0273-2289/02/98-100/1009/$13.75

Hybrid Neural Modeling of Bioprocesses Using Functional Link Networks

Layse H. P. Harada, Aline C. da Costa,* and Rubens Maciel Filho

DPQ/FEQ/UNICAMP,
Cx. Postal 6066, CEP 13081-970,
Campinas, SP, Brazil,
E-mail: accosta@feq.unicamp.br

Abstract

The objective of this work was to develop a model for an extractive ethanol fermentation in a simple and rapid way. This model must be sufficiently reliable to be used for posterior optimization and control studies. A hybrid neural model was developed, combining mass and energy balances with neural networks, which describe the process kinetics. To determine the best model, two structures of neural networks were compared: the functional link networks and the feedforward neural networks. The two structures are shown to describe well the process kinetics, and the advantages of using the functional link networks are discussed.

Index Entries: Extractive alcoholic fermentation; functional link networks; hybrid model; bioprocess; modeling.

Introduction

One of the most difficult problems in the control and optimization of biotechnological processes is the construction of reliable models of the system. For these processes, the development of detailed models based on fundamental principles and intense kinetic studies is frequently expensive and time-consuming. Thus, it would be of great advantage to find a simple and rapid way to describe them, accurately enough for optimization and control. Many methods have been proposed in recent years to achieve this goal. One is the use of neural networks, which offers a tool for direct use of process data to generate input-output relationships *(1,2)*. The training of a neural network, however, requires many experimental data. Furthermore, the interpretation of such models is difficult.

*Author to whom all correspondence and reprint requests should be addressed.

Another alternative is the use of hybrid neural models, in which the aspects of the problem whose quantitative behavior is well understood are described by deterministic mathematical equations, while neural networks describe the kinetics *(3–6)*. These models are expected to perform better than "black-box" neural network models, since generalization and extrapolation are confined only to the uncertain parts of the process, and the basic model is always consistent with first principles. In addition, significantly fewer data are required for their training.

Many neural network structures are cited in the literature, and there are no methods proposed to define the best structure to be used for a given case. In the majority of studies applying neural networks to bioprocesses, the feedforward neural network (FNN) is used. A structure that has not been much explored is the functional link network (FLN) *(7)*. This network has been shown to have a good nonlinear approximation capability, although the estimation of its weights is linear. Because of this linear estimation, the FLN's training is rapid, requires low computational effort, and the convergence is guaranteed *(4,5)*, giving such a structure a large potential for implementation of online control and optimization.

In the present work, the performance of the FLNs and FNNs in describing the kinetics of an extractive alcoholic fermentation *(8)* are compared. The objective was to develop hybrid models for the process combining mass balance equations with neural networks. These models can be used for optimization and/or control of the considered process, and if the FLNs are used, the implementation of adaptive schemes is quite simple.

Materials and Methods

Extractive Alcoholic Fermentation

Because the conventional alcoholic fermentation process is inhibited by ethanol, many methods have been proposed for the selective extraction of this product during fermentation *(9)*. Silva et al. *(8)* have shown that the fermentation combined with a vacuum flash vessel presents many positive features and better performance than a conventional industrial process *(10)*.

The extractive process proposed by Silva et al. *(8)* is shown in Fig. 1. It consists of four interlinked units: fermentor (ethanol production unit), centrifuge (cell separation unit), cell treatment unit, and flash vessel under vacuum (ethanol-water separation unit). This scheme attempts to simulate industrial conditions, with the difference that only one fermentor is used in spite of a cascade system, and the flash vessel is used to extract part of the ethanol. In fact, in a conventional industrial process the usual arrangement is to have four interlinked fermentors with the measurements made at the exit of the last tank *(10)*.

In the cell treatment unit, the cell suspension is diluted with water, and sulfuric acid is added in order to avoid bacterial contamination. The flash vessel is operated in a temperature range from 28 to 30°C, which is chosen

