COMPARISON OF FEED-FORWARD AND RECURRENT NEURAL NETWORKS FOR BIOPROCESS STATE ESTIMATION

M. N. KARIM and S. L. RIVERA

Agricultural and Chemical Engineering Department,
Colorado State University, Fort Collins, Co. 80523

ABSTRACT

The application of artificial neural networks to the estimation of bioprocess variables will be discussed. In fermentation processes, direct on-line measurements of primary process variables usually are unavailable. The state of the cultivation, therefore, has to be inferred from measurements of secondary variables and any previous knowledge of process dynamics. This research investigates the learning, recall and generalization characteristics of neural networks trained to model the nonlinear behavior of a fermentation process. Two different neural network methodologies are discussed, namely, feed-forward and recurrent neural networks, which differ in their treatment of time dependence. The neural networks are trained by backpropagation using a conjugate gradient technique, which provides a dramatic improvement in the convergence speed. The objective is to use environmental and physiological information available from on-line sensors to estimate concentrations of species in a bioreactor. Results of the neural network estimators are presented, based on experimental data available from the ethanol production by *Zymomonas mobilis* fermentation. The feed-forward and recurrent neural network methodologies are demonstrated to perform suitably as unmeasurable state estimators. Both networks offer comparable abilities of recall, but recurrent networks perform better than feed-forward networks in generalization.

KEYWORDS

Biotechnology; estimation; fermentation processes; nonlinear systems; neural networks; recurrent neural networks.

INTRODUCTION

The successful operation, control and optimization of fermentation processes rely heavily on the availability of a fast and accurate evaluation of the system performance. This is turn requires reliable real-time process variable information. The scarcity of biosensors for direct on-line measurements of primary process variables, such as biomass, substrate and product concentrations, makes the identification procedure difficult. Most industrial fermentation control policies are based upon the use of infrequent off-line sample analysis, leading to poor process operability and regulation. The state of the fermentation, therefore, has to be inferred from measurements of secondary variables and any previous knowledge of the process dynamics.

Recent research has been directed towards developing estimation techniques for unmeasurable process variables. Among these, artificial neural networks have been of interest because they

enable the direct modelling of nonlinear processes, without requiring a pre-specified detailed dynamic relationship. Neural networks can *learn* to approximate large classes of nonlinear functions (Rumelhart *et al.*, 1986) during the *training* procedure on a historical database of the process. Examples of this approach have been presented by Bhat and McAvoy (1990), Lant *et al.* (1990) and Van Breusegem *et al* (1991).

In this study, two neural network estimators were developed to predict the main process variables involved in the fermentative production of ethanol by the anaerobic bacteria *Zymomonas mobilis.* The neural network methodologies studied are the feed-forward and the recurrent neural nets, which differ in their treatment of time dependence. A conjugate gradient optimization is used in the network training algorithm since it has shown to require less computational time and space than the traditionally-used steepest descent. The estimators use the process information available from on-line sensors to determine the current fermentation states, like biomass, substrate and product concentrations. Estimation results will be compared with experimental data obtained from a laboratory-scale fermenter run in batch mode. The learning, recall and generalization abilities of both neural network estimators are compared and discussed.

NEURAL NETWORKS AS STATE ESTIMATORS

Artificial neural networks consist of a large number of simple interconnected nonlinear processing units (neurons) having adjustable connection strengths (weights). During the network training procedure, the interconnection weights are adjusted so that the application of a set of inputs produces the desired set of outputs (Fig. 1). Inputs to the neural network estimator may consist of the manipulative inputs to the plant (temperature, pH, flow rates, etc.), together with other measured process variables, like carbon dioxide evolution rate. The corresponding process outputs (biomass, substrate and product concentrations) provide the desired *teacher* signal, which trains the network. At present, in chemical and biochemical engineering applications, the most widely used neural net training method is *backpropagation*, a generalization of the Least Mean Squares Error algorithm. Backpropagation uses an iterative gradient search technique to minimize a performance function, equal to the mean difference between the desired and the actual network outputs.

