

PROCESS MODELING USING STRUCTURED NEURAL NETWORKS

Dimitris C. Psychogios and Lyle H. Ungar
 Dept. of Chemical Engineering, University of Pennsylvania
 Philadelphia, PA 19104-6393

Abstract

A modeling approach is developed which combines a partial first principles model, incorporating the available prior knowledge about the process being modeled, with a neural network which serves as a non-parametric estimator of unmeasured process parameters that are difficult to model. This hybrid model is superior to standard "black-box" neural network models in that it interpolates and extrapolates much more accurately, is easier to analyze and interpret, and requires significantly fewer training examples. The hybrid network model, when used to model a fedbatch bioreactor, gives estimates of the unobserved process parameters and can be used to make predictions. This approach can also be applied when only part of the state is measured by using a state reconstruction method for the first principles component of the hybrid model.

1. Introduction

Artificial neural networks are connectionist models that have the ability to "learn". Learning is the process where the network approximates the underlying function mapping the system inputs to its outputs, given a set of observations of inputs and corresponding outputs. A typical engineering application of this kind is developing process models of dynamic systems, which are often used for control [6,14]. Modeling without using *a priori* knowledge has often proved successful [1,14,16] and is the only possible method when no process knowledge is available. However, the ability of neural networks to learn non-parametric (structure-free) approximations to arbitrary functions is also a weakness. A typical neural network involves hundreds of internal parameters, and this ample generality can lead to "overfitting" - fitting of the noise as well as the underlying function - and poor generalization. Furthermore, physical interpretation of such non-parametric models is difficult [12].

As a result, there has been an increasing interest in developing modeling methods that at least partially alleviate these problems. Since redundancy (excess degrees of freedom) may result in poor models, one route 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 networks' performance [7,13]. These methods, however, 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. One possibility is to incorporate a known linear model within a nonlinear neural network [5]. An alternative method is to impose internal structure on the neural network model by constructing its topology based on prior analysis of the system's behavior [12].

We believe that it is advantageous to *a priori* structure the neural network models; in the paper we develop a modeling strategy that combines first principles knowledge, in the form of equations such as mass and energy balances, with neural networks as non-parametric estimators of important process parameters. The resulting models can be thought of as structured neural networks which contain some known constraints, such as mass and energy balances or equations which contain "parameters" whose dependence on state variables is modeled by neural networks. The use of an approximate (or equivalently, partial) first principles model provides a better starting point for learning, and also allows for both structural and parametric uncertainty. Our goal is to develop hybrid neural network process models that generalize and extrapolate better than classical "black-box" neural

networks and which, at the same time, are more reliable and easier to interpret.

Section 2 describes the test problem on which we applied our modeling method. Section 3 explains in more detail the hybrid and standard neural network modeling approaches. Sections 4 and 5 describe the training procedures for both methods and their relative performance. The assumption of full state accessibility is relaxed in section 6; a nonlinear exponential observer is developed and incorporated in the hybrid neural network model, and results are discussed. Section 7 presents an application of the hybrid model to process optimization and, more specifically, to the calculation of optimal feed policies for fedbatch reactors. A discussion on the potential impact of the proposed modeling method and a summary of our results is given in the final section.

2. Modeling problem : fedbatch bioreactor

Consider a dynamic system described by the following general representation

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

$$p = g(x, u) \quad (2)$$

where x denotes the state vector of the system, u the control vector and p a vector of process parameters. This representation implies that the overall dynamic behavior of the system is determined by the set of equations $f()$ which can be, for example, simple mass balances. The parameters p essentially represent the process kinetics and are related to the system variables through the set of equations $g()$, which are often difficult to derive from first principles.

