ELSEVIER

Chemical Engineering Journal 127 (2007) 95–109

Chemical Engineering Journal

www.elsevier.com/locate/cej

Model based control of a yeast fermentation bioreactor using optimally designed artificial neural networks

Zoltan Kalman Nagy

Department of Chemical Engineering, Loughborough University, Loughborough, Leics LE11 3TU, United Kingdom

Received 25 March 2006; received in revised form 5 September 2006; accepted 3 October 2006

Abstract

Artificial Neural Networks (ANN) have been used for a wide variety of chemical applications because of their ability to learn system features. This paper presents the use of feedforward neural networks for dynamic modeling and temperature control of a continuous yeast fermentation bioreactor. The analytical model of this nonlinear process is also presented and it was used to generate the training data. Different ANNs were trained using the backpropagation learning algorithm. To avoid over-fitting of the data and achieve the best prediction ability with the simplest structure possible, a pruning algorithm is proposed for topology optimization of the ANN. The resulting ANNs were introduced in a Model Predictive Control scheme to test the control performance of the structure. The robustness of this control structure was studied in the case of setpoint changes and noisy temperature measurement, when the network used for prediction had been trained including noisy data in the training set. Results obtained with Linear Model Predictive Control (LMPC) as well as with proportional-integral-derivative (PID) control are also presented and compared with those obtained with the neural network model based predictive control (NNMPC) strategy. The use of inverse neural models for dynamic modeling and control of this process is also discussed and exemplified via simulations.

Keywords: Artificial neural networks; Model based control; Fermentation bioreactor model

1. Introduction

Model Predictive Control (MPC) is one of the most widespread advanced process control strategies in the chemical industries [1]. The main idea of the MPC algorithm is to solve an optimization problem to find the control vector trajectory, which optimizes some kind of performance objective over a future prediction horizon. Predicted values of the controlled parameters are obtained from the process model. Most of the model based control algorithms are based on linear models, because of the numerous techniques available for identification and controller design and optimization. Linear Model Predictive Control (LMPC) is preferred for practical implementation also because of the favorable computation time requirements. However, most of the chemical processes are highly nonlinear, with widely varying operating conditions. In these situations LMPC technology cannot be applied readily. The drawbacks of the LMPC can be avoided using for prediction a nonlinear model of the process instead the linear one. In the Nonlinear Model Predictive Control (NMPC) techniques predictions are usually obtained by integrating the analytical model of the process, described for example by a set of differential equations. However, this approach has two main disadvantages compared to LMPC methods:

(a) It requires the elaboration of a complex, analytical model of the process with good accuracy, which in the case of the most chemical processes can be a very arduous task.
(b) The optimization problem in the NMPC, that requires the repeated solution of the analytical model, might require great computational effort and time, which for large scale, complicated processes can become prohibitively large.

These shortcomings can be avoided using Artificial Neural Networks (ANNs) as the nonlinear model used in the control movement computation. The advantageous properties of neural networks, such as parallel computation, nonlinear mapping and learning capabilities make them an alluring tool in many chemical engineering problems. In the past 30 years there has been a growing interest in the field of artificial intelligence [2]. Neural networks have been successfully used for a wide variety of chemical engineering applications, such as detections and loca-

E-mail address: z.k.nagy@lboro.ac.uk.

1385-8947/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.cej.2006.10.015

Nomenclature

A_j	the output of the jth layer of the network ($j = \overline{0, N}$)
A_N	the output of the neural network
A_T	heat transfer area (m^2)
A_1, A_2	preexponential factors in Arhenius equation
C	discrete state space matrix used in linear models
$C_{heat,ag}$	heat capacity of cooling agent ($J\,g^{-1}\,K^{-1}$)
$C_{heat,\,r}$	heat capacity of mass of reaction ($J\,g^{-1}\,K^{-1}$)
c_j	concentration of ion j (j = Na, Ca, Mg, Cl, CO_3, etc.)
c_{O_2}	oxygen concentration in the liquid phase (mg/l)
$c^*_{O_2}$	equilibrium concentration of oxygen in the liquid phase (mg/l)
$c^*_{O_2,0}$	equilibrium concentration of oxygen in distilled water (mg/l)
c_p	product (ethanol) concentration (g/l)
c_S	substrate (glucose) concentration (g/l)
$c_{S,\,in}$	glucose concentration in the feed flow (g/l)
c_X	biomass (yeast) concentration (g/l)
D	output data from the training set
E	sum squared error of the network
E_{a1}, E_{a2}	apparent activation energy for the growth, respectively, denaturation reaction
F_{ag}	flow of cooling agent ($l\,h^{-1}$)
F_e	outlet flow from the reactor ($l\,h^{-1}$)
F_i	flow of substrate entering the reactor ($l\,h^{-1}$)
F_j	transfer function of the jth layer of the net ($j = \overline{0, N}$)
h	number of learning epoch
H	Gauss–Newton Hessian of the unregularized error criterion in the OBS algorithm
$\tilde{H}$	Gauss–Newton Hessian of the regularized error criterion in the OBS algorithm
H_i	specific ionic constant of ion i (i = Na, Ca, Mg, Cl, CO_3, etc.)
I	input data from the training set
I_i	ionic strength of ion i (i = Na, Ca, Mg, Cl, CO_3, etc.)
I_j	jth unit vector
k	discrete time
(k_1a)	product of mass-transfer coefficient for oxygen and gas-phase specific area (h^{-1})
$(k_1a)_0$	product of mass-transfer coefficient at 20 °C for O_2 and gas-phase specific area (h^{-1})
K_{O_2}	constant of oxygen consumption (g/l)
K_P	constant of growth inhibition by ethanol (g/l)
K_{P1}	constant of fermentation inhibition by ethanol (g/l)
K_S	constant in the substrate term for growth (g/l)
K_{S1}	constant in the substrate term for ethanol production (g/l)
K_T	heat transfer coefficient ($J\,h^{-1}\,m^{-2}\,K^{-1}$)
l_r	learning rate
m	momentum parameter used in the learning algorithm (0.95)
m_i	quantity of inorganic salt i (i = NaCl, $CaCO_3$, $MgCl_2$) (g)
M_i	molecular/atomic mass of salt/ion i (g/mol)
$n(i,j)^{(h)}$	weighting factor from the ith input variable to the jth output variable in the hth learning epoch
N	number of layers in the neural network (input layer is not counted)
P	prediction horizon
Q	number of sets of training input–output data
Q_r	regularization matrix
r_{O_2}	rate of oxygen consumption ($mg\,l^{-1}\,h^{-1}$)
R	universal gas constant ($8.31\,J\,mol^{-1}\,K^{-1}$)
R_{SP}	ratio of ethanol produced per glucose consumed for fermentation
R_{SX}	ratio of cell produced per glucose consumed for growth
S_j	number of neurons in the jth layer ($j = \overline{0, N}$)
t	time (h)
T_{ag}	temperature of cooling agent in the jacket (°C)
T_{in}	temperature of the substrate flow entering to the reactor (°C)
$T_{in,ag}$	temperature of cooling agent entering to the jacket (°C)
T_r	temperature in the reactor (°C)
T_{sp}	setpoint temperature (°C)
U	vector of the manipulated variables in linear models
V	volume of the mass of reaction (l)
V_j	volume of the jacket (l)
X	state vector in linear models
Y	output vector in linear models
Y_{O_2}	yield factor for biomass on oxygen (mg/mg), defined as the amount of oxygen consumed per unit biomass produced
z	ionic charge of ion i

Greek symbols

$\delta w(i,j)^{(h)}$	variation of the weighting factor in the hth learning epoch
ΔH_r	reaction heat of fermentation (kJ/mol O_2 consumed)
Φ, Γ	discrete state space matrices used in linear models
λ_j	Lagrange multipliers
μ_{O_2}	maximum specific oxygen consumption rate (h^{-1})
μ_P	maximum specific fermentation rate (h^{-1})
μ_X	maximum specific growth rate (h^{-1})
θ^*	vector with all weights and biases of the reduced network
ρ_{ag}	density of cooling agent (g/l)
ρ_r	density of the mass of reaction (g/l)
ξ_j	saliency for weight j corresponding to the ANN structure

tions of gross errors in process systems [3], fault detection in control systems [4] and optimal design of chemical processes [5]. ANNs were successfully used in many applications as nonlinear input–output maps for process data, in the identification and modeling of linear and nonlinear systems [6–8], and in various process control [9–11], and pattern recognition [12,13] applications. Besides the above-mentioned contributions, which mainly focus on the applications of neural networks, there has been a recent increase in the number of studies concerning the control-relevant properties of neural networks [14], as well as the improvement of network training by using different network structures, transfer functions and learning algorithms [6,15–17]. The performance of the neural network model strongly depends on its structure. Besides the aforementioned papers focusing on the applications of ANNs or the training of ANNs, there has been a recent increase in the number of studies concerning the improvement of network training and prediction by determining the optimal network topology [18].

Chemical processes in general, and biochemical fermentation systems in particular have strongly nonlinear features. Additionally, biochemical process models have a large number of parameters, which have to be determined experimentally [19,20]. Due to these features, on one hand the linear model approach is not appropriate for such processes, and on the other hand an accurate analytical model development can be very arduous. Consequently, neural networks represent a helpful tool in biochemical process modeling and control [21].

In this paper a detailed analytical model for a continuous fermentation reactor is presented first, which incorporates various nonlinear characteristics of the process, such as the oxygen mass transfer, detailed energy balance, complex reaction kinetics, the temperature dependence of the kinetic parameters as well as the effect of ionic strength and temperature on the mass transfer coefficient of oxygen. This is the first time when such a comprehensive yet control relevant model is derived for the fermentation bioreactor, which can be used as a process simulator to analyze or optimize the dynamic behavior of the system or for the design and evaluation of other control strategies.

