ESAT - SISTA
report 41 - 1992.

Neural Optimal Control of Fed-batch Fermentation Processes

Johan Suykens, Jan Van Impe[1]

Bart De Moor[2], Joos Vandewalle

K.U. Leuven - Department Electrical Engineering - ESAT

Kardinaal Mercierlaan 94, B-3001 Leuven, BELGIUM

tel: 32/16/22 09 31 - fax: 32/16/22 18 55

e-mail: suykens@esat.kuleuven.ac.be

or: vanimpe@esat.kuleuven.ac.be

December, 1992

Submitted to

AUTOMATICA

BRIEF PAPER

[1]Senior research assistant with the N.F.W.O. (Belgian National Fund for Scientific Research)

[2]Research associate with the N.F.W.O. (Belgian National Fund for Scientific Research)

Abstract

We propose a neural optimal control strategy for a fed–batch fermentation process. The principle of parameterized nonlinear static state feedback is applied by means of neural control. Learning of the interconnection weights is done by optimization of a simulation result of interest. A procedure is proposed of gradually extending the network with hidden layers in order to improve the performance of the system. To illustrate the method some simulation results for a penicillin G fed–batch fermentation process are given. As we come very close to the theoretical optimal control result (in the sense of Pontryagin), this method may be useful in calculating optimal profiles which are analytically intractable.

Keywords: multilayer feedforward neural networks, nonlinear optimization, bioreactor control, fed–batch fermentation process

1 Introduction

The design of high performance model based control algorithms for biotechnological processes is hampered by two major problems which call for adequate engineering solutions. First, the process kinetics are most often poorly understood nonlinear functions, while the corresponding parameters are in general time varying. Second, up till now there is a lack of reliable sensors suited to real time monitoring of process variables which are needed in advanced control algorithms. Therefore, the earliest attempts at control of a biotechnological process used no model at all. Successful state trajectories from previous runs which had been stored in the process computer were tracked using open–loop control. Many industrial fermentations are still operated using this method.

During the last two decades, two trends for the design of monitoring and control algorithms for fermentation processes have emerged (Bastin 1991). In a first approach, the difficulties in obtaining an accurate mathematical process model are ignored. In numerous papers classical methods (e.g. Kalman filtering, optimal control theory, ...) are applied under the assumption that the model is perfectly known. Due to this oversimplification, it is very unlikely that a real life implementation of such controllers —very often this implementation is already hampered by e.g. monitoring problems— would result in the predicted simulation results. In a second approach, the aim is to design specific monitoring and control algorithms without the need for a complete knowledge of the process model, using concepts from e.g. adaptive control and nonlinear linearizing control. A comprehensive treatment of these ideas can be found in the textbook by Bastin and Dochain (1990) and the references therein.

Van Impe (1992a and 1993) has shown how to combine the best of both trends into one unifying methodology for optimization of biotechnological processes : *optimal adaptive control.* This is motivated as follows. Model–based *optimal control* studies provide a theoretical realizable optimum. However, the real life implementation will fail in the first place due to modeling uncertainties. On the other hand, model–independent *adaptive controllers* can be designed, but there is a priori no guarantee for at least suboptimality of the results obtained. The gap between both approaches is bridged in two steps. First, heuristic control strategies are developed with near optimal performance under all conditions. These suboptimal controllers are based on biochemical knowledge concerning the process and on a careful mathematical analysis of the optimal control solution. In a second step, implementation of these profiles in an adaptive model–independent way combines excellent robustness properties with near optimal performance.

The main contribution of this paper is the following. We illustrate that, as an alternative to applying Pontryagin's minimum principle (see e.g. Bryson and Ho 1975) in the first step of the above procedure, the determination of the theoretical optimal profiles can be done with an excellent accuracy by using a neural network. In this way, the new methodology of optimal adaptive control could be easily extended to biotechnological processes which are analytically intractable. This is for instance the case for fermentation processes involving mixed cultures (and thus more than one substrate and/or biomass), for which a complete optimal control solution does not exist up to now.

