A Prototype Neural Network Supervised Control System for *Bacillus thuringiensis* Fermentations

Qin Zhang, John F. Reid, J. Bruce Litchfield,* Jinliang Ren, and Shiuan-Wu Chang
Department of Agricultural Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois

Received August 3, 1993/Accepted October 21, 1993.

This article discusses the development of a prototype neural network-based supervisory control system for *Bacillus thuringiensis* fermentations. The input pattern to the neural network included the type of inoculum, operation temperature, pH value, accumulated process time, optical density in fermentation medium, and change in optical density. The output from the neural network was the predicted optical density for the next sampling time. The control system has been implemented in both a computer simulation and a laboratory fermentation experiment with promising results. © 1994 John Wiley & Sons, Inc.
Key words: microbial fermentation control • neural network simulation • backpropagation • network topology design • fermentation

INTRODUCTION

It is desirable to control microbial fermentation processes automatically at their optimal conditions for improving process yields and reducing production costs. However, microbial fermentation processes involve living systems which include a large number of complex and highly interacting biochemical reactions and transport phenomena. There are three problems in developing a reliable automatic control system for microbial fermentation processes: (1) there is a lack of reliable mathematical models to describe the cell growth and metabolite production during the fermentation process,[9] (2) there is a lack of appropriate on-line sensors which can detect the important process state variables,[11] and (3) the microorganisms themselves have a complex regulatory system within the cells and the external control system can only manipulate the extracellular environment in hopes of affecting the intracellular mechanisms.[8]

The development of new and more sophisticated systems for monitoring microbial fermentation processes enables the monitoring of a large number of process variables including physical and chemical environmental variables.[1] However, the relations between most measurable state variables and process outputs consists of uncertainty and are poorly understood. A neural network-based control system would be suitable for monitoring those data-rich and knowledge-poor processes because it can (1) perform rational reasoning without a precise mathematical model of the process, (2) work with incomplete process information, and (3) handle a large amount of information quickly through parallel processing.

One of the most attractive features of a neural network is the ability to train itself through the underlying relations in the training data, which provides a tool to estimate poorly understood processes such as microbial fermentation processes based on acquirable process data. To apply neural networks, it is necessary to have appropriately coupled input-output data sets, so-called training patterns, of a process. The accuracy of the neural network estimation is strongly restricted to the completeness and the preparation method of training patterns as well as the structure of the neural network.

This research investigated the feasibility of applying neural network techniques in microbial fermentation process control. This was accomplished through studies on (1) the effects of network structure on simulation accuracy, (2) the effects of training data preparation on training results, (3) the development and implementation of a prototype neural network supervised control system for laboratory scale *Bacillus thuringiensis* fermentation. The control objective of this system was to regulate the laboratory fermentor to achieve the maximum microorganism growth rate during the exponential growth phase.

MATERIALS AND METHODS

Neural Networks

Neural networks consist of many processing elements called neurons interconnected by information channels. Each neuron may receive multiple input signals but gives only one output signal. The input signals are amplified or dampened by a weight associated with each information channel. The neuron then sums all weighted inputs and passes them through a threshold to determine the activation value (the fired output signal) of this neuron. The interneuron activity can be modeled by an activation function. In a

* To whom all correspondence should be addressed.

CCC 0006-3592/94/060483-07

backpropagation neural network, the function is commonly in the form of a sigmoid function:

$$y_j = -\left[1 + \exp\left(\sum_{i=1}^{n} x_i w_{ij} + \theta_j\right)\right]^{-1} \tag{1}$$

where x_i is an input signal, y_j is the fired output signal, w_{ij} is a weight associated with the input signal x_i, and θ_j is the threshold value of neuron j.

All neurons are linked in terms of certain network topologies. The connection of a backpropagation neural network is based on a multilayer and feed-forward network topology. The input layer, which may consist of many receptors, receives input information from external signal sources, and the output layer, which may also consist of many transmitters, provides output information to the user. One or more hidden layers connect between the receptors and the transmitters with different connection strengths.

