# A Hybrid Neural Network-First Principles Approach to Process Modeling

**Dimitris C. Psichogios and Lyle H. Ungar**
Dept. of Chemical Engineering, University of Pennsylvania, Philadelphia, PA 19104

*A hybrid neural network-first principles modeling scheme is developed and used to model a fedbatch bioreactor. The hybrid model combines a partial first principles model, which incorporates the available prior knowledge about the process being modeled, with a neural network which serves as an estimator of unmeasured process parameters that are difficult to model from first principles. This hybrid model has better properties than standard "black-box" neural network models in that it is able to interpolate and extrapolate much more accurately, is easier to analyze and interpret, and requires significantly fewer training examples. Two alternative state and parameter estimation strategies, extended Kalman filtering and NLP optimization, are also considered. When no a priori known model of the unobserved process parameters is available, the hybrid network model gives better estimates of the parameters, when compared to these methods. By providing a model of these unmeasured parameters, the hybrid network can also make predictions and hence can be used for process optimization. These results apply both when full and partial state measurements are available, but in the latter case a state reconstruction method must be used for the first principles component of the hybrid model.*

## Introduction

The term "artificial neural networks" is a generic description for a wide class of connectionist representations inspired by the models for brain activity. The most common task of these models is to perform a mapping from an input space to an output space. A typical multilayered feedforward neural network (Rumelhart et al., 1986) is shown in Figure 1. It consists of massively interconnected simple processing elements ("neurons" or "nodes") arranged in a layered structure, where the strength of each connection is given by an assigned weight; these weights are the internal parameters of the network. The input neurons are connected to the output neurons through layers of hidden nodes. Each neuron receives information in the form of inputs from other neurons or the world and processes it through some—typically nonlinear—function (the "activation" function); in this way the network can perform a nonlinear mapping. It has been shown that, under some mild assumptions, such networks, if sufficiently large, can approximate any nonlinear continuous function arbitrarily accurately (Stinchcombe and White, 1989).

These connectionist models have the ability to "learn" the frequently complex dynamic behavior of a physical system. Learning is the process where the network approximates the function mapping from system inputs to outputs, given a set of observations of its inputs and corresponding outputs. This is done by adjusting the network's internal parameters, typically in such a way as to minimize the squared error between the network's outputs and the desired outputs. One such method is the error back-propagation algorithm (Werbos, 1974; Rumelhart et al., 1986), which is essentially a first-order gradient descent method. The ability to approximate unknown functions through presentation of their instances makes neural networks a useful and potentially powerful tool for modeling in engineering applications.

Neural networks have typically been used as "black-box" tools, that is, no prior knowledge about the process was assumed; the goal was to develop a process model based only on observations of its input-output behavior. Modeling without using *a priori* knowledge has often proved successful (Bhat and McAvoy, 1990; Psichogios and Ungar, 1991; Willis et al., 1991) and is the only possible method when no process knowl-

Correspondence concerning this article should be addressed to L. H. Ungar.

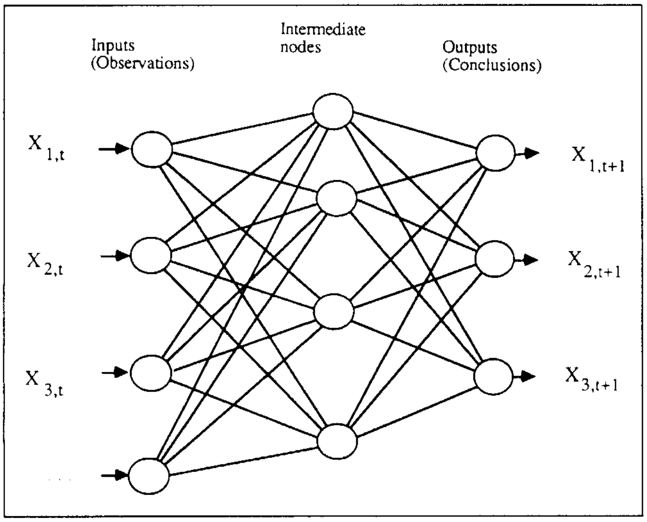

**Figure 1. Multilayered feedforward neural network.**

edge is available. The ability of neural networks to learn nonparametric (structure-free) approximations to arbitrary functions is their strength, but it is also a weakness. A typical neural network involves hundreds of internal parameters, which can lead to "overfitting"—fitting of the noise as well as the underlying function—and poor generalization. Furthermore, interpretation of such models is difficult (Mavrovouniotis, 1992).

As a result, there has been an increasing interest in developing modeling methods that address these problems. Since redundancy (excess degrees of freedom) may result in poor models, one approach has been to decrease the redundancy of the neural network model by developing algorithms that "prune" the weights that have no significant effect on the network's performance (McAvoy and Bhat, 1990; Karnin, 1990; Mozer and Smolensky, 1989). These methods either penalize model complexity or examine the sensitivity of the prediction error to the network's weights, and eliminate these weights (connections) that least affect the fit. However, they do not address the issue of lack of internal model structure and do not use prior knowledge about the process being modeled.

A different approach has focused on imposing internal structure in the neural network model, typically by using some prior knowledge about the process. The common feature of methods that follow this approach is that clearly identifiable different parts of the resulting network model perform different tasks, and it is this interpretation that we give to the term internal structure in the remainder of the article. One possibility is to create a structured network that combines a known linear model with a nonlinear neural network (Haesloop and Holt, 1990). The basic idea behind this technique is that the nonlinear part of the network will model the process nonlinearities, thus enabling the complete model to capture more complex dynamic behavior than the linear part of the network alone. An alternative is to construct a neural network model which can be considered as a hierarchical, sparsely connected, network of smaller subnetworks that perform some local calculation. The task that each of these smaller networks is assigned to perform, as well as the connectivity among them, is based on empirical guidelines and analysis of the system's behavior (Mavrovouniotis, 1992).

We believe that it is advantageous to *a priori* structure the neural network models; in machine learning terminology, this is characterized as imposing "inductive bias" on the final model. In this article we follow a variation of this approach and develop a modeling strategy that combines first principles knowledge, in the form of equations such as mass and energy balances, with neural networks as nonparametric estimators of important process parameters. This approach provides hybrid models with internal structure, where each part of the final model performs a different task. These clearly identifiable parts are the process parameter estimator (neural net) and the partial (first principles) model. The partial model provides a better starting point than "black-box" neural networks and, at the same time, allows for both structural and parametric uncertainty. Since neural networks can approximate arbitrary functions for which no *a priori* parameterization is known, they can ideally complement the basic model and account for the uncertainty. The resulting models can be thought of as structured neural networks which contain some known constraints, such as mass and energy balances; alternatively, they can be thought of as equations which contain "process parameters" whose dependence on state variables is modeled by neural networks. Our goal is to develop hybrid neural network process models which are more flexible than classical parameter estimation schemes and which generalize and extrapolate better than classical "black-box" neural networks, in addition to being more reliable and easier to interpret.

We evaluate this hybrid neural network modeling scheme by comparing its prediction accuracy with standard neural networks. We also discuss and compare the performance of other estimation methods, such as extended Kalman filtering and nonlinear programming (NLP) techniques, under the assumption that proper parameterization of the process parameters is not *a priori* available. Both Kalman filtering and NLP schemes are used to directly estimate the unknown parameters of the known first principles model, either in a stochastic problem formulation (Kalman filter) or in a least-squares minimization approach (NLP estimation).

This article comprises three parts: first, we compare the standard ("black-box") and hybrid neural network modeling methods; secondly, we compare the hybrid modeling method to NLP estimation and Kalman filtering; and thirdly, we extend the concept of hybrid neural network modeling to the partial state information case. All competing methods were tested on the modeling of a fedbatch bioreactor. The bioreactor is discussed as well as the challenges and inherent difficulties involved in modeling such systems. Then, the hybrid neural network and the standard neural network modeling approaches are explained, as well as the generation of a training data set and the training procedures for both methods. Subsequently presented are the method of parameter and state estimation through nonlinear optimization, and extended Kalman filtering combined with least-squares parameter estimation whose results are compared with our approach. The assumption of full state accessibility is relaxed, and a nonlinear exponential observer is developed and incorporated in the hybrid neural network model, and results are discussed. Afterward, the application of nonlinear optimization modeling and extended Kalman filtering to the incomplete measurement case are presented, followed by an application of the hybrid model to process optimization, specifically to the calculation of optimal feed policies for fedbatch reactors. Further insight on the estimation methods considered in this article and a summary of our results is given in the final section.