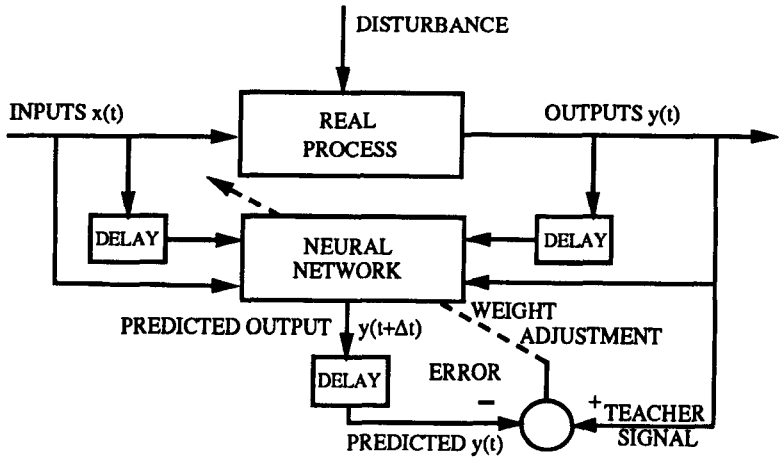

Fig. 1. Block diagram of the neural network estimator.

Feed-forward Neural Networks

Consider the layered feed-forward neural network in Fig. 2(a). In general, it consists of N neurons

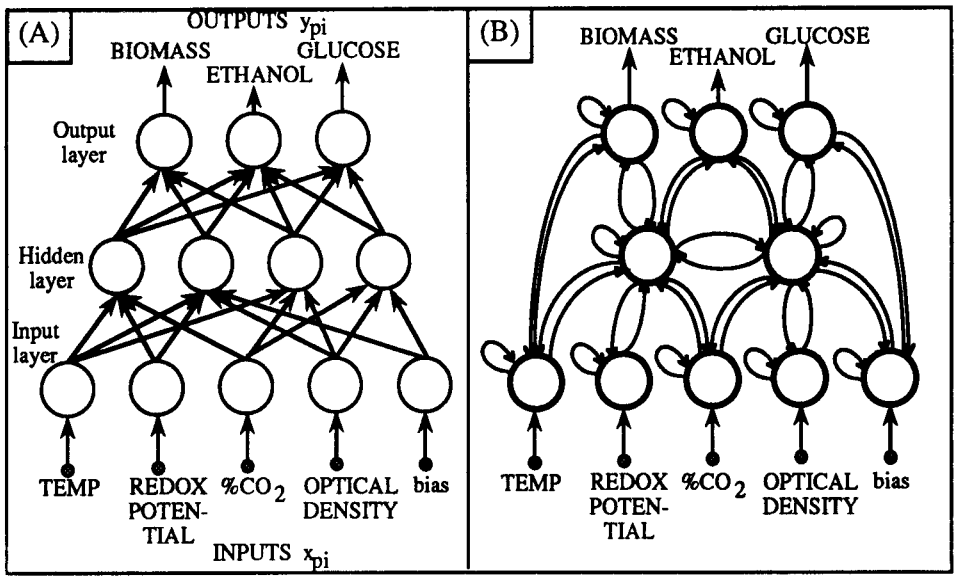

Fig. 2. Sample configurations for neural-based state estimation of ethanol fermentation. (a) Feed-forward neural network (b) Recurrent neural network.

distributed in one input layer, one hidden layer and one output layer. When an input vector $\mathbf{x}_p$ from a pattern p is presented and propagated through the network, a corresponding output vector $\mathbf{y}_p$ is obtained from the output layer. Each connection between neurons has a weight w_{ij} associated with it. The bias unit provides a threshold for the activation of the neuron. A typical neuron j performs two calculations: a weighted linear combination of its inputs to obtain the activation S_{pj} and a nonlinear transformation of this activation value, usually by means of a non-decreasing and differentiable sigmoid function $f(\sigma) = 1/(1 + e^{-\sigma})$:

$$y_{pj} = f(S_{pj}) = f(\sum_{i=1}^{N} w_{ij} x_{pi} + w_{N+1,j}) \tag{1}$$

The set of training examples consists of P input/output vector pairs $(\mathbf{x}_p, \mathbf{d}_p)$. Weights are initially randomized to small values, and thereafter, selected so as to minimize the objective function defined as the total squared estimation error $E(\mathbf{w})$ for all output units and all patterns:

$$E(\mathbf{w}) = \sum_{p=1}^{P} \frac{1}{2} \sum_{k=1}^{L} (d_{pk} - y_{pk})^2 \tag{2}$$

