is developed using the concept of Evolutionary Identification Systems, combining methodologies developed by several authors as: Functional Networks and self GMDH

(Ivakhnenko, 1999, Ivakhnenko et al. 1998; Ivakhnenko GA, 2001; Kondo and Ueno, 2009) Genetic Programming (Mark and Xin, 2002) (Ferreira, 2001, 2004) and GMDH-.Type with negative feedback (Kondo and Ueno, 2006).

The structure of this paper is as follows: Section 2 describes the basic features of the GMDH algorithm. Section 3 is devoted to representation of a GMDH neural network using GA and then move on to Section 4 to describe the steps required to design the model proposed. Section 5 provides an example of applying the model in a Feed-Batch Fermentation type. These results are compared with other algorithms in Section 6. Finally a section with conclusions is presented.

2. THE GMDH ALGORITHM

The GMDH algorithm was developed for identifying nonlinear relationships between inputs and outputs. This provides an optimal structure, obtained in an iterative procedure of partial descriptions of the data by adding new layers. The number of neurons in each layer, the number of layers and the input variables are automatically determined to minimize a criterion of prediction error and thus an optimal neural network architecture is organized using a self-heuristics, which is the basis of this algorithm. (Ivakhnenko, 1971). This method is particularly successful in solving modeling problems with multiple entries and a single output (Mutasem, 2004). The neural networks developed by using this algorithm are called GMDH Type Neural Networks and are classified within the group of Polynomial Neural Networks, PNN.

The data set used to design the network contains vectors consisting of independent input variables ($x_1, x_2, \ldots x_n$), and output variable S_t.

The first step to implement a GMDH algorithm is to divide the data into two groups, one for training, which is used to estimate parameters of each neuron to obtain a partial description of the process, and the second one to evaluate the results and to select those neurons that describe the process more efficiently. In this way it's obtained a structure that allows a full description of the process.

The n input variables are grouped in pairs (xi, xj) such that:

$$\{(x_1, x_2), (x_1, x_3), \cdots, (x_1, x_n), \cdots, (x_{n-1}, x_n)\}$$

resulting $n\,(n\text{-}1)\,/\,2$ combinations.

These combinations are entries to similar number of polynomial equations, describing the process partially.

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j \tag{1}$$

The coefficients can be calculated by regression analysis for the training data set.

An external criterion, usually the mean square error, MSE, also known as Criterion for Regularity, CR, is calculated using the equation for each set of test data.

$$CR = \frac{1}{T}\sum_{n=1}^{T} (\hat{y}_n - y_n)^2 \tag{2}$$

Where:

T number of vectors composing the test data set,

$\hat{y}_n$ actual output variable, such that $\hat{y}_n$ = St,

y_n model output variable.

Only those equations which fulfil a minimum CR survive and the others are eliminated. This method ensures that only units having the greatest ability to approximate will be selected.

Subsequently a new layer is created using the selected outputs from previous layer as inputs and the process starts again forming pairs of entries. These steps are repeated to generate new layers until the error criterion stops decreasing. When this happens the output will be one with the obtained lowest CR value in previous layer above and so will complete the design process. Figure 1 show how a GMDH-type network is structured from the optimum selection of units.

To estimate the parameters on the GMDH algorithm using a feedback loop, the output variable St is used as input, representing the time t as t-1, in this way St-1 is used as input. Then, the model is evaluated using the output of the test data set as St-1 (Kondo 2003, Kondo and Ueno, 2006).

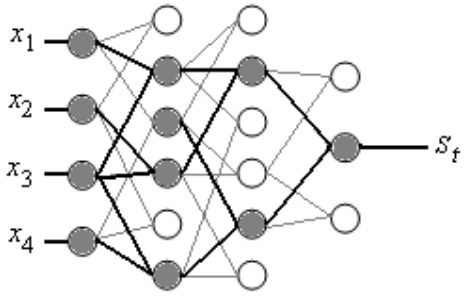
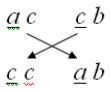


Figure 1. Example of neural network structure obtained by GMDH algorithm.