## Modeling Problem: Fedbatch Bioreactor

Biological reactors exhibit a wide range of dynamic behaviors and offer many challenges to modeling, as a result of the presence of living organisms (cells) whose growth rate is described by complex kinetic expressions. We will illustrate the hybrid modeling method on the identification of such a system.

Consider the dynamic system which can be described by the following general representation:

$$\frac{dx}{dt}=f(x, u, p) \tag{1}$$

$$p=g(x, u) \tag{2}$$

where $x$ denotes the state vector of the system, $u$ the control vector and $p$ a vector of process parameters. These parameters $p$ essentially represent the process kinetics and are related to the system variables through the set of equations $g(\ )$. The functionality that relates the process parameters to state variables and control variables, such as the reactor pH and temperature (Rivera and Karim, 1990), is difficult to derive from first principles reasoning and typically unknown; however, it is the presence of this complex unknown functionality that makes biological reactions highly nonlinear systems. In other systems, the parameters $p$ might be reaction kinetics or viscosities which vary in complex ways with temperature and chemical composition.

Bioreactors operating in a fedbatch (nonstationary) mode can achieve high product concentrations (Gostomski et al., 1990) and are quite difficult to model, since their operation involves microbial growth under constantly changing conditions. Nevertheless, knowledge of process parameters (such as growth rate kinetics) under a wide range of operating conditions is very important in efficiently designing optimal reactor operation policies.

A fedbatch stirred bioreactor can be described by the following equations (Dochain and Bastin, 1990):

$$\frac{dX}{dt}=\mu(t)\cdot X(t)-\frac{F(t)}{V(t)}\cdot X(t) \tag{3}$$

$$\frac{dS}{dt}=-k_1\cdot\mu(t)\cdot X(t)+\frac{F(t)}{V(t)}\cdot[S_{\text{in}}(t)-S(t)] \tag{4}$$

$$\frac{dV}{dt}=F(t) \tag{5}$$

where $X(t)$ is the biomass concentration and $S(t)$ is the substrate concentration. These mass balances on the reacting species provide a partial model. The kinetics of the process are lumped in the term $\mu(t)$ which accounts for the conversion of substrate to biomass. This term, known as specific growth rate is, as noted by Dochain and Bastin (1990), typically a complex function of the biochemical, biological, and physicochemical variables of the system. As a result, a large number of models has been proposed to describe these kinetics and so the choice of a growth model for a particular fedbatch fermentation process is not at all straightforward. In the following we will assume that the "true" but unknown and unmeasured growth rate is described by the Haldane model:

$$\mu(t)=\frac{\mu^*\cdot S(t)}{K_m+S(t)+\frac{S(t)^2}{K_i}}. \tag{6}$$

This expression will only be used to simulate the "true" process model; for all modeling techniques described in the remainder of the article, the above expression describing the cell growth rate will be completely unknown. Furthermore, the inlet substrate feed concentration $S_{\text{in}}$ will be the manipulated input, and the flow rate $F(t)$ will be held constant.

## Standard and Hybrid Neural Network Models

Neural networks have been successfully used as "black-box" models of dynamic systems and, more specifically, as process variable estimators in bioreactor modeling applications (Lant et al., 1990; Thibault et al., 1990). In these efforts the process was operating in a continuous mode; however, identification of batch processes is much more difficult, since a wide range of operating regimes is involved and less data may be available. This section discusses the advantages of structured neural network modeling and describes the development of both a standard and a hybrid neural network model of the bioreactor system.

As discussed in the previous section, it is quite straightforward to derive an approximate model of the bioreactor (Eqs. 3-5) from simple first principles considerations such as mass balances on the process variables. However, the critical factor in determining the dynamic behavior of the process is the unknown kinetics (growth rate model) of the conversion of substrate to biomass. The central idea of this article, then, is to integrate the available approximate model with a neural network which approximates the unknown kinetics, in order to form a combined model structure which can be characterized as a hybrid (or structured) neural network process model.

This approach offers significant advantages over a "black-box" neural network modeling methodology. The hybrid neural network model has internal structure which clearly determines the interaction among process variables and process parameters, and as a result is easier to analyze than standard neural networks. The first principles partial model specifies process variable interactions from physical considerations; the neural network complements this model by estimating unmeasured process parameters in such a way as to satisfy the first principles constraints; nonparametric estimation is needed since no knowledge is available about these parameters. Such structured models are expected to perform better than "black-box" neural network models in process identification tasks, since generalization and extrapolation are confined only to the uncertain parts of the process while the basic model is always consistent with first principles and does not allow aphysical variable interactions.

Furthermore, in data reconciliation and adaptive modeling and control it is very important to be able to correctly identify

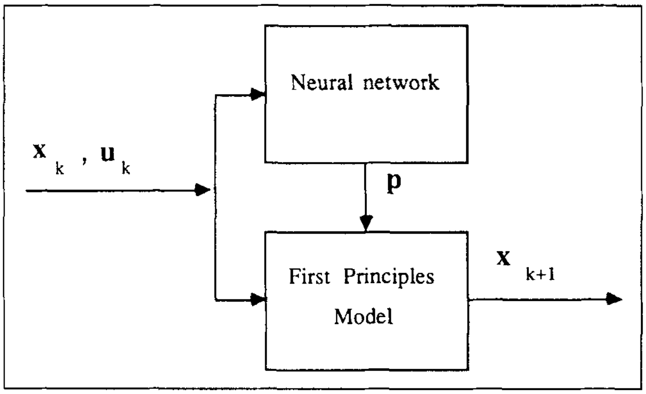

**Figure 2. Hybrid (structured) neural network model; the neural network component estimates the process parameters *p*, which are used as input to the first principles model.**

which part (or parts) of the process model are responsible for erroneous predictions and thus need to be updated. Traditional neural network process models can be adapted as new data become available, but the generality of such an adapted model is questionable (Hernandez and Arkun, 1990) because all of the model's internal parameters are updated since all are considered partially responsible for the error. In contrast, the internal structure of a hybrid neural network model clearly identifies the contribution of each part of the model to its predictions. As a result, the number of potential error sources can be drastically reduced and the adaptation can be more focused.

A schematic representation of the hybrid neural network model is shown in Figure 2. The neural network component receives as inputs the process variables and provides an estimate of the current parameter values, in this case the cell growth rate. The network's output serves as an input to the first principles component, which produces as output the values of the process variables at the end of each sampling time. The combination of these two building blocks yields a complete hybrid neural network model of the bioreaction system.

For the standard neural network modeling approach, development of a process model for the bioreactor is straightforward. Given as inputs observations of the process variables (state) and the manipulated inputs, the neural network model predicts the state of the system at the next sampling instant. Since a set of target outputs is available for every set of inputs presented to the neural network model, a supervised training method can be used to calculate the error signal used to change the model's internal parameters (weights).

However, for the hybrid neural network model target outputs are not directly available, as the cell growth rate is not measured. In this case, the known partial process model can be used to calculate a suitable error signal that can be used to update the network's weights. The observed error between the structured model's predictions and the actual state variable measurements can be "back-propagated" through the known set of equations, essentially by using the partial model's Jacobian, and translated into an error signal for the neural network component. The intuition behind this is that the process parameters should be changed proportionally to their effect on the state variable predictions, multiplied by the observed error in the state predictions.

The generation of a training data set and details of the training procedure for both structured and standard neural network models will be addressed in the following section.