A neural network solves problems by adapting to the nature of input data, namely the underlying information the data carried.[5] This adapting (or training) process changes weights associated with all the information channels between neurons. In backpropagation networks, the process is executed according to an error feedback method, by which it will first update activity values of all the neurons corresponding to input data based on current weights and then adjust the weights according to the error between fired outputs and desired outputs to reduce the error. Let E represent the summation of output errors,

$$E = \frac{1}{2} \sum_j (y_j - d_j)^2 \tag{2}$$

where y_j is the fired output and d_j is the desired output from neuron j. According to the maximum gradient scheme, a common scheme used for neural network training, each connection weight w_{ik} is changed by Δw_{ik},

$$\Delta w_{ik} = -\alpha \frac{\partial E}{\partial w_{ik}} \tag{3}$$

where α is a positive constant controlling the speed of learning.

The Prototype System

The prototype system was a supervisory control system that consisted of a digital measurement and control unit (DCU) and a neural network supervisory unit (NNSU) (Fig. 1). The NNSU was used to estimate process conditions and to supervise the DCU operation according to the estimated process state. This is an attractive approach because many bioreactors have well-developed stand-alone controllers which are capable of measuring and regulating process variables such as medium temperature, pH value, dissolved oxygen concentration, and fermentor stirrer speed.

The digital measurement and control unit was a stand-alone PID (proportional-integral-differential) control system for the bench scale fermentor. The control software, at the fermentor control board (FCB) in the unit, could measure all the above process variables and perform PID control to those variables. The DCU interface could display process variables and allowed the user to interrupt the PID controllers at any time to change set-points. The DCU also allowed the user to adjust set-points remotely so it could be used as the local controller in a supervisory control system.

The neural network supervisory unit was a decision-making system which searched for the most appropriate set-points for PID controllers in the DCU during the course of operation. The key component in the NNSU was a neural network process estimator which could predict the desired

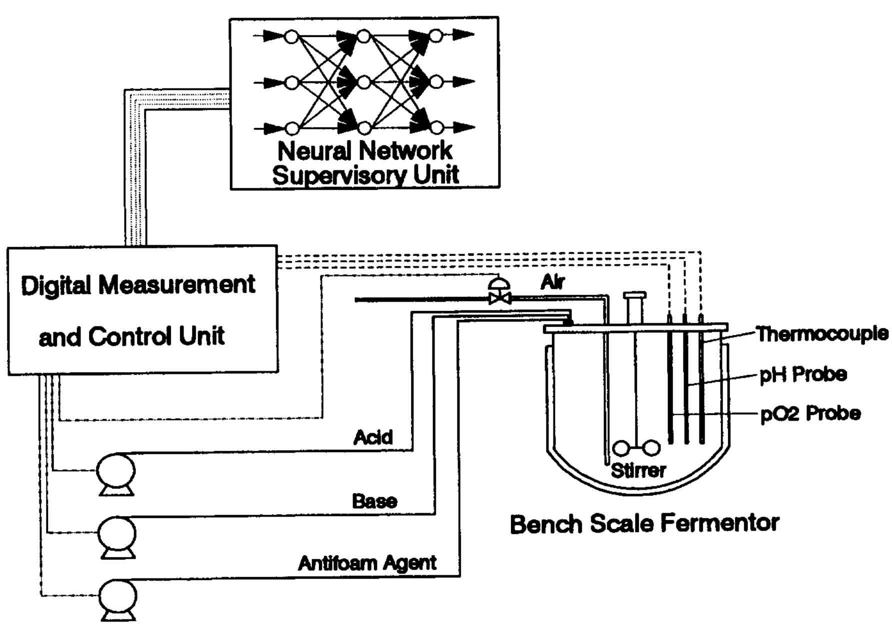

Figure 1. Neural network supervised fermentation control system.

process output based on measured state variables. This neural estimator was constructed based on a backpropagation algorithm and trained according to a forward model, by which the neural network was trained with the process inputs as the network inputs and the process outputs as the network outputs.