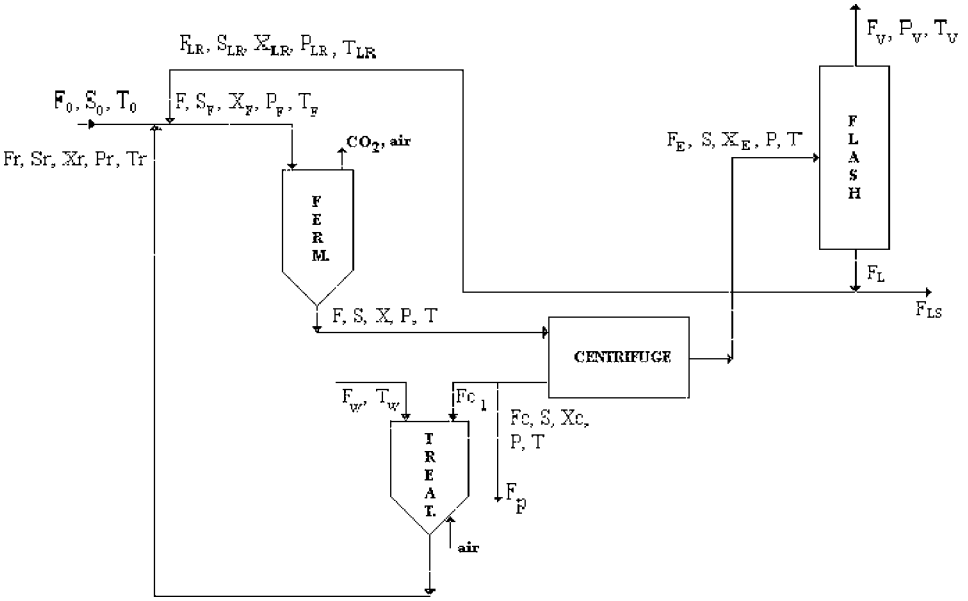

Fig. 1. Extractive alcoholic fermentation.

to eliminate the necessity for a heat exchanger. This reduces drastically the fixed as well as the maintenance costs of the process, since the heat exchangers are expensive pieces of equipment in an alcoholic fermentation plant *(10)*. The associated pressure is in the range of 4 to 5.33 kPa. The process was shown to be able to maintain suitable conditions for the growth of *Saccharomyces cerevisiae*,by keeping a constant temperature, which may be controlled without a heat exchanger *(8,9)*. The vaporized stream leaving the flash vessel is sent to a rectification column with part of the liquid stream, while the other fraction of the liquid returns to the fermentor. This is adjusted to maintain the ethanol concentration in the fermentor at such a value that it acts as an antiseptic. According to practical knowledge in industrial units, this alcohol concentration is about 40 kg/m^3, which has a low inhibitory effect for fermenting yeast but is highly inhibitory for most contaminating microorganisms *(8)*.

Mathematical Model of Fermentor and Flash Vessel

The balance equations in the fermentor are as follows:

$$\frac{dX}{dt} = \mu X - D(X - X_F) \tag{1}$$

$$\frac{dS}{dt} = D(S_F - S) - \sigma X \tag{2}$$

$$\frac{dP}{dt} = \pi X - D(P - P_F) \tag{3}$$

$$\frac{dT}{dt} = D(T_F - T) + \frac{\Delta H\, \sigma X}{\rho\, Cp} \tag{4}$$

in which X, S, and P are the biomass, substrate and product concentrations, respectively; T is the fermentor temperature; μ, σ, and π are the specific rates of growth, substrate consumption, and product formation, respectively, given by Eqs. 5–7; X_F, S_F, and P_F are the biomass, substrate, and product feed concentrations, respectively; T_F is the feed temperature; ΔH is the reaction heat; ρ is the medium density; Cp is the heat capacity; and D is the dilution rate.

The kinetic rates were obtained for S. cerevisiae and are given by Silva et al. *(8)*:

$$\mu = \mu_{max} \left(\frac{S}{K_s + S}\right)\left(1 - \frac{P}{P_{max}}\right)^n \tag{5}$$

$$\sigma = \frac{\mu}{Y_{X/S}} \tag{6}$$

$$\pi = \mu\, \frac{Y_{P/S}}{Y_{X/S}} \tag{7}$$

Table 1 shows the kinetic parameters used. The yield coefficients ($Y_{X/S}$ and $Y_{P/S}$) are not expressed as functions of the temperature but were obtained from industrial data *(10)* and are valid in the considered operational range.

The values of the constants in the energy balance equation (Eq. 4) are given by Silva et al. *(8)*: $\Delta H = 2.167 \times 10^5$ J/kg; $\rho = 1000$ kg/m^3; $Cp = 4.183 \times 10^3$ J/(kg K). The mass balances over the flash tank are given by:

$$F_E = F_V + F_L \tag{8}$$

$$F_E x_{Ei} = F_V y_i + F_L x_i \tag{9}$$

The vapor-liquid equilibrium of the ethanol-water mixture was calculated by Eq. 10, the value of p_i^{sat} was calculated by Antoine's equation (the assumption was made that the light phase was a binary mixture of ethanol and water), and the value of γ_i was calculated using the nonrandom two-liquid model *(8)*.

$$K_i = \frac{y_i}{x_i} = \gamma_i \frac{p_i^{sat}}{p} \tag{10}$$

A program written in Fortran was developed to solve the mathematical model equations. Equations 1–4 were integrated using an algorithm based on the fourth-order Runge-Kutta method.