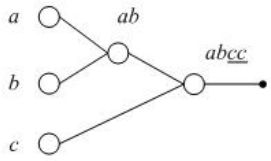
3 REPRESENTATION WITH GENETIC ALGORITHM

Incorporating Genetic Algorithm to GMDH-type neural networks, each neuron is represented as a string, which can be mutated or crossed with each other to form new generations. The procedure to define a chromosome as described in (Nariman-Zadeh et al., 2003) can be modified easily to be included in the proposed GMDH algorithm, which is achieved by repeating the name of the neuron that is moved directly from a layer to another. This means that in GMDH-type neural network with genetic structure the connections between neurons may occur through different non-adjacent layers (Bagheri et al., 2007). In the conventional GMDH neural networks these connections occur only between adjacent layers.

Figure 2 shows an example, where the connection of the neuron c is established directly with the output. This connection is represented by including the input c twice, abcc, which generates a virtual neuron whose inputs are equal cc (Nariman-Zadeh et al., 2005). In other words the crossover and mutation among populations permit that new virtual neurons with repeated inputs are generated, moving the input directly to new layer. Considering $\tilde{n}$ the number of layers moving on, the number of repetitions is calculated as $2^{\tilde{n}}$.



(a). Crossover



(b). Alter crossover

Figure 2. Generation of a virtual neuron through crossover

The quality of each generation (Φ) is represented by

$$\Phi = 1 / CR \tag{3}$$

where CR is obtained by equation (2). This value is minimized through the selection process for increasing the quality of the generation (Φ).

Considering only those descendants that are new generations, M new descendants are obtained. This number is equal to the maximum number of layer input. This increases the number of input variables of the next layer, expressed as:

$$n = F + n_{ant} \tag{4}$$

Where n_{ant} is the number of inputs in previous layer and F is the number of selected outputs. This causes the number of input variables are increased from one layer to another, so it is necessary to provide a stopping criteria for not over sizing the network. Two stopping criteria can be defined:

• Minimum value of CR.
• Maximum number of layers N.

Figure 3 shows this behaviour in one layer. Similar occurs in the others layers.

.

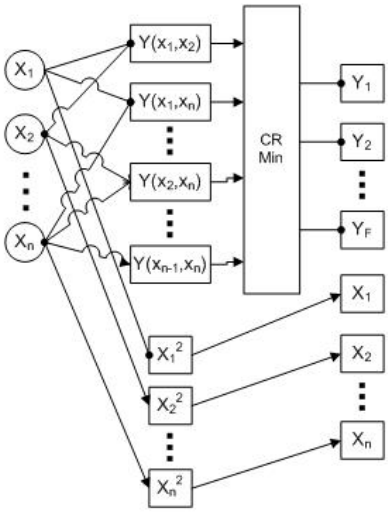


Figure 3. Diagram of distribution of input variables of the adjacent layer.

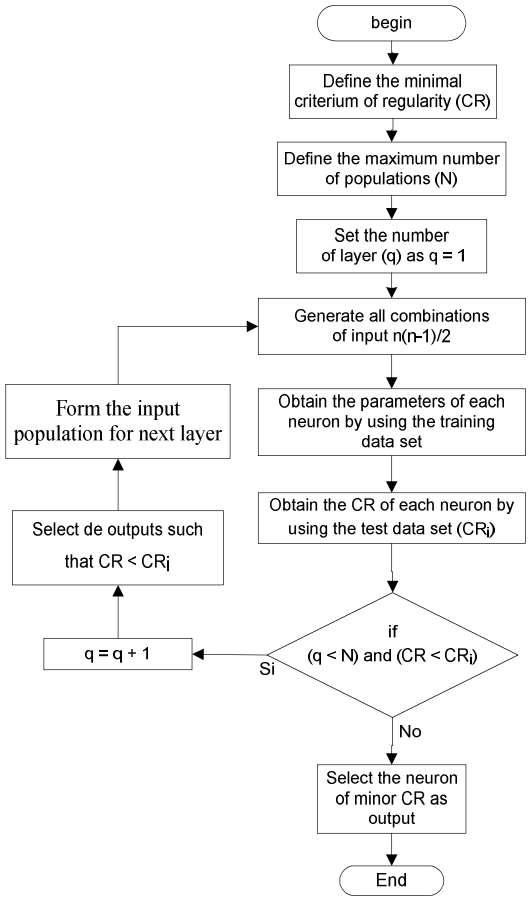


Figure 4. Algorithm.