Biological reactors are a good example of such systems, as the kinetic behavior of the living organisms (cells), reflected by $g()$, is difficult to derive from first principles reasoning and typically unknown. However, it is the presence of this complex unknown functionality that makes biological reactors highly nonlinear. A typical fedbatch stirred bioreactor can be described by the following equations [4]:

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

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

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

where $X(t)$ is the biomass concentration and $S(t)$ is the substrate concentration. These mass balances on the reacting species provides 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 typically a complex function of the biochemical, biological and physicochemical variables of the system [4]. 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 our development it was assumed that the "true" but unknown and unmeasured growth rate was described by the Haldane model

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

the inlet substrate feed concentration S_{in} was the manipulated input, and the flowrate $F(t)$ was held constant. Eqn. 3-6 with parameter values $k_1 = 1$, $K_m = 10$, $K_i = 0.1$ and $\mu^* = 5$ 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/l, 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. The initial reactor volume was set equal to a nominal value of 10 l 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_{in} and flowrate 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 comprising of 50 data points. All measurements were corrupted with normal Gaussian noise $N(0,0.01)$.

3. Standard and hybrid neural network models

Neural networks have been used as "black-box" process variable estimators in bioreactor modeling applications [10,15]. 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.

The central idea of this paper is to integrate an approximate model, derived from simple first principles considerations, with a neural network which approximates the unknown kinetics, in order to develop a hybrid neural network process model. This approach offers significant advantages over a "black-box" neural network modeling methodology. In a hybrid neural network model the interaction among process variables is clearly determined by the internal structure imposed by the approximate first principles model. In contrast, the dense interconnection of a traditional neural network allows interaction of all process variables, even if such interactions are aphysical. Due to its meaningful internal structure, the hybrid neural network model is easier to analyze. The first principles part allows physical interpretation of the process variable interactions and their impact on the model's predictions, while non-parametric estimation is restricted to terms for which little or no knowledge is available. Such structured models are expected to perform better than "black-box" neural network models in process identification tasks.

Furthermore, in data reconciliation and adaptive modeling and control it is very important to be able to correctly identify 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 [6] because all of the model's internal parameters are updated since all are considered partially responsible for the error. In contrast, the physically meaningful 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 of the bioreactor of section 2 is shown in Fig. 1. The neural network component receives as inputs the process variables and provides an estimate of the process 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 forms the 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 network's weights. However, for the hybrid neural network model such an error signal is not immediately available, as the neural network's output (microbial growth rate) is not directly measured. In this case, the known partial process model can be used to calculate a suitable error signal to update the network's weights: The observed error between the hybrid model's predictions and the actual 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. Stated more formally, we perform gradient descent on the output error subject to a set of constraints represented by the first principles partial model. The details of the training procedure for both hybrid and standard neural network models are addressed in the following section.

4. 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. Since the state variables, particularly the biomass concentration, undergo changes of over an order of magnitude which can cause problems when using neural networks, we defined dimensionless biomass concentration and substrate concentration variables. 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_{in} . The desired network's outputs were the dimensionless forms of the state variables X and S .

The hyperbolic tangent was chosen as activation function, and the training method was the error back-propagation algorithm, with weights being 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, E , over both process variables (biomass and substrate concentration) was minimized as with the standard neural network model

$$E = (X - X')^2 + (S - S')^2 \quad (7)$$

where the prime denotes measured values of the corresponding variables. The neural network's output (cell growth rate μ) does not appear explicitly in the above expression. However, if it is considered constant for each sampling instant, the gradient of the hybrid model's output with respect to this internal parameter can be calculated through integration of the system's sensitivity equations [2]. Thus the gradient of the performance measure (Eqn. 7) with respect to the network's output is readily available and can be used to generate an error signal that is used to update the network's weights.