Hybrid Neural Model

As noted by Psichogios and Ungar *(3)*, to deriving an approximate model of a bioreactor from simple first principles considerations such as

Table 1
Kinetic Parameters

Parameter	Expression or value	Reference
μ_{max}	$Z \exp[-E/(R_g T)]$	10
P_{max}	$638.1 \exp(-0.05741\ T)$	11
n	3.0	10
$Y_{X/S}$	0.033	10
$Y_{P/S}$	0.445	10
K_s	1.6	10
E	1.54×10^4	10
Z	4.5×10^{10}	10

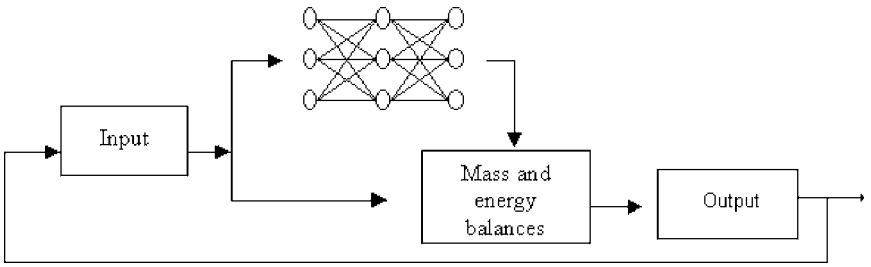

Fig. 2. Hybrid neural model.

mass and energy balances is a straightforward task. The critical factor in modeling the process is the unknown kinetics. Thus, in the hybrid neural model (see Fig. 2), the aspects of the problem whose quantitative behavior is well understood are described by deterministic mathematical equations (mass and energy balances for the whole process of Fig. 1), while neural networks describe the kinetics. The performance of two different structures of neural networks will be tested to describe the kinetics: the FNN and the FLN.

In this work only the specific growth rate (μ) is described by a neural network. Equations 6 and 7 describe the specific rates of substrate consumption (σ) and product formation (π). As can be seen from these equations, σ and π are described by constants multiplied by μ, and if a neural network describes μ well, the development of neural networks to describe σ and π is straightforward.

Feedforward Neural Network

A neural network consists of a large number of simple interconnected nonlinear processing units, called nodes or neurons, arranged in layers and operating in parallel. The weights, which define the strength of connection between the nodes, are estimated to yield good performance.

There are different topologies of neural networks proposed in the literature, but the most widely known is the FNN (see Fig. 3) *(12)*. In the

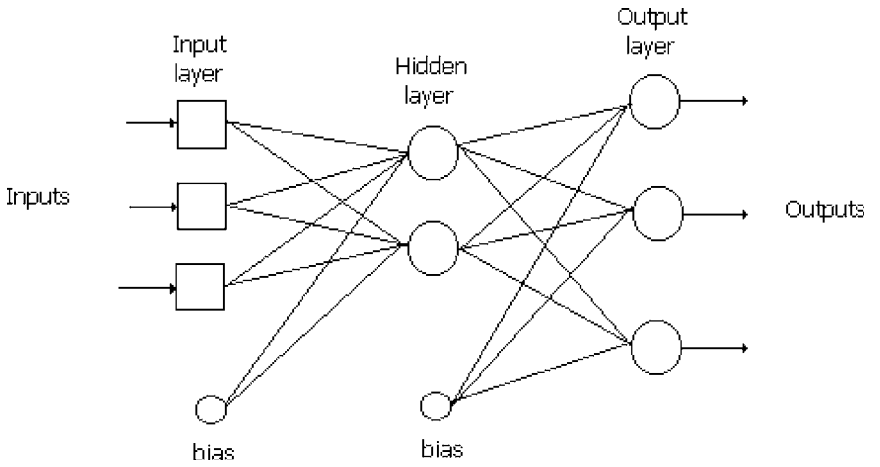

Fig. 3. Topology of an FNN.

FNN's the nodes are arranged in layers: the input layer, the output layer, and the hidden layer(s). All the nodes in a layer are connected to all the nodes of the adjacent layers, and there are no connections among the nodes in a same layer.

The number of nodes in the input layer corresponds to the number of input variables. The number of nodes in the output layer is set by the number of output variables, and the number of hidden layers and nodes in these layers can be chosen by trial and error, but systematic procedures to identify the adequate number have been shown to be an interesting alternative approach *(13)*.