4 STEPS FOR THE DESIGN OF THE MODEL

Before starting the design process the initial values of the feedback variable S_{t-1} for training and testing must be established. Then the following steps are executed:

Step 1: To establish value for stopping criteria, initial minimum *CR* and maximum value of populations or layers *N*.

Step 2: To generate all possible combinations of inputs $m = n\,(n\text{-}1) / 2$.

Step 3: To calculate the parameters of each neuron using the training data.

Step 4: To check stopping criteria: *N* and CR_{min}.

Step 5: To select the output lesser than CR_{min} and to form the input population for next layer, plus combinations obtained by cloning and crossover (virtual neurons).

Step 6: To set CR_{min} = lower *CR* obtained.

Step 7: Go to Step 3 and repeat until satisfying a stopping criterion.

Figure 4 show a detailed diagram of the proposed algorithm.

5 APPLICATION IN THE PROCESS OF MODELLING OF FERMENTATION.

5.1 Description of the process.

Obtain maximum biomass concentration, is a principal objective in a biotechnological production. By measuring this variable is possible to control the cell growth and to optimize the process. Therefore a precise, continuous and on-line measure of this variable is an objective for any control and supervision system in the biotechnological process. Among the most known methods for determining the concentration of biomass are: samples for gravimetric (dry weight of cells) and test of spectroscopy (optical density) (Reed et al., 2000; Känsäkoski et al., 2006). These methods don't allow appropriate actions to control because they are executed out of the process (off-line) and this imply loss of information, take a long time to obtain results and human effort (Royce, 1993).

The fermentations never are equals, even when have strict control of their initial conditions, substrate and instrumentation. For this reason, a robust system with a good capacity of generalization is required for the estimation of this process, (Jenzsch et al., 2006). The use of models is an important alternative for application in these cases.

The studied process, in this application, was a feed-batch fermentation type to produce a vaccine by recombinant method using *Pichea pastoris* yeast, which has two stages in its growth in response to the supply of substrate (Figure 5). The first stage occurs in glycerol where no substrate is added, and the second one takes place in methanol, with a controlled flow of substrate in correspondence with the rate of cell growth.

To facilitate the training of the network and to protect the information of the process, the values of each variable have been normalized. In the figure 5 the behavior of each one in the different stages of growth that characterize this microorganism can be observed. These are represented in the same scale, normalized between 0 and 1. This allows that can be easily observed.

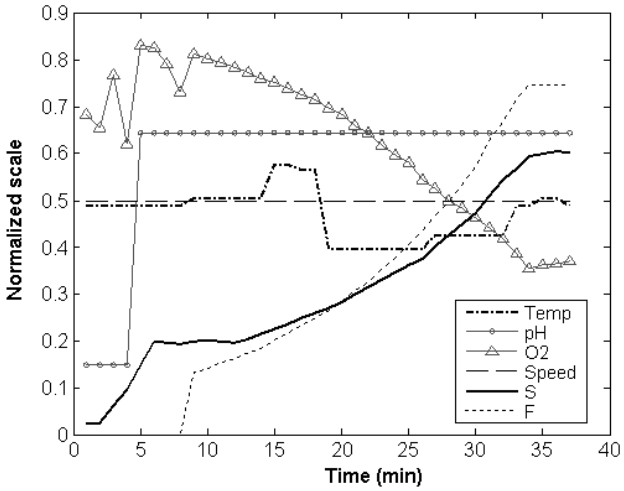


Figure 5. Typical behaviour of the process variables.

The measured variables are: Flow Substrate (*F*), Temperature (*Temp*), pH (*pH*), Dissolved oxygen (O_2) and Stirring Speed (*Speed*) (McNeil and Harvey, 1990). Considering S_{t-1} the output value a step backwards, then the input variables are expressed as follows:

$x_1 = F$ $x_2 = Temp$ $x_3 = pH$
$x_4 = O_2$ $x_5 = Speed$ $x_6 = S_{t-1}$

S_t is the concentration of biomass (output variable). It's measured by the technique of wet cell weight, out of the process, which reduces at 24 samples per fermentation the number of data. The models have been obtained using the measured data from different batch.