## Training of Standard and Hybrid Neural Network Models

A standard neural network model was developed which, given as inputs observations of the state and manipulated variables, predicted the state of the system at the next sampling time. The state variables, particularly the biomass concentration, undergo changes of over an order of magnitude. This large variation can cause problems when using neural networks, so we defined dimensionless biomass concentration $\chi$ and dimensionless substrate concentration $\sigma$ as:

$$\chi=\frac{k_1\cdot X}{S_{\text{avg}}} \tag{7}$$

$$\sigma=\frac{S_{\text{avg}}-S}{S_{\text{avg}}} \tag{8}$$

where $S_{\text{avg}}$ is the average value of the substrate feed concentration $S_{\text{in}}$. Note that the scaling is linear, and should therefore have no effect on a squared error criterion. The inputs to the neural network were the natural logarithm of the biomass concentration $X$ and the substrate concentration $S$, and a scaled value of the manipulated variable $S_{\text{in}}$. The desired network's outputs were the dimensionless biomass concentration and substrate concentration $\chi$ and $\sigma$ respectively. A sigmoid with output ranging from $(-1, +1)$ as given by

$$o_k=\frac{1-e^{-(\Sigma_j w_{jk}\cdot o_j+b_k)}}{1+e^{-(\Sigma_j w_{jk}\cdot o_j+b_k)}} \tag{9}$$

was chosen as activation function. Here, $o_k$ represents the output of neuron $k$, $w_{jk}$ the weight from neuron $j$ to neuron $k$ which multiplies neurons' $j$ output, and $b_k$ the bias of neuron $k$. The training method was the error back-propagation algorithm as described in Rumelhart et al. (1986): A set of inputs was presented to the network, and a set of the network's outputs was obtained by propagating these inputs through the layers of the network (shown in Figure 1). An error signal was obtained by comparing the network's outputs with the actual process outputs that corresponded to this set of inputs, and this error signal was used to change the network's weights. The errors for each input-output example were accumulated and the weights were updated after each complete presentation of the training data set to the neural network. To avoid overfitting of the training data, at frequent intervals during the training session the network's weights were frozen and the mean square prediction error, on a separate testing data set, was calculated. Training was stopped when it was determined that the network's prediction accuracy would deteriorate upon continued training.

For the hybrid neural network model, the squared prediction error over both process variables (biomass and substrate concentration) and for all training patterns $N$ was minimized as with the standard neural network model:

$$\mathrm{MSE}=\frac{1}{N}\sum_{i}^{N}[(\chi_i-\chi_i')^2+(\sigma_i-\sigma_i')^2] \qquad (10)$$

The neural network's output (cell growth rate $\mu$) does not appear explicitly in the above expression. However, if it is considered constant for each sampling instant, the gradient of the structured model's output with respect to this internal parameter can be calculated through integration of the sensitivity equations (Caracotsios and Stewart, 1985)

$$\frac{dG_1}{dt}=\left(\mu(t)-\frac{F(t)}{V(t)}\right)\cdot G_1(t)+X(t) \qquad (11)$$

$$\frac{dG_2}{dt}=-k_1\cdot\mu(t)\cdot G_1(t)-\frac{F(t)}{V(t)}\cdot G_2(t)-k_1\cdot X(t) \qquad (12)$$

$$G_1(t)=\frac{\partial X(t)}{\partial\mu} \quad G_2(t)=\frac{\partial S(t)}{\partial\mu} \qquad (13)$$

with initial conditions $G_1(t)=0$, $G_2(t)=0$. Thus the gradient of the structured model's output with respect to the network's output can be calculated through use of Eqs. 11-13 and, as a result, the gradient of the performance measure (Eq. 10) with respect to the network's output is readily available. This gradient information will generate an error signal that is used to update the network's weights.

Both state variables (biomass and substrate concentration) were used as inputs to the neural network model component of Figure 2. Obviously, since the "true" kinetics only depend on the substrate concentration (Eq. 6), the biomass concentration input is merely a noisy input to the network; however, it is interesting to examine the behavior of the hybrid neural network model under these conditions. The inputs were scaled in the same way as for the standard neural network model (natural logarithm of biomass and substrate concentration taken), and the network's output was an estimate of the growth rate for the current sampling time. This output was scaled as $\mu = 2\cdot\mu_{\max}\cdot\tilde{\mu}$, where $\tilde{\mu}$ was the network's output and $\mu_{\max}$ a constant, and was used as input to the first principles model component (Eqs. 3-5) in order to predict the system's state for the next sampling time. The outputs of this hybrid model were dedimensionalized according to Eqs. 7-8 to allow a direct comparison with the standard "black-box" neural network model.

The hybrid neural network model was trained as follows. A set of inputs was presented to the model, namely the current values of the biomass ($X$), substrate ($S$) and substrate feed ($S_{\mathrm{in}}$). Scaled values of $X$ and $S$ were propagated through the network part of the structured model; its output was an estimate of the growth rate $\mu$, which along with $X$, $S$ and $S_{\mathrm{in}}$ were "propagated" through the first principles part of the hybrid model (Eqs. 3-5) to obtain an estimate of the process variables for the next sampling time. At the same time, the set of the sensitivity equations was integrated, and an error signal (ES) was calculated according to the following relationship:

$$ES_i=\frac{(\chi_i-\chi_i')\cdot g_{i,1}+(\sigma_i-\sigma_i')\cdot g_{i,2}}{\sqrt{[(g_{i,1})^2+(g_{i,2})^2]}} \qquad (14)$$

where

$$g_{i,1}=\frac{2\cdot\mu_{\max}\cdot k_1}{S_{\mathrm{avg}}}\cdot G_{i,1} \qquad (15)$$

$$g_{i,2}=\frac{2\cdot\mu_{\max}\cdot k_1}{S_{\mathrm{avg}}}\cdot(-G_{i,2}) \qquad (16)$$

that is, $g$ represents a dimensionless gradient and subscript $i$ stands for the i-th input-output pattern in the training data set. The error signals $\mathrm{ES}_i$ for the neural network output were summed over all input-output examples and the weights were updated using back-propagation after a complete presentation of the training data set.

Equations 3-6 with parameter values given by Dochain and Bastin (1990) were used to develop a training data set. The substrate feed inlet concentration was randomly perturbed within 50% of a nominal value of 60 g/lt, following a uniform distribution. The initial state of the process was also chosen in such a way as to explore the state space as much as possible. For each of the two state variables (biomass concentration $X$ and substrate $S$) three levels of low (=0.1 g/lt), medium (=0.5 g/lt), and high (=0.9 g/lt) concentrations were considered, and the initial state of the process comprised all possible combinations of these initial conditions. The initial reactor volume of the reactor was set equal to a nominal value of 10lt in all simulated batch runs. Data were sampled every 0.2 hours, and the batch policy consisted of a feeding period of 15 hours and a subsequent "quenching" period of 5 hours, where the substrate feed $S_{\mathrm{in}}$ and flow rate $F$ were set equal to zero. A total of nine data sets, each consisting of 100 data points, were created in this way; two additional runs with shorter feeding time (5 hours) and different initial conditions were created, each consisting of 50 data points. All measurements were corrupted with normal white Gaussian noise N(0, 0.01) added to the dimensional state variables.

The presence of noise in the measurements of process variables $X$ and $S$ corrupts the gradient information obtained through integration of the sensitivity equations with colored noise. A practical advantage of the batch mode of learning, implemented in the training of the hybrid neural network model, is that it provides some degree of noise smoothing: Misleading individual error signals are lumped together with all other error signals providing a total error signal that more closely represents the "true" gradient.

## Comparison of Hybrid and Standard Neural Networks

An important consideration in neural network modeling is how the size of the available training data set affects the accuracy of the learned model. For sufficiently large data sets, a standard neural network should perform arbitrarily well in approximating the dynamic system. When the training data set size is small then the state space is not sampled sufficiently densely, and a traditional neural network relies heavily on interpolation to approximate the dynamic system. Furthermore, a standard neural network relies only on the data to infer the complete process model. On the contrary, a hybrid neural network already contains a partial model, which is also used to reduce the error signal to a subspace that can be ex-

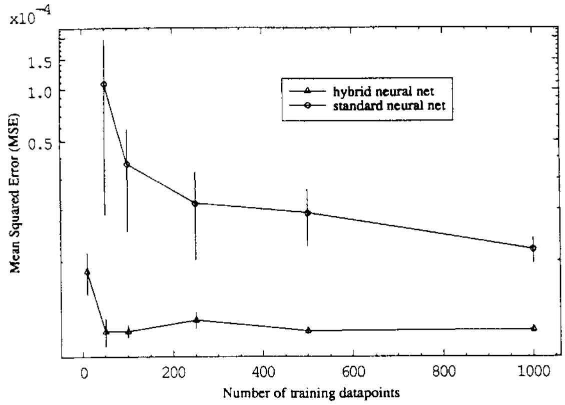

**Figure 3. Logarithm of Mean Squared Error (MSE) on both state variables (Eq. 10) vs. number of available training examples, for the hybrid and standard neural networks.**

Vertical bars indicate standard deviation, calculated from 10 training sessions.

plored with fewer training examples. In other words, the network component of the hybrid model relies on the data to approximate only part of the model (the unmeasured parameters).