The hybrid neural network model was trained as follows: Both state variables (biomass and substrate concentration) were used as inputs to the neural network model component of Fig. 1. Obviously, since the "true" kinetics only depend on the substrate concentration (Eqn. 6), the biomass concentration input is merely a noisy input to the network. The inputs were

scaled in the same way as for the standard neural network model. The network's output was a (dimensionless) estimate of the growth rate for the current sampling time and was used as input to the first principles model component (Eqn. 3-5) along with S_{in} to predict the system's state for the next sampling time. The outputs of this hybrid model were also dedimensionalized to allow for direct comparison with the standard neural network model's outputs. At the same time, the sensitivity equations were integrated, and a dimensionless error signal was calculated. The error signals for the neural network component of the hybrid model were summed over all input-output examples in the training data set, and this total error was used to update the weights using back-propagation.

5. Comparison of hybrid and standard neural networks

It is often important to be able to accurately identify a system with a relatively small training data set, as for example when only few batch runs have been performed. 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 or the state space is not sampled sufficiently densely, it relies heavily on interpolation to approximate the dynamic system. On the contrary, since a hybrid neural network model already contains a partial model, the size of the training data set should, in principle, have less effect on its performance.

We investigated this issue by comparing 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 1000 data points respectively, randomly drawn from the available training database. The networks were trained as described above, with each of the five data sets randomly partitioned (with a 70% : 30% ratio) to two smaller subsets used for training and testing respectively. Fig. 2 shows that the prediction error of the hybrid network is an order of magnitude lower than that of the standard network model and is relatively unaffected by the training data set size; standard deviations of the prediction error are indicated by the vertical bars. However, the hybrid network's prediction error increases as the data set becomes very small. The approximation accuracy of the standard network, as expected, decreases monotonically 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.

Hybrid neural network models also extrapolate better than standard networks, as a result of the partial first principles model they contain. Poor extrapolation has been the plague of traditional neural networks [11], and is even more important for processes such as batch reactors which operate in a non-stationary mode. The superior extrapolation of the hybrid neural network compared to the standard neural network is shown in Fig. 3, 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. These results are due to the fact that the hybrid network generalizes and extrapolates only on the unobserved parameter while observing the underlying model structure (mass balances).

The ability of the hybrid network model to provide meaningful estimates of the growth rate is shown in Fig. 4, for the same simulation as above. When the noise level is reasonable compared to the substrate concentration (high S regime), the prediction accuracy is remarkable. The combination of high noise level (low S regime) and irrelevant input (biomass concentration) of high magnitude slightly decrease prediction accuracy.

6. Incomplete state measurement - hybrid network model

In the development of the hybrid neural network model discussed in section 3, the availability of the gradient of the state with respect to process parameters is essential in

calculating the derivative of the performance measure of the hybrid model with respect to the neural network's output. When some state variables are not measured, gradient information may not be directly calculated as described in section 4. One way to overcome this problem is to estimate the unmeasured state variable using a state observer, which computes estimates of the state of a dynamic system given estimates of the initial state x_0 . Following the analysis of [8], we can construct an exponential closed-loop observer, with desirable performance characteristics [9], for the fedbatch bioreactor system of section 2 under the assumption that only substrate concentration measurements are available. Based on Eqn. 3-4 and since

$$y = h(x) = [0 \quad 1] \begin{bmatrix} X \\ S \end{bmatrix} \quad (8)$$

we can choose the observer's gain matrix B such that

$$B = \begin{bmatrix} -\mu(t) \\ \mu(t) \end{bmatrix} \quad (9)$$

Then the matrix

$$\nabla f - B \cdot \nabla h = \begin{bmatrix} \mu(t) - \frac{F(t)}{V(t)} & \mu(t) \\ -k_1 \cdot \mu(t) & -\frac{F(t)}{V(t)} - \mu(t) \end{bmatrix} \quad (10)$$

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

$$\frac{d\hat{x}}{dt} = f(\hat{x}, p, u) - B(S' - \hat{S}) \quad (11)$$

$$\hat{X}(0) = X_0, \quad \hat{S}(0) = S_0 \quad (12)$$

is an exponential observer for the fedbatch bioreactor, where the hat denotes the observer estimates of the state variables. Eqn. 11-12 comprise the first principles part of the hybrid neural network model; the other component consists of a standard neural network that estimates the microbial growth rate μ , given as input only measurements of the substrate concentration, in contrast to the hybrid model of section 4. As done there, we can derive a set of observer sensitivity equations which provide the gradient of the performance measure with respect to the network's output. However, the performance measure in this case is the squared prediction error of the substrate concentration only. The training procedure is otherwise the same as the one discussed in section 4.

