Neural-Network Modeling and Optimization of Induced Foreign Protein Production

Arun Tholudur and W. Fred Ramirez
Dept. of Chemical Engineering, University of Colorado, Boulder, CO 80309

An experimental verification and validation of the neural network parameter function approach to modeling dynamic systems is provided. The neural-network parameter-function modeling scheme utilizes some a priori process knowledge (usually material balances) and experimental data to develop a dynamic neural-network model. Other models based on fundamental principles are also developed. The experimental system under consideration is the host-vector system Escherichia coli D1210 and plasmid pSD8, which produces the foreign protein β-galactosidase under the effect of the inducer IPTG. Optimal operational conditions are derived and the neural-network-based model is shown to better predict the dynamics and optimum for protein production than the proposed fundamental kinetic models.

Introduction

The use of recombinant bacteria to produce foreign protein is one method to produce large quantities of the desired protein in a relatively short period of time. There has been a significant research effort to describe the foreign production dynamics in recombinant bacteria (Georgiou, 1988; Glick and Whitney, 1987; Zabriskie et al., 1986). In order to optimize such a system, a model that describes the growth kinetics and the protein production dynamics is necessary. Such a model has to be necessarily macroscopic in order to be suitable for use in optimal control studies.

Even though well-formulated structured and unstructured models have been developed (Bentley and Kompala, 1989; Peretti and Bailey, 1986; Shu and Shuler, 1989), none of them is appropriate as a model for use in optimization routines. Certain unstructured models (Bailey and Ollis, 1986; Williams, 1967), did not include the effect of an inducer on the protein production rate as well as on the cell growth rate. Lee and Ramirez (1992) introduced a generalized mathematical model for recombinant bacteria that included the effect of inducer on cell growth and protein production rates.

While investigating the modeling capabilities of neural networks, we developed the method of parameter-function neural-network modeling (Tholudur and Ramirez, 1996) and applied it to foreign-protein production systems. Simulation results for fed-batch studies showed that the neural-network parameter-function modeling technique was a powerful method to model dynamic systems using the combination of process knowledge and the approximation capabilities of neural networks. In this work, we provide an experimental verification and validation of the neural-network parameter-function approach to model such dynamical systems. Alternative fundamental models were also derived using the same experimental data, and optimal operational conditions were developed from these models, compared, and verified.

Correspondence concerning this article should be addressed to W. F. Ramirez.

Neural-Network Parameter-Function Modeling

In Tholudur and Ramirez (1996), we presented the method of neural-network parameter-function modeling of dynamic systems. The motivation for the work arose from the fact that biotechnology reactions are difficult to model with unknown reaction mechanisms and order that have to be postulated and experimentally verified. The resulting fundamental model is only as good as the postulated functional forms in the model. Neural networks, which are universal-function approximators, can instead be used to learn these functions in dynamic models. Thus, the utilization of prior process knowledge (usually in the form of conservation equations) coupled with the approximation capabilities of neural networks constitutes the neural-network parameter-function modeling approach. An example is illustrated below and further elaborated upon in our prior work.

A typical set of coupled differential equations that govern the dynamics of a batch process are

$$\dot{x}_1 = f_1(x_1, x_2)\, x_2 \tag{1}$$

$$\dot{x}_2 = f_2(x_1)(x_1 - x_2), \tag{2}$$

where x_1 and x_2 are the state variables of the system, and f_1 and f_2 are the unknown parameter functions. These types of differential equations are readily derived from some fundamental knowledge of the process and from simple conservation principles such as material and energy balances.

In parameter-function modeling, we attempt to capture the functional mappings f_1 and f_2. Process knowledge tells us that the function f_1 is dependent on both the state variables x_1 and x_2, while the function f_2 depends only upon x_1. If the states x_1 and x_2 are measurable, it is very easy to design a set of experiments that vary the initial conditions $x_1(0)$ and $x_2(0)$ and obtain the state-variable measurements by sampling at periodic intervals. Then, we can rewrite the basic differential equations as follows:

$$f_1(x_1, x_2) = \frac{\dot{x}_1}{x_2} \tag{3}$$

$$f_2(x_1) = \frac{\dot{x}_2}{x_1 - x_2}. \tag{4}$$

This implies that once we have the state measurements and the state derivatives at each sampling instant, we can use the preceding equations to obtain the values for the parameter functions at that instant. Two different neural networks can then be trained to mimic these parameter functions: the first one, which has two inputs, x_1 and x_2, and one output, f_1 and the second network, which has one input, x_1, and one output, f_2. Once these neural networks are trained, they can be put back in the original differential equations, and this combined neural-network–differential-equation model can be used for optimization.

Conservation Equations for a Batch Reactor

For the protein production system under consideration, the material balances across the reactor take the following form:

$$\dot{X} = \mu X \tag{5}$$

$$\dot{G} = -\mu X/Y \tag{6}$$

$$\dot{I} = 0 \tag{7}$$

$$\dot{P} = R_f X, \tag{8}$$

where X is the cell density, G is the glucose (substrate) concentration, I is the inducer concentration, and P is the activity of the protein produced. The parameter functions in these balances are the specific growth rate (μ), the yield coefficient (Y), and the foreign-protein production rate (R_f). The modeling problem reduces to one of estimating these unknown parameter functions.

There are two approaches to the modeling problem. The classic approach has been to postulate functional forms for these unknown parameter functions and estimate the constant parameters in these functions by minimizing the error between the model predictions and the observed experimental data. In this approach, we are limited by our ability to assume an appropriate functional form that is able to explain the observed experimental characteristics. The other approach of neural-network parameter-function modeling, as presented earlier, is an attractive alternative because of the universal function approximation capabilities of neural networks being integrated with a model skeleton based on fundamental conservation principles.

Experimental Work and Design

Batch experiments are performed in shake flasks to generate data in order to estimate the various parameters in the models presented earlier. A recombinant *Escherichia coli* strain D1210 transformed with the plasmid pSD8 (Gold and Stromo, 1990) is used in this work. The plasmid codes for the protein β-galactosidase and also imparts resistance to ampicillin. The cultures are grown in 2000-mL Erlenmeyer flasks with a working volume of 200 mL. The media used is the M9 minimal media supplemented with proline, leucine, and thiamine. Ampicillin at a concentration of 0.1 g/L is used to remove plasmid-free cells. Isopropylthiogalactoside (IPTG) is used as the inducer, and the protein produced is β-galactosidase, which hydrolyzes β-D-galactosides.

The cell mass is determined by optical-density measurement by absorption spectrometry at 600 nm. The glucose concentration is determined using the Hexokinase Assay Kit (Sigma Catalog # 16-1000). A modified procedure is used here to provide increased sensitivity by doubling the amounts of the assay reagent and sample used for the assay. Absorption spectrometry at 340 nm gives an estimate of the glucose concentration, which is determined by comparing the absorbance of the sample with that of a glucose standard of known concentration. The protein β-galactosidase hydrolyzes *o*-nitrophenyl-β-D-galactoside (ONPG) and the protein level is measured by absorption spectrometry at 420 nm. The actual protein activity is obtained by comparing with a standard β-galactoside of known activity.