The model is used in this work to simulate the real process and to generate the training data required for the neural network based empirical models. Neural networks with optimal structure were designed to model the dynamics of the process. It is well accepted that the prediction capability of the ANN based model highly depends on the number of neurons/connections used in its structure. Although an ANN with sufficiently large number of connections can learn any input–output data dependency, if the number of connections is too high the so called over-fitting phenomena occurs, characterized by a very low prediction quality. For the determination of the optimal network topology a novel pruning approach is proposed, which is based on a systematic optimal brain surgeon algorithm. The ANN model with the optimal structure is used then in the MPC algorithm as the internal model for prediction of the control movements. The performances of the Neural Network Model based Predictive Control (NNMPC) of the reactor temperature for the setpoint change were compared with those obtained with LMPC and proportional-integral-derivative (PID) control. The robustness

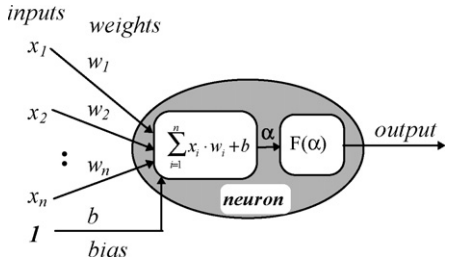

Fig. 1. An individual processing element of a neural net.

of the NNMPC against noisy temperature measurement is also evaluated. Results using Inverse Neural Network Model based Predictive Control (INNMPC) of the process are also presented. The paper provides the first comprehensive simulation study of the application of an optimally designed neural network model based predictive control algorithm to a fermentation bioreactor.

2. Artificial neural networks

A neural network is a computer program architecture for nonlinear computations, which is composed of many simple elements operating in parallel. These elements, called processing elements, are inspired by biological nervous systems, and they are highly interconnected. An individual processing element (neuron) can have any number of inputs, but only one output that is generally related to the inputs by a transfer function. The most frequently used transfer functions are: sigmoid function, hyperbolic tangent function, sine function, linear and saturated linear transfer function. The argument of the transfer function is the sum of the input elements of the corresponding neuron, each input being multiplied by the associated weight, which shows the strength of the connection between two neurons. A neuron usually has an additional input, called bias, which is much like a weight corresponding to a constant input of 1. Fig. 1 shows a schematic representation of an individual processing element (neuron).

The neurons are typically grouped into subsets, called layers, in which usually all the process units have the same bias and transfer function. Among the various architectures proposed for neural networks, the multilayer, feedforward network with Backpropagation Learning algorithm (BPN) has been used most frequently for dynamic modeling and process control applications. A typical feedforward neural network has one input layer (usually with the identity transfer function; thus it only

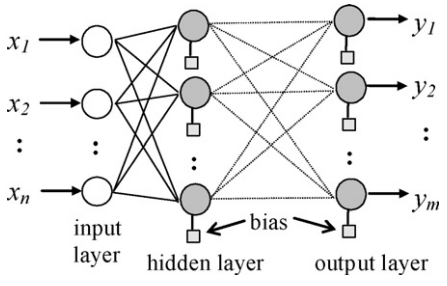

Fig. 2. The structure of a generic feedforward neural network.

distributes the inputs to the neurons from the next layer), one output layer and one or more hidden layers. The structure of a general feedforward neural network is presented in Fig. 2.

The outputs of the neurons from a layer represents the inputs for the next layer. The architecture of a network consists of a description of how many layers the network has, the number of neurons in each layer, the transfer function used in each layer and how the layers are connected to each other.

$$A_j = \begin{bmatrix} F_j\left(\sum_{i=1}^{S_{j-1}} w_j(1,i)A_{j-1}(i,1) + B_j(1)\right) \\ \vdots \\ F_j\left(\sum_{i=1}^{S_{j-1}} w_j(S_j,i)A_{j-1}(i,1) + B_j(S_j)\right) \end{bmatrix}$$

The most frequently used learning algorithm to train feedforward networks is the backpropagation learning algorithm. During the training phase the connection weights and biases are modified, using the backpropagation learning rules, so that the network will learn the process features.

The backpropagation learning algorithms belong to the class of supervised training algorithms, i.e., there is a set of input–output data, which is repeatedly presented to the network when the weights are adjusted in order to minimize the error between the net output (A_N) and the desired training output (D). The more the number of sets of input–output data (Q), the better the network will learn the process. Hence, for a network the training data can be represented by the following matrices:

$$P = \begin{bmatrix} P(1,1) & P(1,2) & \cdots & P(1,Q) \\ P(2,1) & P(2,2) & \cdots & P(2,Q) \\ & \vdots & & \\ P(S_0,1) & P(S_0,2) & \cdots & P(S_0,Q) \end{bmatrix} \quad (1)$$

$$P = \begin{bmatrix} D(1,1) & D(1,2) & \cdots & D(1,Q) \\ D(2,1) & D(2,2) & \cdots & D(2,Q) \\ & \vdots & & \\ D(S_N,1) & D(S_N,2) & \cdots & D(S_N,Q) \end{bmatrix} \quad (2)$$

The backpropagation algorithm can be summarized as follows:

- initialization of the weight coefficients with random values;
  `do`
  `for` (each training input–output pair)
- the input array is presented to the network and the activation flux is propagated layer by layer through the net (forward step);
- an error criterion is calculated and it is propagated back through the net adjusting the weights in order to minimize the error criterion (backward step).

`while` (error is above the error goal).

2.1. Forward step

In this step the output of the net is calculated. For a feedforward network with N layers (the input layer is not counted), with S_j neurons in the jth layer and with the same transfer function (F_j) in one layer the output of the jth layer can be computed with the following matrix with recurrent terms:

$$\begin{bmatrix} \cdots & F_j\left(\sum_{i=1}^{S_{j-1}} w_j(1,i)A_{j-1}(i,Q) + B_j(1)\right) \\ & \vdots \\ \cdots & F_j\left(\sum_{i=1}^{S_{j-1}} w_j(S_j,i)A_{j-1}(i,Q) + B_j(S_j)\right) \end{bmatrix} \quad (3)$$

with $F_0 = \Im$ (identity function: $\Im(x) = x$) and $A_0 = P$. For $j = N$ the output of the network is obtained. The transfer function used in our work was the sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

2.2. Backward step

The most frequently used error criterion, calculated in this step is the sum squared error of the network, defined as:

$$E = \sum_{i=1}^{S_N}\sum_{j=1}^{Q}(D(i,j) - A_N(i,j))^2 \quad (5)$$

The adjustment of the network weights and biases is done by continuously changing their values in the direction of steepest descent with respect to error. There are several improved methods to perform this step more efficiently. One of these algorithms is the backpropagation learning with momentum [22]. Momentum allows the network to ignore shallow local minimums in the error surface. Momentum (m) can be added to backpropagation learning by making weight changes equal to the sum of a fraction of the last weight change and the new change suggested by the backpropagation rule. This is expressed mathematically below:

$$w(i,j)^{(h)} = w(i,j)^{(h-1)} + l_r\,\delta w(i,j)^{(h)} \quad (6)$$

where

$$\delta w(i,j)^{(h)} = m\,\delta w(i,j)^{(h-1)} + (1-m)\frac{\partial E}{\partial w(i,j)} \quad (7)$$

These two steps (forward and backward) are repeated until the sum squared error (E) becomes less then the error goal.

3. First principle model of the continuous fermentation bioreactor

Alcoholic fermentation is one of the most important biochemical processes. The attention directed to this process has increased for the last two decades because its product, the ethanol, could represent an alternative energy source being used as a partial substitute for gasoline as a fuel. There are numerous

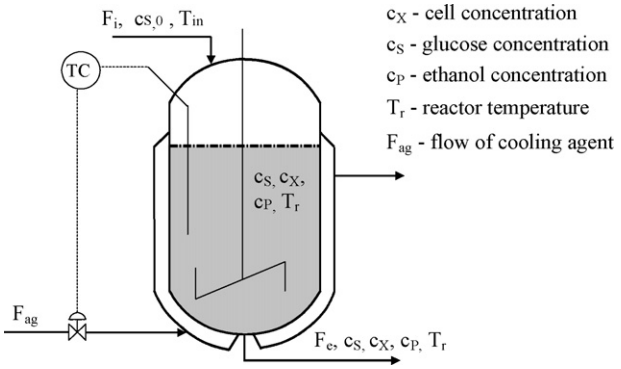

Fig. 3. The continuous fermentation reactor.

models of this process based on different kinetic considerations [23–25]. However, most of these models focus only on the kinetics of the process. The model presented below, used in simulations, besides the detailed kinetic model, involves equations, which express the heat transfer, the dependence of kinetic parameters on temperature, the mass transfer of oxygen, as well as the influence of temperature and ionic strength on the mass transfer coefficient. The kinetic equations used in the presented bioreactor model are modifications of the Monod equations based on the Michaelis–Menten kinetics, proposed by Aiba et al. [26]:

$$\frac{dc_X}{dt} = \mu_X c_X \frac{c_S}{K_S + c_S} e^{-K_P c_P} \tag{8}$$

$$\frac{dc_P}{dt} = \mu_P c_X \frac{c_S}{K_{S1} + c_S} e^{-K_{P1} c_P} \tag{9}$$

$$\frac{dc_S}{dt} = -\frac{1}{R_{SX}} \frac{dc_X}{dt} - \frac{1}{R_{SP}} \frac{dc_P}{dt} \tag{10}$$

where R_{SX} and R_{SP} are yield factors defined as the ratios of cell and ethanol produced per the corresponding amount of glucose used for growth or ethanol production, respectively. These equations express the production or consumption of the main components taking into account the inhibitory effect of ethanol. The continuous fermentation reactor is shown schematically in Fig. 3.

The reactor is modeled as a continuous stirred tank with constant substrate feed flow. There is also a constant outlet flow from the reactor that contains the product, substrate as well as biomass. The reactor contains three distinct main components: (i) the biomass, which is a suspension of yeast fed into the system and evacuated continuously, (ii) the substrate, which is solution of glucose, which feeds the micro-organism (*Saccharomyces cerevisiae*) and (iii) the product (ethanol), which is evacuated together with the other components. In order to have a quasi steady-state regarding the biomass, a low dilution rate (F_e/V) is necessary, that is, the dilution rate must not exceed the biomass production rate. Consequently, the process has a very slow dynamics. Together with the yeast, inorganic salts are added. These are necessary compounds for the formation of coenzymes. The inorganic salts due to the "salting-out" effect have also strong influence upon the equilibrium concentration of oxygen in the liquid phase. This influence of the dissolved inorganic salts as well as that of the temperature upon the equilibrium concentration of oxygen in the liquid phase are modeled in detail by Eqs. (11)–(29).

The mathematical model of the system is presented below:

The initial data of the system are:

- inorganic salts in the reaction medium:

$$m_{NaCl} = 500\,\text{g}$$

$$m_{CaCO_3} = 100\,\text{g}$$

$$m_{MgCl_2} = 100\,\text{g}$$

- the pH of the liquid phase:

$$\text{pH} = 6$$

- the inputs of the system:

$$F_i = F_e = 511\,\text{h}^{-1}$$

$$T_{in} = F_e = 25\,^\circ\text{C}$$

$$c_{S,in} = 60\,\text{g/l}$$

$$T_{in,ag} = 15\,^\circ\text{C}$$

Molar concentrations of ions in the reaction medium are calculated as follows, taking into account that the ion of Cl^- is present in two salts (NaCl and $MgCl_2$):

$$c_{Na} = \frac{m_{NaCl}}{M_{NaCl}} \frac{M_{Na}}{V} \tag{11}$$

$$c_{Ca} = \frac{m_{CaCO_3}}{M_{CaCO_3}} \frac{M_{Ca}}{V} \tag{12}$$

$$c_{Mg} = \frac{m_{MgCl_2}}{M_{MgCl_2}} \frac{M_{Mg}}{V} \tag{13}$$

$$c_{Cl} = \left[\frac{m_{NaCl}}{M_{NaCl}} + 2\frac{m_{MgCl_2}}{M_{MgCl_2}}\right] \frac{M_{Cl}}{V} \tag{14}$$

$$c_{CO_3} = \frac{m_{CaCO_3}}{M_{CaCO_3}} \frac{M_{CO_3}}{V} \tag{15}$$

$$c_H = 10^{-\text{pH}} \tag{16}$$

$$c_{OH} = 10^{-(14-\text{pH})} \tag{17}$$

The ionic strength of the ion i is calculated using Eq. (18):

$$I_i = \tfrac{1}{2} c_i z_i^2 \tag{18}$$

$$I_{Na} = 0.5 c_{Na} (1)^2 \tag{19}$$

$$I_{Ca} = 0.5 c_{Ca} (2)^2 \tag{20}$$

$$I_{Mg} = 0.5 c_{Mg} (2)^2 \tag{21}$$

Table 1
Parameters of the process model

$A_1 = 9.5 \times 10^8$	$H_{Cl} = 0.844$	$R_{SP} = 0.435$
$A_2 = 2.55 \times 10^{33}$	$H_{CO_3} = 0.485$	$R_{SX} = 0.607$
$A_T = 1\,m^2$	$H_{HO} = 0.941$	$V = 10001$
$C_{heat,ag} = 4.18\,J\,g^{-1}\,K^{-1}$	$(k_l a)_0 = 38\,h^{-1}$	$V_j = 501$
$C_{heat,r} = 4.18\,J\,g^{-1}\,K^{-1}$	$K_{O_2} = 8.86\,mg/l$	$Y_{O_2} = 0.970\,mg/mg$
$E_{a1} = 55{,}000\,J/mol$	$K_P = 0.139\,g/l$	$\Delta H_r = 518\,kJ/mol\,O_2$
$E_{a2} = 220{,}000\,J/mol$	$K_{P1} = 0.070\,g/l$	$\mu_{O_2} = 0.5\,h^{-1}$
$H_{Na} = -0.550$	$K_S = 1.030\,g/l$	$\mu_P = 1.790\,h^{-1}$
$H_{Ca} = -0.303$	$K_{S1} = 1.680\,g/l$	$\rho_{ag} = 1000\,g/l$
$H_{Mg} = -0.314$	$K_T = 3.6 \times 10^5\,J\,h^{-1}\,m^{-2}\,K^{-1}$	$\rho_r = 1080\,g/l$
$H_H = -0.774$		

$$I_{Cl} = 0.5 c_{Ca}(-1)^2 \tag{22}$$

$$I_{CO_3} = 0.5 c_{CO_3}(-2)^2 \tag{23}$$

$$I_H = 0.5 c_H(1)^2 \tag{24}$$

$$I_{OH} = 0.5 c_{OH}(-1)^2 \tag{25}$$

The global effect of the ionic strengths is given by Eq. (26):

$$\sum H_i I_i = H_{Na} I_{Na} + H_{Ca} I_{Ca} + H_{Mg} I_{Mg} + H_{Cl} I_{Cl} + \cdots + H_{CO_3} I_{CO_3} + H_H I_H + H_{OH} I_{OH} \tag{26}$$

The dependence of the equilibrium concentration of oxygen with temperature in distilled water is given by the below empirical equation obtained from the experimental data presented by Sevella [27]:

$$c^*_{O_2,0} = 14.6 - 0.3943 T_r + 0.007714 T_r^2 - 0.0000646 T_r^3 \tag{27}$$

Due to the fact that salts are dissolved in the medium the equilibrium concentration of oxygen in liquid phase is obtained from the following Setchenov type equation:

$$c^*_{O_2} = c^*_{O_2,0} \times 10^{-\sum H_i I_i} \tag{28}$$

Mass transfer coefficient for oxygen as temperature function is given by the following empirical equation [27]:

$$(k_l a) = (k_l a)_0 (1.024)^{T_r - 20} \tag{29}$$

The rate of oxygen consumption is:

$$r_{O_2} = \mu_{O_2} \frac{1}{Y_{O_2}} c_X \frac{c_{O_2}}{K_{O_2} + c_{O_2}} \tag{30}$$

The expression of the maximum specific growth rate (Eq. (31)) involves the resultant of the growth rate that increases with the temperature and the effect of the heat denaturation:

$$\mu_X = A_1 e^{-(E_{a1}/R(T_r+273))} - A_2 e^{-(E_{a2}/R(T_r+273))} \tag{31}$$

The balance for the total volume of the reaction medium is:

$$\frac{dV}{dt} = F_i - F_e \tag{32}$$

The mass balances for the biomass, product, substrate and dissolved oxygen are expressed by Eqs. (33)–(36):

$$\frac{dc_X}{dt} = \mu_X c_X \frac{c_S}{K_S + c_S} e^{-K_p c_p} - \frac{F_e}{V} c_X \tag{33}$$

$$\frac{dc_p}{dt} = \mu_P c_X \frac{c_S}{K_{S1} + c_S} e^{-K_{P1} c_P} - \frac{F_e}{V} c_P \tag{34}$$

The first terms in Eqs. (33) and (34) represent the quantity of biomass and product, respectively, produced in the fermentation reactions. The last terms describe the amount of yeast and ethanol, respectively, leaving the reactor.

$$\frac{dc_S}{dt} = -\frac{1}{R_{SX}} \mu_X c_X \frac{c_S}{K_S + c_S} e^{-K_P c_P} - \frac{1}{R_{SP}} \mu_P c_X \frac{c_S}{K_{S1} + c_S} \times e^{-K_{P1} c_P} + \frac{F_i}{V} c_{S,in} - \frac{F_e}{V} c_S \tag{35}$$

The first and second terms in Eq. (35) represent the amount of substrate consumed by the biomass for growth and ethanol production, respectively. The third term is the quantity of glucose entering the reactor with the fresh substrate feed, while the last term is the quantity of glucose leaving the reactor.

The concentration of the dissolved oxygen in the reaction medium is the resultant of the quantity of oxygen entering in the reaction medium due to the mass transfer, expressed by the first term in Eq. (36), and the amount consumed in the fermentation reactions (last term):

$$\frac{dc_{O_2}}{dt} = (k_l a)(c^*_{O_2} - c_{O_2}) - r_{O_2} \tag{36}$$

The energy balances for the reactor and jacket are given by Eqs. (37) and (38), respectively.

$$\frac{dT_r}{dt} = \frac{F_i}{V}(T_{in} + 273) - \frac{F_e}{V}(T_r + 273) + \frac{r_{O_2} \Delta H_r}{32 \rho_r C_{heat,r}} + \frac{K_T A_T (T_r - T_{ag})}{V \rho_r C_{heat,r}} \tag{37}$$

$$\frac{dT_{ag}}{dt} = \frac{F_{ag}}{V_j}(T_{in,ag} - T_{ag}) + \frac{K_T A_T (T_r - T_{ag})}{V_j \rho_{ag} C_{heat,ag}} \tag{38}$$

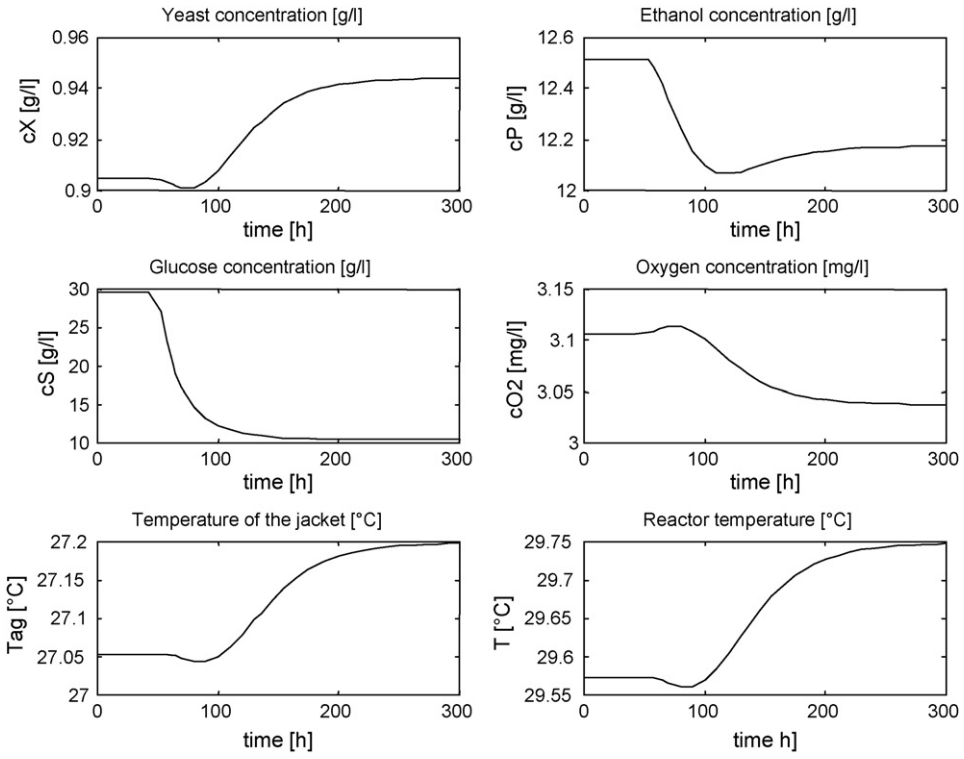

Fig. 4. Dynamic response of the system in the case of step change in the input substrate concentration (40 → 60 g/l).

The parameters of the model are presented in Table 1 [28].

The above-described model was used for the study of the dynamic behavior of the bioreactor in the case of different disturbances. The disturbances considered were: step change in the inlet flow temperature and in the substrate concentration. The first disturbance can occur due to the ambient temperature variation, while the second one because of quality changes of the

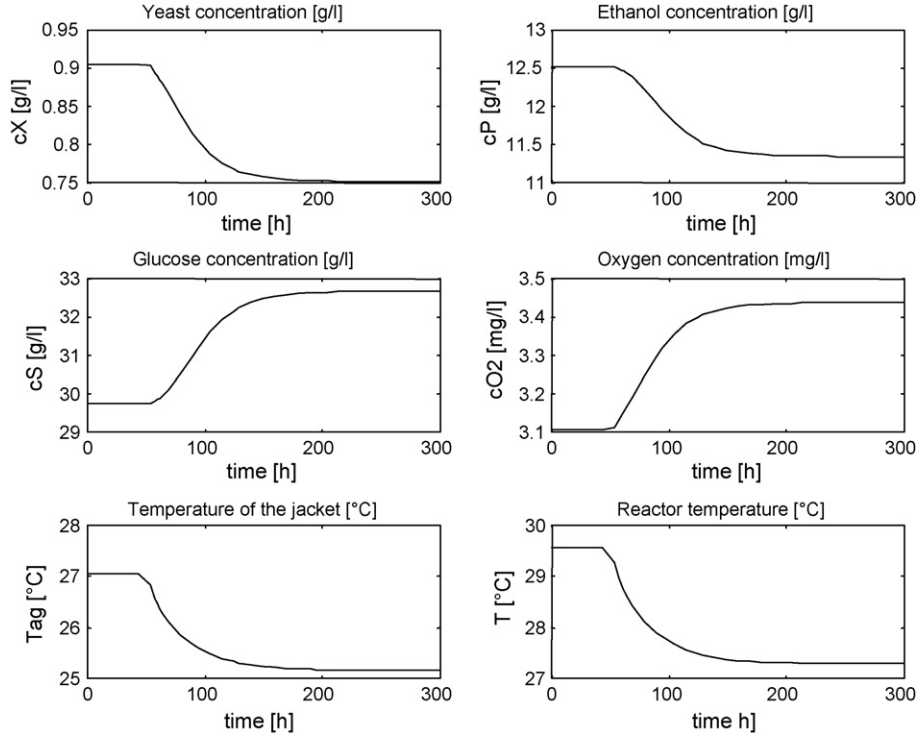

Fig. 5. Dynamic response of the system in the case of step change in the temperature of input flow (25 → 23 °C).

substrate flow. The model was implemented as a MATLAB S-function and the simulator is available upon request from the author.

The dynamic behavior of the system for the disturbances studied is shown in Figs. 4 and 5.

These figures show that variations in the input concentration have no significant effects on the ethanol concentration, therefore will not be considered as an important disturbance in this study. According to Fig. 5 the effect of the change in the inlet temperature is much more significant. A step of only 2 °C causes an important variation of the ethanol concentration.

The presented model can be a useful tool to test various control methods. In this work two artificial neural network model based NMPC control techniques are presented andassessed in comparison to PID and linear model based control.

4. Linear model identification of bioreactor

In order to evaluate, how well the linear approximation describes the process, two linear models were identified based on the simulated data obtained from the analytical model described in the previous section:

(1) *Linear state space model*, expressed by the equations below:

$$X(k+1) = \Phi X(k) + \Gamma U(k), \qquad Y(k) = CX(k) \tag{39}$$

where Φ, Γ and C are discrete state space matrices for the corresponding sampling time, $X(k)$ the state vector, $Y(k)$ the output vector of the linear model and $U(k)$ is the vector of the manipulated variables at moment k. With this model the process nonlinearity is demonstrated by the simulation results presented in Fig. 6. The procedure of obtaining the plotted data in Fig. 6 was the following: at the steady-state operating point (where $F_{ag} = 18\,\mathrm{l\,h^{-1}}$ and $T_r \cong 30\,^{\circ}\mathrm{C}$) a sequence of step inputs (ΔF_{ag}) was given. The changes in output (ΔT_r) after one sampling period are plotted showing the clear difference between the nonlinear model (represented by circles) and the linearized one (solid line).

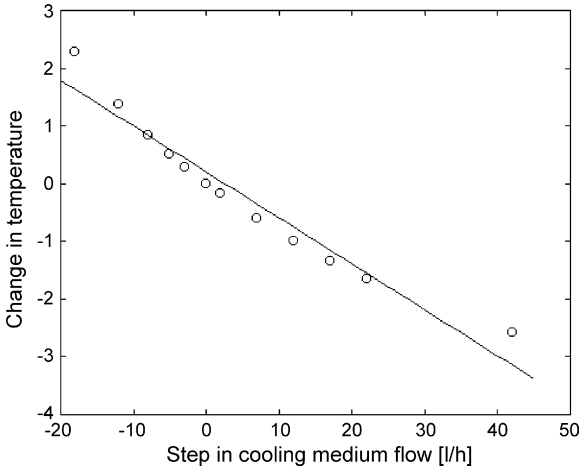

Fig. 6. Process nonlinearity. The solid line indicates the temperature changes for a linear approximation of the process.

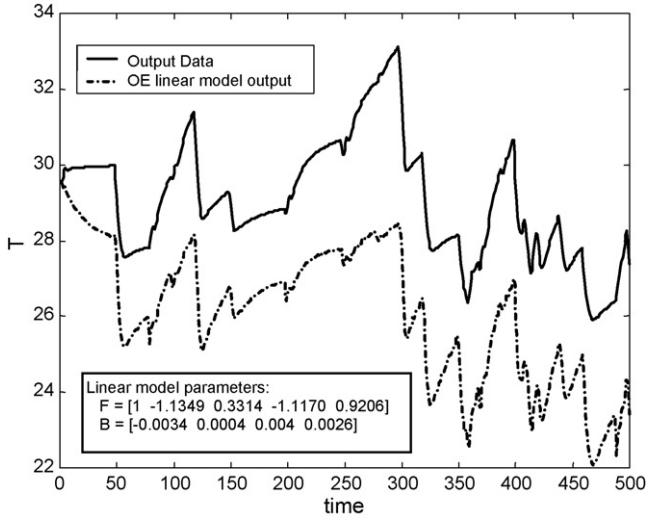

Fig. 7. OE linear model prediction.

(2) *Output-Error* (OE) model, from the polynomial linear model category was also identified using the simulated input–output data pairs:

$$\begin{aligned} y(k) = &-f_1(y(k-1) - \mathrm{e}(k-1)) - f_2(y(k-2) \\ &-\mathrm{e}(k-2)) - \cdots - f_{nf}(y(k-nf) - \mathrm{e}(k-nf)) \\ &+\cdots + b_1 u(k-nk) + b_2 u(k-nk-1) \\ &+\cdots + b_{nb} u(k-nk-nb+1) + \mathrm{e}(k) \end{aligned} \tag{40}$$

where $f_1, f_2, \ldots, f_{nf}, b_1, b_2, \ldots, b_{nb}$, are the coefficients of the model. The structure of the model is defined by giving the time delay nk, and the order of the polynomials nf and nb, respectively. In order to assure a sufficient complexity of the model, a structure with parameters $nk = 1$, $nf = nb = 4$, was identified. Fig. 7 shows that the obtained linear model is not able to model accurately the process. Although the overall dynamic characteristics are captured, and a proper feedback model/plant correction can be used to minimize modeling errors proper nonlinear model is needed for accurate control as it will be shown in the following sections.

5. Artificial Neural Network based dynamic model and control of the bioreactor

5.1. Identification of the ANN model of the bioreactor

In this part of the work the primary goal was to obtain a dynamic ANN model, which describes the variations of the reactor temperature (y) as a function of the cooling agent flow (u). For this, a random input signal was generated and applied to the system. The simulated response of the system together with the random input signal was used to train the ANN. Once the ANN model is identified, it can be used as an internal model in an advanced nonlinear model predictive control algorithm. For this, it is crucial to have a network with very good generalization properties. One way to obtain a network with appropriate generalization properties is to choose a structure with sufficient

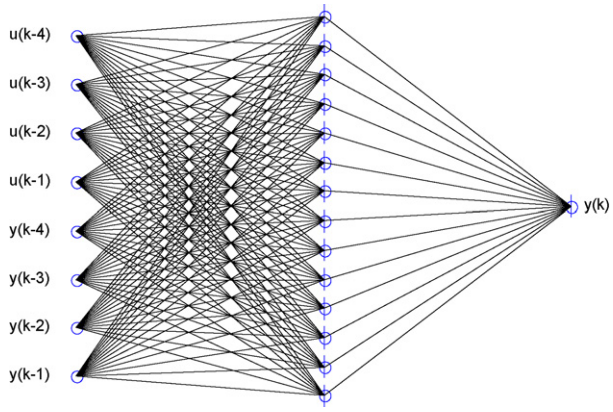

Fig. 8. The initial structure of the ANN.

number of parameters, which assure that the ANN learns the training data and then optimize the topology of the network until the best generalization properties are achieved. A feedforward neural network, with the same *nk*, *nf* and *nb* parameters as in the case of the linear OE model was chosen. Thus, in the input layer the network has 8 neurones and 1 neurone in the output layer, with linear transfer function. One hidden layer with 14 neurones with the hyperbolic tangent sigmoid transfer function was used. The fully connected initial network is presented in Fig. 8. Fig. 9 shows the input–output data sequence that was used for training the network. Figs. 10 and 11 demonstrate that the network was able to learn the training data with a very good accuracy.

In order to test the ANN capability of generalization, other random input sequence was obtained by simulation, from the first principle model of the system. The test data set is presented in Fig. 12. Figs. 13 and 14 demonstrate the very poor generalization performance of the ANN model. In this case very high prediction errors were obtained. By comparing the plots for training and test set, it is quite obvious that the reason of the poor generalization is the overfitting of the data. It is concluded therefore that the model structure selected contains too many neurons (weights). Consequently, for improving the generalization performance of the ANN model, it is necessary to remove the superfluous weights from the network.

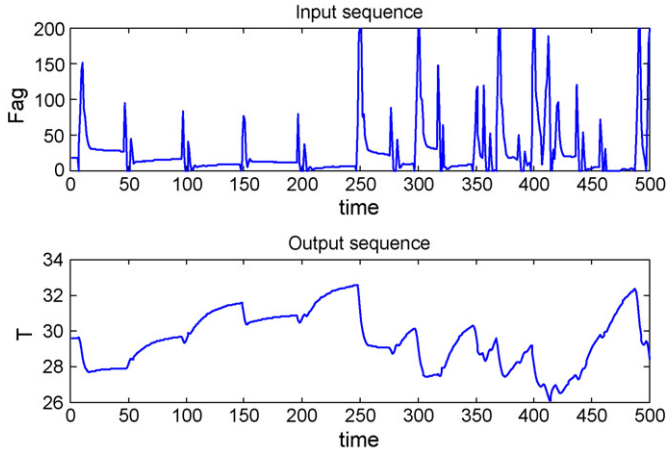

Fig. 9. The training data.

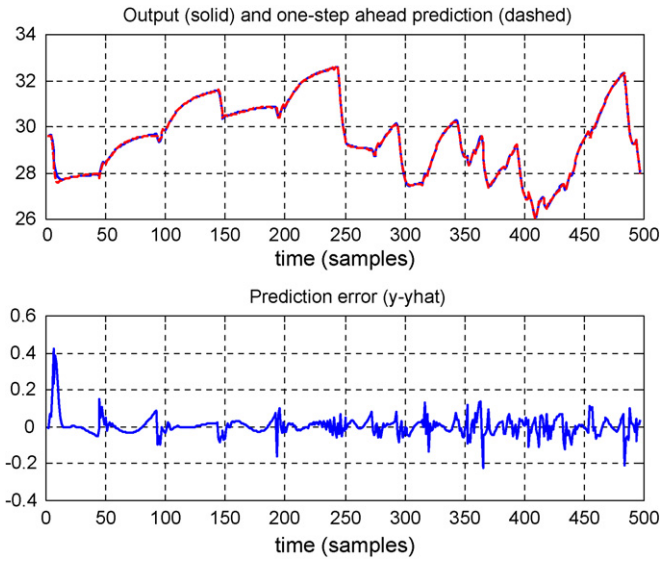

Fig. 10. ANN prediction and prediction error for the training data.

5.2. Determination of optimal topology of the ANN

One of the most important parameters of the ANN is the number of connections among the neurons. As it can be seen on the simulation results presented in the previous section, this parameter determines the learning and especially the generalization performances of the ANN. The so-called Optimal Brain Surgeon (OBS) is one of the most important strategy for pruning neural networks. This algorithm determines the optimal network architecture by removing the superfluous weights from the network in order to avoid the overfitting of the data by the ANN. A variant of the OBS algorithm proposed by Hansen and Pedersen

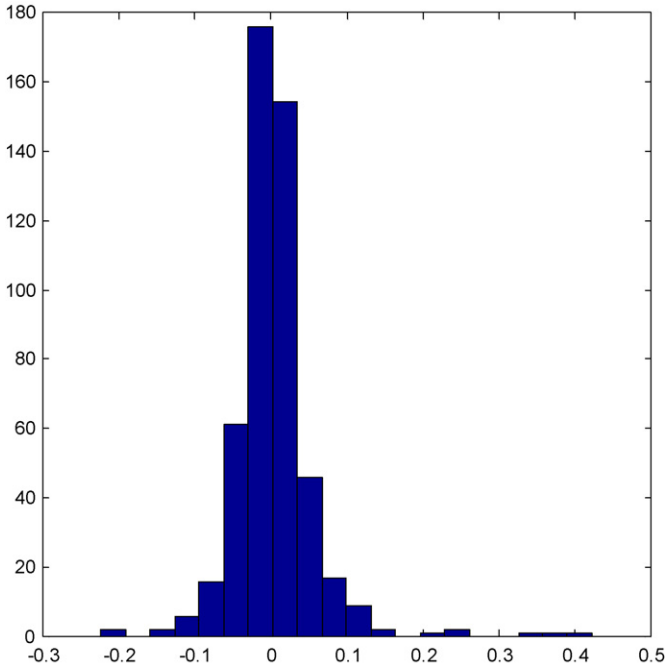

Fig. 11. Histogram of prediction errors for the training data.

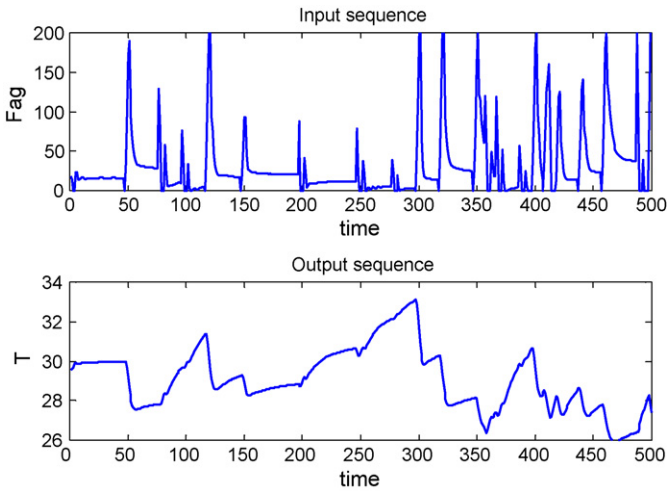

Fig. 12. Data set to test the ANN model.

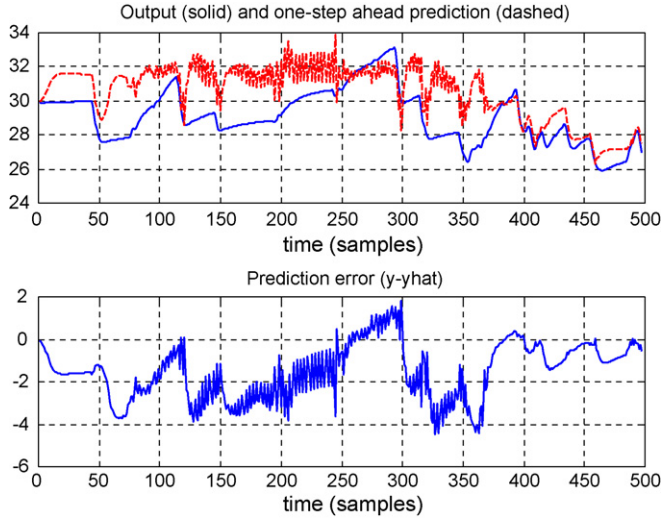

Fig. 13. ANN prediction and prediction error for the testing data.

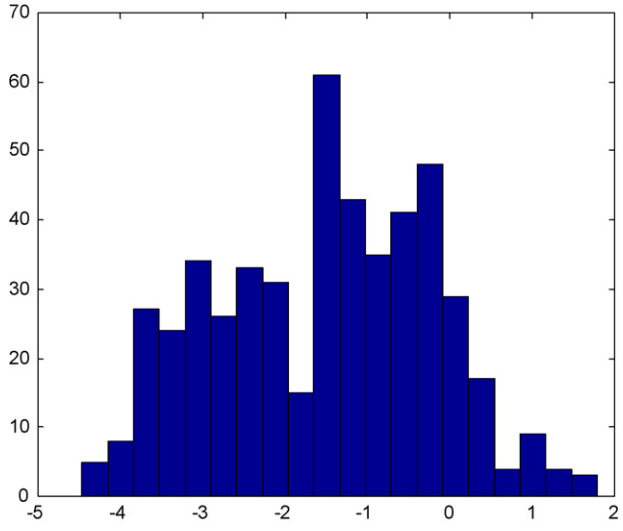

Fig. 14. Histogram of prediction errors for the testing data.

[29] was developed and implemented with the modification to take into account that it should not be possible to have networks where a hidden unit has lost all the weights leading to it, while there still are weights connecting it to the output layer, or vice versa.

In this algorithm a saliency is defined as the estimated increase of the unregularized error criterion, when a weight is eliminated. The saliency for weight j is defined by:

$$\xi_j = \lambda_j I_j^T \tilde{H}^{-1}(\theta^*)\frac{1}{N}Q_r\theta^* + \frac{1}{2}\lambda_j^2 I_j^T \tilde{H}^{-1}(\theta^*)H(\theta^*)\tilde{H}^{-1}(\theta^*)I_j \tag{41}$$

where θ^* is a vector with all the weights and biases of the reduced network and I_j is the jth unit vector. The Gauss–Newton Hessian of the regularized criterion is calculated with the equation:

$$\tilde{H}(\theta^*) = H(\theta^*) + \frac{1}{N}Q_r \tag{42}$$

where H is the Hessian of the unregularized error criterion, and Q_r is the regularization matrix. The Lagrange multipliers λ_j are calculated from the following equation:

$$\lambda_j = \frac{\theta_j^*}{\tilde{H}_{j,j}^{-1}(\theta^*)} \tag{43}$$

The constrained minimum (the minimum when weight j is 0) is then found from:

$$\delta\theta = \theta^* - \theta = -\lambda_j \tilde{H}^{-1}(\theta^*)I_j \tag{44}$$

In the beginning, the saliencies are calculated and the weights are pruned as described above. However, when a situation occurs where a unit has only one weight leading to or one weight leading from it, the saliency for removing the entire unit is calculated instead, by setting all weights connected to the unit to zero. With the proposed enhanced OBS algorithm the computational time necessary to obtain the optimal topology was reduced in some

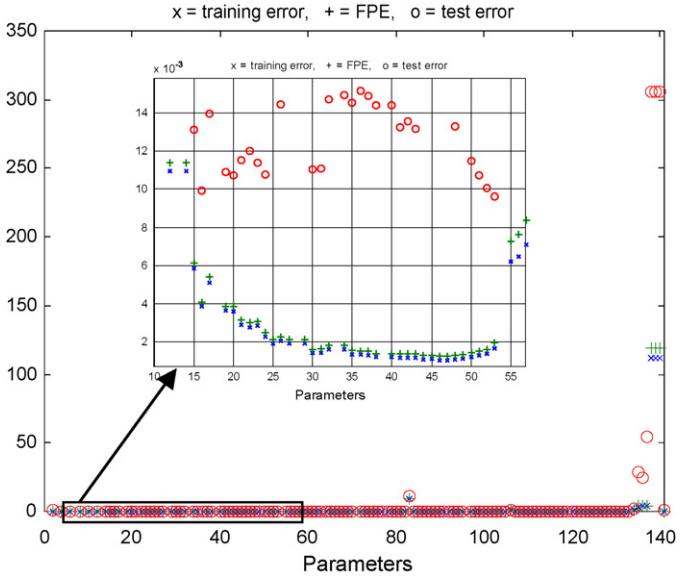

Fig. 15. Results of pruning the ANN with OBS.

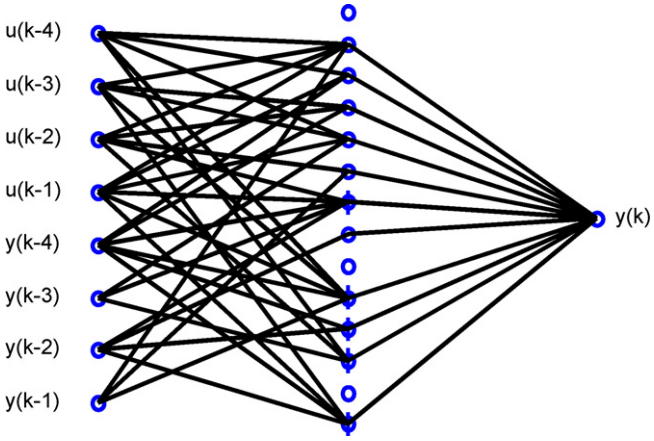

Fig. 16. The architecture of the optimal ANN (with 53 weights) obtained by the OBS algorithm.

cases with 30%. The network can be retrained after each weight elimination or after a certain percent of the weights was eliminated. The error criterion used by the algorithm is calculated for the test data set. This algorithm was implemented in MATLAB and successfully applied to the above-obtained ANN models.

The fully connected feedforward ANN, used in our simulations, contains a total number of 141 parameters (weights and biases). The OBS algorithm was used in order to prune the network. After each weight elimination the network was retrained for 50 iterations. Fig. 15 presents the results obtained by the OBS algorithm. In this figure the error criteria for both the training data and testing data together with the final prediction error (FPE) is presented. The FPE is estimated from the training set and is very useful when a test set is not available. The test error is the most reliable estimate of the generalization error; therefore, the OBS algorithm selects the network with the smallest test error. The OBS algorithm gave as the final network the one with 53 weights (a reduction with 88 weights, i.e., 62%). The architecture of the selected network is presented in Fig. 16. A considerable reduction of the network structure was achieved. The number of the weights was reduced with 62%. Three of the

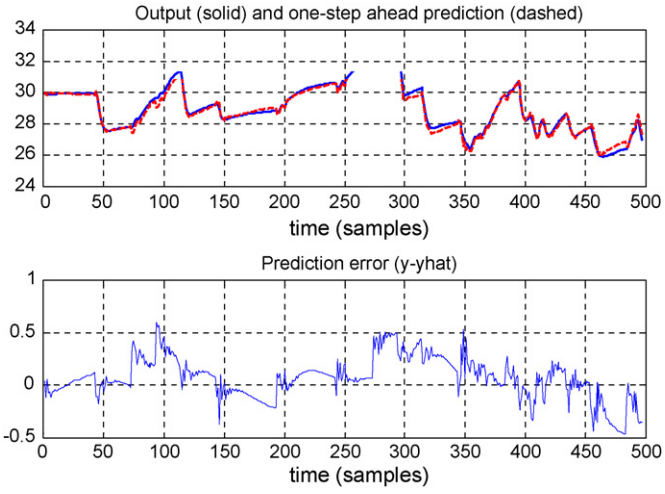

Fig. 17. Generalization performances of the reduced ANN (with 53 weights).

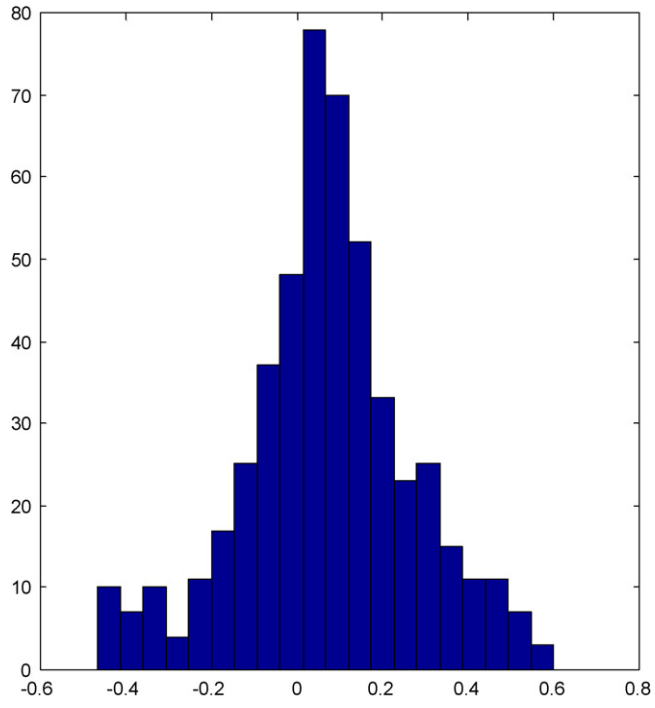

Fig. 18. Histogram of prediction errors for the testing data (ANN with 53 weights).

neurons from the hidden layer were completely eliminated. The performances on the testing data (Figs. 17 and 18), obtained with the reduced ANN are significantly better than in the case of the original structure.

Studying further the results obtained with the OBS algorithm, presented in Fig. 15, one can observe that the network with the second best test error has a much simpler architecture (only 16 weights). The test error is very close to the value obtained with the ANN with 53 weights, but the topology is reduced additionally with 37 weights. The structure of this ANN is presented in Fig. 19. This network suffered a considerably simplification of its structure. A number of 125 weights were removed from the total number of 141 weights (a reduction of 88%). From the input

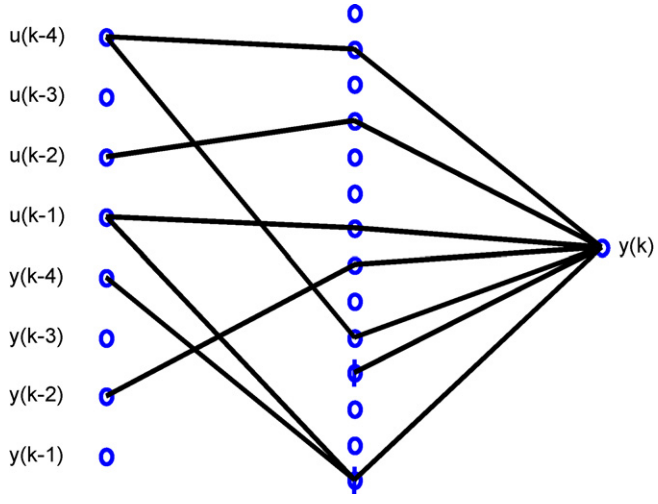

Fig. 19. The architecture of the ANN with 16 weights.

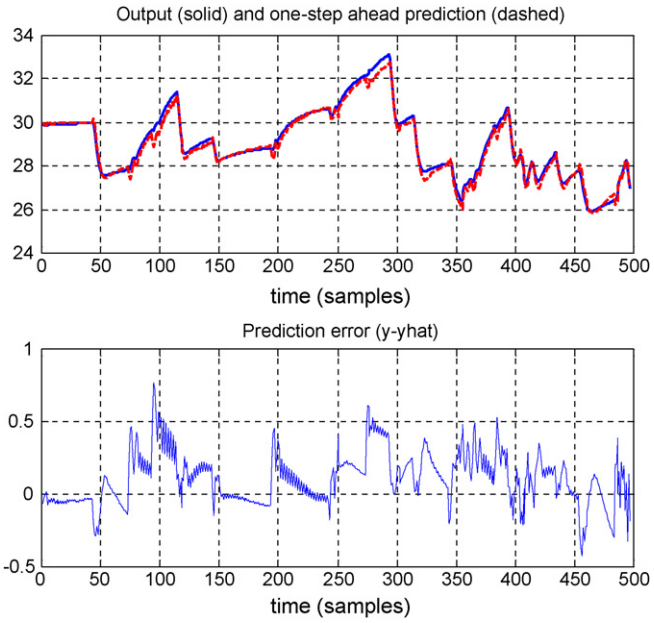

Fig. 20. Generalization performances of the reduced ANN (with 16 weights).

layer 3, and from the hidden layer 7 neurons were eliminated completely. This structure shows that for an accurate modeling of the dynamic behavior of the system it is not necessary to use the output measurements with 1 and 3 steps in the past and the input value with 3 sampling time in the past.

Despite the considerable reduction of the complexity of the network structure this simple structure has very good generalization properties, as can be seen in Fig. 20. The simple structure also facilitates the fast data processing. Consequently, the pruned networks can be used in different nonlinear model predictive or optimal control algorithms as the internal model for prediction.

5.3. ANN model based nonlinear predictive control of the bioreactor

Once the ANN model is identified and the structure with the best generalization properties is selected, it can be used in different NMPC algorithms. In NMPC usually multi-step-ahead prediction is needed to foresee the behavior of the process in advance in the future. An ANN model with one-step-ahead prediction can be used repeatedly or another structure with more then one future output parameter in the output layer can be identified. The latter one has the advantage of faster computation of the predicted values but in this case the prediction horizon usually is fixed when the network structure is chosen and for different prediction horizons different networks with different structures have to be trained.

The network with the optimal topology (Fig. 16) was introduced in a model predictive control scheme as the internal model used for prediction during the control movement calculation. The neural network model based predictive control structure is presented in Fig. 21.

In each sampling period the current temperature measurement is obtained ($T_r(k)$), and considering that the past temperature measurements and control actions are known, the next control action is calculated by solving an optimization problem. The next control action is selected such that the predicted outcome of the control action is optimum in the sense of minimizing the square of the deviation from the setpoint trajectory over a finite horizon (P). Prediction over the horizon P is achieved by repeatedly applying the ANN model. Consequently, the optimization problem for this particular case can be formulated as follows:

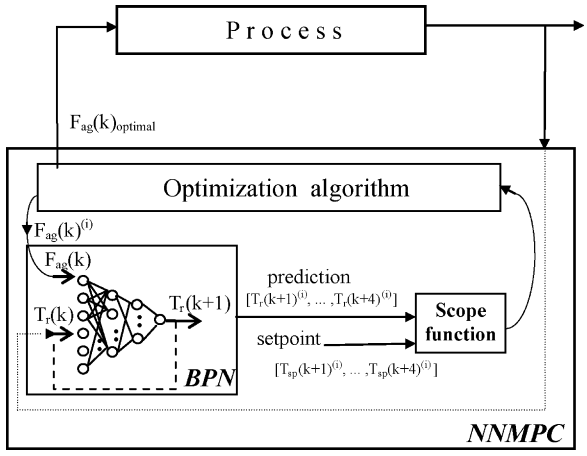

Fig. 21. Block diagram of NNMPC of the process.

$$\min_{F_{ag}(k)} \left\{ \sum_{i=1}^{P} [T_r(k+i) - T_{sp}(k+i)]^2 \right\} \tag{45}$$

where:

$$T_r(k+i)|_{i=\overline{1,P}} = f_{NN}(F_{ag}(k), F_{ag}(k-1), F_{ag}(k-2), T_r(k), \times T_r(k-1), T_r(k-2)) \tag{46}$$

With this control structure, an excellent control of the process was achieved. Here a control horizon $P=4$ was used. For comparison the PID and LMPC control of the process are also presented in Fig. 22. The superiority of NNMPC can be clearly seen.

The robustness of the NNMPC structure was studied in the case of noisy temperature measurement. The amplitude of the white noise considered in the simulation was 1.5 °C. To make the network capable of controlling the process in the case of noisy temperature measurement, the ANN model was trained including noisy data into the training set. If $\mathbf{I}$ and $\mathbf{D}$ are the training input and output, respectively, obtained from the analytical model of the process, the training set for training the net with noise can be constructed as follows:

$$\text{-input data}: \quad \mathbf{I}^* = [\mathbf{P}\,\mathbf{P} + \text{noise}] \tag{47}$$

$$\text{-corresponding target data}: \quad \mathbf{D}^* = [\mathbf{D} \quad \mathbf{D}]^* \tag{48}$$

Once the network had been trained with these training data it was used in the above-described NNMPC scheme. The scenario considered in this study is the case when the measurement is subject to high frequency noise (e.g., due to magnetic fields that would affect the temperature sensor, or variations in the hydrodynamics

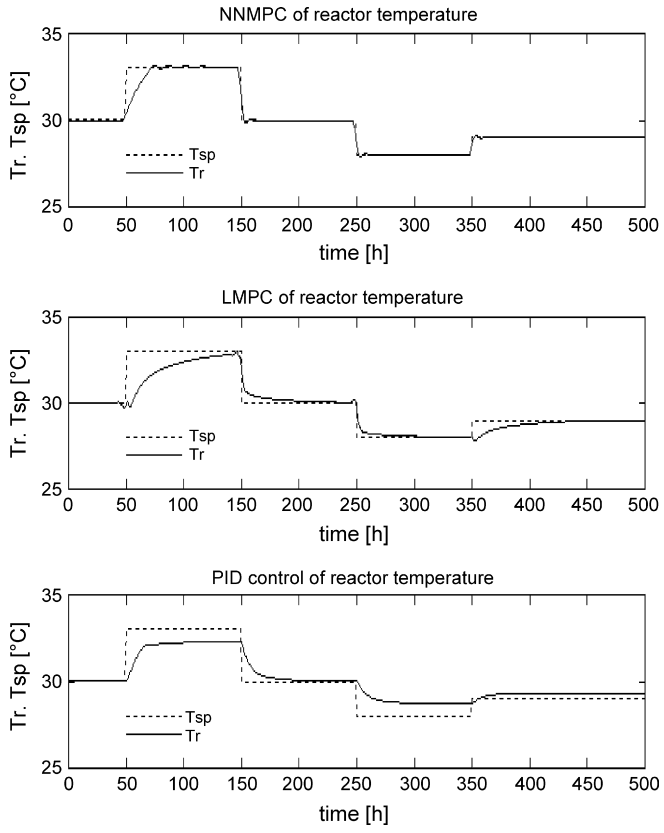

Fig. 22. Simulation results with ANMPC, LMPC and PID control of the process.

in the reactor due to nonideal agitation). In this case a lowpass filter could be used to obtain the non-noisy measurements required for the training of the network. The results obtained in the case of noisy temperature measurement are shown in Fig. 23. It can be seen that a fairly good control was achieved. In this case, PID

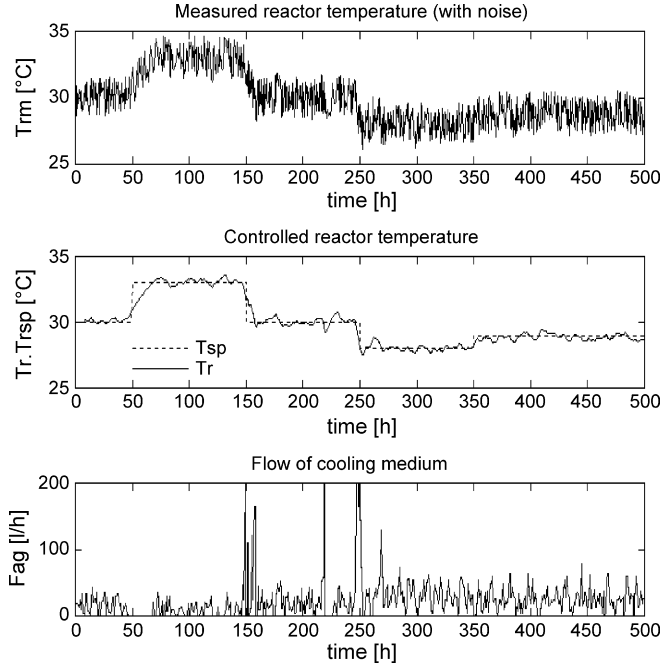

Fig. 23. NNMPC of the process with noisy temperature measurement.

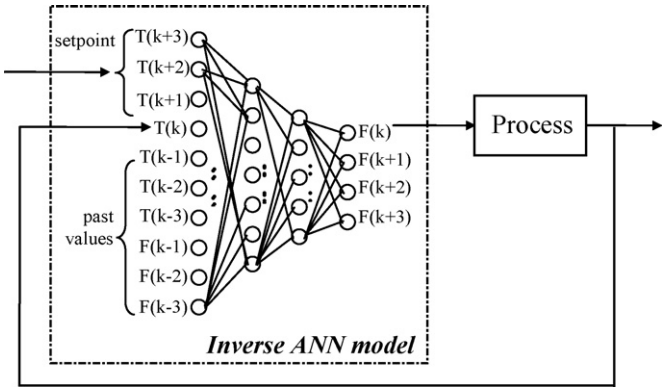

Fig. 24. Block diagram of inverse-ANN based control of the process.

control gave worse results than NNMPC and LMPC fails for all filter designs tried.

As an alternative way to use neural networks for process control, the use of an inverse neural network model was also considered. In the case of inverse neural network models, the outputs of the network correspond to the future values of the process inputs while the input layer of the net contains, besides the past values of the process inputs and outputs, and the current process output measurement, also the future values of controlled variables (process outputs). The inverse neural network due to its structure eliminates the optimization algorithm from the control movement computation. Using the past values of the controlled and manipulated variables as well as the current measurement, the control movement can be directly obtained from the net when the setpoint values are presented to the network as the future values of the controlled variables. The block diagram of the inverse neural network model based predictive control (INNMPC) of the process is presented in Fig. 24.

The network used in these simulations had two hidden layers (with 7 and 5 neurons, respectively), 10 neurons in the input layer and 4 neurons in the output layer. The ANN model can be represented as follows:

$$[F_{ag}(k),\ F_{ag}(k+1),\ F_{ag}(k+2),\ F_{ag}(k+3)] = f_{\text{inv}NN}\,(T_r(k+3), T_r(k+2)\ , T_r(k+1), \ldots, T_r(k), T_r(k-1), T_r(k-2), T_r(k-3), F_{ag}(k-1), F_{ag}(k-2), F_{ag}(k-3)) \quad (49)$$

The network was trained using a historical database obtained from the analytical model. In the training phase the future values of the temperature ($T_r(k+\mathrm{i})$) are known. After the network had been trained, it was used for control, when in each sampling time, for the future temperature inputs of the network the setpoint values were used:

$$T_r(k+i)|_{i=\overline{1,3}} = T_{sp}(k+i)|_{i=\overline{1,3}} \quad (50)$$

Fig. 25 shows that a very good control performance was achieved with this control structure. Note that both control algorithms are designed for setpoint tracking therefore the current formulation would result in a bias in the controlled output in the case of unmeasured disturbances. However, the bias can be easily

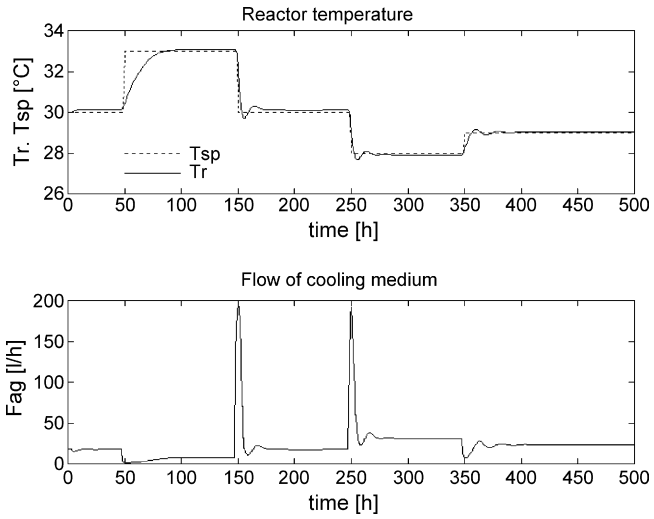

Fig. 25. Inverse-NNMPC of the process.

eliminated with a simple error feedback loop in the control architecture, that would provide the necessary integral action to the controller.

6. Conclusions

The paper successfully demonstrates the ability of artificial neural networks to model complex nonlinear biochemical processes, such as the alcoholic fermentation. The detailed analytical model of the continuous fermentation bioreactor is presented. This model is more complex than those used generally to test different control systems involving more nonlinear characteristics of the process. Therefore it can be a useful tool to test various nonlinear control methods. Using the data obtained from the analytical model, artificial neural network based models were also developed. An efficient new algorithm, the enhanced Optimal Brain Surgeon, is also presented as a pruning algorithm for the determination of the optimal ANN topology. With the OBS algorithm a reduction of the number of weights from 141 to 53 (62%) in the first step, and finally to 16 (88%) was achieved. Simulation results are presented to demonstrate that this very simple network structure can achieve a better generalization than the initial, fully connected structure. The pruned networks have very good generalization properties. The simple structure also facilitates the fast data processing. Therefore, the pruned networks can be used in different nonlinear model predictive algorithms as the internal model used for prediction.

Two ANN model based control schemes are presented and tested via simulations. The results were compared with those obtained with linear MPC and PID control. The superiority of the NNMPC structure was demonstrated. The main advantage of the NNMPC compared with the NMPC, which uses the analytical model of the system, is that the former does not need detailed knowledge about the process, which is a feature that might be of crucial importance in the case of complex biochemical processes. The nonlinear model used in the NNMPC can be obtained from experimental input–output data without the modeling burden required by the derivation of the analytical model. The simulations presented also demonstrate how neural networks can be trained and used for nonlinear model predictive control of a process when measurements are affected by noise. Additionally, the development and application of a predictive control scheme based on the inverse neural network model of the process is also illustrated. The main advantage of this control structure is that it needs a very simple mathematical apparatus for the control movement calculation. This algorithm is no longer iterative thus the required computational time is very short, making it preferable for real-time applications.

References

[1] C.E. Garcia, D.M. Prett, M. Morari, Model predictive control: theory and practice—a survey, Automatica 25 (1989) 335–348.
[2] K.J. Hunt, D. Sbarbaro, R. Zbikowski, P.J. Gawthrop, Neural networks for control systems—a survey, Automatica 28 (6) (1992) 1083–1112.
[3] C. Aldrich, J. van Deventer, Comparison of different artificial neural nets for the detection and location of gross errors in process systems, Ind. Eng. Chem. Res. 34 (1) (1995) 216–224.
[4] L.H. Ungar, B. Powel, S.N. Kamens, Adaptive network for fault diagnosis and process control, Comput. Chem. Eng. 14 (1990) 561–572.
[5] D.R. Baughman, Y.A. Liu, An expert network for predictive modeling and optimal design of extractive bioseparations in aqueous two-phase systems, Ind. Eng. Chem. Res. 33 (11) (1994) 2668–2687.
[6] J.F. Pollard, M.R. Broussard, D.B. Garrison, K.Y. San, Process identification using neural networks, Comput. Chem. Eng. 16 (4) (1992) 253–270.
[7] N.V. Bhat, T.J. McAvoy, Use of neural nets for dynamic modeling and control of chemical process systems, Comput. Chem. Eng. 14 (4–5) (1990) 573–585.
[8] Y. Cheng, T.W. Karjala, D.M. Himmelblau, Identification of nonlinear dynamic processes with unknown and variable dead time using an internal recurrent neural network, Ind. Eng. Chem. Res. 34 (5) (1995) 1735–1742.
[9] B. Joseph, F.W. Hanratty, Predictive control of quality in a batch manufacturing process using artificial neural network models, Ind. Eng. Chem. Res. 32 (9) (1993) 1951–1961.
[10] J.C. Hoskins, D.M. Himmelblau, Process control via artificial neural networks and reinforcement learning, Comput. Chem. Eng. 16 (4) (1992) 241–251.
[11] M.J. Willis, G.A. Montague, C. Di Massimo, M.T. Tham, A.J. Morris, Artificial neural networks in process estimation and control, Automatica 28 (6) (1992) 1181–1187.
[12] J.R. Whiteley, J.F. Davis, Knowledge-based interpretation of sensor patterns, Comput. Chem. Eng. 16 (4) (1992) 329–346.
[13] S.N. Kavuri, V. Venkatasubramanian, Combining pattern classification and assumption-based techniques for process fault diagnosis, Comput. Chem. Eng. 16 (4) (1992) 299–312.
[14] E. Hernandez, Y. Arkun, Study of the control-relevant properties of backpropagation neural network models of nonlinear dynamic systems, Comput. Chem. Eng. 16 (4) (1992) 227–240.
[15] J. Chen, D.P. Bruns, WaveARX neural network development for system identification using a systematic design synthesis, Ind. Eng. Chem. Res. 34 (12) (1995) 4420–4435.
[16] M.A. Kramer, Autoassociative neural networks, Comput. Chem. Eng. 16 (4) (1992) 313–328.
[17] M.L. Mavrovouniotis, S. Chang, Hierarchical neural networks, Comput. Chem. Eng. 16 (4) (1992) 347–369.
[18] N.V. Bhat, T.J. McAvoy, Determining model structure for neural models by network stripping, Comput. Chem. Eng. 16 (4) (1992) 271–281.
[19] F.-S. Wang, J.-W. Sheu, Multiobjective parameter estimation problems of fermentation processes using a high ethanol tolerance yeast, Chem. Eng. Sci. 55 (2000) 3685–3695.

[20] G.-Y. Zhu, A. Zamamiri, M.A. Henson, M.A. Hjorts, Model predictive control of continuous yeast bioreactors using cell population balance models, Chem. Eng. Sci. 55 (2000) 6155–6167.
[21] C. Di Massimo, G.A. Montague, M.J. Willis, M.T. Tham, A.J. Morris, Towards improved penicillin fermentation via artificial neural networks, Comput. Chem. Eng. 16 (4) (1992) 283–291.
[22] H. Demuth, M. Beale, Neural Network Toolbox User's Guide, The MathWorks Inc., 1993.
[23] J.A. Roels, Mathematical models and the design of biochemical reactors, J. Chem. Technol. Biotechnol. 32 (1982) 59–72.
[24] B. Volesky, L. Yerushalmi, J.H.T. Luong, Metabolic-heat relation for aerobic yeast respiration and fermentation, J. Chem. Technol. Biotechnol. 32 (1982) 650–659.
[25] G.H. Cho, C.Y. Choi, Y.D. Choi, M.H. Han, Ethanol production by immobilised yeast and its CO_2 gas effects in a packed bed reactor, J. Chem. Technol. Biotechnol. 32 (1982) 959–967.
[26] S. Aiba, M. Shoda, M. Nagatani, Kinetics of product inhibition in alcoholic fermentation, Biotechnol. Bioeng. 10 (1968) 846–864.
[27] B. Sevella, Bioengeneering Operations, Technical University of Budapest, Tankonykiado, Budapest, 1992.
[28] F. Godia, C. Casas, C. Sola, Batch alcoholic fermentation modelling by simultaneous integration of growth and fermentation equations, J. Chem. Technol. Biotechnol. 41 (1988) 155–165.
[29] L.K. Hansen, M.W. Pedersen, Controlled growth of cascade correlation nets, in: M. Marino, P.G. Morasso (Eds.), Proceedings ICANN'94, Sorrento, Italy, 1994, pp. 797–800.