The performance of a hybrid neural network model trained using only substrate concentration measurements is shown in Fig. 5, where its predictions of the (unobserved) growth rate are given 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. We found competing state and process parameter estimation techniques, such as nonlinear programming optimization and extended Kalman filtering, to give significantly less accurate parameter estimates than the hybrid network. 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 that the signal-to-noise ratio for the substrate concentration measurements is very low in this regime. More importantly, this partial state hybrid network model, like the full state hybrid network, gives estimates for the unmeasured biomass concentration. In contrast, standard neural networks can only predict the measured process output (see the autoregressive model, in [14] or [1]).

7. Process operation scheduling using a hybrid neural network model

It was emphasized above that the most important advantage of the hybrid neural network model is that it provides a model of the unobserved growth kinetics. A practical application in

showing the usefulness of such a model is the optimization of fedbatch process operating schedules, i.e., determining the substrate feeding policy that maximizes product yield. One way to obtain such an optimal substrate feed policy is to use optimization techniques to maximize a suitable objective function. We formulated the problem of determining the optimal feed policy for the bioreactor of section 2 as follows:

$$\max_{S_m} J = X(T) \cdot V(T) \quad (13)$$

subject to Eqn. 3-5 and

$$\mu(t) = \pi(X, S) \quad (14)$$

$$S_{in,j} = S_{in,j+1} = S_{in,j+2} \quad (15)$$

This formulation implies that the objective function to be maximized is the cell mass in the reactor at the end of some feeding period T . Eqn. 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. Eqn. 15 implies that the substrate feed was 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 led 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 Successive Quadratic Programming [3].

The substrate feed optimal profile, for a batch run with initial conditions of $X=0.5$ g/l, $S=0.1$ g/l and $V=10$ lt, is shown in Fig. 6; also, for comparison, the optimal profile obtained if the same problem is solved using Eqn. 6 (the "true" growth rate model) instead of Eqn. 14. The substrate feed is initially set at a high value, so that the substrate concentration can be rapidly increased to a value of about 1g/l 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 batch runs.

8. Discussion

A hybrid neural network modeling approach was presented and used to model a fedbatch bioreactor. This hybrid model is comprised of two parts: A partial first principles model, which reflects *a priori* knowledge, and a neural network component, which serves as a non-parametric 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 (e.g., 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 considerations. This produces models which are more reliable and which generalize and extrapolate more accurately

than standard neural networks. Furthermore, hybrid networks are much easier to interpret than unstructured ones, since they predict meaningful parameters (hybrid nets) rather than distributing knowledge of the entire model among a large number of weights (standard nets). Of equal importance is that significantly less data are required for training hybrid neural networks. The partial first principles model dramatically reduces the number of functions that the neural network has to choose from so that hybrid networks give far better approximation accuracy, for the same number of data points, than standard neural networks.

It was also demonstrated that a hybrid model can be constructed even when only partial state measurements are available, and its prediction accuracy is not significantly affected. This is an improvement over standard network models which can be used only as autoregressive models in such a case. An additional advantage of the hybrid neural network, particularly when compared to Kalman filtering or NLP parameter estimation, is that it serves as a model of the process parameters and can, thus, give predictions. This feature is important for applications such as process optimization, and on-line multistep predictive or gain scheduling control.