Fermentation and Process Data Set

This study was carried out in a bench scale fermentor (BIOSTAT-MD) which has a working volume of 2 L. A stirrer, driven by a dc motor, was installed at the center of the fermentor. Medium temperature was monitored by a thermocouple and was regulated by controlling water temperature through a glass jacket. Medium pH value was detected by a pH probe and adjusted by supplying acid or alkali through two peristaltic pumps. Dissolved oxygen concentration was measured by a pO_2 probe and was controlled by changing the stirrer speed automatically or the air supply rate manually. The foam level was checked by a foam sensor and controlled by pumping antifoam solution into the vessel as needed.

A C-2 recipe of the growth medium[4] (suggested by Ecogen, Longhorne, PA), which consisted of a carbon source, nitrogen source, phosphates, and mineral salts, was used in all the experiments. The inocula used in those experiments were either vegetative cells or spores of *B. thuringiensis.* The size of the inocula was around 1% of the effective growth medium volume at the beginning. The growth environment ranged from 26°C to 32°C for medium temperature, from 6.0 to 8.0 for medium pH level, and from 2.5% to 30% for dissolved oxygen concentration in the growth medium. The growth environment for each experiment was constant.

The neural network estimator was trained based on a set of historic process data obtained from 18 fermentation experiments. This data set consisted of the type of inoculum, the accumulated process time, optical density of growth medium, medium temperature, pH level, and dissolved oxygen concentration measured at each hour of the process. Since antifoam was always added before the inoculation, and during the cultivation as needed, the effects on the growth of *B. thuringiensis* was not considered; therefore, the amount of antifoam was not recorded in the historic data set.

RESULTS AND DISCUSSION

Effects of Network Topology

When applying neural network techniques to simulate a complex biochemical process, like microbial fermentation, the input information, represented by process data, is important to obtain reliable process estimation. For every process variable in the input pattern of the training set, there should be a receptor in the input layer of the network. Similarly, corresponding to each output variable in the output pattern, there should be a transmitter in the output layer of the network. The associated variables contained in both input and output patterns were defined as the input–output data pairs. Selection of the input–output data pairs could affect the ability to train the network.

As discussed in the previous section, several process variables were recorded from a series of preliminary fermentation experiments. Some were critical for making process control decisions, some were difficult to be predicted, and some could be ignored for process control purposes. One of the significant advantages of applying neural network technology is its ability to evaluate the informational value of input variables. This feature of neural networks, which could be observed from network training error converging curves, could be used to select the input variables for the network.

In this study, for instance, the neural network estimator was designed to predict optical density of the growth medium. Different patterns of network input variables, including accumulated processing time, medium temperature, pH value, dissolved oxygen, and optical density, were investigated. The type of inoculum was considered as an important process variable since the fermentation inoculated with spores of *B. thuringiensis* would have a longer lag phase than when inoculated with vegetative cells. Figure 2 presents a topology of a network which consisted of six variables in its input pattern and one variable in the output pattern. The variables involved in the input pattern were the type of inoculum, accumulated process time, medium temperature, pH value, optical density of the current sample, and change in optical density since the previous sampling. The variable in the output pattern was the predicted optical density at the subsequent sampling time.

The complete process information is usually represented by a combination of many variables.[2] Therefore, it is important to involve as many informational variables and ignore those less informational variables in the input pattern. To perform pattern analysis, several patterns were evaluated through network training. For instance, pattern 1 contained

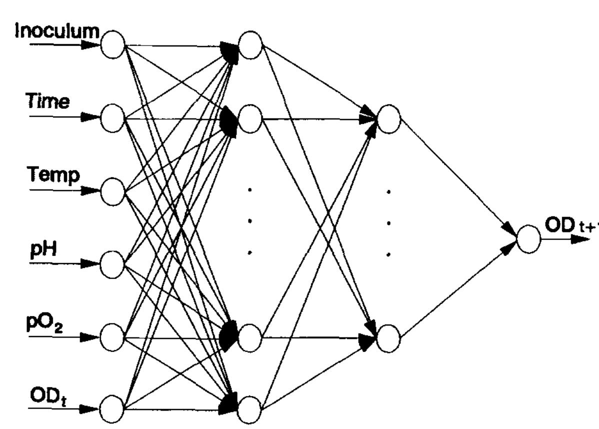

Figure 2. Topology of the neural network with a normalized real output pattern.

all available variables, including the type of inoculum, accumulated fermentation time, medium temperature, pH value, optical density, and dissolved oxygen concentration. Patterns 2 and 3 excluded optical density and dissolved oxygen concentration, respectively. Pattern 4 replaced dissolved oxygen concentration with the change in optical density since the previous sample. To obtain satisfactory results, up to 20,000 cycles of training were performed in trial training based on a few input patterns, and stable errors were observed after 5000 cycles of training for all trail training. Therefore, subsequent analyses were based on 5000 cycles of training, which required training periods of 2 to 6 h on a SUN SPARCstation 1 + workstation) depending upon the topology of the network.