It is known that multilayer neural networks with one or more hidden layers can approximate any continuous nonlinear function (Funahashi, 1989; Hornik, 1989). This means that these networks can be used also in order to parameterize the control input to a fermentation process. The neural network represents then a nonlinear mapping from the state space to the control space and acts as nonlinear static state feedback. The method for learning of the interconnection weights is based on a simulation approach, where the simulation result of interest is to be optimized (see also Suykens and De Moor, 1992). A procedure is proposed that adds hidden layers to the neural network making use of the previous results in order to improve the performance.

In this paper the proposed methodology is applied to the optimization of the penicillin G fed–batch fermentation process, for which a complete optimal control solution is described in the literature (see e.g. Van Impe *et al.*, 1992b). The paper is organized as follows. In section 2 the penicillin fed-batch fermentation process is briefly described. In section 3 the principle of parameterized static state feedback by multilayer neural networks is introduced. Section 4 gives a method for learning of the interconnection weights through a simulation approach. Some simulation results and a comparison with optimal control are given in section 5 and 6.

2 Process model and optimization problem

In this section we introduce a commonly used model for the penicillin G fed–batch fermentation process, and formulate an optimization problem in order to maximize the final amount of product.

2.1 Process model

According to Bajpai and Reuß (1981), the penicillin G fed–batch fermentation process can be described by the following nonlinear mathematical model :

$$\begin{aligned} \frac{dS}{dt} &= -\sigma X + C_{s,in}u & \frac{dX}{dt} &= \mu X \\ \frac{dP}{dt} &= \pi X - k_h P & \frac{dV}{dt} &= u \end{aligned} \tag{1}$$

where the state variables S, X, P, and V are respectively the amount of substrate (glucose) in broth [g], the amount of cell mass in broth [g DW] (DW stands for dry weight), the amount of product (penicillin) in broth [g], and the volume of the liquid phase in the fermentor [L]. Concentrations C_s, C_x and C_p are defined as S/V, X/V and P/V. The input u of the system is the volumetric substrate feed rate [L/h]. $C_{s,in}$ (expressed in [g/L]) is the (constant) substrate concentration in the feed stream u, while k_h [1/h] is the product hydrolysis or degradation constant. Observe that this model structure is representative for a whole class of fermentation processes with product formation.

σ, μ and π are respectively the specific substrate consumption rate [g/g DW h], the specific growth rate [1/h] and the specific production rate [g/g DW h]. These are interrelated by :

$$\sigma = \frac{\mu}{Y_{x/s}} + m + \frac{\pi}{Y_{p/s}}$$

with $Y_{x/s}$ the biomass on substrate yield coefficient [g DW/g], $Y_{p/s}$ the product on substrate yield coefficient [g/g], and m the specific maintenance demand [g/g DW h]. In the case of the penicillin G fed–batch fermentation process, μ and π are modeled using the following nonlinear functions of the concentrations C_s and C_x :

Specific growth rate μ : *Contois*–kinetics

$$\mu = \mu_C \frac{C_s}{K_x C_x + C_s} \tag{2}$$

with μ_C the maximum specific growth rate [1/h] and K_x the Contois saturation constant for substrate limitation of biomass production [g/g DW].

Specific production rate π : *Haldane*–kinetics

$$\pi = \pi_m \frac{C_s}{K_p + C_s + C_s^2/K_i} \tag{3}$$

with π_m the specific production constant [g/g DW h], K_p the Monod saturation constant [g/L] and K_i the substrate inhibition constant [g/L].

Observe that the process model (1) can be written in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{b}\, u \tag{4}$$

where the state vector $\mathbf{x}$ is defined as (^{t} denotes the transpose) $\mathbf{x} \equiv [x_1\ x_2\ x_3\ x_4]^t = [S\ X\ P\ V]^t$, and $\mathbf{f} = [-\sigma(\mathbf{x})x_2\ \ \mu(\mathbf{x})x_2\ \ \pi(\mathbf{x})x_2 - k_h x_3\ \ 0]^t$, $\mathbf{b} = [C_{s,in}\ 0\ 0\ 1]^t$.