The neurons in the input layer distribute the input patterns to the subsequent layers. From the second layer, the information will be processed and propagated through the output layer, through Eqs. 11 and 12:

$$z_i^{(k)} = \sum_{j=1}^{(nk-1)} w_{ij}^{(k)} yr_j^{(k)} + \theta_i^{(k)} \tag{11}$$

$$yr_j^{(k)} = f\left(z_i^{(k)}\right) \tag{12}$$

in which $w_{ij}^{(k)}$ is the weight of connection between the nodes, $yr_j^{(k-1)}$ is the output of node j from the prior layer, $\theta_i^{(k)}$ is the threshold value (bias), and $f(z_i^{(k)})$ is the activation function.

The model proposed by Silva et al. *(8)* was used to generate data for training and validation of the neural network. The validation data set contained input patterns different from the training data set and was used to test the generalization capacity of the neural network. Random disturbances were introduced every 2 h in the inlet substrate concentration (S_0) and inlet temperature (T_0). The disturbances used to generate the training

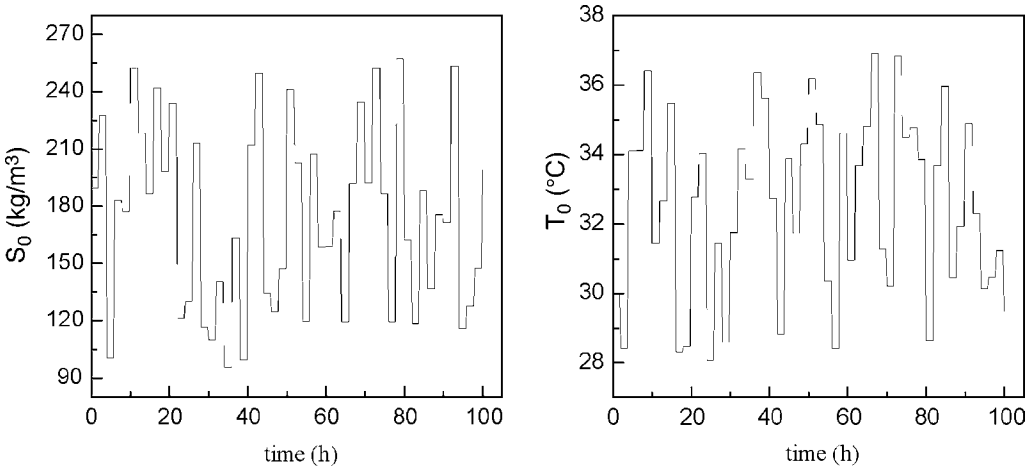

Fig. 4. Random disturbances in S_0 and T_0 to generate training data set.

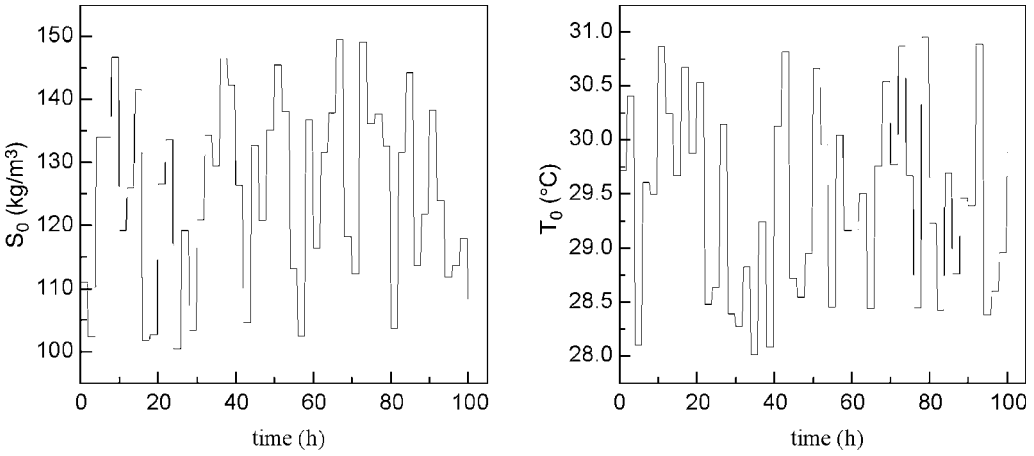

Fig. 5. Random disturbances in S_0 and T_0 to generate validation data set.

data set are shown in Fig. 4 and those used to generate the validation data set in Fig. 5.

The neural network was trained using the neural network toolbox of Matlab 5.3 with the Levenberg-Marquardt training algorithm. The activation function was the sigmoidal function. The inputs to the neural network were ethanol and substrate concentrations and temperature (P, S and T), and the output was the specific growth rate (μ). Only one hidden layer was used, and the number of nodes in this layer was determined by trial and error to minimize validation error. The neural network selected has four hidden neurons.