In addition to the amount of inducer, the time of induction of the culture has been found to be a very critical variable. Preliminary experiments were performed to establish bounds on these variables and a simple design with five levels of times of induction and three levels of amount of inducer was developed. This is shown in Figure 1. Each experiment consisted of three inducer levels induced at a given time. A control shake flask with no inducer is always included in each experiment. Thus, a total of 20 shake flask cultures are grown.

Each experiment is run for a total of 14 h and samples of 2.5 mL are taken every 2 h. Optical-density measurements are performed immediately using 1 mL of the sample to determine the cell mass. The remaining 1.5 mL is centrifuged and stored for next-day glucose and protein analyses. The glucose assay is performed as explained before. Since β-galactosidase is an intracellular protein, the cells have to lysed in order for the protein levels to be determined. Sonication has been traditionally used for this purpose, but we use a

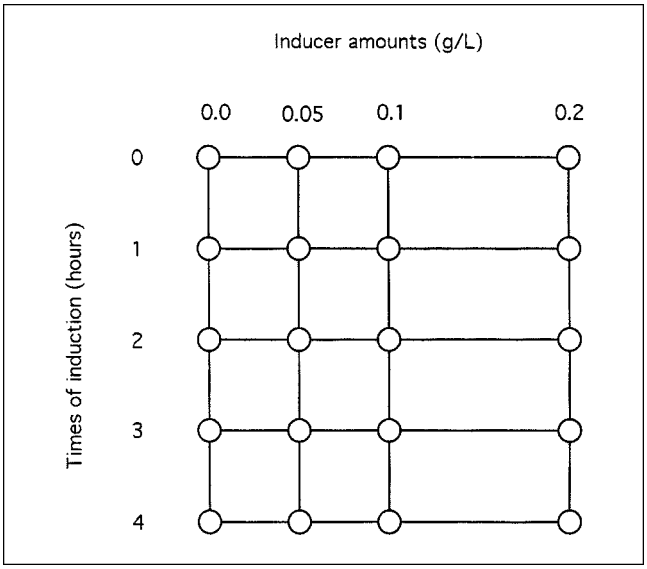


Figure 1. Experimental design.

chemical lysis technique based on detergent breaking of the cell envelope (Putnam and Koch, 1975), which yields much more consistent results.

Fundamental Models for Protein Production

Utilizing the framework introduced in the previous sections, we develop various models of protein production. First, an attempt is made to estimate parameters in the model developed by Lee and Ramirez (1992). The inability of this model to account for the experimental data observed under various times of induction motivated the need for a different model. A second fundamental model is developed, parameters estimated, and is found to perform better than the original model of Lee and Ramirez (1992).

Constant inducer model

Model Development. Lee and Ramirez (1992) presented a mathematical model for the production of β-galactosidase using recombinant *E. coli*. They assumed the existence of shock and recovery terms, which accounted for the inducer shock on the specific growth rate and its subsequent recovery. They proposed the following equations for the specific growth rate:

$$\mu = \frac{\mu_{\max} G}{K_{CG} + G}(k_s + k_r R_R), \tag{9}$$

where k_s is the shock rate effect, k_r is the recovery rate effect, and R_R is the recovery ratio. The shock and recovery dynamics were modeled by first-order kinetic expressions:

$$\frac{dk_s}{dt} = -k_1 k_s \qquad k_s(0) = 0,\ k_s(t_{\text{ind}}) = 1 \tag{10}$$

$$\frac{dk_r}{dt} = k_2(1 - k_r) \qquad k_r(0) = 1, k_r(t_{\text{ind}}) = 0, \tag{11}$$

where the rate parameters k_1 and k_2 were assumed to be functions of inducer concentration and described by Monod expressions:

$$k_1 = \frac{k_{11} I}{K_{IX} + I} \tag{12}$$

$$k_2 = \frac{k_{22} I}{K_{IX} + I}. \tag{13}$$

It is assumed that there is no shock on the culture as long as inducer is not present. However, once the inducer is added to the system, the shock term k_s is set to unity and the recovery term k_r to zero. The first-order dynamics then result in an increasing recovery factor and a decreasing effect of the shock term on the growth rate. The recovery ratio R_R was assumed to have the following functional form:

$$R_R = \frac{K_R}{K_R + I}. \tag{14}$$

The yield coefficient (Y) was assumed to be a constant. The foreign production rate was assumed to be a function of the inducer and glucose levels. The following functional form was proposed for the protein production rate:

$$R_f = \left(\frac{f_{\max} G}{K_{CP} + G}\right)\left(\frac{f_{Io} + I}{K_I + I}\right). \tag{15}$$

Parameter Estimation. The specific growth-rate model parameters $\mu_{\max}$ and K_{CG} and the yield coefficient Y are determined from the experimental growth data without IPTG. The parameters in the differential-equation model are estimated by minimizing the sum-of-squares error between the model predictions and the experimental data of cell mass and glucose concentrations. The parameters are estimated as $\mu_{\max} = 0.436\ \text{h}^{-1}$, $K_{CG} = 0.226$ g/L, and $Y = 0.869$ A/g/L. Keeping the already determined growth rate and yield parameters constant, the shock, recovery, and the recovery ratio parameters are estimated using the data with inducer and minimizing the sum of squares error between the model predictions and the experimental data of the cell mass concentration. Interestingly, this parameter estimation results in $K_{IX} = 0$ g/L. The other parameters are estimated as $k_{11} = 0.103\ \text{h}^{-1}$, $k_{22} = 0.00142\ \text{h}^{-1}$, and $K_R = 6.10$ g/L. Using a similar procedure and utilizing the entire data set (with and without inducer), the parameters in the foreign-protein production rate are estimated by minimizing the sum-of-squares error between the model-predicted protein activity and the experimental data. This results in $f_{\max} = 3580\ \text{h}^{-1}$, $K_{CP} = 72.1$ g/L, $f_{Io} = 0.00412$ g/L, and $K_I = 0.0554$ g/L.

Model Evaluation. The estimated value of $K_{IX} = 0$ implies that the shock and recovery dynamics are independent of the inducer concentration. Lee and Ramirez (1992) postulated that the shock and recovery dynamics vary with inducer concentration. Furthermore, while the model predictions of the data without inducer and those induced at $t = 0$ hour are good, the predictions are very poor at higher induction times. While the model can predict the cell growth dynamics well, it

consistently predicts lower protein activities at higher times of induction. These facts can be seen in Figure 2.