5.2 Design of the network.

After establishing the values of S_{t-1} for training and test data respectively, the initial values of the following stopping criteria are adopted as: $CR = 0.01$ and $N = 5$.

Then all input combinations are formed according to the proposed model and the parameters of polynomial equations (3) are calculated. Due to the characteristics of the process the polynomial was restricted to order 2. Afterwards, the stopping criteria are evaluated and then, the pairs that better represent the process are selected. Those combinations over minimum CR are rejected. The selected outputs are part of the input population in the intermediate layer. Then the new population of input variables is completed by adding the input of the previous layer.

Once selected the output neuron the network can be constructed following the input combinations involved in the process. The model results are shown in Figure. 6.

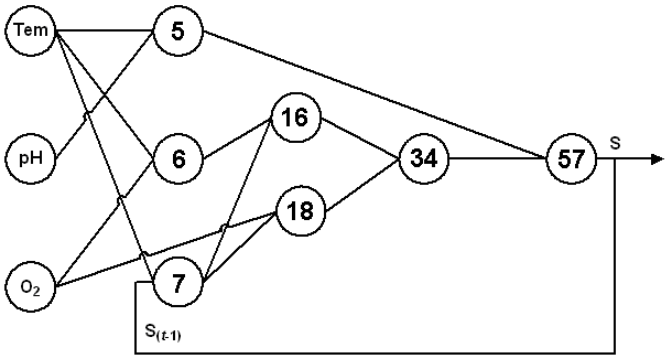


Figure 6. Network architecture design.

The input variables that do not appear in the model were rejected during the design process. Table 1 shows the parameters obtained by regression methods. These parameters belong to the intermediate layer equations. There are only those equations that describe the process efficiently. To calculate these parameters the training data set was used.

Table 1. Coefficients for equations of the intermediate layer.

Ne	a_1	a_2	a_3	a_4	a_5	a_6
5	3.22E-01	1.25E+00	-8.59E-01	-1.22E+00	0.00E+00	6.08E-01
6	1.62E-03	5.50E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7	1.62E-02	-6.57E-02	1.56E+00	-5.11E-01	0.00E+00	-2.85E-01
16	-9.69E-04	1.42E-02	1.00E+00	-6.82E-02	0.00E+00	3.08E-02
18	1.08E-03	9.92E-01	0.00E+00	0.00E+00	1.11E-02	0.00E+00
34	2.79E-02	4.57E+01	-4.49E+01	-3.28E+04	1.63E+04	1.65E+04
57	1.73E-04	9.57E-01	4.35E-02	9.35E-01	-3.95E-01	-5.41E-01

Ne: Number of neuron (see figure 6).

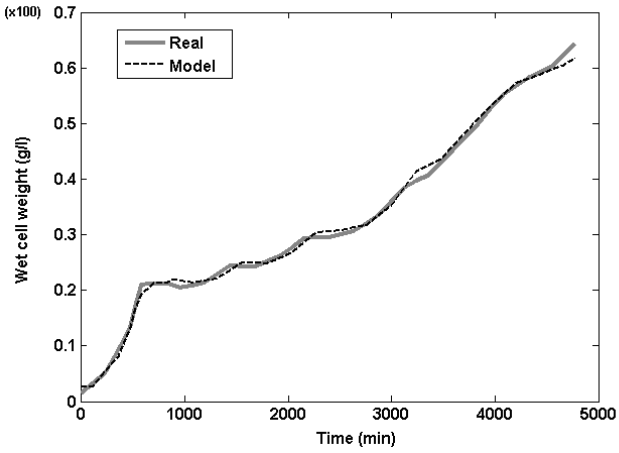


Figure 7. Simulation of a model.

5.3 Analysis of results.

To demonstrate the generalization capacity and the robustness of the model is necessary to prove its behavior with different batch fermentations and to introduce perturbations in the input variables.