The training error converging rate was used as the criterion for input pattern evaluation (Fig. 3). The error converging rate should be high if an input pattern contained a set of complete informational variables. In this study, pattern 1 was treated as the basis of the evaluation since it contained all the available variables of the process. The error converging curves showed that pattern 2, which excluded medium optical density from the pattern, showed a noticeably slower converging rate than pattern 1, whereas pattern 3, which excluded medium dissolved oxygen concentration, had an error convergence rate close to that of pattern 1 after 5000 cycles of training. These results indicated that optical density was a more informational variable than the dissolved oxygen concentration in the ranges evaluated from the neural network training point of view, even though both variables are considered important from the physiological point of view. An interesting result was that if the dissolved oxygen concentration was replaced by the change in optical density, pattern 4 could achieve the lowest training error after 5000 cycles of training among all evaluated patterns. Therefore, pattern 4 was selected as the input pattern for the rest of the study.

The number of hidden layers and the number of neurons in each hidden layer play an important role in training the neural network accurately and efficiently. However, there is a lack of a standard techniques for designing neural network topology. A practical method of trial-and-error, by which several neural networks with different topology would be trained and the training accuracy would be compared, was recommended for topology design.[3] In this study, neural networks with different numbers of hidden layers and neurons were arbitrarily designed and tried. Figure 4 shows error converging curves of a few networks with different topology, and Figure 5 presents the recall capability of those networks. The results demonstrated that both training accuracy, indicated by the total recall error, and training efficiency, indicated by the training error converging rate, varied with network topology design. For instance, the network of 6–30–15–1 (input neurons–first hidden layer neurons–second hidden layer neurons–output neuron) had the best topology among the networks tested (Figs. 4 and 5). Increasing the number of hidden layers and neurons was not always helpful in improving the training accuracy but did always decrease the training efficiency. In fact, too many hidden layers in a backpropagation neural network would be harmful to the training accuracy because of the decay of the error information through the hidden layers.[3]

Data Preparation

The process data used for the topology design of the neural network discussed in the previous section were represented in the form of real-value inputs and normalized outputs, namely, all the input variables were in their natural form as they were obtained, and the output variables were normalized according to their maximum possible values. One problem revealed in this method of data preparation was the poor capability of replicating the decline in optical density at the stationary phase. As the optical density curves indicated (Fig. 4), none of the networks could replicate the decline in optical density.

To solve this problem, a method of input variable normalization was first tried using the network topology 6–30–15–1. All the input variables were normalized according to their possible ranges, that is, the real-valued

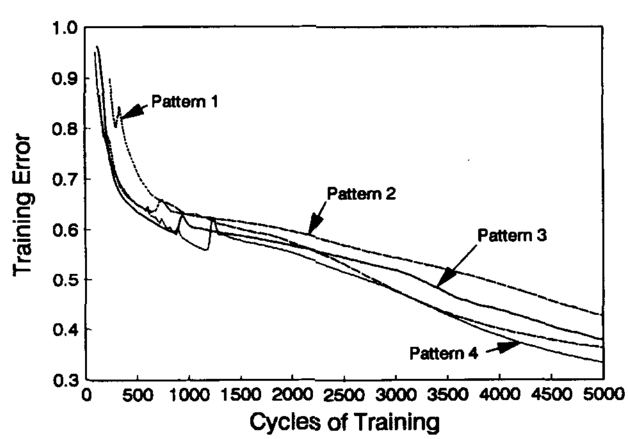

Figure 3. Training error convergence curves from neural networks with different input patterns (pattern 1 contained all available variables, 2 excluded OD, 3 excluded pO_2, and 4 replaced pO_2 by ΔOD).