To investigate this argument we compared the approximation accuracy of the standard and hybrid neural networks as a function of the training data set size. Five cases were considered, with training data set sizes of 50, 100, 250, 500 and 1,000 data points respectively. For all but the last case the corresponding number of data points was randomly drawn from the 1,000 training examples available (see the previous section). The networks were trained as described above, with each of the five data sets randomly partitioned (with a 70%:30% ratio) into two smaller subsets used for training and testing respectively. Figure 3 shows that the mean squared prediction error of the hybrid network on both state variables is an order of magnitude lower than that of the standard network model and is relatively unaffected by the training data set size. However, the hybrid network's prediction error increases as the data set becomes very small (see Figure 3). The performance of the standard network, as expected, improves significantly as the size of the data set increases and should asymptotically approach the hybrid network's prediction error as the data set becomes very large.

The hybrid neural network model should also extrapolate better than the standard network model, as a result of the partial first principles model it contains. Poor extrapolation has been the plague of traditional neural networks (Leonard et al., 1991), and is even more important for processes such as batch reactors which operate in a nonstationary mode. The superior extrapolation of the hybrid neural network compared to the standard neural network is shown in Figure 4, where both were required to predict the state of the system when it operated in a state space regime that was not sampled by the training data set. The standard network model fails to extrapolate correctly, whereas the hybrid model gives quite accurate predictions. The hybrid model also interpolates better, as illustrated in Figure 5.

In conclusion, the hybrid network model appears to reject the noisy input (biomass concentration, which does not affect the growth kinetics), as well as the noise in the process variables measurements, and gives very good predictions of the system's state. Furthermore, these predictions are achievable even with only a few data points available for training.

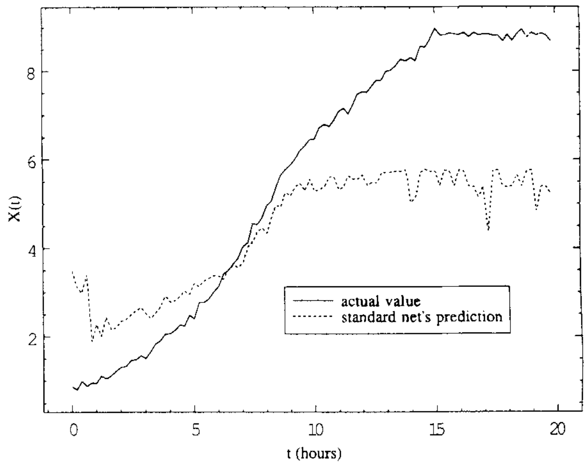

**Figure 4a. Biomass concentration (*X*) prediction vs. time for the standard network process model.**

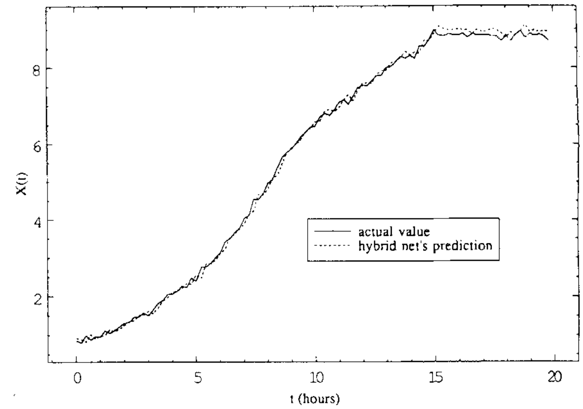

**Figure 4b. Biomass concentration (*X*) prediction vs. time for the hybrid network process model.**

## State and Parameter Estimation Using an NLP Technique

Hybrid neural networks offer a method of estimating unobserved process parameters and process variables, as demonstrated in the previous sections; however, a variety of other methods has been extensively applied to the problem of state and parameter estimation. In recent years, optimization methods have come to be increasingly used in such problems (Bequette and Sistu, 1990; Brengel and Seider, 1989; Jang et al., 1986). These methods use a process model and typically assume some parameterization (model with fixed coefficients) of the process parameters. The unknown fixed coefficients are treated as decision variables that can be estimated in such a way as to minimize prediction error. However, as discussed in the introduction, proper *a priori* parameterization is not always possible.

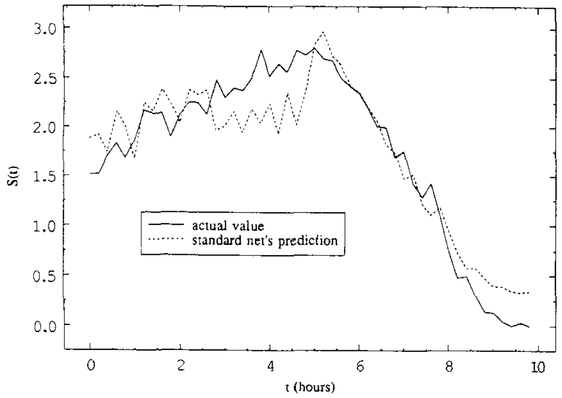

**Figure 5a. Substrate concentration (*S*) prediction vs. time for the standard neural network process model.**

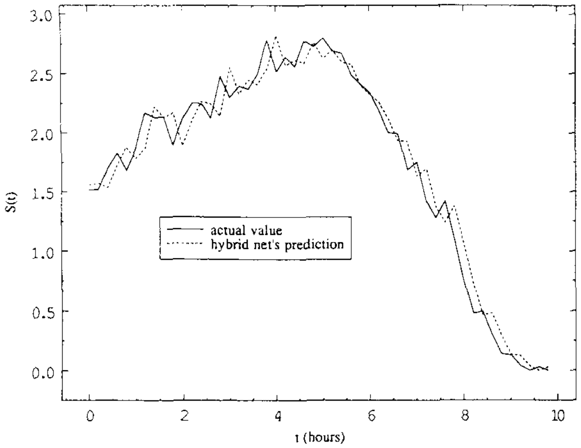

**Figure 5b. Substrate concentration (*S*) prediction vs. time for the hybrid neural network process model.**

We investigated the performance of the NLP estimation approach on the bioreactor problem under the assumption that the functional relationship describing the cell growth rate is completely unknown. Thus the (discretized) growth rate was treated as a time-varying unknown parameter to be directly estimated; the decision variables represented the estimates of the growth rate for the corresponding sampling times. A gradient-based optimization strategy, Successive Quadratic Programming (Cuthrell and Biegler, 1985), was used to solve the problem. The gradient of the objective function—squared prediction error—can be analytically calculated through integration of the sensitivity equations, as described in the section on comparing hybrid and standard neural networks. However, this requires substantial computation, and a simpler alternative of using numerical derivatives was used.

Under this formulation, this parameter estimation method essentially performs local fitting of the unknown cell growth rate. As a result, the presence of noise in the process variables measurements degrades the estimator's performance and produces erroneous parameter estimate values. The parameter estimates were also significantly affected by the magnitude of the upper bound, indicating that the method is essentially unstable in estimating the time-varying growth rate under this formulation. The apparent conclusion is that the NLP approach is not suitable for directly estimating nonconstant process parameters of dynamic systems such as the fedbatch bioreactor considered here. An alternate approach in this case would be to experiment with different parameterized models for the unmeasured process parameter, in order to determine which is most suitable for the specific problem at hand. The advantage of the hybrid neural network model is that, by providing a very general parameterization of the process parameter (neural network component), it helps avoid this experimentation.

## Kalman Filtering

### *Extended Kalman filtering for state estimation*

When noise in the process measurements is present, a widely used modeling method for *stochastic* systems, that is, systems whose state is perturbed by random process noise, and whose observations are corrupt with random measurement noise, is the Kalman filter. The discrete linear Kalman filter provides optimal (in the sense of maximum likelihood) estimates of a linear dynamic system's states in the presence of additive white Gaussian noise (Anderson and Moore, 1981). Extended Kalman filtering applies this method to nonlinear time-varying systems. For a nonlinear dynamic system with the discrete time representation:

$$x_{k+1}=f_k(x_k,\ p_k,\ u_k)+w_k \tag{17}$$

$$z_k=h_k(x_k)+v_k \tag{18}$$

where the noise vectors $w_k$ and $v_k$ represent white Gaussian state and measurement noise processes, with

$$E[v_k\cdot v_k^T]=R_k,\ E[w_k\cdot w_k^T]=Q_k, \tag{19}$$

then the extended Kalman filter formulation (Wells, 1971) is given by the following set of equations:

*Prediction step*:

$$\hat{x}_{k/k-1}=f_{k-1}(\hat{x}_{k-1/k-1},\ p_{k-1},\ u_{k-1}) \tag{20}$$

$$\hat{z}_{k/k-1}=h_k(\hat{x}_{k/k-1}) \tag{21}$$

$$P_{k/k-1}=\Phi_{k-1}\cdot P_{k-1/k-1}\cdot\Phi_{k-1}^T+Q_{k-1} \tag{22}$$

*Correction step*:

$$\hat{x}_{k/k}=\hat{x}_{k/k-1}+W_k\cdot(z_k-\hat{z}_{k/k-1}) \tag{23}$$

$$\hat{z}_{k/k}=h_k(\hat{x}_{k/k}) \tag{24}$$

$$P_{k/k}=(I-W_k\cdot H_k)\cdot P_{k/k-1} \tag{25}$$

$$W_k=P_{k/k-1}\cdot H_k^T\cdot(H_k\cdot P_{k/k-1}\cdot H_k^T+R_k)^{-1} \tag{26}$$

where

$$\Phi_{k-1}=\left(\frac{\partial f_{k-1}}{\partial x_{k-1}}\right)_{\hat{x}_{k-1/k-1}} \tag{27}$$

$$H_k = \left(\frac{\partial h_k}{\partial x_k}\right)_{\hat{x}_{k-1/k-1}} \qquad (28)$$

The term $W_k$ in Eq. 26 represents the estimator's gain which essentially specifies how much to weigh new information about the evolution of the system, as obtained through the measurements $z_k$, in updating the current state estimate. The recursive equations are initialized by assigning *a priori* values to the state estimate and the state covariance matrices. The recursive nature of the Kalman filter estimator is an important advantage, since the covariance matrix $P$ and the filter's gain $W$ are updated at each sampling time interval as new data become available.

### *Extended Kalman filtering combined with parameter estimation*

The above development of the extended Kalman filter estimator was based on the assumption that the parameter vector $p$ is known. When this is not the case, then an appropriate parameter estimation method has to be incorporated. Since the microbial growth rate in the fedbatch bioreactor changes with time, a suitable parameter estimation algorithm should have the ability to track the parameter changes as the process evolves. To accomplish this, we implemented a two step estimation scheme as follows:

1) Given an estimate of the unknown process parameter at the current time, use Eqs. 20-26 to update the state estimates

2) Given these state estimates, update the parameter estimate with an appropriate algorithm. A sequential nonlinear least-squares algorithm was used to estimate the parameters; for multiple observations, the algorithm is described by the following set of equations (Goodwin and Sin, 1984):

$$\hat{p}_k = \hat{p}_{k-1} + \Gamma_{k-1}\cdot\Psi_{k-1} \cdot[\Psi^T_{k-1}\cdot\Gamma_{k-1}\cdot\Psi_{k-1}+R_k]^{-1}\cdot[z_k-\hat{z}_{k/k}] \qquad (29)$$

$$\Gamma_{k-1} = \Gamma_{k-2} - \Gamma_{k-2}\cdot\Psi_{k-1} \cdot[\Psi^T_{k-1}\cdot\Gamma_{k-2}\cdot\Psi_{k-1}+R_k]^{-1}\cdot\Psi^T_{k-1}\cdot\Gamma_{k-2} \qquad (30)$$

where

$$\Psi_{k-1} = \left(\frac{\partial \hat{z}_{k/k}}{\partial p}\right)_{\hat{p}_{k-1}} \qquad (31)$$

In order to use the above estimation technique to model the fedbatch bioreactor, the dynamic system has to be represented in a discrete form; equations (Eqs. 3-5) were discretized using Euler's method. The discretization error can be accounted for with a judicious choice of the process state noise $w_k$, as noted by Wells (1971). We did not assume any prior parameterization of the growth rate, which was considered constant within each sampling time and was the parameter $p$ to be directly estimated, given measurements of the process variables: biomass concentration $X$ and substrate concentration $S$. Measurement noise and process state noise were both assumed white Gaussian, with $R_k = \mathrm{N}(0, 0.01)$ and $Q_k = \mathrm{N}(0, 0.001)$ respectively; these matrices were diagonal, and the choice of values reported here gave the best filter performance.

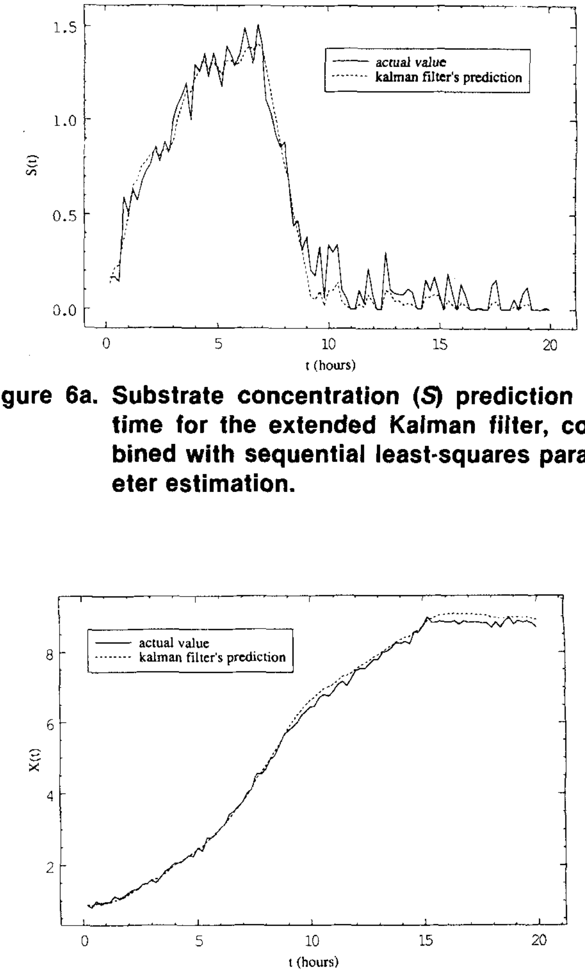

**Figure 6a. Substrate concentration (*S*) prediction vs. time for the extended Kalman filter, combined with sequential least-squares parameter estimation.**

**Figure 6b. Biomass concentration (*X*) prediction vs. time for the extended Kalman filter, combined with sequential least-squares parameter estimation.**

The performance of the extended Kalman filter combined with the sequential least-squares estimation algorithm, on the same simulated batch run used in the section on state and parameter estimation using an NLP technique, is shown in Figure 6. As can be observed, the estimator performs quite well and is able to predict the process state quite accurately. A small error in the prediction of the biomass concentration is observed at the "quenching" period of operation of the bioreactor (after $t = 15$ h), due to an error in the estimation of the growth rate; the "true" growth rate is zero for this time period, since the substrate in the bioreactor has all been consumed. Not surprisingly, the noise in the predictions of the system's state variables is rejected. The performance of a trained hybrid neural network, for the same simulated batch run, is shown in Figure 7. The hybrid neural network gives slightly better predictions in this operating regime since it provides a better estimate of the growth kinetics; however, the state variable predictions are noisy. The unobserved growth rate estimates for both methods is shown in Figure 8. The transient behavior of the growth rate estimate is poor for the Kalman filter; nevertheless, when the process operates such that the growth rate is approximately constant, the estimation accuracy

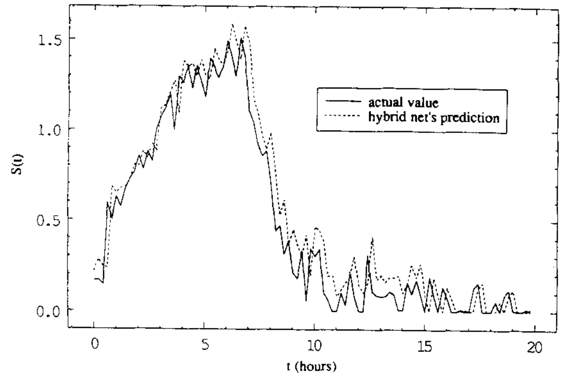

**Figure 7a. Substrate concentration (*S*) prediction vs. time for the hybrid neural network process model.**

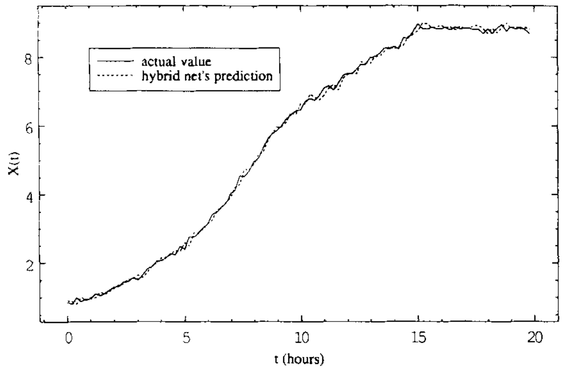

**Figure 7b. Biomass concentration (*X*) prediction vs. time for the hybrid neural network process model.**

is much improved. In contrast, the growth rate prediction of the hybrid model is in general more accurate.