Varying inducer model

Model Development. While the constant-inducer model predicts the cell growth and glucose consumption curves reasonably well, it fails to predict the protein dynamics, especially at higher induction times. In that model, it was assumed that the concentration of the inducer remained the same throughout the batch. This resulted in the need to introduce the fictitious state variables of shock and recovery in order to account for the varying effect of inducer. An alternative to using the shock and recovery dynamics is the concept of inducer inactivation over a period of time. Introducing this concept leads to the varying inducer model that has the same governing differential equations for the cell density, glucose, and protein as introduced in the constant inducer model, but has a modified differential equation for the inducer concentration as follows:

$$\frac{dI}{dt} = -R_I X, \tag{16}$$

where R_I is the inducer inactivation rate and I now stands for the active inducer concentration. The functional forms postulated for the various rate terms in the model are as follows:

$$\mu = \frac{\mu_{\max} G}{K_{CG} + G}\left(\frac{1+\beta_1 I}{1+\beta_2 I}\right) \tag{17}$$

$$R_I = \frac{R_{\max} I}{K_{CI} + I} \tag{18}$$

$$R_f = \left(\frac{f_{\max} G}{K_{CP} + G}\right)\left(\frac{f_{Io} + I}{K_I + I}\right). \tag{19}$$

Parameter Estimation. The estimation of the parameters in this model is performed as a two-step process. First, the parameters in the specific growth rate and the inducer inactivation rate and the yield coefficient are estimated by minimizing the sum-of-squares error between the model predictions and the experimental data of cell mass and glucose concentrations. The inducer concentration does not figure in the least-square performance index to be minimized because it is not a measured variable. This results in $\mu_{\max} = 0.513$ h^{-1}, $K_{CG} = 0.594$ g/L, $Y = 0.890$ A/g/L, $\beta_1 = -127$ L/g, $\beta_2 = -180$ L/g, $R_{\max} = 1.08$ h^{-1}, and $K_{CI} = 8.91$ g/L. Once these functions are estimated, the parameters in the protein production rate are obtained by minimizing the sum-of-squares error between the model predictions and the experi-

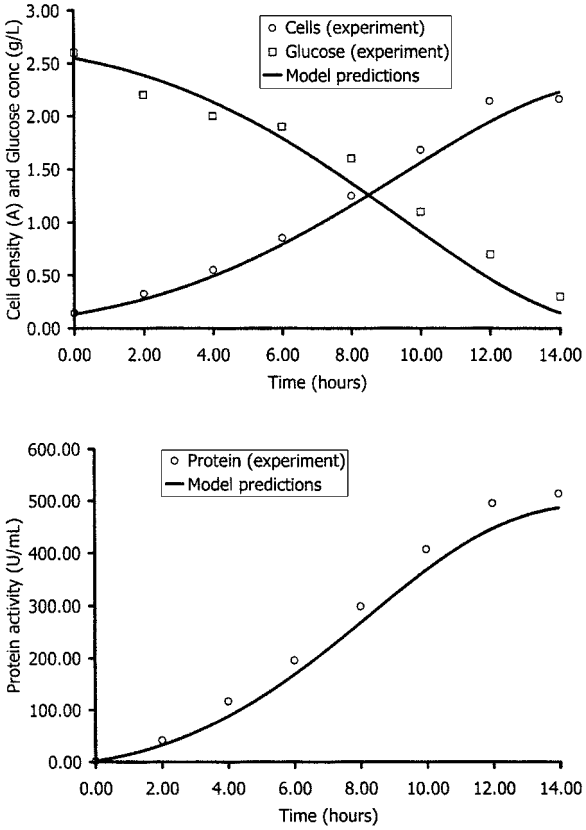


Figure 2a. Growth and production kinetics predicted by the constant-inducer model ($I = 0.1$ g/L, $t_{ind} = 0$ h).

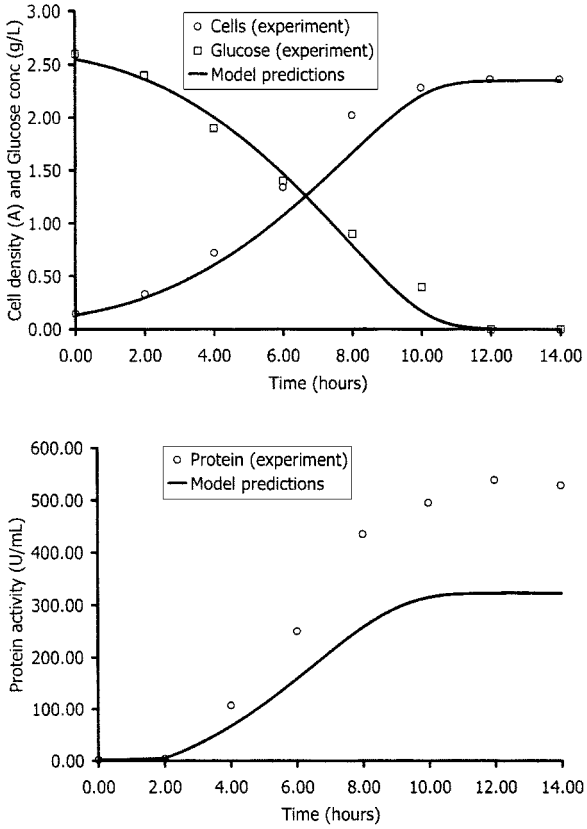
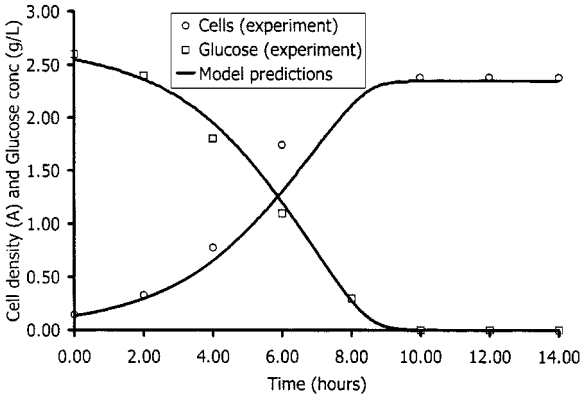


Figure 2b. Growth and production kinetics predicted by the constant-inducer model ($I = 0.1$ g/L, $t_{ind} = 2$ h).



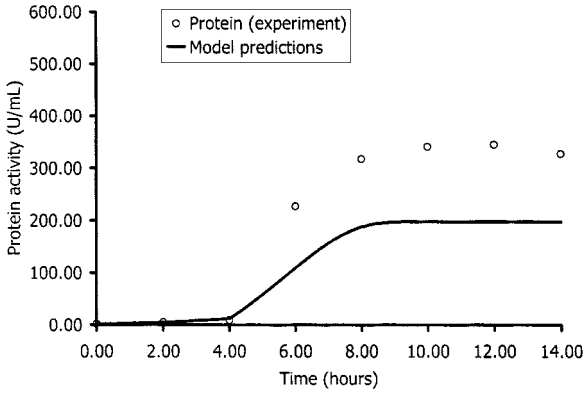
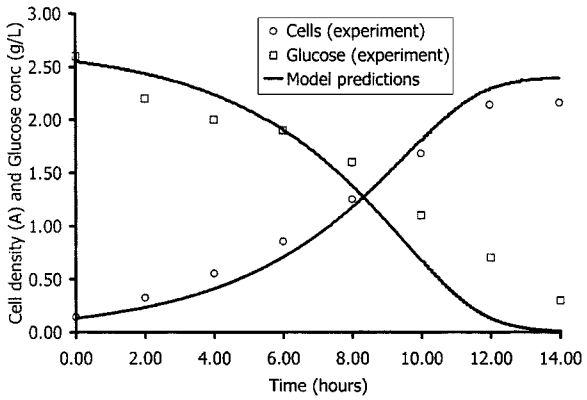


Figure 2c. Growth and production kinetics predicted by the constant-inducer model ($I = 0.1$ g/L, $t_{ind} = 4$ h).