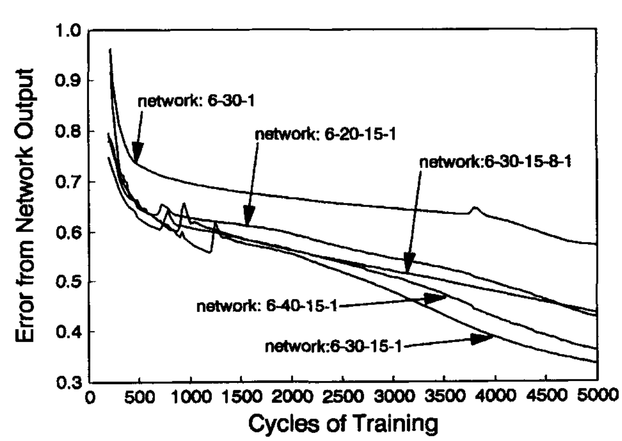

Figure 4. Training error convergence curves from neural networks with different topology.

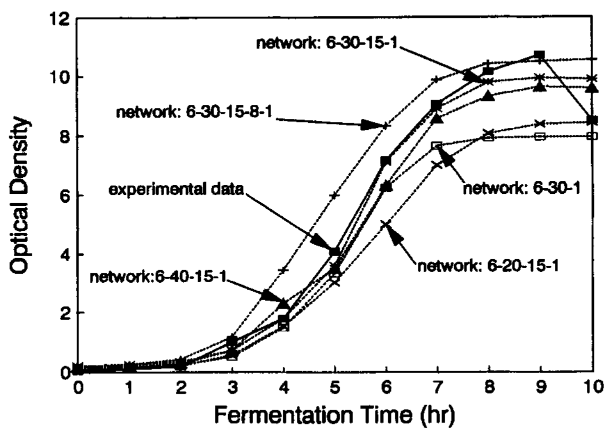

Figure 5. Comparisons of neural network recall capability with different topology.

input variables were converted into a range of [0, 1] based on their possible maximum values. Therefore, both the input and the output variables of the neural network were treated in the same way. No significant improvement on the accuracy of process replication was achieved through this method.

The next effort was the qualitative representation method, by which the output variable was represented by grades, as in fuzzy logic.[12] The network output pattern contained the weights of each predicted grade of optical density. The real-valued prediction, therefore, would be the weighted sum of the median values of those qualitative grades. Several different numbers (6 to 12) of qualitative grades, namely the number of transmitters in the network output pattern, with different defined ranges corresponding to the real value of optical density were tried. The results showed that a 9-grade network (Fig. 6) with finer definition ranges in the lag phase was comparable to the 12-grade network in the error converging rate, whereas the training time was significantly shorter.

Based on the nature of a sigmoid threshold function, the neural network outputs of the predicted grades were decimal values within [0, 1]. In other words, the network would provide an indirect presentation of the predicted optical density. Therefore, the interpretation of the network output would be critical for obtaining an accurate prediction. In this study, three methods of network output interpretation were investigated. The first method, entitled the natural sum method, was to treat network outputs as the weights of corresponding grades. The predicted optical densities were the sum of the products of network outputs and the median values of corresponding grades. As shown in Fig. 7, this method was capable of replicating the optical density decline in the stationary phase of the fermentation, but the prediction error was large within this phase. Hence, this method was inappropriate for such a case.

The second method was the so-called threshold method. A threshold value equal to the maximum value of the network outputs was defined so it would screen out all the weights smaller than the threshold value and revalue the remaining weights to unity. The predicted optical density would be the median value of the qualitative grade corresponding to the maximum network output value. In most cases, this method could provide a prediction with fair accuracy (Fig 7). However, a big error would result if two identical maximum values were obtained from the network output.

To avoid such an error, a third method was investigated. By this method, called the normalized sum, all the network outputs were normalized according to the sum of those outputs. Thus, the modified weights were real weights of qualitative grade, the portions in the sum of network outputs. This method could avoid the shortcomings of both previous methods and predict optical densities with reasonable accuracy. (Fig. 7). All three of these methods were based on the crisp data conversion method, by which the median value was used to represent the grade.