References

- [1] Bhat, N. and McAvoy, T., "Use of Neural Nets for Dynamic Modeling and Control of Chemical Processes," *Comp. Chem. Engng.* 14, 573-583 (1990).
- [2] Caracotsios, M. and Stewart, W. E., "Sensitivity Analysis of Initial Value Problems With Mixed ODE'S and Algebraic Equations," *Comp. Chem. Engng.* 9, 359-365 (1985).
- [3] Cuthrell, J. E. and Biegler, L. T., "Improved Infeasible Path Optimization for Sequential Modular Simulators. II: The Optimization algorithm," *Comp. Chem. Engng.* 9, 257-267 (1985).
- [4] Dochain, D. and Bastin, G., "Adaptive Control of Fedbatch Bioreactors," *Chem. Eng. Comm.* 87, 67-85 (1990).
- [5] Haesloop, D. and Holt, B. R., "A Neural Network Structure for System Identification," *Proceedings of the American Control Conference*, 2460-2465 (1990).
- [6] Hernandez, E. and Arkun, Y., "Neural Network Modeling and an Extended DMC Algorithm to Control Nonlinear Systems," *Proceedings of the American Control Conference*, 2454-2459 (1990).
- [7] Karnin, E. D., "A Simple Procedure for Pruning Back-Propagation Trained Neural Networks," *IEEE Transactions on Neural Networks* 1, 239-242 (1990).
- [8] Kou, S. R. et al., "Exponential Observers for Nonlinear Dynamic Systems," *Information and Control* 29, 204-216 (1975).
- [9] Kravaris, C. and Chung, C. B., "Nonlinear State Feedback Synthesis by Global Input/Output Linearization," *AIChE J.* 33, 592-603 (1987).
- [10] Lant, P. A. et al., "A Comparison of Adaptive Estimation With Neural Based Techniques for Bioprocess Application," *Proceedings of the American Control Conference*, 2173-2178 (1990).
- [11] Leonard, J. A. et al., "A Neural Network Architecture That Computes Its Own Reliability," *Comp. Chem. Engng.*, submitted (1991).
- [12] Mavrouniotis, M., "Hierarchical Neural Networks for Process Monitoring," *Comp. Chem. Engng.*, submitted (1990).
- [13] Mozer, M. C. and Smolensky, P., "Skeletonization: A Technique For Trimming The Fat From A Network via Relevance Assessment," in *Advances in Neural Information Processing* D.S. Touretzky, Ed. Morgan Kaufmann, (1989).
- [14] Psychogios, D. C. and Ungar, L. H., "Direct and Indirect Model Based Control Using Artificial Neural Networks," *Ind. Eng. Chem. Res.* 30, 2564-2573 (1991).
- [15] Thibault, J. et al., "On-Line Prediction of Fermentation Variables Using Neural Networks," *Biotechnology and Bioengineering* 36, 1041-1048 (1990).
- [16] Willis, M. J. et al., "Artificial Neural Networks in Process Engineering," *IEE Proceedings-D* 138, 256-266 (1991).