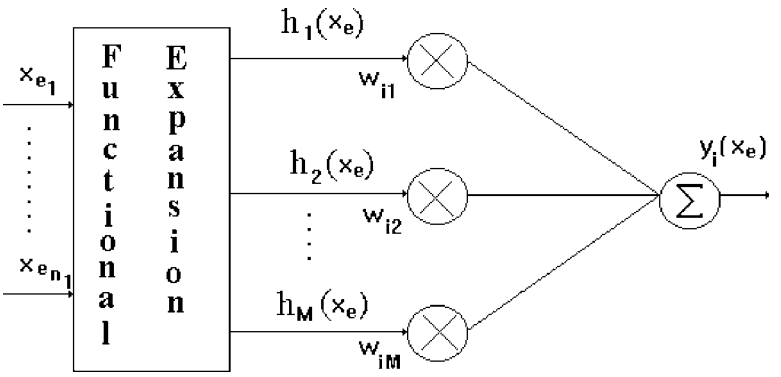

Fig. 6. General structure of an FLN.

Functional Link Network

In the FNN, the inputs to a node are linearly weighted before the sum is passed through some nonlinear activation function that ultimately gives the network its nonlinear approximation ability. The same nonlinearity, however, creates problems in the estimation of the network weights, and because nonlinear learning rules must be used, the learning rate is often slow and local minima may cause problems *(7)*.

One way to avoid nonlinear learning is to use FLNs. In these networks, a nonlinear functional transform or expansion of the network inputs is initially performed and the resulting terms are combined linearly. The structure obtained has good nonlinear approximation capability, and the estimation of network weights is linear.

The general structure of an FLN is shown in Fig. 6, where $\boldsymbol{x}_e$ is the input vector and $y_i(\boldsymbol{x}_e)$ is an output. The hidden layer performs a functional expansion on the input data that maps the input space of dimension n_1 onto a new space of increased dimension, M ($M > n_1$). The output layer consists of m nodes, each one, in fact, a linear combiner. The input-output relationship of the FLN is as follows:

$$y_i(\boldsymbol{x}_e) = \sum_{j=1}^{M} w_{ij}\, h_j(\boldsymbol{x}_e), \quad 1 \le i \le m \tag{13}$$

Henrique *(14)* proposed a modification on the structure of the FLNs, which the output given by Eq. 13 is transformed by an invertible nonlinear activation function. The new output is

$$y_i(\boldsymbol{x}_e) = f_i\left[\sum_{j=1}^{M} w_{ij}\, h_j(\boldsymbol{x}_e)\right], \quad 1 \le i \le m \tag{14}$$

in which f_i is an invertible continuously differentiable nonlinear function such as the sigmoidal function.

Table 2
Polynomial Expansion of Degree Six

Degree	Monomials
0	1
1	z_i $i = 1, n_z$
2	$z_i z_j$ $i = 1, n_z; j = i, n_z$
3	$z_i z_j z_k$ $i = 1, n_z; j = i, n_z; k = j, n_z$
4	$z_i z_j z_k z_l$ $i = 1, n_z; j = i, n_z; k = j, n_z; l = k, n_z$
5	$z_i z_j z_k z_l z_m$ $i = 1, n_z; j = i, n_z; k = j, n_z; l = k, n_z; m = l, n_z$
6	$z_i z_j z_k z_l z_m z_n$ $i = 1, n_z; j = i, n_z; k = j, n_z; l = k, n_z; m = 1, n_z; n = m, n_z$

Another modification in the FLNs used in this work is that the real inputs, x_e, are transformed into a greater number (n_z) of auxiliary inputs, z, before the functional expansion is performed *(5)*. These modifications increase the nonlinear approximation ability of the network, while estimation of the parameters remains a linear problem.

A polynomial expansion of degree six is then performed on the new inputs. The generated monomials [$h_j(z)$] are shown in Table 2. Once the monomials are generated, the orthogonal least squares estimator proposed by Billings et al. *(15)* is used to calculate the network weights (w_{ij}) and to eliminate the monomials that are not significant in explaining the output variance *(14)*. This reduces the size and complexity of the neural network and avoids overfitting of the data.