Another significant factor which would affect the prediction accuracy in all three methods was the width of the defined ranges of the qualitative grades, especially within the lag phase of the fermentation. The main reason was

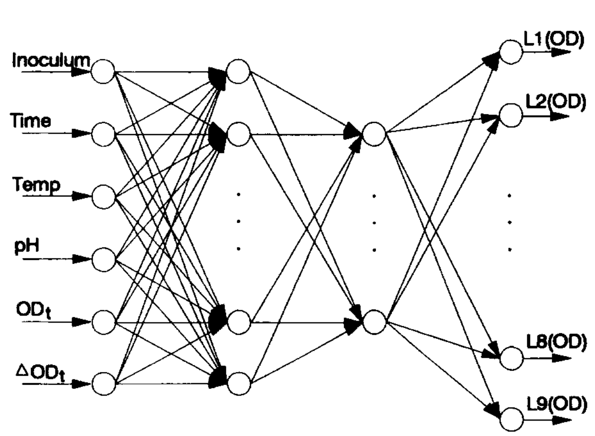

Figure 6. Topology of the neural network with qualitative grade output pattern.

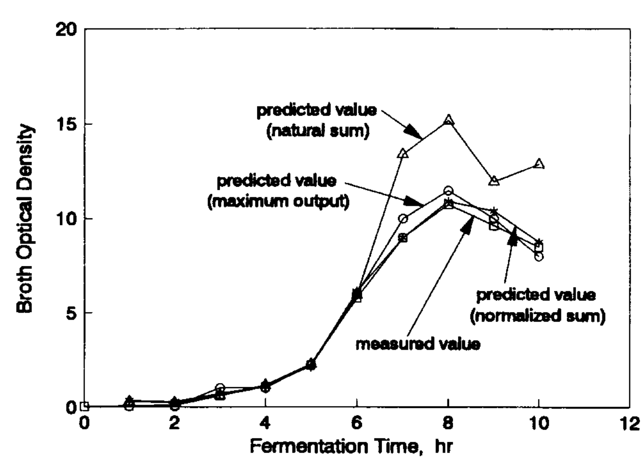

Figure 7. Comparisons of measured and predicted medium optical densities from a fermentation process. The predicted values were derived from natural sum of outputs, the threshold (maximum output), and normalized sum of outputs.

that the median value of a qualitative grade was used in all those methods. One way to improve the predication accuracy for those methods was to define finer ranges for qualitative grades at low optical density. However, the effectiveness of this method would be limited by the capacity as well as the training time of the neural network. A fuzzy representation method could map real-valued data into two or more qualitative grades to specify the degrees of the real-valued data involved in those grades. A standard triangle fuzzy membership function[6] was appropriate for mapping real-valued data into qualitative grades in this case. Figure 8 shows the defined ranges and the fuzzy memberships of real-valued broth optical density in nine qualitative grades. For example, if the real-value optical density was 6.74, the coincident qualitative grades were G6 with a fuzzy membership of 0.63 and G7 with a fuzzy membership of 0.37.

The fuzzy method could represent the real-value data better in qualitative grades than the crisp method. If the fuzzy representation method was introduced into the normalized method of network output interpretation discussed earlier, the neural network estimator could provide a more accurate prediction than the crisp method, especially within both the lag and the stationary phases of the fermentation (Fig. 7). The above discussion demonstrates that the methods of data representation and network output interpretation both play important roles in improving prediction accuracy of a neural network estimator.[7]

Implementation of Supervisory Control Systems

The neural network-based supervisory control system was implemented for both computer simulation and operation of a laboratory scale fermentor to evaluate the effectiveness of the system. The control objective was to achieve the highest optical density, which was an indication of microbe growth in this study, within the shortest fermentation time. The input process variables to the neural network estimator were type of inoculum, accumulated process time, fermentation medium temperature, pH value, optical density, and change in optical density from the previous sample. The tuned control set-points in the supervisory controller were medium temperature and pH value. *Bacillus thuringiensis* spores were used as the inoculum. Operations were terminated 1 h after the fermentation turned into the stationary phase.