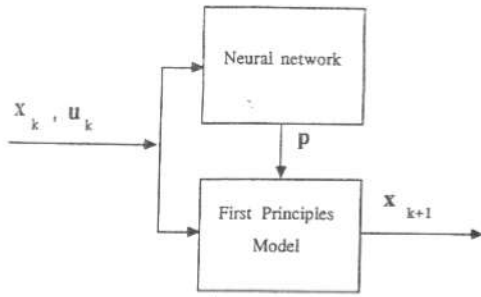


Figure 1. Hybrid (structured) neural network model

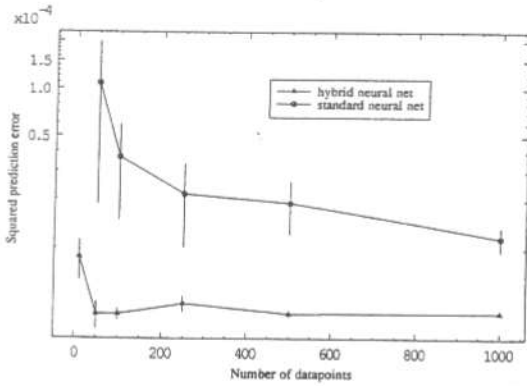


Figure 2. Mean squared prediction error vs number of training examples

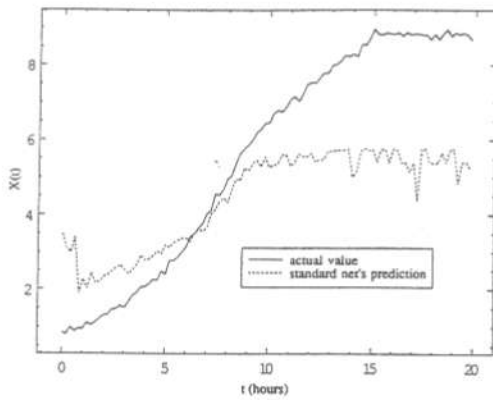


Figure 3a. Biomass concentration vs time for a standard network model

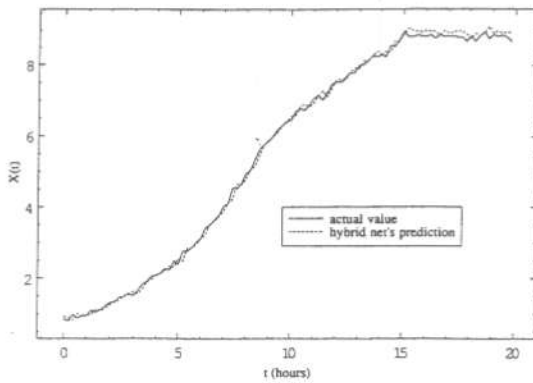


Figure 3b. Biomass concentration vs time for a hybrid network model

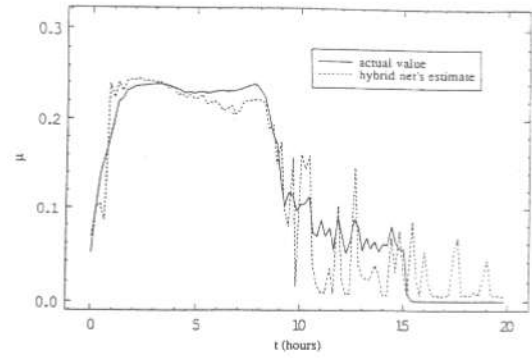


Figure 4. Cell growth rate estimate for the hybrid network of section 4

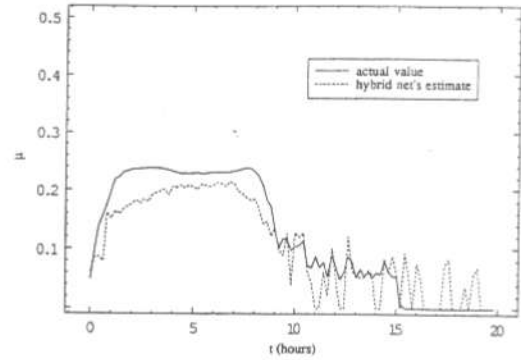


Figure 5. Cell growth rate estimate for the hybrid network of section 6

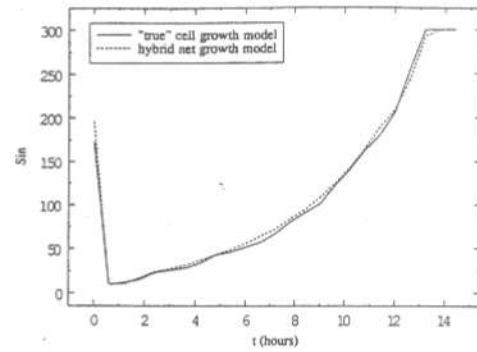


Figure 6. Optimal substrate feed policy for the fedbatch bioreactor