The same data sets used for the training and validation of the FNN were used for the FLN. The inputs and output of the FLN are also the same as for the FNN. Vector $\boldsymbol{z}$ was chosen as

$$\boldsymbol{x} = \left[S \; \frac{1}{S} \; P^3 \; \frac{1}{P} \; \ln(T) \; \frac{1}{\ln(T)} \right] \tag{15}$$

and the activation function as

$$f\left[\sum_{j=1}^{M} w_{ij} \, h_j(\boldsymbol{z}) \right] = \frac{1}{\sum_{j=1}^{M} w_{ij} \, h_j(\boldsymbol{z})} \tag{16}$$

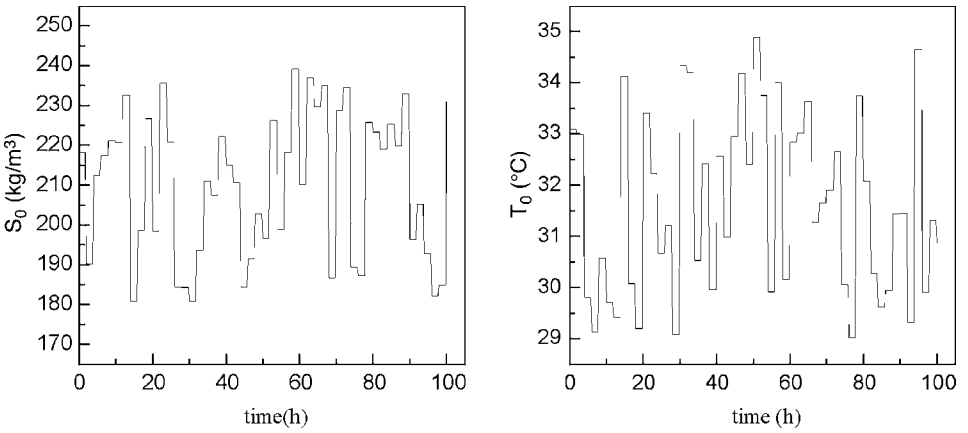

Fig. 7. Random disturbances in S_0 and T_0 to test performance of hybrid models.

The activation function and vector z were chosen by trial and error to minimize the fitting residuals during the training phase. Physical knowledge of the process can be used to help in the selection of these functions.

A program in Matlab, developed by Henrique *(15)*, was used for the training and validation of the FLN.

Results

The performance of the two hybrid neural models proposed was tested with a new data set generated by performing new random disturbances in S_0 and T_0. These disturbances are shown in Fig. 7. Figures 8–11 show the results for the two hybrid neural models compared with the deterministic model. It can be seen that both hybrid models, with the FFN and FLN, described well the dynamic behavior of the process.

The quality of the prediction of the hybrid models can be characterized using the residual standard deviation (RSD), Eq. 17, which provides an indication of the accuracy of the prediction, as suggested by Cleran et al. *(16)*:

$$\text{RSD} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - y_{pi})^2}{n}} \tag{17}$$

in which y_i is the "real" value (calculated by the deterministic model), y_{pi} is the value predicted by the hybrid model and n is the number of points.

Since the magnitude of the RSD will vary depending on the magnitude of the variable to be predicted, it is easier to analyze the RSD written as a percentage of the average of the real values $\overline{y_i}$:

$$\text{RSD (\%)} = \frac{\text{RSD}}{\overline{y_i}} \cdot 100 \tag{18}$$

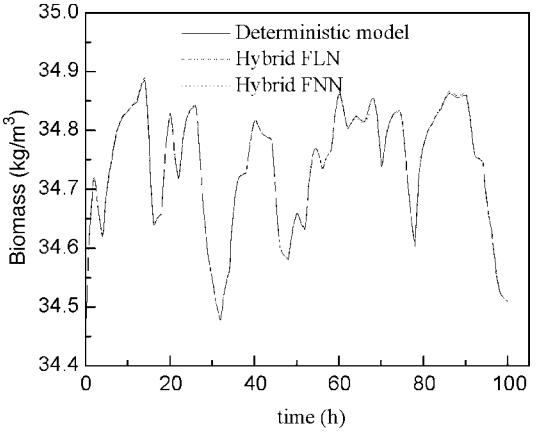

Fig. 8. Biomass concentration described by deterministic model and by hybrid neural models using FFN and FLN.

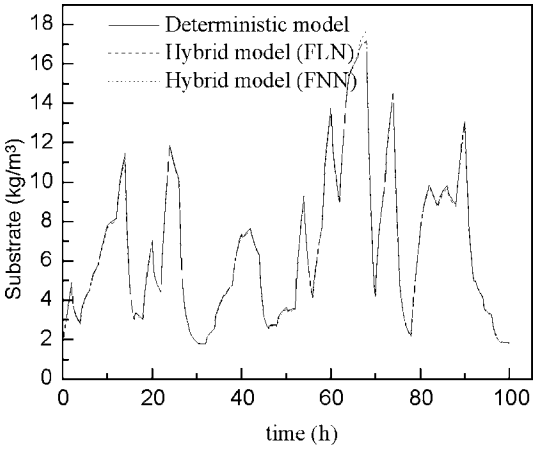

Fig. 9. Substrate concentration described by deterministic model and by hybrid neural models using using FFN and FLN.