2.2 Optimization problem

The optimization problem considered in this paper is to maximize the final amount of product $P(t_f)$ for a given total amount of substrate, or equivalently, to minimize a performance index J :

$$J[u(t), x_1(0), t_f] = -x_3(t_f) \tag{5}$$

under the following set of constraints :

1. The final volume V is fixed :

$$x_4(t_f) = V_f \tag{6}$$

   Observe that, due to the fourth model equation, this means that the total amount of substrate available is fixed a priori.

2. The initial amount of biomass $x_2(0)$ and of product $x_3(0)$ is given, while the initial substrate amount $x_1(0)$ is free. The initial volume $x_4(0)$ follows from :

$$x_4(0) = V_* + \frac{x_1(0)}{C_{s,in}} \tag{7}$$

   where V_* is the (known) initial volume without any substrate. Remember that substrate is added as a solution with concentration $C_{s,in}$.

Observe that the final time t_f is free and can be used as an additional control variable. Furthermore, all state variables are assumed to satisfy $x_i(t) \geq 0$ $(\forall i, t)$. We do not include this condition as a constraint in the optimization problem, but we assume that the model (1) is modified such that it is automatically fulfilled.

3 Neural controller structure

For the neural controller we apply the principle of *parameterization*. The control signal u is the output of a feedforward neural network with a predefined and fixed structure. We will start with a very simple neural network consisting of one neuron and we will demonstrate that good results can be obtained with this network. It will also be illustrated how the performance can be improved by introducing hidden layers.

3.1 Neural network without hidden layers

In the 'one neuron case' (Fig.1) the control signal u can be parameterized as

$$u = s[u_{MAX} \cdot tanh(\mathbf{w}^t\mathbf{z} + \theta)] \tag{8}$$

where $\mathbf{z} = [x_1/x_4 \quad x_2/x_4]^t \equiv [C_s\ C_x]^t$ and $\mathbf{w} = [w_1\ w_2]^t$ is the vector of interconnection weights. In other words, we assume that the control is a (nonlinear) feedback law of the concentrations C_s and C_x. This assumption is inspired by the the analytical structure of the specific rates μ (2) and π (3), and by the fact that product P does not explicitly appear at the right–hand side of the differential equations for S, X and V (1).

We consider a neuron with zero threshold $\theta = 0$. The parameter u_{MAX} is the maximum achievable value of the control signal u. Two nonlinear functions appear in control law (8): the tangent hyperbolic function $tanh(\alpha) = (1 - exp[-2\alpha])/(1 + exp[-2\alpha])$, and the function $s(\alpha)$ with $s = \alpha$ if $\alpha \geq 0$ and $s = 0$ if $\alpha < 0$ which guarantees that $u(t) \geq 0$ for all t (Fig.2). Observe that $\mathbf{w}^t\mathbf{z} = 0$ is a linear switching line that divides the (C_s, C_x)–plane into two parts: a part where $u = 0$ and another where $u = u_{MAX} \cdot tanh(\mathbf{w}^t\mathbf{z})$.

3.2 Optimization problem after parameterization

With this specified structure for the control signal the optimization problem translates into minimization of :

$$J[\mathbf{w}, u_{MAX}, x_1(0), t_f] = -x_3(t_f) \tag{9}$$

under the same constraints (6) and (7) as for (5). Observe that, as compared to (5), the number of degrees of freedom has been drastically reduced, because only w_1, w_2 and u_{MAX} are to be determined instead of the whole input profile $u(t)$ ($\forall t$).