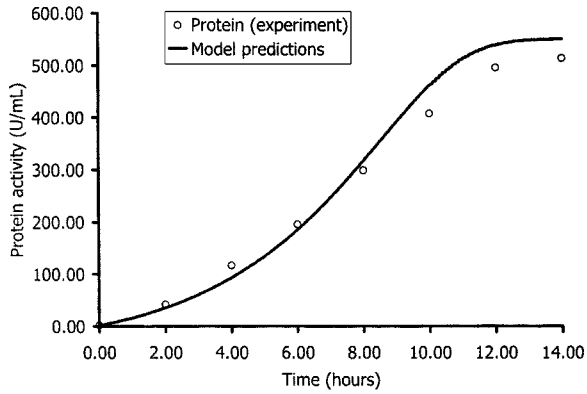


Figure 3a. Growth and production kinetics predicted by the variable-inducer model ($I = 0.1$ g/L, $t_{ind} = 0$ h).

mental data of protein activities. The protein production-rate parameters are $f_{\max} = 231\ \text{h}^{-1}$, $K_{CP} = 3.09$ g/L, $f_{Io} = 0.000337$ g/L, and $K_I = 0.00847$ g/L.

Model Evaluation. Figure 3 graphically depicts the performance of this model. Compared to the constant-inducer model, the protein activity predictions of the variable-inducer model are superior. However, the cell growth and glucose-consumption dynamics are predicted to be slower than the actual experimental data. This model also performs better at lower times of induction while demonstrating slower dynamics at higher induction times. An interesting feature to note in both the constant- and variable-inducer models is the expression for R_f that predicts protein synthesis even when no inducer is present. This feature is included because there is a small amount of protein being produced even in the absence of the inducer. When the inducer is added to the culture, the protein is overexpressed, resulting in a substantial increase in protein activity.

Neural-Network Parameter-Function Model

As can be seen from the first two attempts at modeling of this system, it is very difficult to come up with a fundamental model that explains the experimental data accurately. We are limited by the functional forms we choose for the various parameter functions. It is possible that in reality, the functional forms are totally different from those postulated. The use of neural networks to capture these parameter functions seems an attractive approach. This approach to neural-network modeling is described by Tholudur and Ramirez (1996).

Model development

The original differential equations constituting the material balances across a batch reactor can be rewritten and solved for the parameter functions as follows:

$$\mu = \frac{\dot{X}}{X} \tag{20}$$

$$\frac{\mu}{Y} = \frac{\dot{G}}{X} \tag{21}$$

$$R_f = \frac{\dot{P}}{X} \tag{22}$$

The lefthand sides of these equations constitute the unknown parameter functions that need to be estimated. Using the measurements of cell density (X), glucose concentration (G), and protein activity (P), the appropriate derivatives can be estimated and the parameter functions obtained. Neural networks can then be trained to approximate these functions.

In general, the yield coefficient Y could be time varying and a function of the other state variables. As a simplification, it is assumed to be a constant that will be estimated

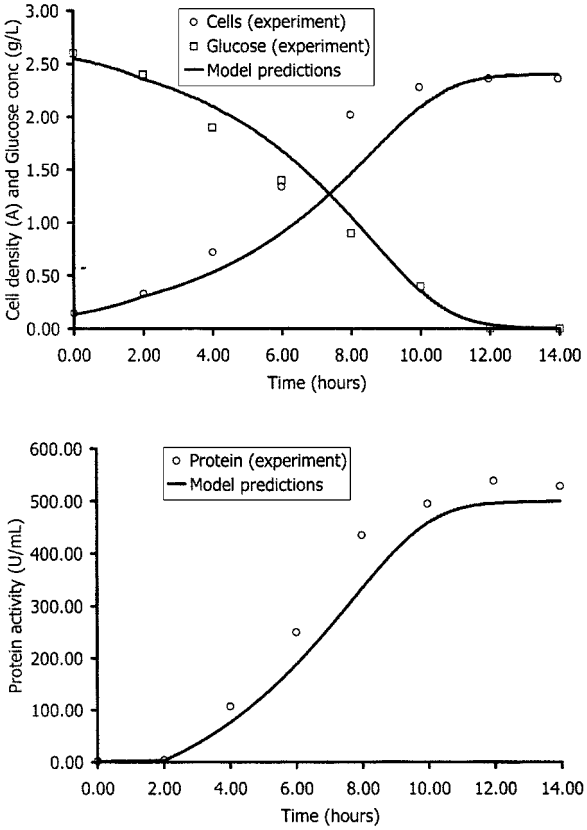


Figure 3b. Growth and production kinetics predicted by the variable-inducer model ($I = 0.1$ g/L, $t_{ind} = 2$ h).

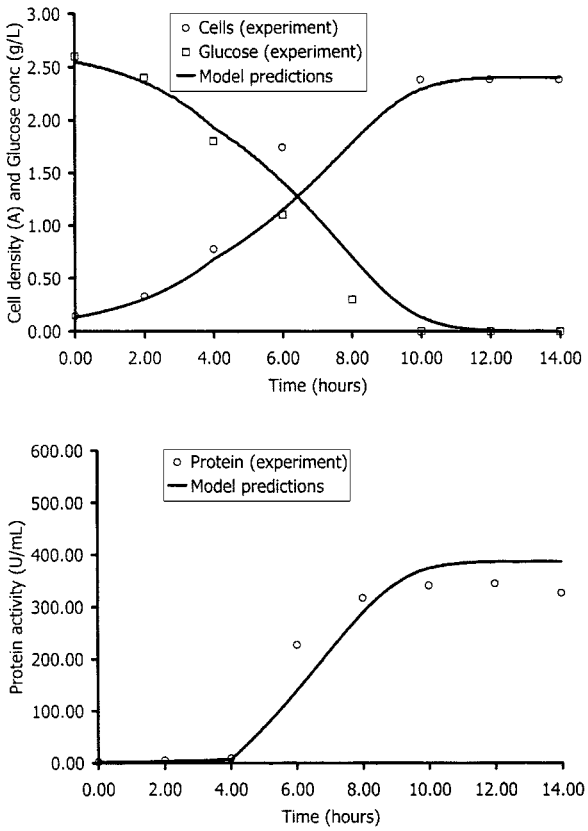


Figure 3c. Growth and production kinetics predicted by the variable-inducer model ($I = 0.1$ g/L, $t_{ind} = 4$ h).

from the experimental data. This assumption leads us to the fact that the parameter function μ can be obtained from either Eq. 20 or 21. We prefer to use Eq. 20 because the cell-density measurements are more accurate than the glucose measurements and hence the derivative estimates are expected to be more reliable.

Derivative estimations

In order to obtain values for the parameter functions μ and R_f, the values of $\dot{X}$ and $\dot{P}$ are needed. The trends in experimental data suggest that both the cell growth curve and the protein production curve have sigmoidal characteristics and so can be fitted by sigmoidal functions. While various sigmoidal functions are available (Zullinger et al., 1984) and can be chosen, a simple three-parameter logistic equation of the following form is used to fit the data:

$$V(t) = \frac{c_1}{1 + e^{-c_2(t - c_3)}}. \tag{23}$$

For each shake-flask run, a spline interpolation of the experimental data followed by a least-squares fit of the logistic function to the interpolated data is performed. Analytical differentiation of this logistic function and evaluation at interpolated times yields the derivatives of the data. This procedure also results in the smoothing of the data so that the derivative estimations are reliable.

Parameter estimations

Using the derivatives and the actual data, the values for the parameter functions μ and R_f can be obtained. This procedure is repeated for each of the shake-flask runs to give us a set of parameter functions that will accurately track the experimental data. In order to obtain a value for the yield coefficient, only data from the experiments without inducer are considered. The parameter functions μ can be obtained for the various no-induction runs as explained earlier. Using these values for μ, the state equations for cell density and glucose concentration are integrated and the yield coefficient Y estimated such that the least-squares error between the model predictions and the experimental data for the glucose concentrations is minimized. This results in a value of $Y = 0.893$ A/g/L. It is assumed that this yield coefficient remains constant for all the other shake-flask runs as well.

In the process of estimation of the parameter functions, it is desirable to scale the variables in the model so that all the variables have similar orders of magnitude. For example, in this problem, the protein activity is two orders of magnitude higher than the cell density or glucose concentration. Thus, all the variables are normalized by dividing by a normalization factor ($t_n = 16$ h, $X_n = 3$ A, $G_n = 3$ g/L, $I_n = 0.2$ g/L, and $P_n = 600$ U/mL), and the parameter estimations are per-

Table 1. Optimal Network Parameters for the Parameter Functions

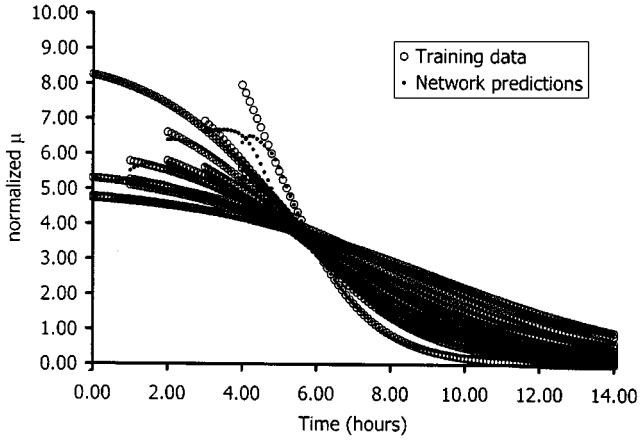
Parameter Function	Optimal Hidden Neurons	Training RMSE	Validation RMSE	Testing RMSE
μ	33	8.739×10^{-3}	5.827×10^{-3}	6.403×10^{-3}
R_f	34	5.066×10^{-3}	4.249×10^{-3}	3.898×10^{-3}

formed in the normalized variable space. The model equations are

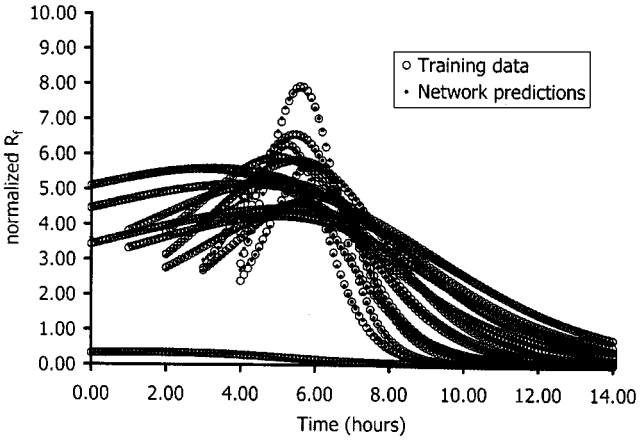
$$\frac{d\tilde{X}}{d\tilde{t}} = \tilde{\mu}\tilde{X} \qquad \tilde{\mu} = \mu t_n \tag{24}$$

$$\frac{d\tilde{G}}{d\tilde{t}} = -\tilde{\mu}\tilde{X}/\tilde{Y} \qquad \tilde{Y} = \frac{YG_n}{X_n} \tag{25}$$

$$\frac{d\tilde{I}}{d\tilde{t}} = 0 \tag{26}$$



(a)



(b)

Figure 4. Network predictions vs. training data for each parameter function: (a) $\tilde{\mu}$ and (b) $\tilde{R}_f$.

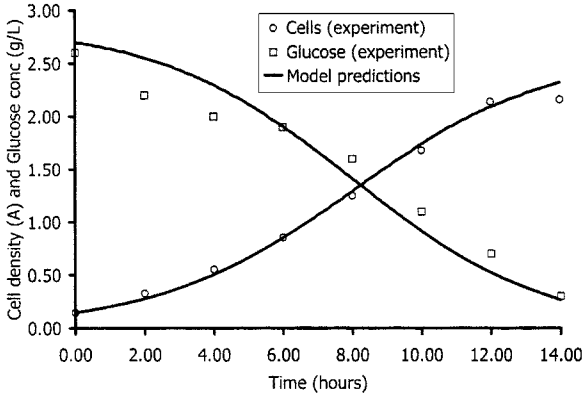
$$\frac{d\tilde{P}}{d\tilde{t}} = \tilde{R}_f\tilde{X} \qquad \tilde{R}_f = \frac{R_f X_n t_n}{P_n}. \tag{27}$$

The model evaluations are also performed in the normalized variable space and the results scaled back.

Neural-network training

Once the parameter functions have been estimated, neural networks need to be trained to learn the mappings. One of the key issues in neural-network training is the concept of generalization. While neural networks are very good function approximators and can learn input–output mappings, the utility of a network that can predict just the data it has been trained on is very limited. In other words, the neural network should try to capture the underlying trends in the data rather than "memorizing" the data points. A measure of the capabilities of the neural network to predict based on data it has not been exposed to is the generalization error (Geman et al., 1992; Moody, 1992). A lower generalization error implies that the neural network has captured the underlying trend in the data better.

While there are many different methods to incorporate generalization capabilities in neural networks, the most common methodology is the cross-validation approach (Schenker and Agarwal, 1996). In this method, the entire data set is divided into three sets: a training set, a validation set, and a



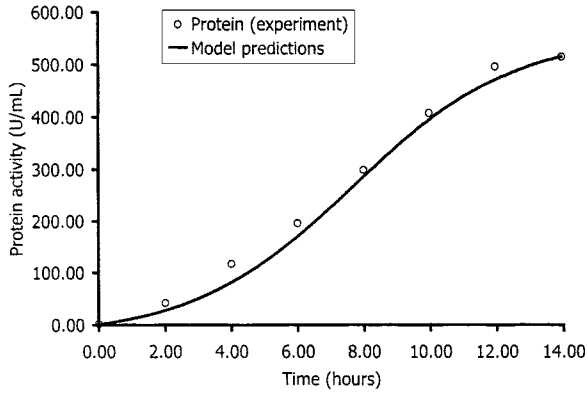


Figure 5a. Growth and prediction kinetics predicted by the neural-network model (I = 0.1 g/L, t_{ind} = 0 h).

testing set. Many different neural networks are trained on the training set, and the network with the lowest validation error is chosen as the best network. A measure of the generalization performance is then obtained by evaluating the error on the testing set.