Figure 8 show an example of the behavior of the network using other batch fermentation and adding a white noise in the dissolved O_2 input. In it's, the difference on the fermentation time and on the curve form can be observed, due to non-linear behavior inherent to the organisms alive. This non-linearity has been absorbed by the model, which represent the real process with a good approximation. For this case, a satisfactory model performance has been obtained.

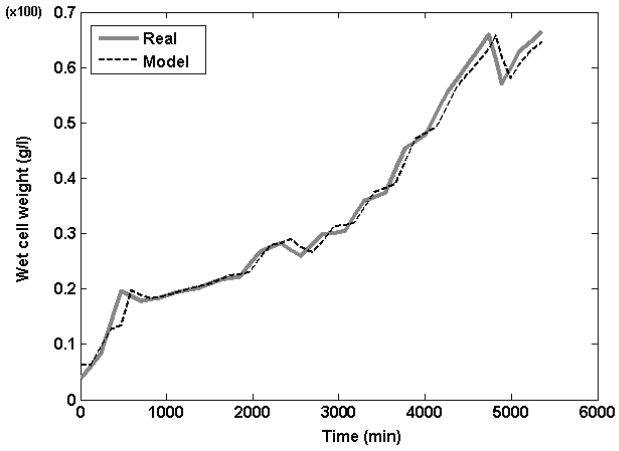
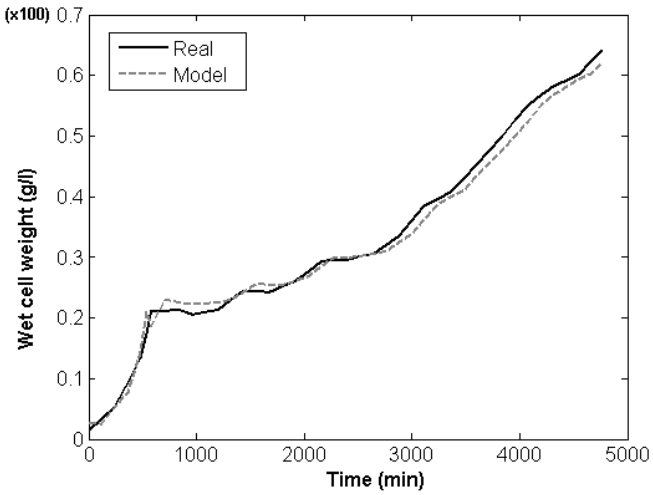


Figure 8. Network response to other batch fermentation and in presence of noise.

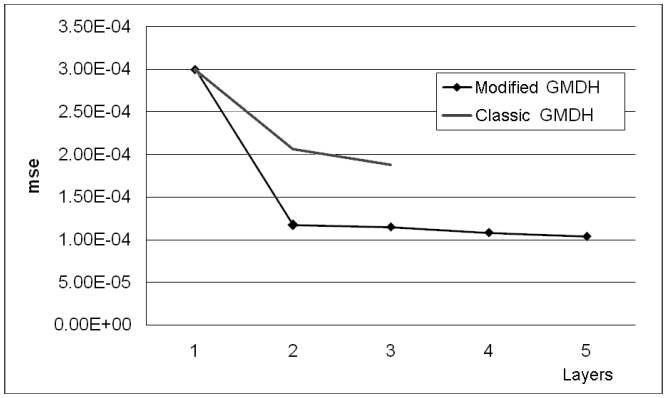
6. COMPARISON WITH OTHER ALGORITHMS

The mean square error (MSE) (Ramsey, 1994; and Ekonomou Pappas, 2006) was used as criterion for comparison with other algorithms. Figure 9 (a) shows the results using a classical GMDH algorithm, and the comparison with error convergence is presented in Figure 9 (b).

Figure 10 shows a comparison of the proposed model with other methods among them, the GMDH algorithm in its classical form, the Elman neural network which contains a recurrent loop too and a classic feed-forward neural network. For the studied process, it's clear that the proposed GMDH algorithm with negative feedback has better performance, showing a lower average error than the others methods.