3.3 Neural network with hidden layers

Once an optimal solution is found to (9) we can try to further decrease the value of cost function J by introducing hidden layers into the network. In the case of one hidden layer (Fig.1) the control signal becomes then :

$$u = s[u_{MAX} \cdot tanh(\mathbf{w}^t tanh(\mathbf{T}\ \mathbf{z}))]$$

where $\mathbf{w}$ ($\in \mathbb{R}^{2\times 1}$) and $\mathbf{T}$ ($\in \mathbb{R}^{2\times 2}$) contain the interconnection weights of the output neuron and of the hidden neurons respectively. In the case of m hidden layers this becomes :

$$u = s[u_{MAX} \cdot tanh(\mathbf{w}^t tanh(\mathbf{T}_1 \cdots tanh(\mathbf{T}_m\ \mathbf{z}) \cdots))]$$

The increased number of parameters allows to realize a nonlinear switching curve. In the following section we will show how the results to (9) of the one neuron case can be used for learning of the weights when the network is extended with hidden layers.

4 Learning algorithm for the interconnection weights

The solution to the optimization problem (5) that we propose here consists of the following three major steps :

Step 1 Random search for the weights in the one neuron case.

Step 2 Application of a (local) nonlinear optimization method starting from Step 1.

Step 3 Extension of the neural network with hidden layers.

We discuss now each of these steps in detail.

4.1 Step 1 : Random search for the one neuron case

In the one neuron case we have to solve the optimization problem (9). We simplify this problem first to an unconstrained minimization problem

$$J[\mathbf{w}] = -x_3(t_f) \tag{10}$$

by fixing u_{MAX}, $x(0)$, t_f. The performance index J is calculated from the simulation result of the differential equations (1) using certain values for $\mathbf{w}$, $x(0)$ and t_f. An estimate for

u_{MAX} can be found from the equation $\dot{x}_4 = u$ combined with the constraint $x_4(t_f) \leq V_f$ (7). We propose then random interconnection weights w_1 and w_2 : normal distribution with zero mean and a standard deviation chosen such that $\mathcal{O}\{E\{\mathbf{w}^t\mathbf{z}\}\} = 1$ (see also Fig.2), where $E\{.\}$ denotes the expectation operator and $\mathcal{O}$ the order of magnitude. This procedure is repeated then for some other values of u_{MAX}, $x(0)$, t_f.

For the unconstrained minimization problem (10) other methods like genetic algorithms can be used. But in our case of two weights random search turns out to work fairly well. In any case we need a global search procedure because (10) may have many local optima.

4.2 Step 2 : Local optimization of the one neuron case

The results of Step 1 can now be used as starting point for a (local) nonlinear optimization routine (e.g. *constr* of Matlab, that uses a technique of sequential quadratic programming). We solve then a constrained minimization problem of the form

$$J[\mathbf{w}, u_{MAX}, x_1(0), t_f] = -x_3(t_f) \tag{11}$$

with constraint $x_4(t_f) \leq V_f$ (7). The file where the objective function and the constraint are evaluated consists of the following calculations

[f,g]=xprime(inpar)

% input arguments: inpar: vector containing $\mathbf{W}, u_{MAX}, x_1(0), t_f$

% output arguments: f: performance index , g: constraint

- compute $x_4(0)$ from $x_1(0)$
- apply an integration rule to the system of differential equations
- evaluate the performance index J from the simulation result
- evaluate the constraint from the simulation result

4.3 Step 3 : Extension of the network with hidden layers

Once we have found the optimal linear switching line $\mathbf{w}^t\mathbf{z} = 0$ in Step 2, we can try to find a better (nonlinear) switching characteristic by extending the neural network with hidden layers. It is possible then to use the optimal results of Step 2 as starting point for a constrained minimization problem. In the case of one hidden layer we have :

$$J[\mathbf{w}, \mathbf{T}, u_{MAX}, x_1(0), t_f] = -x_3(t_f)$$

under constraint (6). The starting values for the interconnection weights are then :

$$\mathbf{w} := \frac{1}{\alpha}\mathbf{w}$$
$$\mathbf{T} := \alpha\,\mathbf{I}_2$$