The neural network estimator predicted the future (next sample) optical density of the medium based on measured process variables with possible set-points for manipulating variables. The supervisory controller searched for desired set-points yielding the possible highest increase in optical density according to the estimated results and then turned the slave controller, installed in the standby DCU, in terms of the selected set-points. Both the simulation and experimental results of the implementation were plotted (Fig. 9). The predicted optical densities matched the measured values well within the lag and log phases (the first 7 h period) of the fermentation. The mean square of the error between predicted and measured optical densities were 0.293 with the largest error being 0.880 at the 6 h operation time. Control set-points for medium temperature and pH value were selected through a multivariable search process[10] according to the predicted optical density. The neural estimator stopped at the accumulated process time of 7 h because the measured optical density surpassed the upper limit (12.0) of the estimator. Those results confirmed that the neural network estimator worked well to predict the fermentation medium optical density, and the supervisory controller could provide correct control supervision for stand-alone PID controllers in the digital measurement and control unit.

The supervisory control system does not replace PID controllers but improves the performance of PID controllers by tuning the set-points according to process conditions. So, even though not comparable, it is interesting to see that a supervisory control system could achieve improved performance over the PID control alone. Figure 10 shows that the fermentation controlled by a neural network-based supervisory control system was capable of achieving both a high growth and high yield, whereas the fermentations

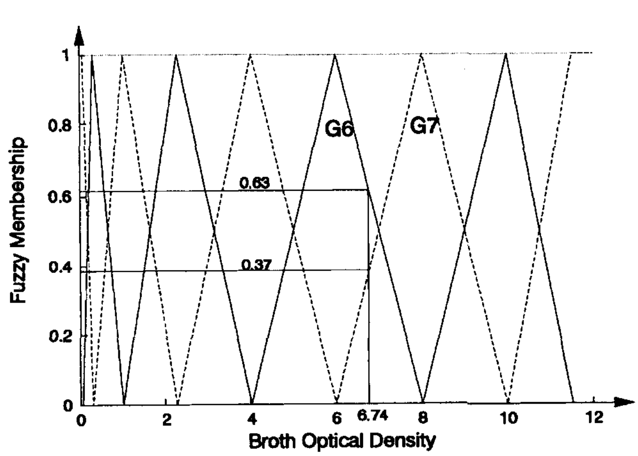

Figure 8. Fuzzy membership of nine discrete groups of broth optical density in the range 0.0 to 1.0.

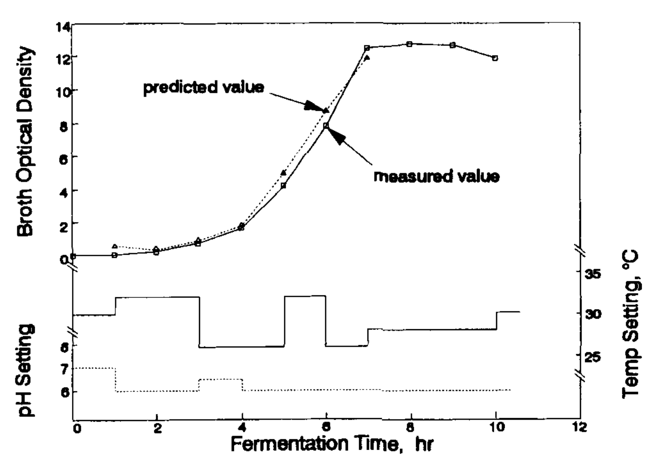

Figure 9. Predicted and measured medium optical density vs. fermentation time obtained from the neural network supervised control experiment.