(a)



(b)

Figure 9. Comparing classic and modified GMDH. (a) Modeling using classic GMDH. (b) Comparing convergence of MSE between classic and modified GMDH.

The neural networks developed using the GMDH algorithm have an important difference in comparison with the models whose neurons are determined beforehand, due to their processing units have an active role because the algorithm is executed within each unit, representing a new variable, which is generated by selection, (Balance et al., 1998).

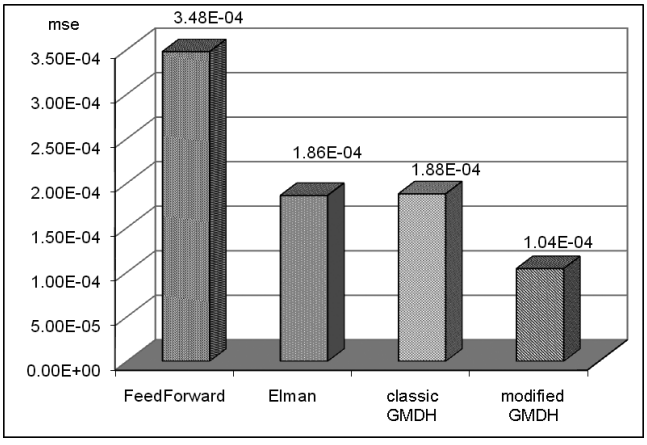


Figure 10. Comparing average of the error between different methods

7. CONCLUSIONS

This paper presented a neural network using the GMDH algorithm, incorporating a feedback loop and increasing the number of input combinations by Genetic Algorithm. It was demonstrated its ability to estimate the cell growth in a Batch Feed type fermentation. This allows the implementation of a virtual sensor (Soft-sensor) to estimate on-line the biomass concentration and to enable in this way, an appropriate control of the variables and to optimize the results.

The number of feedback loops and the input variables are automatically determined so that minimizes the criterion RC. In comparison with other methods of neural modelling, a better performance can be observed, demonstrating its ability to be successfully used in modelling of a batch fermentation process.

The use of recurrent loops and Genetic Algorithms applied to neural network designed by using the GMDH algorithm opens new horizons. This allows developing complex structures with multiple internal loops, formed by neurons that are able to link in several directions, forward and backward. Therefore, models with greater potential for identifying non-linear processes can be obtained, such as the fermentation process.

8. REFERENCES

Bagheri A., Nariman-Zadeh N., Babaei M., and Jamali A., 2007, Polynomial Modeling of the Controlled Rack-Stacker Robot Using GMDH-type Neural Networks and Singular Value Decomposition. *International Journal of Nonlinear Sciences and Numerical Simulation*, **8(3)**, 301-310.

Ferreira, C., 2001, Gene Expression Programming in Problem Solving *Complex Systems*

Ferreira, C., 2004, Designing Neural Networks Using Gene Expression Programming. Paper presented at the *9th Online World Conference on Soft Computing in Industrial Applications*.

Happel, B. L. M. and Murre, J.M.J., 1990, Structure Identification of Non Linear Dinamic Systems - A survey on input/output Approaches. *Automatica*, **26(4)**, 651-667.

Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* University of Michigan Press, (second edition: MIT Press, 1992).

Ivakhnenko A. G., K. V. V., Tetko I. V., Luik A.I., Ivakhnenko G.A., Ivakhnenko N.A., 1999, Self-Organization of Neural networks with Active Neurons for Bioactivity of Chemical Compounds Forecasting by Analogues Complexing GMDH Algorithm. Paper presented at the Poster for the ICANN'99 Conference.

Ivakhnenko A.G., 1971, Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(1).

Ivakhnenko A.G., D. W., Ivakhnenko G.A.., 1998, Inductive Sorting-Out GMDH Algorithms with Polynomial Complexity for Active Neurons of Neural Network. from http://come.to/GMDH

Ivakhnenko G.A., 2001, Inductive Self-Organising Algorithm for Maximum Electrical Load Prediction. Ukraina, Kyiv: *International Centre of Informational Technologies and Systems of the National* Ac.Sci.