where α is chosen such that $\mathcal{O}\{E\{\alpha\mathbf{T}\mathbf{z}(t)\}\} << 1$. The motivation for this choice is that the nonlinear switching curve $\mathbf{w}^t tanh(\mathbf{T}\ \mathbf{z}) = 0$ tends to the linear switching line $\mathbf{w}^t\mathbf{z} = 0$ for $\alpha \rightarrow 0$. This trick can be repeated if one wants to add more hidden layers. An advantage of this technique is that it is not needed to relearn the weights from the beginning (Step 1). Instead, one can improve the results of Step 2 by gradually extending the neural network. The degree of improvement can even be quantified based on the decrease in performance index J.

5 Illustrative case study

5.1 Model

For the model (1) we take the following parameters (in [g,L,h]–units) (see Bajpai and Reuß 1981) : $C_{s,in} = 500$, $k_h = 0.01$, $Y_{x/s} = 0.47$, $m = 0.029$, $Y_{p/s} = 1.2$, $\mu_C = 0.11$, $K_x = 0.006$, $\pi_m = 0.004$, $K_p = 0.0001$, $K_i = 0.1$. For the initial state variables we take $x_2(0) = 10.5$, $x_3(0) = 0$, $V_* = 7$. The final volume is $V_f = 10$. For numerical reasons we simulate the system with the state variables x_1 and x_2 expressed in [kg]. In order to assure that $x_i(t) \geq 0$ $(\forall i, t)$, we modify the set of differential equations (4) as follows :

$$\dot{x}_i = \begin{cases} max\{0, f_i + b_i\,u\} & \text{if } x_i \leq 0 \\ f_i + b_i\,u & \text{if } x_i > 0 \end{cases}$$

5.2 Learning algorithm

Step 1 For the unconstrained optimization problem (10) we fixed u_{MAX}, $x_1(0)$ and t_f to 0.015, 500 and 120. After 100 simulations with random weights, we obtained $w_1 = -70.10$ and $w_2 = 91.01$ resulting in a performance index $J = -25.43$.

Step 2 We solved then the constrained optimization problem (11) by means of the *constr* function of Matlab. The input parameter *inpar* for the optimization routine was taken from Step 1 ($inpar$ = $[-70.10\ 91.01\ 0.015\ 500\ 120]^t$). After 50 iterations we obtained as optimal result $w_1 = -69.89$, $w_2 = 91.20$, $u_{MAX} = 0.0198$, $x_1(0) = 500.78$ and $t_f = 126.60$ with performance index $J = -63.67$. The simulation results obtained with these optimal parameters are shown in Fig.3. For a lower initial state $S(0)$ good results can be obtained by the same network structure : with initial parameter vector $inpar = [-70.10\ 91.01\ 0.015\ 20\ 120]^t$, we obtained after 50 iterations as optimal result $inpar = [-68.88\ 117.86\ 0.019\ 35.68\ 171.26]^t$ with $J = -62.50$ (see Fig.4).

Step 3 If we compare the results of the neural controller with the results of optimal control theory from Pontryagin's Minimum Principle, we see that we are very close to the ultimate performance index J. The optimal results derived from the Minimum Principle are (see e.g. Van Impe *et al.* 1992c) : $P(t_f) = 63.846$, $t_f = 132$ and $x_1(0) = 528$. Another observation that can be made is that the performance index J is more sensitive to changes in u_{MAX} and t_f than to changes in $x_1(0)$. Because we are so close to the theoretical optimum, it makes not very much sense to extend the neural network with a hidden layer.

6 Conclusion

In this paper we considered the optimization of a fed–batch fermentation process. We have shown that by parameterization of the control signal by neural networks a solution in feedback form can be obtained for different initial states of the system that are very close to the ultimate achievable goal derived from Pontryagin's Maximum Principle even with a simple network consisting of one neuron and two interconnection weights. A learning algorithm for the interconnection weights is proposed than can gradually improve the performance of the system by adding hidden layers and making use of the simpler network. We believe that neural controllers will become candidate solutions to control even more complicated processes for getting an idea of the optimal input profile by numerical optimization, especially in the case where analytical methods become intractable.