The Kalman filter's performance depends on a number of "tuning" parameters which have to be carefully selected for good performance. The initial state estimate and the initial state covariance affect its response; the values of the measurement noise and process state noise also affect performance. In addition, the estimator Eqs. 20-26 have been derived with the assumption of white Gaussian noise; performance under different and unknown noise statistics may well deteriorate. In contrast, the hybrid neural network model needs no such tuning parameters, makes no assumption about the noise statistics, and is not affected by irrelevant inputs (biomass concentration) to the neural network component of the model. The only cumbersome step in the use of the hybrid model is the training time.

## Incomplete State Measurements

### *Hybrid network model*

In the development of the hybrid neural network model discussed in the section on standard and hybrid neural network models, the availability of the gradient of the state with respect to process parameters (growth rate) is essential. When the sensitivity equations are a function of nonmeasured state variables, as is the case here, gradient information is not directly available. As a result, the derivative of the performance measure (squared prediction error) of the hybrid model with respect to the neural network's (unmeasured) output cannot be analytically calculated.

One way to overcome this problem is to estimate the unmeasured state variables using state observers, which compute estimates of the state of a dynamic system given estimates of the initial state $x_0$ (open-loop observers). When process output measurements are available, then the prediction error can be used as a feedback term, multiplied with an observer gain, to provide better state estimates. This results in a closed-loop observer, which has desirable characteristics (Kravaris and Chung, 1987; Banks, 1981; Bestle and Zeitz, 1983). Following the analysis of Kou et al. (1975), we can construct an exponential closed-loop observer for the fedbatch bioreactor system, as discussed previously. Based on Eqs. 3-5 and the availability of substrate concentration measurements, we have:

$$\nabla f = \begin{bmatrix} \mu(t) - \dfrac{F(t)}{V(t)} & 0 \\ -k_1 \cdot \mu(t) & -\dfrac{F(t)}{V(t)} \end{bmatrix} \tag{32}$$

$$\nabla h = [0 \quad 1] \tag{33}$$

With a choice of the observer's gain matrix $\boldsymbol{B}$ such that

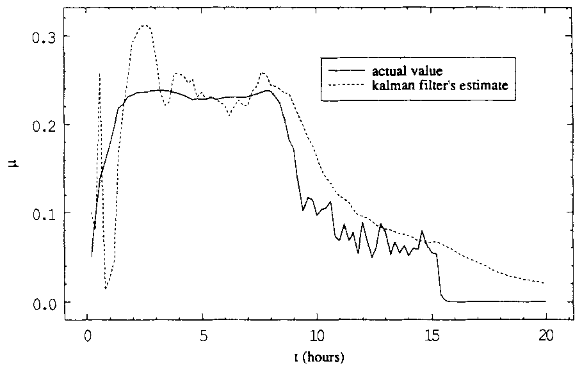

**Figure 8a. Cell growth rate estimate for the extended Kalman filter combined with sequential least-squares estimation.**

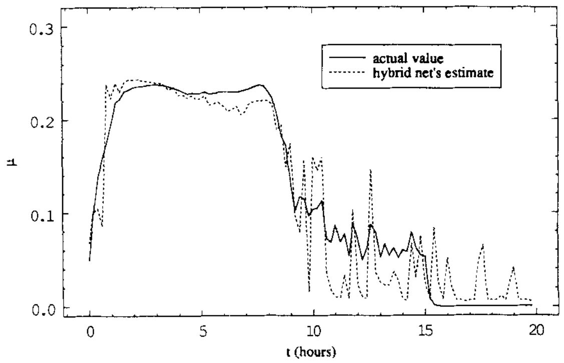

**Figure 8b. Cell growth rate estimate for the hybrid network.**

$$B = \begin{bmatrix} -\mu(t) \\ \mu(t) \end{bmatrix} \tag{34}$$

it can be shown that the matrix

$$\nabla f - B \cdot \nabla h = \begin{bmatrix} \mu(t) - \dfrac{F(t)}{V(t)} & \mu(t) \\ -k_1 \cdot \mu(t) & -\dfrac{F(t)}{V(t)} - \mu(t) \end{bmatrix} \tag{35}$$

is stable and, as a result, the closed-loop observer described by the equations:

$$\frac{d\hat{X}}{dt} = \mu(t) \cdot \hat{X}(t) - \frac{F(t)}{V(t)} \cdot \hat{X}(t) + (-\mu(t)) \cdot [S'(t) - \hat{S}(t)] \tag{36}$$

$$\frac{d\hat{S}}{dt} = -k_1 \cdot \mu(t) \cdot \hat{X}(t) + \frac{F(t)}{V(t)} \cdot [S_{\text{in}}(t) - \hat{S}(t)] + (\mu(t)) \cdot [S'(t) - \hat{S}(t)] \tag{37}$$

$$\hat{X}(0) = X_0, \quad \hat{S}(0) = S_0 \tag{38}$$

is an exponential observer for the fedbatch bioreactor. Equations 36-38 comprise the first principles part of the hybrid neural network model; the other component consists of a neural network that estimates the microbial growth rate $\mu$, given as input only measurements of the substrate concentration. As in the section on the training of standard and hybrid neural network models, we can derive a set of observer sensitivity equations from Eqs. 36-37, which provide the gradient of the performance measure with respect to the network's output. The performance measure in this case is the squared prediction error of the substrate concentration only. Thus Eq. 14 has to be modified so that the error signal used to update the network's weights involves only the prediction error on the substrate concentration. The training procedure is otherwise the same as the one discussed earlier.

The prediction§ for the substrate concentration and the cell growth rate of a hybrid neural network model trained using only substrate concentration measurements is shown in Figure 9, for the same simulated batch run as before. Partial state accessibility does not noticeably degrade the hybrid model's ability to estimate the growth rate; this is not true, as will be seen in the following, for extended Kalman filtering. In most of the operating regime the prediction accuracy is quite good, with the possible exception of the regime where the substrate concentration approaches zero. However, it should be emphasized—in defense of the method—that the signal-to-noise ratio for the substrate concentration measurements is very low in this regime. This problem was also apparent in the full state hybrid network model. More importantly, this partial state hybrid network model, like the full state hybrid network, gives estimates for the unmeasured process variables (biomass concentration). In contrast, standard neural networks can only predict the measured process output (ARMA type models).

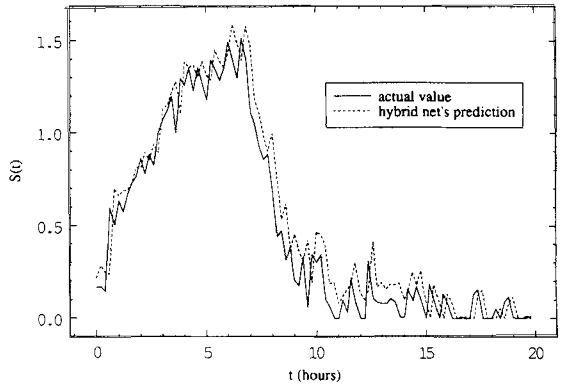

**Figure 9a. Substrate concentration (*S*) prediction vs. time for the hybrid network model using partial state measurements.**

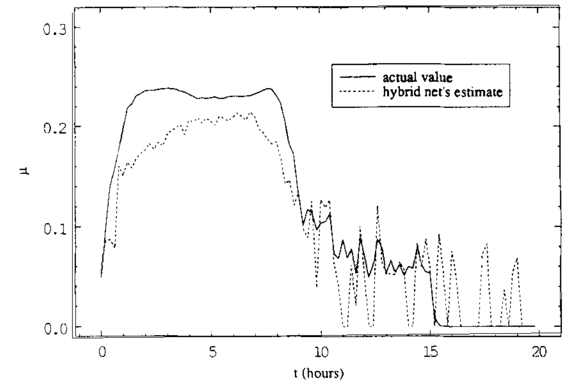

**Figure 9b. Cell growth rate estimate for the hybrid network using partial state measurements.**

### *NLP estimation and extended Kalman filtering*

The NLP optimization method can be applied to the problem of state and process parameter estimation of the fedbatch bioreactor, in the case when only substrate concentration measurements are available, and the basic problem formulation and solution methods remain the same as presented in the section on state and parameter estimation using an NLP technique. However, the objective function involves only the error of the measured output over the process monitoring window. We found the estimator's performance to be similar to the one previously discussed using full state measurements. Furthermore, the observations made in the earlier section about the method's deficiencies when estimating time-varying process parameters, for which properly parameterized models are not available, also apply here.