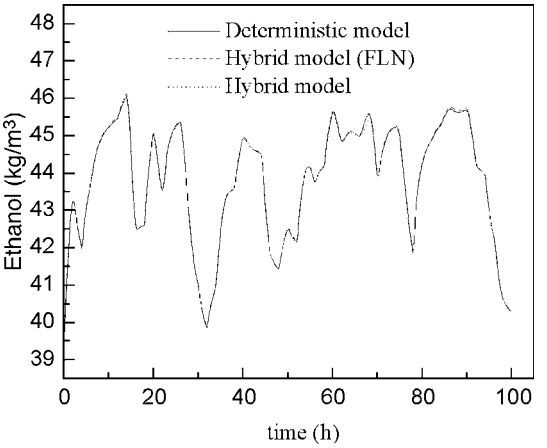

Fig. 10. Ethanol concentration described by deterministic model and by hybrid neural models using FFN and FLN.

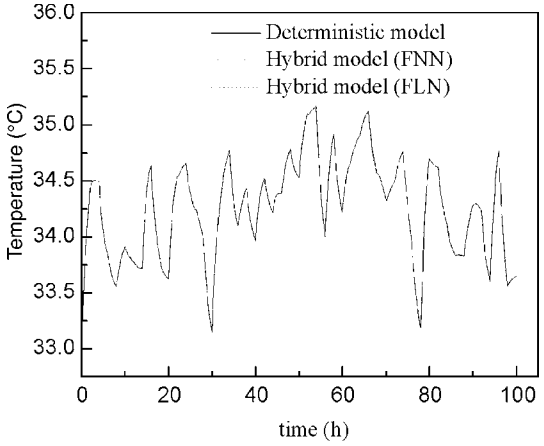

Fig. 11. Temperature described by deterministic model and by hybrid neural models using FFN and FLN.

Table 3
Residual SD Written as a Percentage of Average of "Real" Values

RSD(%)	Hybrid model + FNN	Hybrid model + FLN
Biomass	0.006	0.004
Substrate	1.3	0.9
Product	0.07	0.05
Temperature	0.01	0.007
Specific growth rate	0.1	0.08

Table 3 shows the quality of the prediction of the two hybrid models for biomass, substrate and product concentrations, and temperature. It also shows the quality of the prediction of the specific growth rate by the FFN and FLN. It can be seen that the magnitude of the errors is similar for the two hybrid models, although the hybrid model using the FLN presents slightly lower errors. Note that an FNN with a different structure that presents better performance than the FLN may exist.

Discussion

In many cases detailed models based on fundamental principles and detailed kinetic studies are not readily available for bioprocesses, owing to economic and time constraints. Hybrid neural modeling is a promising tool for the rapid development of reliable models, which are indispensable for the design and implementation of advanced control and optimization strategies.

Although the FLNs are not used much in chemical engineering applications, they present many advantages when compared with FFNs. Because of the linear estimation of the weights, training is rapid, requires

low computational effort, and convergence is ensured. The implementation of adaptive schemes using this network structure is much easier than using the feedforward structure. In an adaptive scheme, the network weights can be reestimated online based on measured process data. In this way, the hybrid model will present a good performance even when there are unexpected changes in the operational conditions or in the metabolism of the microorganism.

Our work demonstrated that the FLN was able to describe the kinetics of the extractive alcoholic fermentation with the same accuracy as the FFN. The hybrid model using FLNs to represent the kinetics of the fermentation process appears to be stable and robust in all the operating conditions considered, so we did not need to perform any further investigation on stability and robustness of the least squares algorithm used to calculate the FLN weights. Note that the FLN approach, because the weights are tuned in a linear fashion, tends to have no stability problems regarding convergence *(7)*.

The polynomial expansion model used in the FLNs is generated based on the auxiliary inputs vector in such a way that a larger number of data are generated by a combination of the inputs. The nature of the model is of an order higher than one, since it is desirable to take into account all possible combinations, which will be automatically discriminated in order to keep the representative pairing of variables or pseudo-variables of the original data set. For the type of activation function used, there is no need to go through scaling procedures of the training data set *(4,5)*. However, if an alternative function is to be used, it may be necessary to apply some scaling procedure.

Further improvement in the proposed approach to generate hybrid models using FLNs may require the use of alternative intelligent algorithms to choose the morphology of the FLN (meaning the selection of different possible internal activation functions, of the degree of the polynomial expan-sion, as well as of the form and size of the auxiliary inputs vector).