Acknowledgments

This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the framework of a Concerted Action Project of the Flemish Community (entitled *Applicable Neural Networks*), and of the Belgian National incentive–program on fundamental research in Life Sciences initiated by the Belgian State — Prime Minister's Office — Science Policy Programming. The scientific responsibility is assumed by its authors.

References

- Bajpai, R.K. and M. Reuß (1981). Evaluation of feeding strategies in carbon–regulated secondary metabolite production through mathematical modelling, *Biotechnol. Bioeng.*, **23**, 717–738
- Bastin, G. and D. Dochain (1990). *On-line estimation and adaptive control of bioreactors*, Elsevier Science Publishing Co.
- Bastin, G. (1991). Nonlinear and adaptive control in biotechnology : a tutorial. *Proceedings of the 1991 European Control Conference*, Grenoble (France), 2001–2012
- Bryson, A.E. Jr and Y.-C. Ho (1975). *Applied optimal control*, Hemisphere, Washington - New York - London
- Funahashi, K.I. (1989). On the Approximate Realization of continuous Mappings by Neural Networks, *Neural Networks*, **2**, 183–192
- Hornik, K. (1989). Multilayer Feedforward Networks are universal approximators, *Neural Networks*, **2**, 359–366
- Miller, W., R. Sutton and P. Werbos (1990). *Neural networks for control*, Cambridge, MA: M.I.T. Press
- Suykens, J. and B. De Moor (1992). Stabilizing neural controllers with application to control of inverted pendulum, submitted to *IEEE Transactions on Neural Networks*
- Van Impe, J.F., G. Bastin, B. De Moor, V. Van Breusegem and J. Vandewalle (1992a). Optimal adaptive control of fed-batch fermentation processes with growth/production decoupling, *Proceedings of the 5th International Conference on Computer Applications in Fermentation Technology and 2nd IFAC Symposium on Modeling and Control of Biotechnical Processes*, March 29–April 2, 1992, Keystone (Colorado), in press
- Van Impe, J.F., B. Nicolaï, P. Vanrolleghem, J. Spriet, B. De Moor and J. Vandewalle (1992b). Optimal control of the penicillin G fed–batch fermentation : an analysis of a modified unstructured model, *Chemical Engineering Communications*, in press
- Van Impe, J.F. (January 1993). *Modeling and optimal adaptive control of biotechnological processes*, PhD thesis, Department of Electrical Engineering, K.U. Leuven, Belgium

Figure captions

Figure 1

Parameterization of the control signal u *by a neural network*

a/ one neuron case *b/ neural network with one hidden layer*

Figure 2

Nonlinear functions tanh(.) and s(.) appearing in the neural network

Figure 3

Optimal result with neural controller that consists of one neuron and $S(0) = 500.78$

a/ substrate and cell mass concentrations *b/ amount of product and volume*

c/ optimal control signal from neural network

Figure 4

Optimal result with neural controller that consists of one neuron and $S(0) = 35.68$