The extended Kalman filtering method combined with a sequential least-squares parameter estimation technique can also be applied to the bioreactor modeling problem, when only substrate concentration measurements are available. The estimator's formulation, as described in the previous section (Eqs. 20-26), remains the same, with the only difference that the measurement vector $z_k$ is now a scalar. This simplifies the expressions that are used to update the parameter estimate (Eqs. 29-31) since $\Gamma$ and $\Psi$ now represent scalar quantities. The performance of the partial state Kalman filter estimator is shown in Figure 10 for the same batch run as in the previous

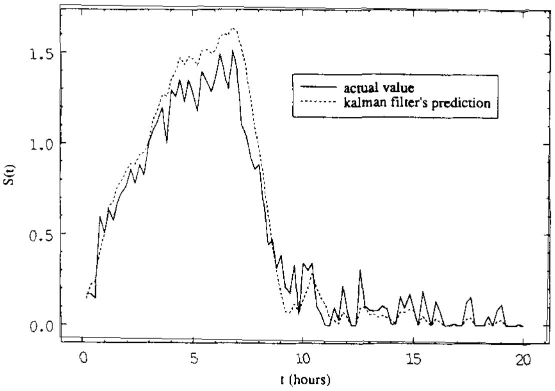

**Figure 10a. Substrate concentration ($S$) prediction vs. time for the extended Kalman filter, combined with sequential least-squares parameter estimation, using partial state measurements.**

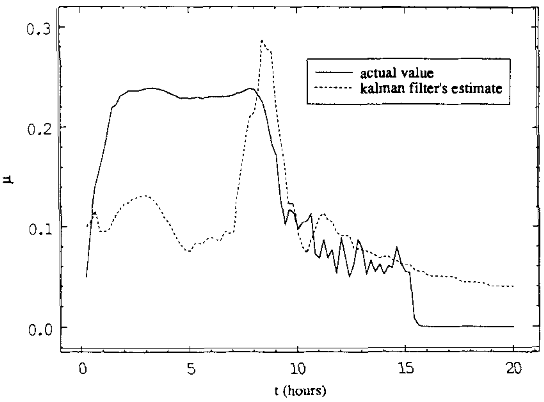

**Figure 10b. Cell growth rate estimate for the extended Kalman filter, combined with sequential least-squares parameter estimation, using partial state measurements.**

section. It is evident that the prediction accuracy decreases when only the substrate concentration is measured; the estimate of the growth rate is inaccurate in most of the operating regime. The hybrid neural network model gives more accurate predictions with partial state measurements than the Kalman filter, particularly for the (unmeasured) biomass concentration.

## Process Operation Scheduling using a Hybrid Neural Network Model

It was emphasized above that an important feature of the hybrid neural network model is that it provides a model for the unobserved growth kinetics. A practical application in showing the usefulness of such a model is the optimization of (fed)batch process operating schedules or, more specifically, determining the substrate feeding policy that maximizes product yield. One way to obtain such an optimal substrate feed policy is to formulate the problem as an optimal control problem and calculate (analytically if the growth rate kinetics are completely known) the substrate profile. An alternative method is to use optimization techniques to maximize a suitable cost function.

We formulated the problem of determining the optimal feed policy for the bioreactor following the latter approach, as:

$$\max_{S_{in}} J = X(T) \cdot V(T) \qquad (39)$$

subject to Eqs. 3-5 and

$$\mu(t) = \pi(X, S) \qquad (40)$$

$$S_{in,j} = S_{in,j+1} = S_{in,j+2} \qquad (41)$$

This formulation implies that the objective function to be maximized is the cell mass in the reactor after the end of some feeding period $T$. Equations 3-5 describe the process model, and $\pi(\ )$ represents the neural network model of the growth rate; together these equations comprise the hybrid network model. Equation 41 simply implies that the substrate feed is held constant for three consecutive time intervals. A simulation for $T = 15$ hours, followed by a subsequent "quenching" period of 5 hours where no substrate is fed in the reactor, was performed. With a sampling time of 0.2 hours, this formulation leads to an optimization problem with 25 decision variables. Upper and lower bounds were imposed on the substrate feed, and numerical derivatives were used to determine the gradient of the objective function; the problem was solved using SQP.

The substrate feed optimal profile, for a batch run with initial conditions of $X = 0.5$ g/lt, $S = 0.1$ g/lt and $V = 10$ lt, is shown in Figure 11; also, for comparison, the optimal profile obtained if the same problem is solved using Eqs. 3-6 (that is, the "true" growth rate model) instead of the hybrid network process model. The substrate feed is initially set at a high value, so that the substrate concentration can be rapidly increased to a value of about lg/lt which is the value that maximizes the cell growth rate. Subsequently, the substrate feed is initially decreased and progressively increased in order to regulate the substrate concentration to this maximum growth rate value. It can be seen that the hybrid model's predictions for the optimal policy are in very good agreement with the predictions using the true model. This suggests that the hybrid neural network model can be used in the design of high product yield

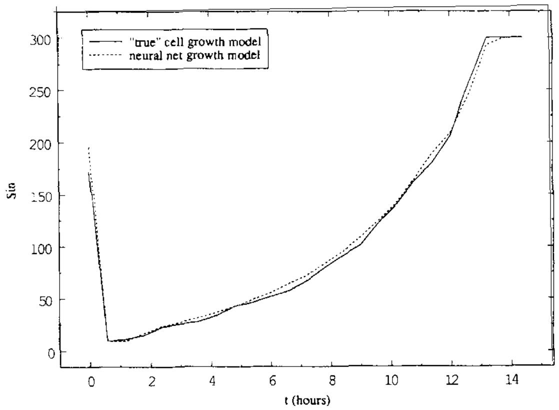

**Figure 11. Comparison of optimal substrate feed policies for the fedbatch bioreactor, as calculated by using the actual process model (Eqs. 3-6) and the hybrid network process model.**

batch runs. Furthermore, with its predictive capability, it can also be used for on-line multistep predictive control when the process is required to follow an optimal feed policy which has been previously calculated off-line, or in gain scheduling control.

## Discussion

A hybrid neural network modeling approach was presented and used to model a fedbatch bioreactor. This hybrid model is comprised of two parts including a partial first principles model, which reflects the *a priori* knowledge, and a neural network component, which serves as a nonparametric approximator of difficult-to-model process parameters. This form of hybrid neural network is useful for modeling processes where a partial model can be derived from simple physical considerations (for example, mass and energy balances), but which also includes terms that are difficult (or even infeasible) to model from first principles. For example, the neural network component of the hybrid model may be used to approximate unknown reaction kinetics, or predict product properties whose correlation with the process variables is difficult to determine from first principles, such as the solution viscosity in polymerization reactors. The bioreactor used to demonstrate this hybrid modeling method only contained one process parameter, but it is in principle straightforward to extend this method when models for multiple parameters are to be learned. We are currently studying such problems.

Such hybrid neural networks have distinct advantages over standard "black-box" neural networks. As was argued above, the hybrid model uses its internal structure to restrict the interactions among process variables to be consistent with physical models. This produces combined models which are more reliable and which generalize and extrapolate more accurately than standard neural networks. Furthermore, interpreting a standard network is difficult, as inferred knowledge of the process is spread among its many internal parameters. In contrast, the nonparametric approximation in the hybrid neural network model is restricted to modeling terms for which *a priori* models are difficult to obtain. Of equal importance, significantly less data are required for training hybrid neural networks. As was argued, the partial model "projects" the error signal to a subspace that is easier to sufficiently explore with a small number of training points. Put differently, use of the partial first principles model reduces the number of functions that the neural network has to choose from to approximate the process parameters. Thus, hybrid network models give far better approximation accuracy—for the same number of data points—than standard neural networks.

The hybrid modeling approach assumes that the partial first principles model has a reasonably correct structure, thereby reducing the identification problem to that of estimating unmeasured process parameters. If the partial model's structure is highly uncertain, or contains a large number of very complex process parameters, then hybrid modeling may not provide a significant advantage over standard "black-box" neural networks. This may also be true when modeling processes operating in a continuous mode, where the behavior of interest is in a small region around a nominal operating point.