Jenzsch, M., Simutis, R., Lübbert, A., 2006, Optimization and Control of Industrial Microbial Cultivation Processes. *Eng. Life Sci.*, **6(2)**, 117-124.

Känsäkoski, M., Kurkinen, Marika, von Weymarn, Niklas, Niemelä, Pentti, Neubauer, Peter, Juuso, Esko, Eerikäinen, Tero, Turunen, Seppo, Aho, Sirkka & Suhonen, Pirkko., 2006, Process analytical technology (PAT) needs and applications in the bioprocess industry. *VTT Technical Research Centre of Finland*, **60**, 99.

Kondo, T., 2003, Revised GMDH-type neural networks with radial basis functions and their application to medical image recognition of stomach. *Systems Analysis Modeling Simulation*, **43(10)**, 1363-1376.

Kondo, Tadashi and Ueno, Junji, 2006, Revised GMDH-Type Neural Network Algorithm With a Feedback Loop Identifying Sigmoid Function Neural Network. *International Journal of Innovative Computing, Information and Control*, **5(2)**, 985—996.

Kondo, Tadashi and Ueno, Junji, 2009, Medical Image Recognition of Abdominal Multi-Organs by Rbf Gmdh-Type Neural Network. *International Journal of Innovative Computing, Information and Control*, **5(1)**, 225-240.

Lau, C., 1992, Artificial Neural Networks: paradigms, applications and hardware implementations. *IEEE Press*, 78.

Leiva, G. A., 2006, Redes Neuronales como Herramienta para la Automatización de Sistemas Complejos. *Paper presented at the EVIC2006*.

Mark S. Voss, Xin Feng, 2002, A new methodology for emergent system identification using Particle Swarm Optimization (PSO) and the Group Method Of Data Handling (GMDH). *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*.

McNeil B., Harvey L. M., 1990, *Fermentation a Practical Approach* (1ra ed.). Oxford: Oxford University Press.

Miroslav Šnorek, P. K., 2006, Inductive Modelling World Wide the State of the Art. Report of investigation, *Dept. of Computer Science and Engineering*, Karlovo nam.

Mutasem Hiassat, N. M., 2004, An evolutionary method for term selection in the Group Method of Data Handling. *Automatic Control & Systems Engineering*, University of Sheffield, 11-14.

Nariman-Zadeh, N., Darvizeh, A., Ahmad-Zadeh, R., 2003, Hybrid Genetic Design of GMDHType Neural Networks Using Singular Value Decomposition for Modelling and Prediction of the Explosive Cutting Process. *Journal of Engineering Manufacture*, **217**, 779-790.

Nariman-Zadeha N., Darvizeha A., Jamalia A., Moeinib A., 2005, Evolutionary design of generalized polynomial neural networks for modeling and prediction of explosive forming process. *Paper presented at the 13th International Scientific Conference on Achievements in Mechanical and Materials Engineering*.

Pappas S. Sp., Ekonomou L., 2006, Comparison of Artificial Intelligence Methods for Predicting the Time Series Problem. Proceedings of the *6th WSEAS International Conference on Simulation, Modelling and Optimization*, 22-24.

Passoni, L. I., 2005, Modelos en Bioingeniería: Caracterización de Imágenes Estáticas y Dinámicas. *Tesis del Doctorado en Ingeniería*, Universidad Nacional de Mar del Plata.

Ramsey, A., 1994, Assessment of the modeling abilities of Neural networks. *U. of Massachusetts*, US.

Reed G., Rehm J., Puhler A., Stadler P., 2000, Biotechnology, Measuring modelling and control. VHC, **4**, 181.

Royce, P. N., 1993, A discussion of recent developments in fermentation monitoring and control from a practical perspective. *Critical reviews in biotechnology*, **13**, 117-149.

Solla, S. A., 1989, Learning and Generalization in Layered Neural Network: the Contiguity Problem. Neural Networks: from Models to Aplications. In L. Personnas and G. Dreyfus Paris: I.D.S.E.T.

Yuan J.Q. and Vanrolleghem P. A., 1999, Rolling learning-prediction of product formation in bioprocesses. *Journal of Biotechnology*, **69**(Elsevier Science B.V.), 47–62.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.