Among the design parameters in the neural networks are the number of neurons in the hidden layer and the activation functions of the neurons in the hidden and output layer. We have used a three-layer network with an input layer, a hidden layer with neurons employing the hyperbolic tangent activation function, and an output layer with logarithmic sigmoidal activation function. Obtaining the weights in the neural networks is an optimization problem with the weights being adjusted such that the sum-of-squares errors between the actual outputs and the network predictions is minimized. As such, the initial weights (analogous to a starting point in the optimization problem) are also important so that the optimizing routine does not get stuck in local minima. The training algorithm is based on the Levenberg-Marquardt algorithm coded in the Neural Network Toolbox in MATLAB. The Appendix to this article gives the procedure that was used to ensure good generalization properties.

The choice of the inputs to the networks is another important factor. This is where prior knowledge of the process is incorporated. We do know that the growth rate and protein production rates are dependent upon the glucose and inducer concentrations. In addition, a correlation analysis of the data suggests that time and cell density are important factors as well. The amount of protein does not affect the parameter functions. Two neural networks are now trained, one to capture the function μ and the other to learn the function R_f. Both the networks have the normalized variables of time, cell density, glucose concentration, and inducer concentration, as the inputs and the corresponding parameter function as the output. The results of employing this method of training are summarized in Table 1. In Figure 4, the ability of neural networks to learn the functional relationship between the state variables and the parameter functions is shown. It is clear that the neural networks have captured the trends in the experimental data well.

Model validation

Figure 5 graphically summarizes the predictive capabilities of the neural-network-based model. Clearly, among the three models presented thus far, the neural-network model performs the best in terms of accurately predicting the dynamics of cell growth and protein production simultaneously. While the predictions of three shake-flask runs have been reproduced in Figure 5, the neural-network-based model does well on all the shake-flask runs. In addition, it also produces very reasonable dynamic responses when presented with situations the network has not been trained on (for example, times of induction of 2.5 or 3.5 h).

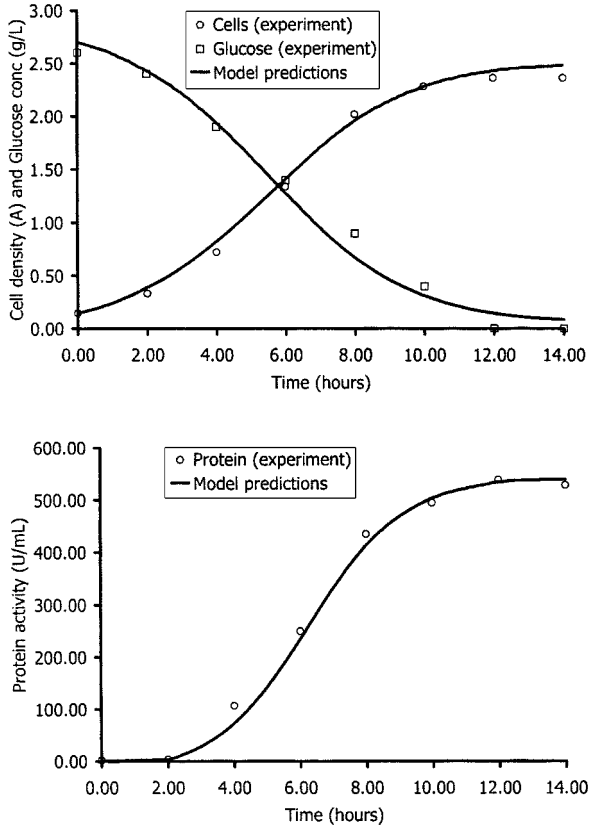


Figure 5b. Growth and production kinetics predicted by the neural-network model (I = 0.1 g/L, t_{ind} = 2 h).

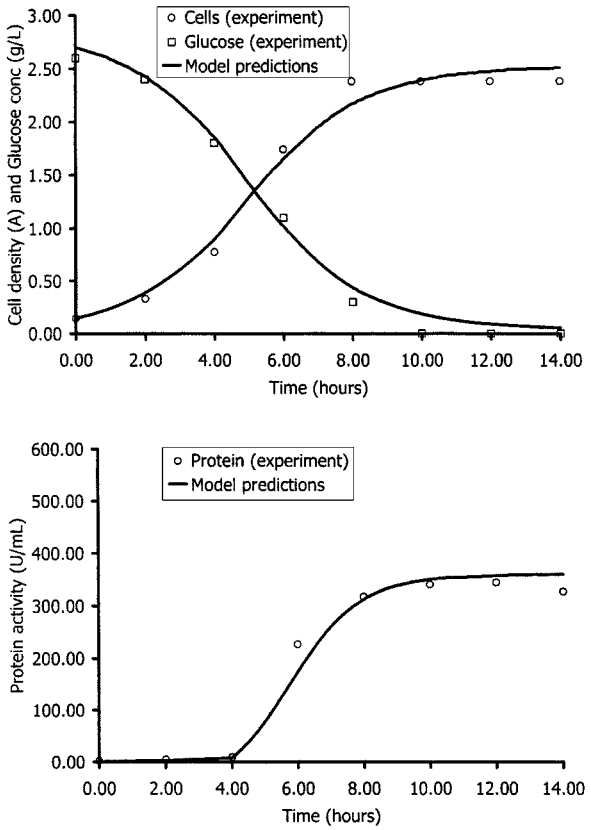


Figure 5c. Growth and production kinetics predicted by the neural-network model (I = 0.1 g/L, t_{ind} = 4 h).

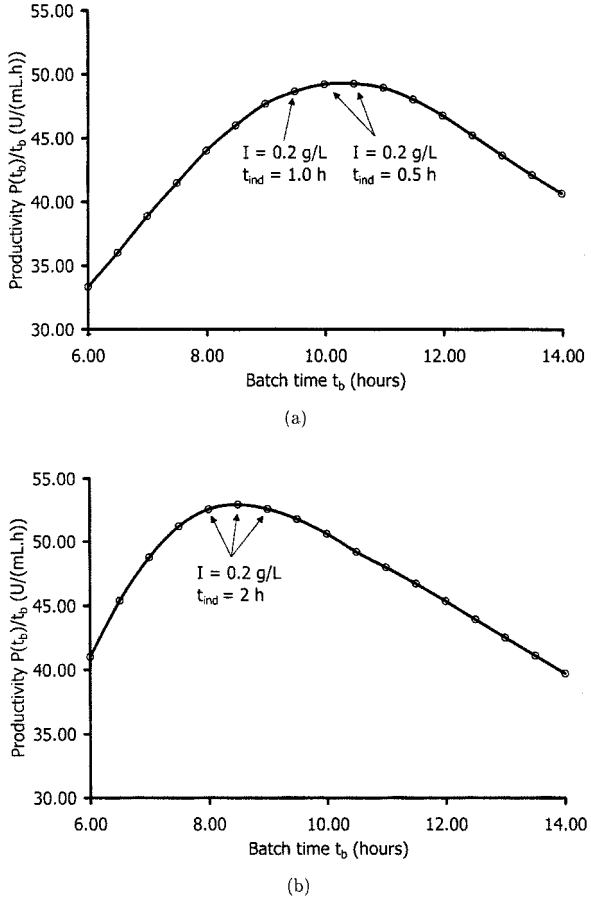


Figure 6. Productivity predictions as a function of batch time using (a) variable-inducer model and (b) neural-network-based model.

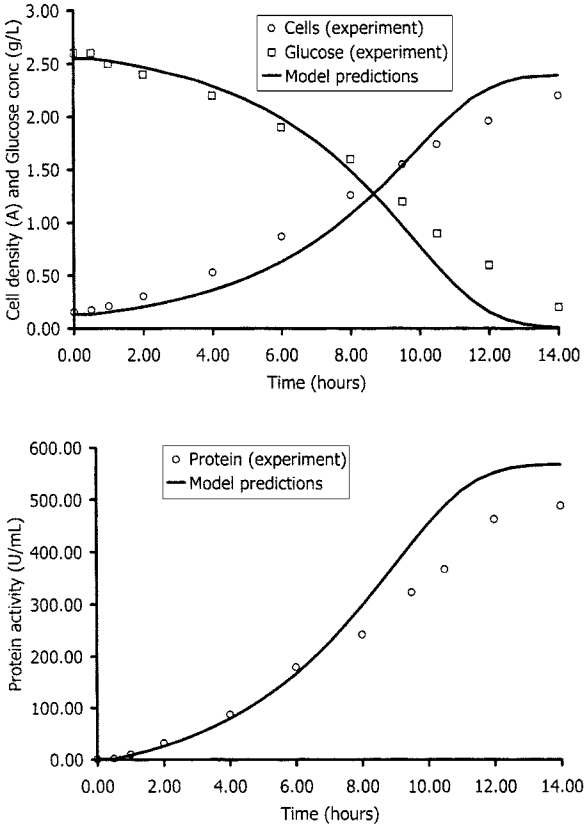


Figure 7a. Variable-inducer model predicted dynamics for the optimum $I = 0.2$ g/L and $t_{ind} = 0.5$ h vs. experimental data.

Optimization of the Models of β-Galactosidase Production

In previous sections, we have developed three models of β-galactosidase production. The constant-inducer model does not adequately represent the protein production dynamics of the process. The varying-inducer model performs better, but the neural-network-based model has the best modeling performance. The motivation for developing dynamic models of processes is so that these models can form the basis of an optimization scheme in order to obtain the best possible performance from the system. This involves, in addition to the development of predictive models, the formulation of a performance index that reflects the trade-offs involved in the process and optimizing this performance index to yield the best performance.

Two variables of interest in this problem are the inducer concentration and the time of induction. In addition, the batch time of the process is another important variable. As can be seen from the dynamic models already developed, early times of induction result in lower growth rates, while later times of induction result in lower amounts of protein. Thus, we formulate the following performance index to be maximized:

$$J = \frac{P(t_b)}{t_b}, \tag{28}$$

where t_b is the batch time of the process, and $P(t_b)$ is the amount of protein produced at the end of the batch. In other

Table 2. Optimal Model Predictions and Actual Experimental Values

Model Name	Optimal Predictions t_b (h)	t_{ind} (h)	I (g/L)	Pred. Product. J_p (U/(mL·h))	Actual Product. J_a (U/(mL·h))	% Diff. in Product. $(J_a - J_p)/J_a$
Variable inducer	9.5	1.0	0.2	48.66	39.58	−22.94
Variable inducer	10.5	0.5	0.2	49.24	40.00	−23.10
Neural network	8.0	2.0	0.1	52.55	55.25	4.89
Neural network	8.5	2.0	0.1	52.91	57.29	7.65
Neural network	9.0	2.0	0.1	52.57	56.67	7.23

words, we are trying to maximize the productivity of the process.

The optimization scheme consists of searching over the space of inducer concentrations (I), times of induction (t_{ind}), and batch time (t_b) to find the one combination that yields the maximum productivity. This is a constrained optimization problem with the following bounds on the variables:

$$0 \le I \le 0.2$$
$$0 \le t_{ind} \le 4$$
$$6 \le t_b \le 14.$$

One approach to solving this problem is by integrating the models for various induction levels, times of induction, and batch times and plotting the maximum productivity (for a given batch time) as a function of batch time. A grid with a resolution level of 0.5 g/L for the inducer concentration and 0.5 h for both the time of induction and batch time was used. Using this grid structure, I and t_{ind} were varied within their bounds for a given batch time, and the maximum productivities (for a given batch time) predicted by the models were obtained. This process was repeated for the entire range of batch times. This results in Figure 6. This figure clearly demonstrates that there exists an optimal batch time that maximizes the productivity of the system. The variable-inducer model predicts a different set of conditions as compared to the neural-network model. In order to verify the optimum obtained using both models, three cases were taken and shake flasks experiments run to experimentally verify the optimum. These conditions are summarized in Table 2. As can be seen, the neural-network model is closer to the experimentally observed productivities than the variable-inducer model. Also, the optimal productivity conditions predicted by the neural-network model result in significantly better performance (57.29 U/(mL·h) vs. 40.00 U/(mL·h)). Figure 7 also suggests that the neural-network-based model is able to predict the dynamics of the system better than the variable-inducer model. No attempt has been made to use the constant-inducer model for these optimization studies, since it is obvious that the model is not good enough to yield any reasonable information.

Conclusions

This article provides an experimental validation of the neural-network parameter-function technique for modeling dynamic systems. An attempt was made to model the protein production process using fundamental models that are simple enough to be used for optimization studies. These models were difficult to develop and were limited by the form of the function postulated for the various rate terms. The neural-network-based model consistently modeled the data better, and the optimum resulting from the optimization was also

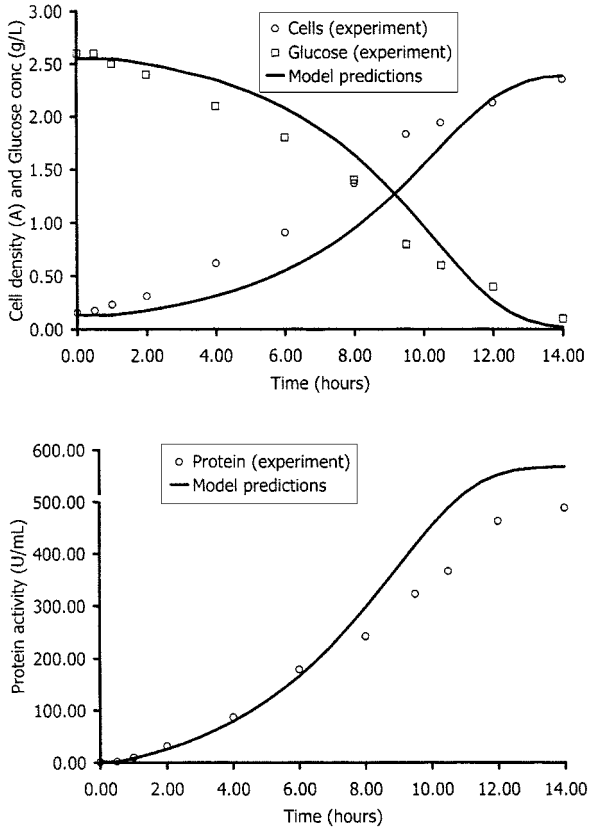


Figure 7b. Variable-inducer model predicted dynamics for the optimum $I = 0.2$ g/L and $t_{ind} = 1.0$ h vs. experimental data.

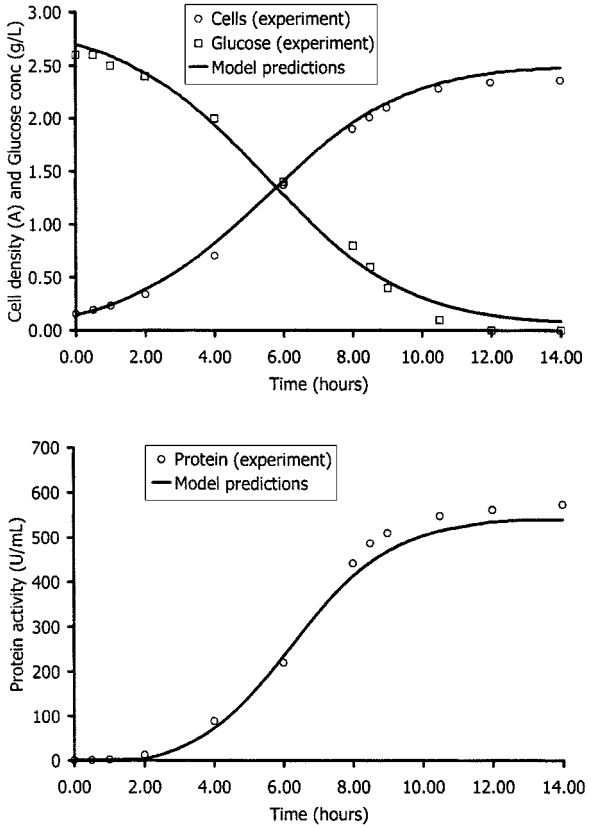


Figure 7c. Variable inducer model predicted dynamics for the optimum $I = 0.2$ g/L and $t_{ind} = 2$ h vs. experimental data.

more accurate than from the fundamental models. The incorporation of conservation principles in the neural-network model results in a framework wherein the strengths of neural networks (namely, function approximation abilities) are being exploited. The method was able to model and optimize a difficult dynamic system using real-world noisy data from a complex biological process.

Acknowledgment

The authors acknowledge the Colorado Institute for Research in Biotechnology for financial support.

Notation

c_2, c_2, c_3 = parameters in logistic function
$\tilde{G}$ = dimensionless glucose concentration
$\tilde{I}$ = dimensionless inducer concentration
J = performance index (productivity) to be maximized, U/(mL · h)
$\tilde{P}$ = dimensionless protein activity
$\tilde{t}$ = dimensionless time
t_n = time normalization factor, h
$\tilde{R}_f$ = normalized specific foreign-protein production rate
$\tilde{X}$ = dimensionless cell density
X_n = cell-density normalization factor, A
$\tilde{Y}$ = dimensionless yield coefficient
$\tilde{\mu}$ = dimensionless growth rate

Literature Cited

Bailey, J. E., and D. F. Ollis, *Biochemical Engineering Fundamentals*, McGraw-Hill, New York (1986).

Bentley, W. E., and D. S. Kompala, "A Novel Structured Kinetic Modeling Approach for the Analysis of Plasmid Instability in Recombinant Bacteria Cultures," *Biotechnol. Bioeng.*, **33**, 49 (1989).

Geman, S., E. Bienenstock and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Comput.*, **4**, 1 (1992).

Georgiou, G., "Optimizing the Production of Recombinant Proteins in Microorganisms," *AIChE J.*, **34**, 1233 (1988).

Glick, B. R., and G. K. Whitney, "Factors Affecting the Expression of Foreign Proteins in *Escherichia coli*," *J. Ind. Microbiol.*, **1**, 277 (1987).

Gold, L., and G. D. Stromo, "High Level Translation Initiation," *Methods Enzymol.*, **185**, 89 (1990).

Lee, J., and W. F. Ramirez, "Mathematical Modeling of Induced Foreign Protein Production by Recombinant Bacteria," *Biotechnol. Bioeng.*, **39**, 635 (1992).

Moody, J. E., "The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems," *NIPS 4*, 847 (1992).

Peretti, S. W., and J. E. Bailey, "Mechanistically Detailed Model of Cellular Metabolism for Glucose-Limited Growth of *Escherichia coli* B/r-A.," *Biotechnol. Bioeng.*, **28**, 1672 (1986).

Putnam, S. L., and A. L. Koch, "Complications in the Simplest Cellular Enzyme Assay: Lysis of *Escherichia coli* for the Assay of β-Galactosidase," *Anal. Biochem.*, **63**, 350 (1975).

Schenker, B., and M. Agarwal, "Cross-Validated Structure Selection for Neural Networks," *Comput. Chem. Eng.*, **20**(2), 175 (1996).

Shu, J., and M. L. Shuler, "A Mathematical Model for the Growth of a Single Cell of *E. coli* on a Glucose/Glutamine/Ammonium Medium," *Biotechnol. Bioeng.*, **33**, 1117 (1989).

Tholudur, A., and W. F. Ramirez, "Optimization of Fed-Batch Bioreactors Using Neural Network Parameter Function Models," *Biotechnol. Prog.*, **12**, 302 (1996).

Williams, F. M., "A Model of Cell Growth Dynamics," *J. Theor. Biol.*, **15**, 190 (1967).

Zabriskie, D. W., and E. J. Arcuri, "Factors Influencing Productivity of Fermentations Employing Recombinant Microorganisms," *Enzyme Microb. Technol.*, **8**, 706 (1986).

Zullinger, E. M., R. E. Ricklefs, K. H. Redford and G. M. Mace, "Fitting Sigmoidal Equations to Mammalian Growth Curves," *J. Mammal.*, **65**(4), 607 (1984).

Appendix

The algorithm used to ensure good generalization properties of the neural network parameter functions is presented below.

NTrainingPoints = 72% of the total number of data points
NValidationPoints = 18% of the total number of data points
NTestingPoints = 10% of the total number of data points
NTrials = 5 (Number of starting points for each network)
StartNHid = 20 (Search over the range of StartNHid to EndNHid)
EndNHid = 35
NEpochs = 1000 (Number of epochs to train)

```
for Trial = 1 to NTrials,
  for NHid = StartNHid to EndNHid,
    Randomly choose training set and validation set
    Initialize neural network weights to random values
    Train the neural network for a fixed number of epochs
    Generate the validation error
    if this is lowest validation error,
      Save the neural network
    end if
  end for
end for
```

Check generalization performance by generating the testing error.

Manuscript received Apr. 22, 1998, and revision received Apr. 30, 1999.