It is possible to use a simple parameterized model (for example, linear or quadratic) in place of the neural network and estimate its unknown coefficients by using the available process data. This approach has been widely used in certain applications and works, to the extent that the correct parameterization is chosen for the process parameter model. However, it is often not possible to *a priori* choose a parameterization that can closely approximate the true process parameter model. For example, any linear parameter model will perform poorly when trying to approximate the growth rate kinetics used in this article. Often the true parameter model is quite complex and extensive experimentation is required to get an acceptable approximation. For example, polymerization reactors are typically described by kinetic expressions which are quite complex and involve a large number of adjustable coefficients, and biological reactions often have growth kinetics described by complex nonlinear expressions, with exponential dependencies on the state variables. Use of the hybrid modeling approach in such problems provides a general and versatile parameterization (neural network) which can approximate arbitrarily complex parameter models and can help avoid the lengthy experimentation to determine a properly parameterized model.

Extended Kalman filtering and NLP estimation were considered as alternative methods of estimating unknown process parameters in a known first principles model, under the assumption that *a priori* parameterization of the process parameter model is not possible. The hybrid network model outperformed both methods in estimating the unobserved process parameter, especially when only partial state measurements were available. In the latter case, a hybrid model can be developed by using a state reconstruction scheme, as shown in the section on the hybrid network model. If a sufficiently detailed process parameter model is available so that the unknown coefficients which it contains are almost constant (such as reaction rate activation energies), use of a hybrid neural network is not appropriate and extended Kalman filtering or NLP estimation are more suitable. However, for process parameters which are rapidly time-varying and are not easily described by a parameterized model, the hybrid neural network gives superior performance.

## Notation

$E(Y)$ = expectation of a random variable $Y$
$F$ = inlet flow rate
$I$ = identity matrix
$k_1$ = substrate to cell conversion coefficient (= 1)
$K_m$, $K_i$ = Haldane growth rate model constants (= 10 and 0.1 respectively)
$P$ = state covariance matrix (Kalman filter)
$Q$ = process state noise covariance matrix
$R$ = measurement noise covariance matrix
$S$ = substrate concentration
$S_{\text{avg}}$ = mean value of $S_{\text{in}}$ over the batch run
$S_{\text{in}}$ = inlet feed concentration
$v$ = process measurement noise vector
$V$ = reactor's volume
$w$ = state variable noise vector
$W$ = Kalman filter's gain matrix
$X$ = biomass concentration
$Y'$ = measured value of the variable $Y$
$\hat{Y}$ = predicted value of variable $Y$ (Kalman filter, exponential observer)
$Y_{i/j}$ = value at time $i$ given information up to time $j$ (Kalman filter)
$Y_k$ = value of variable $Y$ at time $k$
$z$ = vector of process measurements

### *Greek letters*

$\Gamma$ = parameter vector's covariance matrix (Kalman filter)
$\mu^*$ = constant (=5)
$\mu_{max}$ = maximum growth rate (=5/21)
$\sigma$ = dimensionless substrate concentration
$\mu(t)$ = cell growth rate
$\chi$ = dimensionless biomass concentration

## Literature Cited

Anderson, B. D. O., and J. B. Moore, *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, NJ (1979).

Banks, S. P., "A Note on Nonlinear Observers," *Int. J. Control*, **34**, 185 (1981).

Bestle, D., and M. Zeitz, "Canonical Form Observer Design for Non-Linear Time-Variable Systems," *Int. J. Control*, **38**, 419 (1983).

Bhat, N., and T. McAvoy, "Use of Neural Nets for Dynamic Modeling and Control of Chemical Processes," *Comp. Chem. Eng.*, **14**, 573 (1990).

Bhat, N., and T. McAvoy, "Information Retrieval From A Stripped Back-propagation Network," *AIChE Meeting* (1990).

Brengel, D. D., and W. D. Seider, "Multistep Nonlinear Predictive Controller," *Ind. Eng. Chem. Res.*, **28**, 1812 (1989).

Caracotsios, M., and W. E. Stewart, "Sensitivity Analysis of Initial Value Problems With Mixed ODE'S and Algebraic Equations," *Comp. Chem. Eng.*, **9**, 359 (1985).

Cuthrell, J. E., and L. T. Biegler, "Improved Infeasible Path Optimization for Sequential Modular Simulators: II. The Optimization Algorithm," *Comp. Chem. Eng.*, **9**, 257 (1985).

Dochain, D., and G. Bastin, "Adaptive Control of Fedbatch Bioreactors," *Chem. Eng. Commun.*, **87**, 67 (1990).

Goodwin, G. C., and K. S. Sin, *Adaptive Filtering Prediction and Control*, Prentice-Hall, Englewood Cliffs, NJ (1984).

Gostomski, P. A., B. W. Bequette, and H. R. Bungay, "Multivariable Control of a Continuous Culture," *AIChE Meeting* (1990).

Haesloop, D., and B. R. Holt, "A Neural Network Structure for System Identification," *Proc. of the American Control Conf.*, 2460 (1990).

Hernandez, E., and Y. Arkun, "Neural Network Modeling and an Extended DMC Algorithm to Control Nonlinear Systems," *Proc. of the American Control Conf.*, 2454 (1990).

Jang, S. S., B. Joseph, and H. Mukai, "Comparison of Two Approaches to On-Line Parameter and State Estimation of Nonlinear Systems," *Ind. Eng. Chem. Process Des. Dev.*, **25**, 809 (1986).

Karnin, E. D., "A Simple Procedure for Pruning Back-Propagation Trained Neural Networks," *IEEE Trans. on Neural Networks*, **1**, 239 (1990).

Kou, S. R., D. L. Elliot, and T. J. Tarn, "Exponential Observers for Nonlinear Dynamic Systems," *Inf. and Control*, **29**, 204 (1975).

Kravaris, C., and C. B. Chung, "Nonlinear State Feedback Synthesis by Global Input/Output Linearization," *AIChE J.*, **33**, 592 (1987).

Lant, P. A., M. J. Willis, G. A. Montague, M. T. Tham, and A. J. Morris, "A Comparison of Adaptive Estimation With Neural Based Techniques for Bioprocess Application," *Proc. of the American Control Conf.*, 2173 (1990).

Leonard, J. A., M. A. Kramer, and L. H. Ungar, "A Neural Network Architecture That Computes Its Own Reliability," *Comp. Chem. Eng.*, submitted (1991).

Mavrovouniotis, M. L., and S. Chang, "Hierarchical Neural Networks for Process Monitoring," *Comp. Chem. Eng.*, **16**(4), 347 (1992).

Mozer, M. C., and P. Smolensky, "Skeletonization: A Technique For Trimming The Fat From A Network via Relevance Assessment," in *Advances in Neural Information Processing*, D. S. Tourestzky, ed., Morgan Kaufmann, San Mateo, CA (1989).

Psichogios, D. C., and L. H. Ungar, "Direct and Indirect Model Based Control Using Artificial Neural Networks," *Ind. Eng. Chem. Res.*, **30**, 2564 (1991).

Rivera, S. L., and M. N. Karim, "Application of Dynamic Programming for Fermentative Ethanol Production by *Zymomonas mobilis*," *Proc. of the American Control Conf.*, 2144 (1990).

Rumelhart, D., G. Hinton, and R. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition: 1. Foundations*, MIT Press (1986).

Sistu, P., and W. Bequette, "Process Identification Using Nonlinear Programming Techniques," *Proc. of the American Control Conf.*, 1534 (1990).

Stinchcombe, M., and H. White, "Universal Approximation Using Feedforward Networks With Non-Sigmoid Hidden Layer Activation Functions," *Proc. of the Int. Joint Conf. on Neural Networks*, 613 (1989).

Thibault, J., V. Van Berugesen, and A. Chéruy, "On-Line Prediction of Fermentation Variables Using Neural Networks," *Biotech. and Bioeng.*, **36**, 1041 (1990).

Wells, C. H., "Application of Modern Estimation and Identification Techniques to Chemical Processes," *AIChE J.*, **17**, 966 (1971).

Werbos, P., "Beyond Regression: New Tools for Prediction and Analysis in Behavioral Sciences," PhD Thesis, Harvard University (1974).

Willis, M. J., C. DiMassimo, G. A. Montague, M. T. Tham, and A. J. Morris, "Artificial Neural Networks in Process Engineering," *IEE Proc.-D*, **138**, 256 (1991).

*Manuscript received Dec. 5, 1991, and revision received May 13, 1992.*