The main objective of the present work was to verify the potential of the proposed approach to model fermentation processes. The neural networks were validated using simulated data, and the presence of changes in the kinetic parameters and noisy measurements were not considered. Costa et al. *(4)* have used an adaptive hybrid neural model based on FLNs in the optimal control of fed-batch fermentation processes. They have shown that the use of the adaptive hybrid model led to very good results even in the presence of changes in parameters.

Nomenclature

Cp = Heat capacity (J/[kg K])
D = Dilution rate (h^{-1})
E = Activation energy (J/mol)

F_E = Light-phase flow rate to flash tank (m^3/h)
F_L = Liquid outflow from vacuum flash tank (m^3/h)
F_V = Vapor outflow from vacuum flash tank (m^3/h)
K_i = Equilibrium constant
K_s = Substrate saturation constant (kg/m^3)
n = Product inhibition power
P = Product concentration (kg/m^3)
P_F = Feed product concentration (kg/m^3)
p = Pressure (Pa)
p_i^{sat} = Vapor pressure (Pa)
P_{max} = Product concentration when cell growth ceases (kg/m^3)
R_g = Gas constant (8.314 J/[mol.K])
S = Substrate concentration (kg/m^3)
S_0 = Inlet substrate concentration (kg/m^3)
S_F = Feed substrate concentration (kg/m^3)
T = Temperature (°C)
T_0 = Inlet temperature (°C)
T_F = Feed temperature (°C)
X = Biomass concentration (kg/m^3)
X_F = Feed biomass concentration (kg/m^3)
x_i = Component *i* concentration in liquid (mol %)
x_{Ei} = Component *i* concentration in light phase (mol %)
y_i = Component *i* concentration in vapor (mol %)
$Y_{P/S}$ = Yield constant (kg product/kg substrate)
$Y_{X/S}$ = Yield constant (kg cell/kg substrate)
z = Constant (h^{-1})
ΔH = Reaction heat (J/kg)
γ_i = Activity coefficient of component *i*
μ = Specific growth rate (h^{-1})
μ_{max} = Maximum specific growth rate (h^{-1})
π = Specific product formation rate (h^{-1})
ρ = medium density (kg/m^3)
σ = Specific substrate consumption rate (h^{-1})

References

1. Bhat, N. and Mcavoy, T. J. (1990), *Comp. Chem. Eng.* **14,** 573–583.
2. Hussain, M. A. (1999), *Artif. Intell. Eng.* **13,** 55–68.
3. Psichogios, D. C. and Ungar, L. H.(1992), *AIChE J.* **38,** 1499–1511.
4. Costa, A. C., Alves, T. L. M, Henriques, A. W. S., Maciel Filho, R., and Lima, E. L. (1998), *Comp. Chem. Eng.* **22,** S859–S862.
5. Costa, A. C., Henriques, A. S. W., Alves, T. L. M., Maciel Filho, R., and Lima, E. L. (1999), *Braz. J. Chem. Eng.* **16,** 53–63.
6. Zorzetto, L. F. M., Maciel Filho, R., and Wolf-Maciel, M. R. (2000), *Comp. Chem. Eng.* **24,** 1355–1360.

7. Chen, S. and Billings, S. A. (1992), *Int. J. Control* **56,** 319–346.
8. Silva, F. L. H., Rodrigues, M. I., and Maugeri, F. (1999), *J. Chem. Tech. Biotechnol.* **74,**176–182.
9. Costa, A. C., Dechechi, E. C., Silva, F. L. H., Maugeri Filho, F., and Maciel Filho, R. (2000), *Appl. Biochem. Biotechnol.* **84,** 577–593.
10. Andrietta, S. R. and Maugeri, F. (1994), in *Advances in Bioprocess Engineering*, Galindo, E. and Ramirez, O. T., eds., Kluwer Academic, Dordrecht, pp. 47–52.
11. Alves, J. G. L. F. (1996), MSc thesis, Faculdade de Engenharia de Alimentos, UNICAMP, Campinas, SP, Brazil.
12. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986), *Nature*, **323,** 533–536.
13. Henrique, H. M., Lima, E. L., and Seborg, D. E. (2000), *Chem. Eng. Sci.*, **55,** 5457–5469.
14. Henrique, H. M. (1999), MSc thesis, PEQ/COPPE/UFRJ, Rio de Janeiro, RJ, Brazil.
15. Billings, S. A., Chen, S., and Korenberg, M. J. (1989), *Int. J. Control*, **49,** 2157–2189.
16. Cleran, Y., Thibault, J., Cheruy, A., and Corrieu, G. (1991), *J. Ferm. Bioeng.*, **71,** 356–362.