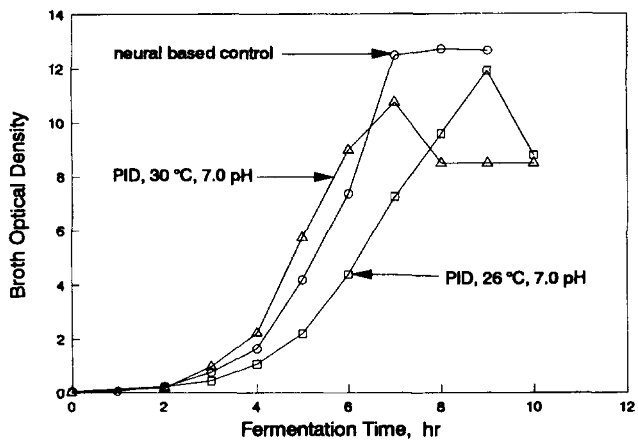

Figure 10. Measured medium optical density vs. fermentation time obtained from the neural network supervised control and PID control experiments.

controlled by PID controllers alone could achieve either a high growth rate or a high yield, but not both. The set-points for those PID controllers were selected so that either a high growth rate or a high peak yield could be achieved based on historic fermentation operations, as discussed earlier in this article.

CONCLUSIONS

A prototype neural network based supervisory control for *B. thuringiensis* fermentation was developed and implemented. The results from the simulation and experimental results of the neural network controller were compared. The neural network based supervisory control technique was capable of improving the control performance of a fermentation process by utilizing underlying information carried by acquirable process variables and regulating the fermentation under its optimal operating conditions throughout the process

Based on the system development, the authors recommend a general approach for designing a neural network-based supervisory control system effectively. This approach is a four-step procedure: (1) select appropriate input–output data pairs for monitoring the process; (2) design a suitable network topology for the neural network-based process simulator; (3) train the simulator according to a training data set obtained from the previous operations of this process; and (4) validate the supervisory control system through computer simulation and experimental operation. It is important to point out that some process variables might be informational from the fermentation kinetics point of view, but not from the neural network training point of view. For increasing training speed, it is helpful to exclude such variables in the input–output data pairs to develop an effective neural network simulator.

A neural network is an attractive tool for automatic control of uncertain biological processes. Since a neural network trains through examples, the reasonable response range of the network depends on the completeness of the training data set. The preparation of training data as well as the design of network topology both played important roles in improving the learning accuracy of the network. Simply increasing the number of hidden layers and the number of neurons in hidden layers did not always improve learning ability. The use of fuzzy data in a neural network showed promising results in improving representation and learning accuracy.

References

1. Armiger, W. B. 1985. Instrumentation for monitoring and controlling bioreactors. pp. 133–148 In: M. Murry (ed.), Comprehensive Biotechnology: The principles, applications, and regulations of *biotechnology in industry, agriculture, and machine.*
2. Bull, D. N. 1985. Instrumentation for fermentation process control. pp. 165–174 In: M. Murry (ed.), Comprehensive Biotechnology: The principles, applications, and regulations of biotechnology in industry, agriculture, and machine. Pregamon, Oxford.
3. Caudill, M. 1991. Neural network training tips and techniques. AI Expert **6**(1): 56–61.
4. Chang, S. W. 1993. Studies of growth kinetics and estimation of oxygen uptake rate for *Bacillus thuringiensis.* Ph.D. Dissertation, University of Illinois at Urbana-Champaign.
5. Cox, E. 1992. Integrating fuzzy logic into neural nets. AI Expert **7**(6): 43–47.
6. Kaufmann, A., Gupta, M. M. 1988. Fuzzy mathematical models in engineering and management science. North-Holland, New York.
7. Lawrence, J. 1991. Data preparation for a neural network. AI Expert **6**(11): 34–41.
8. Rolf, M. J., Lim, H. C. 1985. Systems for fermentation process control. pp. 165–174 In: M. Murry (ed.), Comprehensive biotechnology: The principles, applications, and regulations of biotechnology in industry, agriculture, and machine. Pergamon, Oxford.
9. Shimizu, K., Morisue, T. 1989. Efficient control strategies for bioreactor systems. pp. 1159–1162 In: Proceedings of the 28th SICE annual conference. Society of Instrument and control Engineers, Tokyo.
10. Stoecker, W. F. 1989. Design of thermal systems, 3rd edition. McGraw-Hill, New York.
11. Williams, D. 1990. Problems of measurement and control in biotechnological processes. pp. 167–213 In: M. A. Winkler (ed.), Chemical engineering problems in biotechnology. Elsevier Applied Science, London.
12. Zadeh, L. A. 1976. The concept of a linguistic variable and its application to approximate reasoning–I. Information Sciences. **8**: 199–249