The objective function $E(\mathbf{w})$ is clearly a continuous differentiable function of every weight, therefore a nonlinear numerical optimization technique can be used to minimize it. The *backpropagation* algorithm traditionally optimizes the objective function by the gradient method of steepest descent, requiring calculation of $(\partial E_p)/(\partial w_{ij}^{(q)})$. However, steepest descent is highly inefficient as an optimization method, requiring many iterations for convergence and lacking robustness. Other researchers have investigated alternate methods to train the neural network, like Owens and Filkin (1989). They expressed the training procedure as a problem of solving a set of stiff differential equations. However, the number of iterations required is still large and the calculation of second order derivatives makes the procedure prohibitive for large order networks. Instead, the training algorithm in this study uses Le's Conjugate Gradient method for unconstrained optimization (Le, 1985).

Le's Conjugate Gradient method is a type of conjugate gradient method with dynamic optimization of step size $\alpha^{(q)}$. In each iteration q, the vector of interconnection weights $\mathbf{w}$ is improved by:

$$\mathbf{w}^{(q+1)} = \mathbf{w}^{(q)} + \alpha^{(q)}\mathbf{S}^{(q)} \tag{3}$$

where $\alpha^{(q)}$ is chosen to minimize $\mathbf{w}$ along the search direction $\mathbf{S}^{(q)}$. The algorithm generates n mutually conjugate directions and minimizes a positive definite quadratic function of dimension n in at most n steps, where n is the size of the vector $\mathbf{w}$. The sequence of search directions $\mathbf{S}^{(q)}$ are formed by linear combinations of the current steepest descent direction and the previous search direction.

In summary, the learning procedure consists of two phases (Hoskins and Himmelblau, 1988). First, the inputs are propagated through the network in a feed-forward fashion to produce output values that are compared to the desired outputs, resulting in an error signal for each of the output nodes. Second, the error is propagated backward through the network, and the error gradients are calculated to update all the weights in the network.

Recurrent Neural Networks

Different from the feed-forward neural networks, recurrent networks are more general, in the sense that connections are allowed both ways between a pair of neurons, and even from a neuron to itself, as shown in Fig. 2(b). They are especially able to perform *temporal association.* In this situation, a particular output *sequence* is produced by the network in response to a specific input sequence (Hertz *et al.*, 1991), making the networks suitable for modelling of dynamic systems.

Pearlmutter (1989) has developed an algorithm for training a general recurrent network. The neuron dynamics evolve according to

$$\tau_i \frac{dy_i}{dt} = -y_i + f(\sum_{j=1}^{N} w_{ij} y_j + x_i(t)) \tag{4}$$

where τ_i is the neuron time constant and $f(\sigma)$ is the sigmoid function. The inputs $x_i(t)$ are functions of time, as well as the desired outputs $d_i(t)$. In this case, an appropriate error function for an overall time domain $0 - T$ is given by

$$E(\mathbf{w}) = \frac{1}{2}\int_0^T \sum_{k=1}^{L}(d_k(t) - y_k(t))^2 dt \tag{5}$$

where the sum is only over the output neurons with specified desired values d_k.

The derivative $(\partial E)/(\partial w_{ij})$ needed for gradient descent can be calculated from

$$\frac{\partial E}{\partial w_{ij}} = \frac{1}{\tau_i}\int_0^T z_i f'(h_i) y_j dt \tag{6}$$

where $h_i(t) = \sum_j w_{ij} y_j(t)$ and $z_i(t)$ is the solution of the dynamical equation

$$\frac{dz_i}{dt} = -\frac{1}{\tau_i} z_i + \sum_j \frac{1}{\tau_j} w_{ji} f'(h_j) z_j + (y_k(t) - d_i(t)) \tag{7}$$

with the boundary condition $z_i(T) = 0$ for all i at the end point.

The algorithm can be summarized as follows: first integrate Eq. (4) forwards from $t = 0$ to $t = T$ and store the resulting $y_i(t)$'s. Then, integrate Eq. (7) backwards from $t = T$ to obtain the $z_i(t)$'s. Finally, evaluate the integrals in Eq. (6) to find the appropriate gradient descent increments. Since

our inputs and outputs are discrete time series, rather than continuous signals, all integrations were performed using first order finite difference approximations.

DESCRIPTION OF THE EXPERIMENTS

The *Zymomonas mobilis* bacterial fermentation for ethanol production proceeds anaerobically, where glucose is metabolized to ethanol, carbon dioxide and lactic acid. *Z. mobilis*, strain ATCC 10988, was used in this study. The nutrient medium used and analytical methods for determination of glucose, ethanol and cell-mass concentrations are described in Rivera and Karim (1990).

Experiments were conducted in a 7-liter Chemap stirred fermenter with a working volume of 3 liters at pH=6. Following heat sterilization, the fermenter was seeded with 10% inoculum and made anaerobic by nitrogen gas sparging. The fermenter is equipped with sterilizable Ingold pH and redox potential electrodes. The temperature is measured with a contact J-type thermocouple. Optical density measurements of the fermentation broth were determined every 15 minutes by measuring the absorption at 575 nm of a 5 ml. sample in a Bausch and Lomb Spectronic 21 spectrophotometer. The exhaust fermentation gas is passed through the condenser and through a silica gel filter to remove water vapor. The carbon dioxide content is measured by means of an Infrared CO_2 Analyzer. All the measurement signals are received by an HP3497A Data Acquisition and Control Unit and sent forth to an HP220 microcomputer for data logging and control.

DESCRIPTION OF THE ESTIMATION PROBLEM

Previous research in control and optimization of the fermentative ethanol production revealed the importance of the cultivation temperature for optimum ethanol yields (Rivera and Karim, 1990). Several sets of batch fermentation data were obtained at different temperatures, providing a suitable candidate for training the neural network on the behavior of the process at various environmental conditions. Five data sets were generated at temperatures 30°C, 33°C, 35°C, 37°C and 39°C. The estimator was required to predict current biomass, glucose and ethanol concentrations every 15 minutes, using on-line measurements of temperature, redox potential, % CO_2 in exhaust bioprocess gas, and optical density. Therefore, each data set consisted of 41 time patterns corresponding to 15 minute sampling during 10 hrs. Figure 3 shows an example of the measurements used as inputs to the net, for the data set at temperature of 35°C.

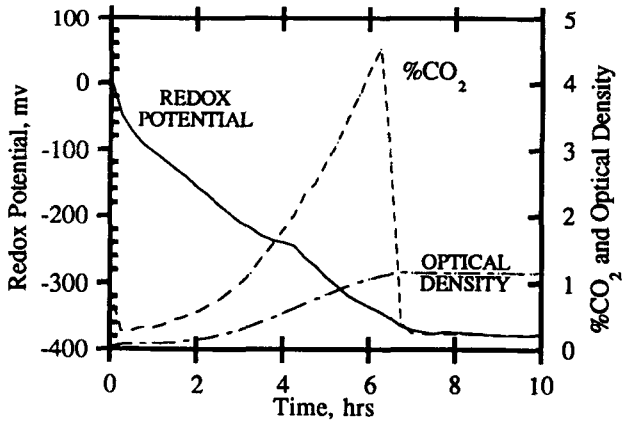

Fig. 3. Inputs for training the neural networks. Experimental set: T=35°C.

The five input-output data sets were normalized between 0.1 and 0.9 and presented to the net-

works sequentially during training. The use of the conjugate gradient optimization for the training procedure decreased the number of iterations required for convergence, compared to those required by normal backpropagation. On average, convergence was achieved in 10–15 iterations, whereas steepest descent requires more than 1000, and convergence is not guaranteed.

The number of inputs and outputs define the number of nodes in the input and output layer of the network (see Fig. 2(a)). One hidden layer was used in this study, since it has been shown (Cybenko, 1989) that one hidden layer is enough to approximate any nonlinear function provided there exists a sufficient number of hidden nodes. To determine the number of nodes in the hidden layer, different number of hidden nodes were proposed and evaluated according to the Mean Squared Error criteria, and the configuration which provided the least error was selected. In the case of the feedforward neural network, the configuration with the least error was found to be the one with four hidden nodes.

Since feed-forward neural networks are usually used to model nonlinear static processes, an improvement in the estimation was expected if the temporal dependence of the data was incorporated in the estimator. This resulted in the use of the recurrent neural network as an estimator of the bioprocess. In the same way as with the feed-forward net, several topologies were tried to find the one which provided the least error in the estimation. Figure 4 shows the sum of squared error in the estimation (SSE) as training proceeds for different number of nodes in the network. The least error in estimation is provided by the network with 10 nodes, being two of them hidden.

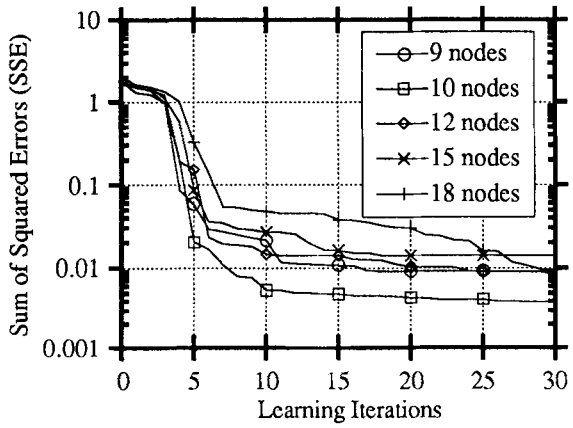

Fig. 4. Sum of squared estimation errors -vs- number of learning iterations for different number of nodes in the recurrent network.

DISCUSSION OF RESULTS

The performance of the neural networks was evaluated based on the sum of squared estimation errors (SSE), using normalized process variables between 0.1 and 0.9. After the training was completed, the networks were tested by presenting a data set which was used during training, i.e. batch run at T=35°C. This enables an evaluation of the network's recall characteristics. Further, the networks were tested using a data set which was not used during training. This data set consists of fermentation data available for a batch run where the temperature was varied twice during the fermentation, providing a time-varying temperature profile. For this data set, the temperature started at 35°C; at 5 hrs. it was decreased to 34°C; and then at 6 hrs, it was again decreased to 33°C, where it remained for the rest of the fermentation. The same on-line measurements were

available for this run, as for the previous runs.

Figure 5(a) shows the real experimental process variables for the data set at temperature 35°C and the feed-forward neural network estimations. The SSE for this case is 0.03. Figure 5(b) shows the estimations for the temperature profile data set. It can be seen that the feed-forward network was clearly unable to generalize the case outside the original training sets, giving an SSE of 0.14.

Figure 6(a) shows the T=35°C data set for the recurrent neural network estimations. The network is fully connected and the time constants of each neuron were kept constant at 0.5. For the T=35°C data set, the SSE is comparable to that obtained in Fig. 5(a). Figure 6(b) shows the results for the temperature profile data set, and it can be seen that the inclusion of recurrency improved the generalization ability of the estimator. However, there still exists a small estimation error throughout the glucose concentration profile.

In trying to improve the performance of the recurrent neural network, the neuron time constants were incorporated in the optimization algorithm so they could also be adjusted during training. The converged values of the time contants were on the average between 1.0 and 2.5. Using these time constants, the two test data sets were tried again. Figure 7(a) shows the estimations obtained for the T=35°C data set. The SSE now is 0.08, which is almost twice as much as the SSE obtained keeping all the time constants at the same value (0.5). However, for the temperature profile data set (Fig. 7(b)), the estimation has improved considerably, decreasing the SSE to half of the value obtained in Fig. 6(b).

CONCLUSIONS

The feed-forward and recurrent neural network methodologies were demonstrated to perform suitably as unmeasurable state estimators. Using raw measurement data, and without requiring a simplified process model derived from first principles, the neural networks are able to approximate the nonlinear behavior of the fermentation process.

Both neural network estimators are similar in the sense that they require the determination of a set of parameters (weights) by the minimization of a sum of squared errors cost function, using error backpropagation. In this study, the conjugate gradient algorithm used for minimization, reduced 100-fold the number of learning iterations required for convergence when compared to traditional backpropagation, which uses steepest descent. However, the recurrent neural network model, being more complex than the feed-forward network, requires more computational time for training. This would be of concern only when on-line weight adaptation is needed. Ideally this shouldn't be required if a representative non-linear neural-based process model can be found.

In general, it was found that both types of neural networks offer comparable abilities of recall. The recurrent networks seem to perform better in generalization situations, where the estimator is tested with patterns it was not originally trained for. This could be due to the inherent inclusion of time-dependence in the recurrent network, a feature which the feed-forward network completely lacks.

The network topologies (number of hidden nodes and number of hidden layers) were found to be adequate in each case. The approach used to define the number of hidden nodes, though partly heuristical, enabled us to observe situations when a high order network doesn't improve the estimation at all, but rather deteriorates it. This occurs because higher order systems map high order nonlinearities of the system, which could be regarded as noise.

From the results obtained, it can be seen that both neural networks are able to estimate the biomass and ethanol concentrations relatively well, whereas the glucose concentration estimate

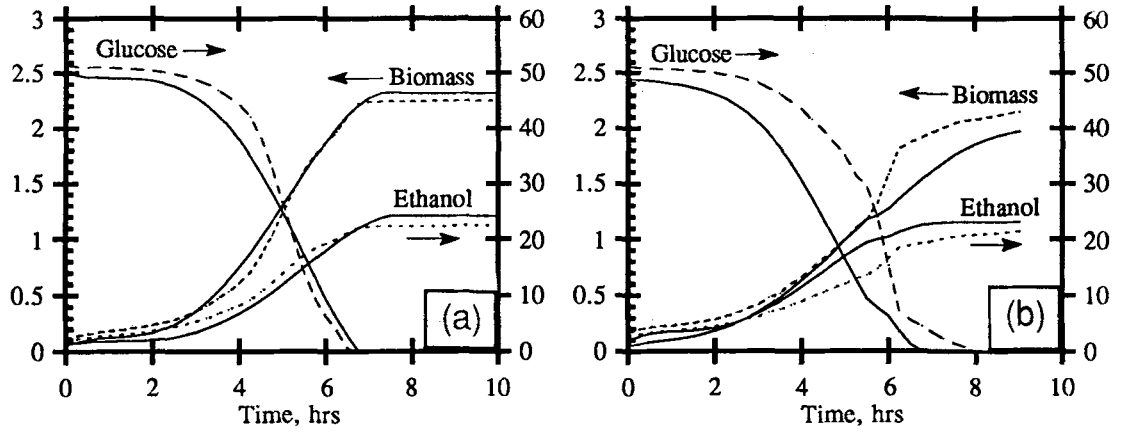

Fig. 5. Feed-forward neural network estimations using data sets at: (a) Temperature 35°C; SSE = 0.03. (b) Temperature profile; SSE = 0.14.

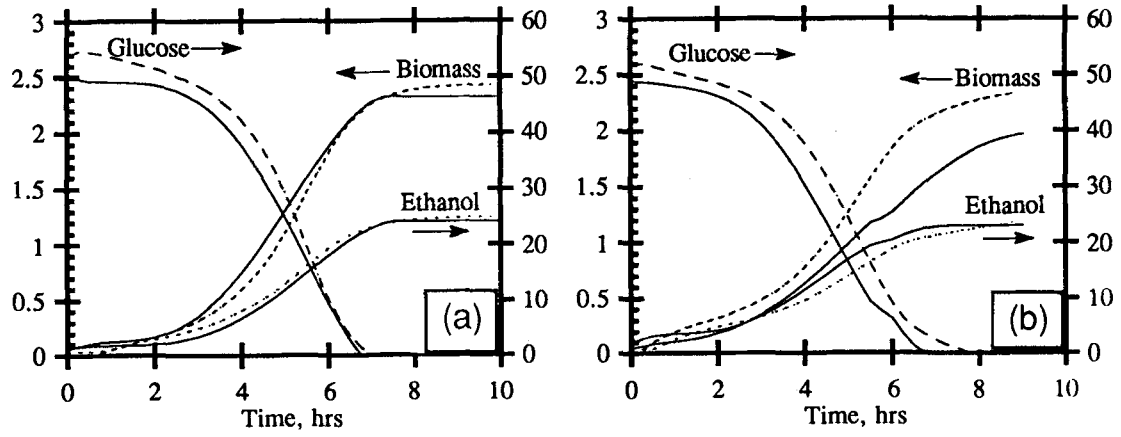

Fig. 6. Recurrent neural network estimations using data sets at: (a) Temperature 35°C; SSE = 0.04. (b) Temperature profile; SSE = 0.10. All neuron time constants are set at 0.5 and the network is fully-connected.

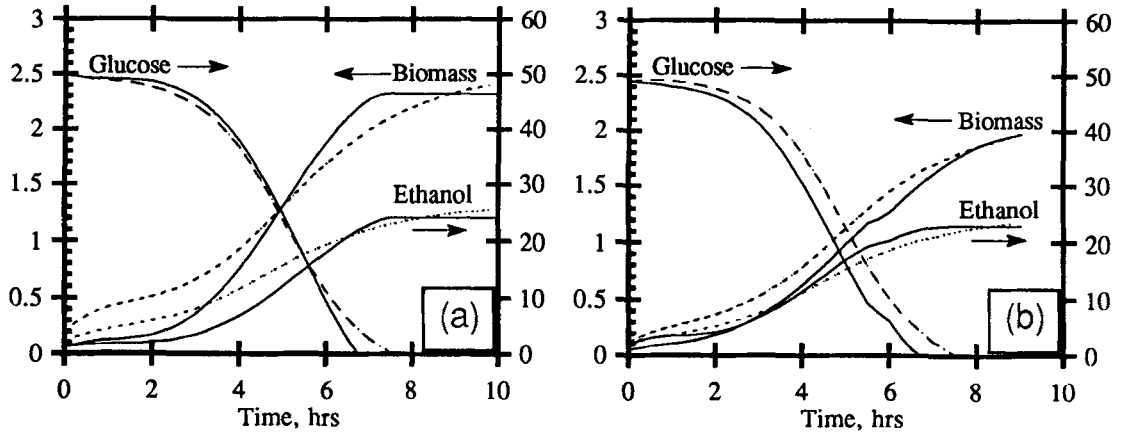

Fig. 7. Recurrent neural network estimations using optimized neuron time constants and fully connected neurons. Data sets at: (a) Temperature 35°C; SSE = 0.08. (b) Temperature profile; SSE = 0.06.

often deviates from its true value. It is apparent that the nonlinear network model is unable to capture and represent all relevant process phenomena from the four measured inputs provided. We believe that the inclusion of an additional measurement, i.e. acid production rate, will provide the additional information required by the network to produce a better estimation.

Researchers have postulated that an asynchronous updating of the neuron activations can prove useful to improve the performance of the recurrent networks. This was partially demonstrated when the neuron time constants were adjusted during training together with the weights. The recall characteristics of the recurrent network deteriorated, but its generalization ability improved. Further studies have to be done to investigate the influence of the time constants in the generalization abilities of the recurrent neural network.

REFERENCES

Bhat, N. and T. J. McAvoy (1990). Use of neural nets for dynamic modeling and control of chemical process systems. *Comp. Chem. Eng.*, 14, 573–583.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Contr. Sign. Syst.*, 2, 303.

Hertz, J., A. Krogh and R. G. Palmer (1991). *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, Ca.

Hoskins, J. C. and D. M. Himmelblau (1988). Artificial neural network models of knowledge representation in chemical engineering. *Comp. Chem. Eng.*, 12, 881–890.

Lant, P. A., M. J. Willis, G. A. Montague, M. T. Tham and A. J. Morris (1990). A comparison of adaptive estimation with neural based techniques for bioprocess application. In: *Proc. 1990 ACC*, May 23–25, San Diego, Ca.

Le, D. (1985). A fast and robust unconstrained optimization method requiring minimum storage. *Math. Prog.*, 32, 41–68.

Pearlmutter, B. A. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1, 263–269.

Rivera, S. L. and M. N. Karim (1990). Application of dynamic programming for fermentative ethanol production by *Zymomonas mobilis.* In: *Proc. 1990 ACC*, May 23–25, San Diego, Ca.

Rumelhart, D. E., G. E. Hinton and R. J. Williams (1986). A general framework for parallel distributed processing. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pp 45, MIT Press, Boston, Mass.

Van Breusegem, V., J. Thibault and A. Chéruy (1991). Adaptive Neural Models for On-line Prediction in Fermentation. *Can. J. Chem. Eng.*, 69, 481–487.