a/ substrate and cell mass concentrations *b/ amount of product and volume*

c/ optimal control signal from neural network

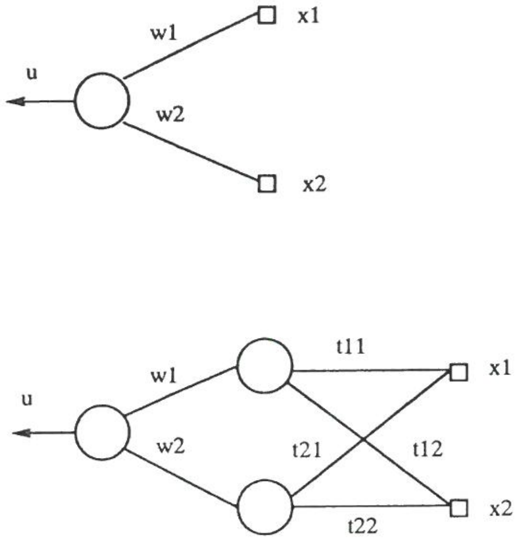
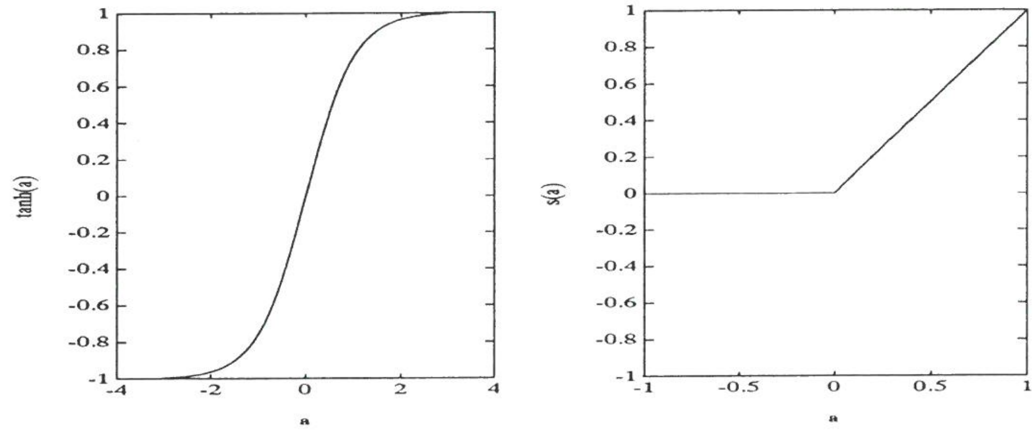


Figure 1:



tanh(a)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-4
-2
0
2
4
a
s(a)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
a

Figure 2:

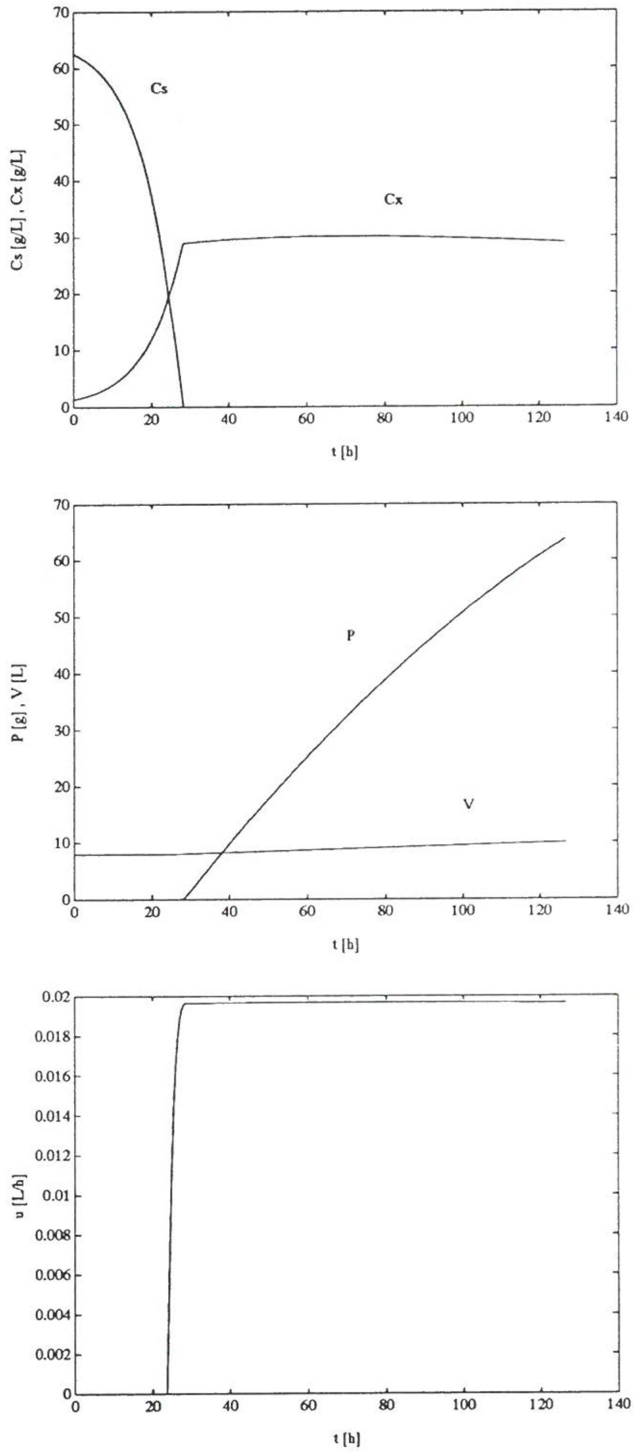


Figure 3:

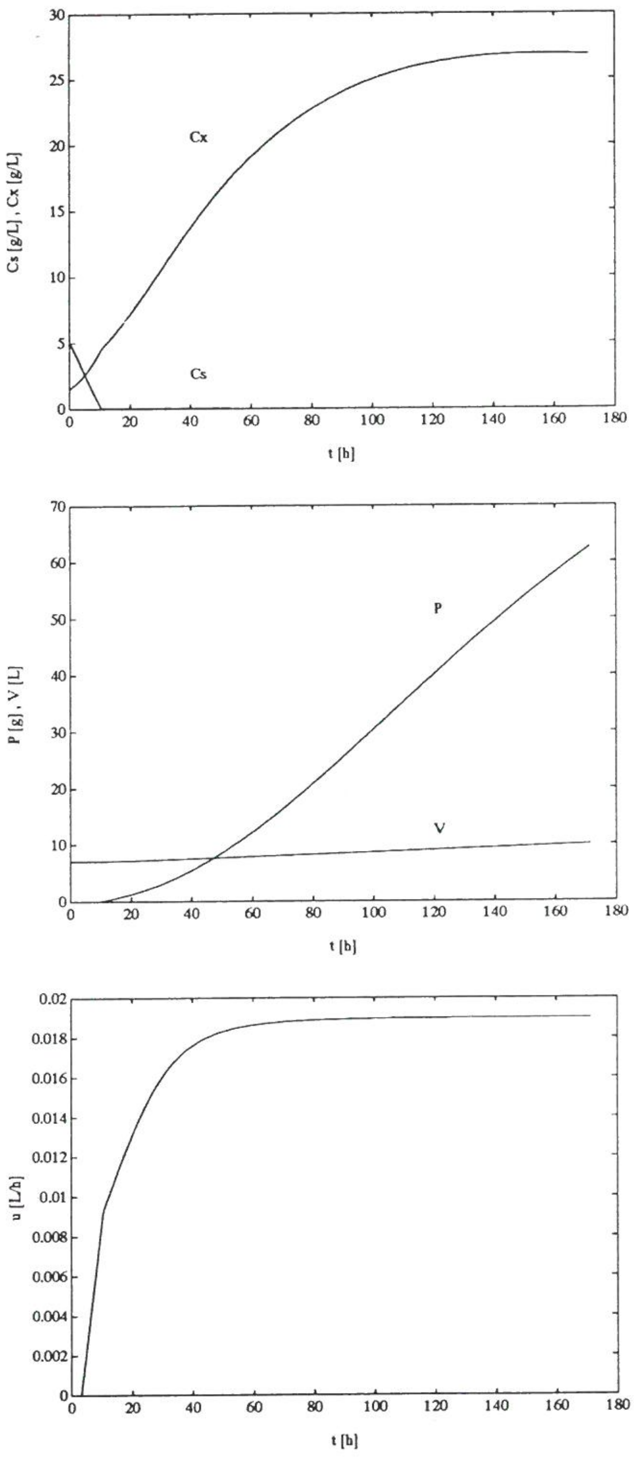


Figure 4: