Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Chemical Engineering 28 (2004) 1499–1509
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

# On-line tuning of a neural PID controller based on plant hybrid modeling

A. Andrášik[a], A. Mészáros[a,*], S. F. de Azevedo[b]

[a] Faculty of Chemical and Food Technology, Slovak University of Technology, Radlinského 9, 81237 Bratislava, Slovak Republic
[b] Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

Received 17 April 2003; received in revised form 7 November 2003; accepted 8 December 2003

**Abstract**

In this paper, a new control technique for nonlinear control based on hybrid neural modeling is proposed. For neural network training, a variant of the well-known gradient steepest descent method is employed where the learning rate is adapted in each iteration step in order to accelerate the speed of convergence. It is shown that appropriate selection of the learning rate results in stable training in Lyapunov sense. The closed-loop control system consists of two neural networks. The first one is a feedforward neural network that is employed as a predictive hybrid model of the controlled plant. The second network is a neural PID-like controller, which has been pre-trained off-line as an inverse black-box model of the controlled process. To ensure offset-free performance, additional on-line tuning of the neural controller is required, especially in the presence of process uncertainties and time-varying parameters. Advantages of the proposed technique are demonstrated through simulation experiments for a case study, investigating the control of a continuous flow stirred biochemical reactor.
© 2003 Elsevier Ltd. All rights reserved.

*Keywords:* Neural control; Hybrid modeling; Fermentation process

## 1. Introduction

Many industrially important fermentation processes exhibit nonlinear and time-varying behavior. Moreover, since the cultivation medium contains living microorganisms, the application of modern control techniques is often hampered by such effects as substrate inhibition, catabolite repression, product inhibition, glucose effect and auto-trophic mutation. Therefore, design of a control strategy that should provide a required control performance of key variables, such as biomass concentration, may be a difficult task.

Successful control requires good knowledge of the controlled process in form of an adequate mathematical model. A considerable amount of both mechanistic and empirical mathematical models describing fermentation processes has been proposed in past decades (e.g. Enfors, Hedenberg, & Olsson, 1990; Castrillo & Ugalde, 1994; Pham, Larsson, & Enfors, 1998). The main disadvantage of these models is their complexity, especially when kinetic, thermodynamic and physical mechanisms are described. Moreover, applications of these models are limited to certain neighborhood of operation point, corresponding to specific fermentation conditions. Application of empirical mathematical models may be further limited by technical factors such as lack of on-line sensors for substrates, biomass and products.

Methods of artificial intelligence have offered new, effective tools of system modeling. Methods based on artificial neural networks (ANN) have strongly established themselves in this field, during the past decade. It has been shown that any continuous function can be approximated as precisely as required by a neural network having enough neurons, at least one hidden layer, and an appropriate set of weights using data samples since they are universal function approximators (Hornik, Stinchcombe, & White, 1989). Neural black-box models have been successfully used in chemical and biochemical applications (Thibault, van Breusegem, & Cheruy, 1990). However, since there is no process knowledge incorporated in the neural model, these methods may lead to predictions which may conflict with fundamental constraints represented by the conservation principles, particularly when outside the domain of training (de Azevedo, Dahm, & Oliveira, 1997). One way to overcome this problem is to include all available knowledge of the process into a

* Corresponding author. Tel.: +421-2-59-32-53-64; fax: +421-2-52-49-64-69.
*E-mail addresses:* andrasik@chtf.stuba.sk (A. Andrášik), ameszaro@cvt.stuba.sk (A. Mészáros), sfeyo@fe.up.pt (S.F. de Azevedo).

0098-1354/$ – see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2003.12.002

**Nomenclature**

| | |
|---|---|
| $C$ | carbon dioxide concentration (mol l$^{-1}$) |
| $D$ | dilution rate (h$^{-1}$) |
| $E$ | ethanol concentration (mol l$^{-1}$) |
| $f_{\mathrm{ic}}$ | induction or repression factor |
| $G$ | gas phase oxygen concentration (mol l$^{-1}$) |
| $h$ | nonlinear function |
| $J$ | sum squared error |
| $k$ | saturation constant (mol l$^{-1}$) |
| $K_{\mathrm{L}}a$ | volumetric mass transfer coefficient based on the liquid volume (h$^{-1}$) |
| $L$ | Lyapunov function |
| $m$ | gas liquid distribution coefficient (mol mol$^{-1}$) |
| $N$ | number of patterns |
| $O$ | dissolved oxygen concentration (mol l$^{-1}$) |
| $q$ | flow rate (l h$^{-1}$) |
| $Q$ | specific consumption or production rate (mol C mol DW$^{-1}$h$^{-1}$) |
| $S$ | substrate concentration (mol l$^{-1}$) |
| $t$ | time (h) |
| $T$ | time constant for the induction of the production of consumption capacity (h) |
| $u$ | system input vector |
| $V$ | volume (l) |
| $X$ | biomass concentration (mol l$^{-1}$) |
| $y$ | system output vector |
| $Y_{ij}$ | yield of component $j$ on $i$ (mol mol$^{-1}$) |

*Greek letters*

| | |
|---|---|
| $\alpha$ | learning rate |
| $\mu$ | specific biomass growth rate (l h$^{-1}$) |

*Subscripts*

| | |
|---|---|
| c | carbon dioxide |
| e | ethanol |
| g | gas phase |
| $i$ | component $i$ |
| in | input |
| I | inhibition |
| $k$ | iteration step |
| l | liquid phase |
| lim | limited capacity |
| max | maximum |
| n | glucose |
| o | oxygen |
| ox | oxidative |
| pr | production |
| red | reductive |

*Superscripts*

| | |
|---|---|
| M | model |
| ox | oxidative |
| red | reductive |

hybrid model, where unknown process parameters are modeled using artificial neural networks (Psichogios & Ungar, 1992).

Having the appropriate model enables to develop a controller based on neural networks. There are two main approaches to neural control design that can be termed as direct and indirect methods. In the direct control scheme, a neural network is incorporated into the control system such that the system results in identity mapping between reference input and system output (e.g. Ordónez & Passino, 2001; Chan & Rad, 2000). The indirect scheme utilizes a model of the plant in order to identify the dynamics of the system and, then, generate a control input according to predefined optimization calculation (e.g. Lizarraga & Etxebarria, 1999; Wang, Oh, & Yoon, 1998). A good review of modeling and control strategies using neural networks has been carried out in (Hunt, Sbarbaro, Zbikowski, & Gawthrop, 1992). During the nineties further progress has been made in the field (Najim, Rusnák, Mészáros, & Fikar, 1997; Mészáros, Andrášik, & Rusnák, 2000), concerning, e.g. predictive control employing in some form ANN based modeling.

The design of most neural control schemes is based on gradient optimization, such as back-propagation, for training. Although some control problems can be well treated using this approach, it may fail for systems with strong non-linearities and large uncertainties.

Herein, we present a new ANN control strategy based on indirect adaptive control, where the controlled process is represented by a hybrid model consisting of a physical part and a neural network black box. In this strategy, three adjustable parameters of the neural controller are adapted on-line using a special training algorithm designed to be stable in the sense of Lyapunov. To demonstrate the feasibility and the performance of this scheme, simulation studies have been carried out on the control of a continuous-flow stirred biochemical reactor. The main goal of the control system is to maintain a desired profile of biomass concentration in the fermentor by manipulating the substrate flow rate. Simulation results demonstrate the usefulness and the robustness of the control system proposed.

## 2. Neural network based hybrid modeling

Combining black box techniques with physical equations has been receiving significant attention since the early 1990s. There are several approaches to hybrid modeling discussed in literature (Schubert, Simutis, Dors, Havlik, & Lübbert, 1994). In Thompson and Kramer (1994), a hybrid model of a fed-batch bioreactor is developed, in which an artificial neural network augments the performance of a parametric model that describes the specific kinetic rates, such as biomass growth and substrate consumption. The combined output of the parametric model and the ANN is processed by an output model that calculates the system state. Thus, a new type of model is created—a hybrid model using neural networks.

This model combines a partial first principles model—which incorporates the available prior knowledge about the process being modeled—with a neural network which serves as an estimator of unmeasured process parameters that are difficult to model from first principles. In comparison with conventional approaches, ANN based hybrid modeling is a powerful tool for process modeling, particularly when highly nonlinear behavior over a large operating domain has to be described and limited theoretical knowledge of the process is available. Examples are models of batch or fed-batch processes, cyclic processes or distributed parameter processes.

However, in majority of cases, in neural network based hybrid modeling, target outputs are not directly available. One possibility how to overcome this problem is to use the so called sensitivity approach proposed by Schubert et al. (1994). The general idea of this approach is to express the overall model, including the neural network part, as a single differential equation. In general terms, the known partial process model can be used to calculate a suitable error signal that can be used to update the network's weights. The observer error between the structured model's predictions and the actual state variable measurements can be “back-propagated” through the known set of equations and translated into an error signal for the neural network component.

Let us consider a process described by differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}t} = h(y(t), u(t), w) \tag{1}$$

where $h$ is a nonlinear vector function of the system inputs, $u(t)$, the system outputs, $y(t)$ and some parameters, $w$ (which we assume to be represented by means of a neural network). In order to train the neural network (black-box part of the hybrid model), pairs of input/output data vectors $(u, y)$ must be available, e.g., as a set of past measurements. The training consists in adaptation of network's weights, $w$, in such a way that the sum of the squared deviations between the output data predicted by the network, $y_i$, and the corresponding target data, $y_i^*$, becomes minimal,

$$J = \frac{1}{2}\sum_{i=1}^{N}(y_i - y_i^*)^2 \tag{2}$$

where $N$ stands for dimension of the data vector. The usual way to minimize $J$ is to use gradient procedures, like the steepest-descent algorithm. Weights in the $k$th step of this iterative process are changed in the direction of gradient,

$$w_{k+1} = w_k - \alpha\frac{\partial J}{\partial w_k} \tag{3}$$

Considering Eq. (2), the derivation of $J$ with respect to $w_k$ gives

$$\frac{\partial J}{\partial w_k} = \sum_{i=1}^{N}\left[(y_i - y_i^*)\frac{\partial y_i}{\partial w_k}\right] \tag{4}$$

Thus, the problem consists in determining the derivatives $\partial y/\partial w$. During the training phase, differential Eq. (1) must be solved for fixed $u(t)$. Differentiation of this equation with respect to weights gives,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial y}{\partial w}\right) = \frac{\partial h \partial y}{\partial y \partial w} + \frac{\partial h}{\partial w} \tag{5}$$

with initial condition

$$\left.\frac{\partial y}{\partial w}\right|_{t=0} = 0 \tag{6}$$

In comparison with conventional techniques, hybrid modeling usually gives results with significantly higher level of accuracy. Since generalization is related only to uncertain parts of the process, the hybrid modeling method gives satisfactory predictions using even a limited dimension of training data sets and the results are only slightly influenced by the number of weights in the network part of the hybrid model (de Azevedo et al., 1997). Furthermore, to obtain the same mapping quality as that in the case of conventional neural network model, less data are needed to train the network and, eventually, it provides a more general process representation (Schubert et al., 1994).

## 3. Hybrid model of the controlled process

The non-linear model describing the response of *Saccharomyces cerevisiae*, known as baker's yeast, has been used for simulation of the process dynamics. This mathematical model, adopted partly from work of (Sweere, 1988) and extended to the present dynamical structure (Mészáros, Brdys', Tatjewski, & Lednický, 1995) is based on limited oxidation capacity of yeast leading to a switch-over from oxidative to oxido-reductive metabolism. Regarding the law of conservation of mass, the model for continuous process can be expressed by the following set of ordinary differential equations.

*Cell mass concentration*

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \left(\mu - \frac{q}{V_1}\right)X \tag{7}$$

*Substrate concentration*

$$\frac{\mathrm{d}S}{\mathrm{d}t} = \frac{q}{V_1}(S_{\mathrm{in}} - S) - Q_{\mathrm{s}}X \tag{8}$$

*Ethanol (product) concentration*

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{q}{V_1}(E_{\mathrm{in}} - E) + (Q_{\mathrm{e,pr}} - Q_{\mathrm{e}})X \tag{9}$$

*Carbon dioxide concentration*

$$\frac{\mathrm{d}C}{\mathrm{d}t} = D_{\mathrm{g}}(C_{\mathrm{in}} - C) - Q_{\mathrm{c}}X \tag{10}$$

*Dissolved oxygen concentration*

$$\frac{\mathrm{d}O}{\mathrm{d}t} = \frac{q}{V_1}(O_{\mathrm{in}} - O) + Na - Q_{\mathrm{o}}X \tag{11}$$

*Gas phase oxygen concentration*

$$\frac{\mathrm{d}G}{\mathrm{d}t} = D_\mathrm{g}(G_\mathrm{in} - G) - Na\frac{V_\mathrm{l}}{V_\mathrm{g}} \tag{12}$$

Mathematical description of the kinetic model mechanisms is arranged in the following table:

| Mechanism | Description |
|---|---|
| Glucose uptake | $Q_\mathrm{s} = Q_{\mathrm{s,max}}\dfrac{S}{k_\mathrm{s}+S}$ |
| Oxidation capacity | $Q_{\mathrm{o,lim}} = Q_{\mathrm{o,max}}\dfrac{O}{k_\mathrm{o}+O}$ |
| Oxidative glucose metabolism | $Q_{\mathrm{s,ox}} = \min\begin{cases} Q_\mathrm{s} \\ Y_\mathrm{os}Q_{\mathrm{o,lim}} \end{cases}$ |
| Reductive glucose metabolism | $Q_{\mathrm{s,red}} = Q_\mathrm{s} - Q_{\mathrm{s,ox}}$ |
| Ethanol uptake | $Q_\mathrm{e} = Q_{\mathrm{e,max}}\dfrac{E}{k_\mathrm{e}+E}\dfrac{k_\mathrm{I}}{k_\mathrm{I}+S}$ |
| Oxidative ethanol metabolism | $Q_{\mathrm{e,ox}} = min\begin{cases} Q_\mathrm{e} \\ (Q_{\mathrm{o,lim}} - Q_{\mathrm{s,ox}}Y_{\mathrm{so}})Y_{\mathrm{oe}} \end{cases}$ |
| Ethanol production | $Q_{\mathrm{e,pr}} = Y_{\mathrm{se}}Q_{\mathrm{s,red}}$ |
| Growth | $\mu = Y_{\mathrm{sx}}^{\mathrm{ox}}Q_{\mathrm{s,ox}} + Y_{\mathrm{sx}}^{\mathrm{red}}Q_{\mathrm{s,red}} + Y_{\mathrm{ex}}Q_\mathrm{e}$ |
| Carbon dioxide production | $Q_\mathrm{c} = Y_{\mathrm{sc}}^{\mathrm{ox}}Q_{\mathrm{s,ox}} + Y_{\mathrm{sc}}^{\mathrm{red}}Q_{\mathrm{s,red}} + Y_{\mathrm{ec}}Q_\mathrm{e}$ |
| Oxygen consumption | $Q_\mathrm{o} = Y_{\mathrm{so}}Q_{\mathrm{s,ox}} + Y_{\mathrm{eo}}Q_{\mathrm{e,ox}}$ |
| Oxygen transfer | $Na = K_\mathrm{L}a\left(\dfrac{G}{m} - O\right)$ |
| Maximum consumption rates where induction or repression factors are as follows | $\dfrac{\mathrm{d}Q_{i,\mathrm{max}}}{\mathrm{d}t} = \dfrac{1}{T_i}\left(Q_{i,\mathrm{max}}^{p}f_{\mathrm{ic}} - Q_{i,\mathrm{max}}\right)$ <br> $f_{\mathrm{oc}} = \dfrac{O}{k_\mathrm{o}+O}\dfrac{2S+E}{k_\mathrm{s}+2S+E}$ <br> $f_{\mathrm{sc}} = \dfrac{S}{k_\mathrm{n}+S}$ <br> $f_{\mathrm{ec}} = \dfrac{E}{k_\mathrm{e}+E}\dfrac{k_\mathrm{I}}{k_\mathrm{I}+S}\dfrac{O}{k_\mathrm{o}+O}$ |

The first principle model, expressed by Eqs. (7)–(12), is well behaved for description of cell growth on glucose as substrate and is suitable for control simulation experiments. For the purposes of this work, it takes the role of the controlled plant and serves for acquisition of data needed for training the neural module of the hybrid model.

The crucial step in modeling of biochemical processes is to identify quantities of specific growth rates. The problem is on the lack of accurate kinetic mechanism description relating to mass balance of biotransformations. Although a fair number of analytical models describing growth rates have been published, each case seems to need a specific approach. A momentum of generalization, in this subject, can be offered by black-box techniques using neural networks. Combining ANN modules for prediction of the biomass growth rate, $\mu$, from Eq. (7) and a structured mass balance module (selected differential equations from Eqs. (7)–(12)), results in a hybrid model. The structure of the proposed hybrid model, consisting of a neural network and a first principles part, is depicted in Fig. 1.

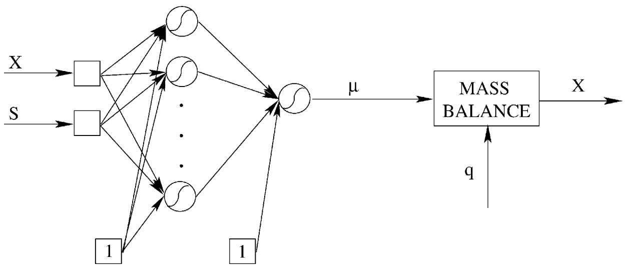

Fig. 1. Hybrid model structure for the fermentation process.

Let us carry out now the mathematical analysis for hybrid model training for our case study, in relation to Eqs. (1)–(6). Since our aim is to control only the biomass concentration at the outlet, $X$, the sensitivity approach is only carried out for the following differential equation:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \mu X - \frac{q}{V_\mathrm{l}}X \tag{13}$$

As $X$ will be chosen as the output controlled variable, with respect to Eq. (1), we can formally rewrite the model as,

$$h = \mu X - \frac{q}{V_\mathrm{l}}X \tag{14}$$

and gradient $\partial h/\partial y$ takes the form,

$$\frac{\partial h}{\partial y} = \frac{\partial h}{\partial X} = \mu + X\frac{\partial \mu}{\partial X} - \frac{q}{V_\mathrm{l}} \tag{15}$$

When determining gradient $\partial h/\partial w$ with respect to Eq. (14), one has to realize that in (14) only $\mu$, as the result of the ANN training (see Fig. 1), is dependent on $w$. It gives,

$$\frac{\partial h}{\partial w} = X\frac{\partial \mu}{\partial w} \tag{16}$$

These two terms are inserted into the sensitivity Eq. (5), leading to

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial X}{\partial w}\right) = \left(\mu + X\frac{\partial \mu}{\partial X} - \frac{q}{V_\mathrm{l}}\right)\frac{\partial X}{\partial w} + X\frac{\partial \mu}{\partial w} \tag{17}$$

where in gradient $\partial X/\partial w$, $X$ refers to the prediction of output.

Thus, we have a differential equation which computes gradients $\partial X/\partial w$, necessary for optimization of weights during iteration steps: Eqs. (3) and (4). The unknown partial

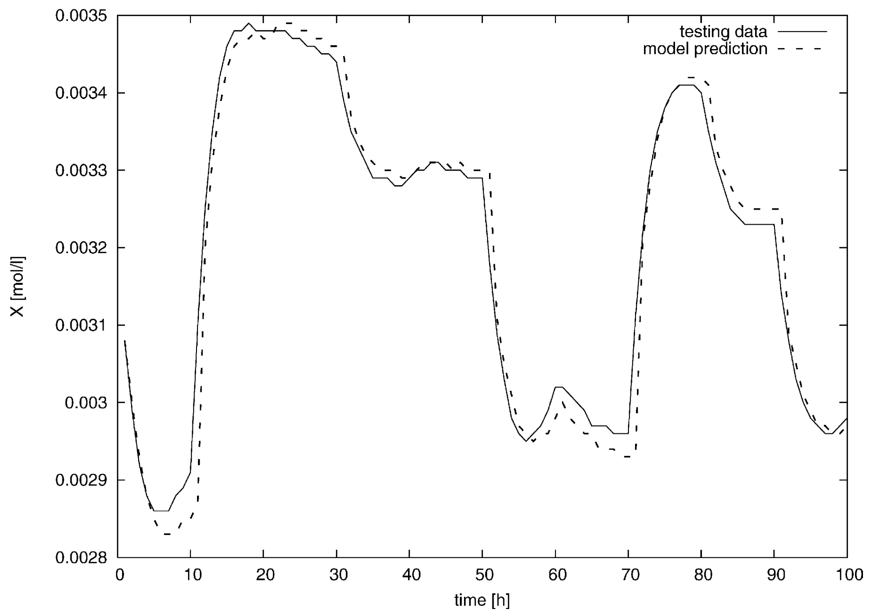

Fig. 2. Hybrid model performance in the presence of noise.

derivatives, $\partial\mu/\partial X$ and $\partial\mu/\partial w$ in sensitivity equation depend on the neural net structure and can be evaluated using the back-propagation method.

The mathematical model of the biochemical process (Eqs. (7)–(12)) was used to generate data sets for training ANN of the hybrid model as well as the neural controller. The substrate flow rate, $q$, was chosen as a manipulated variable for generating data. The hybrid model network of structure introduced in Fig. 1 with 7 hidden neurons was fed with training data set and was trained using the sensitivity approach proposed. To demonstrate the performance of the hybrid model in the presence of noise, a white noise-like disturbance was applied to the process in form of variations in inlet substrate concentration, $S_{\text{in}}$, with a dispersion of 20% about the mean value. The comparison of perturbed target data and hybrid model predictions is depicted in Fig. 2.

## 4. The control system

The objective of this work is to propose a control system using two feedforward ANN models: one incorporated in the hybrid model of the process, providing values for the plant output; the other, performing the task of feedback controller in the form of plant dynamic inverse. Identification of process inverse dynamics is defined as finding the inverse mapping of a system (Pham & Oh, 1999). It is useful to know the inverse dynamics of a plant in order to control it. Then, in an ideal situation, the dynamics of the controller could simply take the form of the plant inverse dynamics.

For our case we shall consider an approach adopted by Psaltis, Sideris, and Yamamura (1989) known as direct or generalized inverse learning. The basic idea is to feed the network with present and past plant outputs and get it trained until it generates the right plant input which caused just those output responses applied. Such a network structure is shown in Fig. 3. The error between the desired and actual output of the network is used to adjust the network weights.

However, inverse dynamic models are prone to give incorrect results especially when process nonlinearities are modeled. To prevent the system from producing permanent control error, an additional controller term is proposed herein to be incorporated into the neural controller. This new controller, augmented by so-called PID neurons, as shown in Fig. 4, is designed to ensure offset-free performance. The basic idea is to replace the common bias signal by three additional nodes, where inputs to the neurons are the control error, the sum of control error and the deviation of control error, respectively. Connections between PID neurons and the output of the neural network are evaluated by weights designated $w_{\text{P}}$, $w_{\text{I}}$ and $w_{\text{D}}$. These weights are trained on-line using the particular optimizing algorithm

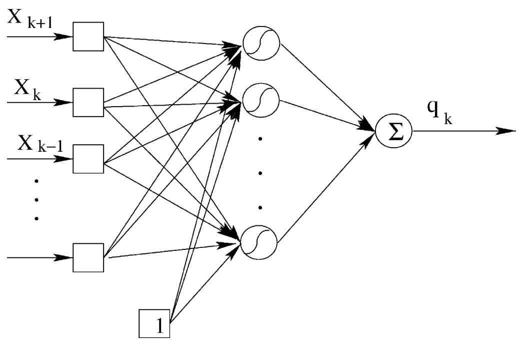

Fig. 3. Architecture of the plant inverse model.

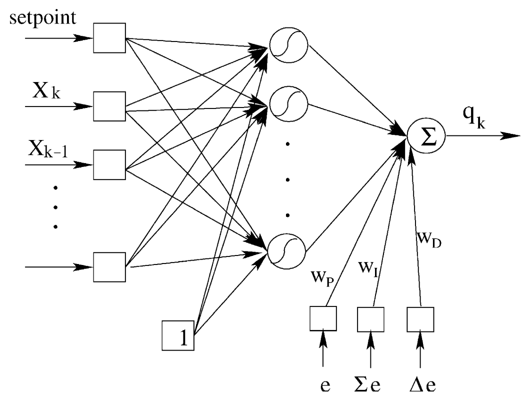

Fig. 4. Neural controller augmented by PID term.

described in the following section. Then, the overall control scheme takes the structure depicted in Fig. 5.

### 4.1. Convergence of the learning algorithm

The strategy of training consists of on-line tuning of the PID weights of the neural controller in such a way that the control error converges to zero. The principle of the proof is adopted from (Hoo, Sinzinger, & Piovoso, 2002) where the stability of neural predictor is based on proper selection of learning rate. Finding the optimal interval for the learning rate is accomplished by using Lyapunov function approach.

Let us consider a discrete Lyapunov function given by

$$L = \tfrac{1}{2}(e^2 + \lambda\,\Delta u^2) = \tfrac{1}{2}[(r - y^{\mathrm{M}})^2 + \lambda(u - u^{\mathrm{old}})^2] \qquad (18)$$

where $r$ is setpoint, $y^{\mathrm{M}}$ is predicted value of the controlled variable (output of the hybrid model), $u$ is estimated manipulated variable (output of the neural controller) and $u^{\mathrm{old}}$ is the last control action applied to the plant. The change in the discrete Lyapunov function at the $k$th iteration step is given by

$$\Delta L_k = L_{k+1} - L_k = \tfrac{1}{2}(e_{k+1}^2 - e_k^2) + \tfrac{1}{2}\lambda(\Delta u_{k+1}^2 - \Delta u_k^2) \qquad (19)$$

where the setpoint error in the next sampling period is estimated as

$$e_{k+1} = e_k + \Delta e_k = e_k + \left(\frac{\partial e_k}{\partial \vec{w}}\right)^{\mathrm{T}} \Delta\vec{w} \qquad (20)$$

As we have to tune only the PID weight coefficients and, with respect to linear activation function at the output neuron (see Fig. 4), the change of the manipulated variable in the next sampling period is given by

$$\Delta u_{k+1} = \Delta u_k + \left(\frac{\partial u_k}{\partial \vec{w}}\right)^{\mathrm{T}} \Delta\vec{w} = \Delta u_k + \vec{B}^{\mathrm{T}}\Delta\vec{w} \qquad (21)$$

where $\vec{B}$ is a vector of inputs to the PID neurons.

The PID weights are adjusted in proportion to the negative gradient of their contribution to the control error. Unlike in the common "steepest descent" algorithm, the learning rate in our case is not constant. The change of the weight coefficients is given by

$$\Delta\vec{w} = -\alpha\frac{\partial L}{\partial \vec{w}} = -\alpha\left(\frac{\partial L}{\partial e_k}\frac{\partial e_k}{\partial \vec{w}} + \frac{\partial L}{\partial u_k}\frac{\partial u_k}{\partial \vec{w}}\right) = -\alpha\left(e_k\frac{\partial e_k}{\partial \vec{w}} + \lambda u_k\frac{\partial u_k}{\partial \vec{w}}\right) \qquad (22)$$

Let us start with Eq. (19). Substituting for $e_{k+1}$ from Eq. (20) and $\Delta u_{k+1}$ from Eq. (21) we have,

$$\begin{aligned}\Delta L &= \frac{1}{2}\left\{\left[e_k + \left(\frac{\partial e_k}{\partial \vec{w}}\right)^{\mathrm{T}}\Delta\vec{w}\right]^2 - e_k^2 + \lambda[(\Delta u_k + \vec{B}^{\mathrm{T}}\Delta\vec{w})^2 - \Delta u_k^2]\right\}\\ &= e_k\left(\frac{\partial e_k}{\partial \vec{w}}\right)^{\mathrm{T}}\Delta\vec{w} + \frac{1}{2}\left[\left(\frac{\partial e_k}{\partial \vec{w}}\right)^{\mathrm{T}}\Delta\vec{w}\right]^2 + \lambda\,\Delta u_k\vec{B}^{\mathrm{T}}\Delta\vec{w}_k + \frac{1}{2}\lambda(\vec{B}^{\mathrm{T}}\Delta\vec{w}_k)^2 \end{aligned} \qquad (23)$$

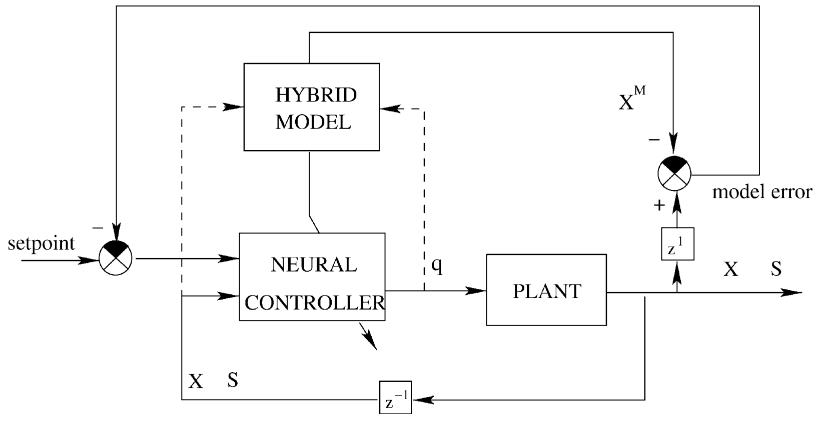

Fig. 5. Block diagram of the proposed control system.

Further, substitution of $\Delta\vec{w}_k$ from Eq. (22) into Eq. (23), leads to

$$\Delta L = -\alpha e_k^2 \left\| \frac{\partial e_k}{\partial \vec{w}} \right\|^2 - \alpha\lambda\, \Delta u_k e_k \left(\frac{\partial e_k}{\partial \vec{w}}\right)^{\mathrm{T}} \vec{B} + \frac{\alpha^2}{2}\left[ e_k \left\| \frac{\partial e_k}{\partial \vec{w}} \right\|^2 + \lambda\, \Delta u_k \left(\frac{\partial e_k}{\partial \vec{w}}\right)^{\mathrm{T}} \vec{B} \right]^2 - \alpha\lambda\, \Delta u_k e_k \vec{B}^{\mathrm{T}} \frac{\partial e_k}{\partial \vec{w}} - \alpha\lambda^2\, \Delta u_k^2 \|\vec{B}\|^2 + \frac{\alpha^2}{2}\lambda \left[ e_k \vec{B}^{\mathrm{T}} \frac{\partial e_k}{\partial \vec{w}} + \lambda\, \Delta u_k \|\vec{B}\|^2 \right]^2 \tag{24}$$

After manipulation we get,

$$\Delta L = -\frac{\alpha}{2} e_k^2 \left\| \frac{\partial e_k}{\partial \vec{w}} \right\|^2 \left[ 2 - \alpha \left( \left\| \frac{\partial e_k}{\partial \vec{w}} \right\|^2 + \lambda \|\vec{B}\|^2 \right) \right] - \frac{\alpha\lambda^2}{2} \Delta u_k^2 \|\vec{B}\|^2 \left[ 2 - \alpha \left( \left\| \frac{\partial e_k}{\partial \vec{w}} \right\|^2 + \lambda \|\vec{B}\|^2 \right) \right] - \alpha\lambda\, \Delta u_k \left(\frac{\partial e_k}{\partial \vec{w}}\right)^{\mathrm{T}} \vec{B} e_k \times \left[ 2 - \alpha \left( \left\| \frac{\partial e_k}{\partial \vec{w}} \right\|^2 + \lambda \|\vec{B}\|^2 \right) \right] \tag{25}$$

which leads to,

$$\Delta L = -\frac{\alpha}{2} \left[ 2 - \alpha \left( \left\| \frac{\partial e_k}{\partial \vec{w}} \right\|^2 + \lambda \|\vec{B}\|^2 \right) \right] \times \left( e_k \frac{\partial e_k}{\partial \vec{w}} + \lambda\, \Delta u_k \vec{B} \right)^2 \tag{26}$$

Since the last term is always positive and $\alpha$ is a positive number, we can conclude that difference of the Lyapunov function is negative only in the case when

$$\left[ 2 - \alpha \left( \left\| \frac{\partial e_k}{\partial \vec{w}} \right\|^2 + \lambda \|\vec{B}\|^2 \right) \right] > 0 \tag{27}$$

$$\alpha < \frac{2}{(\|\partial e_k/\partial \vec{w}\|^2 + \lambda \|\vec{B}\|^2)} \tag{28}$$

### 4.2. On-line tuning of PID weights

In the relevant control implementation, it is necessary to synchronize the following steps:

- data acquisition,
- tuning of PID weights,
- computation of new control action.

The crucial procedure is the on-line tuning of the PID weight coefficients. The key task is to determine the learning rate in order to speed up the optimization process. However first, it is necessary to compute gradient $\partial e_k/\partial \vec{w}$. Assuming that the hybrid model of the plant is accurate enough, we can find this gradient using the stochastic algorithm approach (Kushner & Sanvicente, 1974). The algorithm is iterative and it can be summarized in the following steps:

(1) perturbation of weight coefficient ($w_{\mathrm{P}}$) in positive direction;
(2) computation of neural controller response: $q_t$;
(3) application of $q_t$ to the input of mass balance part of the hybrid model;
(4) computation of hybrid model response $X_{t+1}$, as prediction of the controlled variable at the next sample period;
(5) repeating (2)–(4) for negative perturbation of weight;
(6) repeating (1)–(5) for the other two weight coefficients ($w_{\mathrm{I}}$, $w_{\mathrm{D}}$);
(7) computation of the unknown gradient

$$\frac{\partial e_k}{\partial w} = \frac{X_{t+1}^{-} - X_{t+1}^{+}}{\Delta}$$

where $\Delta$ is the size of perturbation and $+/-$ is an indication of positive and negative direction, respectively;
(8) computation of learning rate at present step

$$\alpha_k = \frac{2c}{(\|\partial e_k/\partial w\|^2 + \lambda \|\vec{B}\|^2)}$$

where $c \in (0, 1)$;
(9) optimization of PID weight coefficients

$$\vec{w}_k = \vec{w}_{k-1} - \alpha \frac{\partial L}{\partial \vec{w}}$$

(10) evaluation of the objective function (Eq. (18)) at actual iteration;
(11) if $L_k <$ (acceptable error) or number of iterations $k > k_{\max} \Rightarrow$ go to the next step; else $k := k + 1$ and go to step (1);
(12) computation of new control action as a response of the neural controller;
(13) application of new control action to the plant.

**Note**: If the hybrid model is not an accurate model of the plant, the approximation method for the error derivative may have problems due to noise presence in the error and the use of Lyapunov function does not guarantee asymptotic exponential stability.

Choice of $c$ coefficient is a result of trade-off between the speed and stability of training algorithm, for each specific case. A lower value of $c$ guarantees more stable adaptation of weights. Since a possible hybrid model mismatch may occur, it is convenient to set $c$ coefficient “small enough”.

## 5. Results

Two different cases concerning the controller structure have been tested for comparison. First, the inverse model of the process was used as a non-adaptive controller, with corresponding control structure shown in Fig. 6. The training data

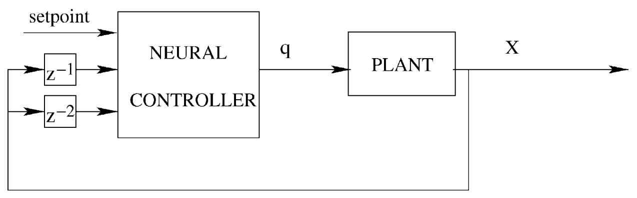

Fig. 6. Block diagram of direct inverse control.

set generated by the mathematical model (Mészáros et al., 1995) was used to obtain the inverse mapping of the process. Inputs to the neural controller (see Fig. 3) were $X_{k+1}$, $X_k$ and $X_{k-1}$, respectively. The controlled process was simulated by the same structured model as that of used for training the network. Two simulation experiments were carried out for this case study. The first one did not consider any disturbance or parameter perturbation. The second experiment was to demonstrate the control performance in presence of noise: the inlet substrate concentration, $S_{\mathrm{in}}$ was considered to be perturbed at each sampling period in the range up to 20% of its steady state value. Simulation results for deterministic and stochastic case are depicted in Fig. 7 and 8, respectively. It is obvious, from simulation results, that a controller of this structure is unable to control satisfactorily the process because of imperfect inverse mapping. The performance is poor especially when parameter perturbation occurs.

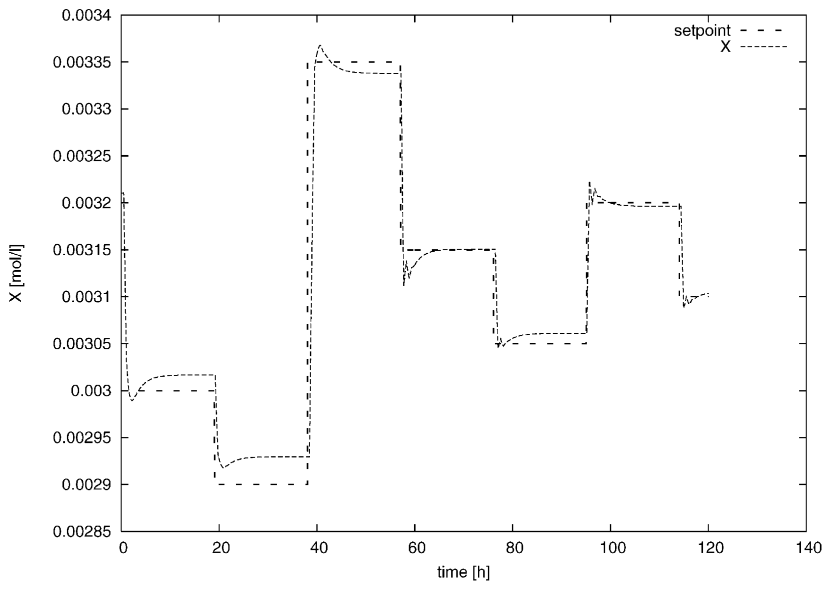

Fig. 7. Direct inverse control performance for deterministic case.

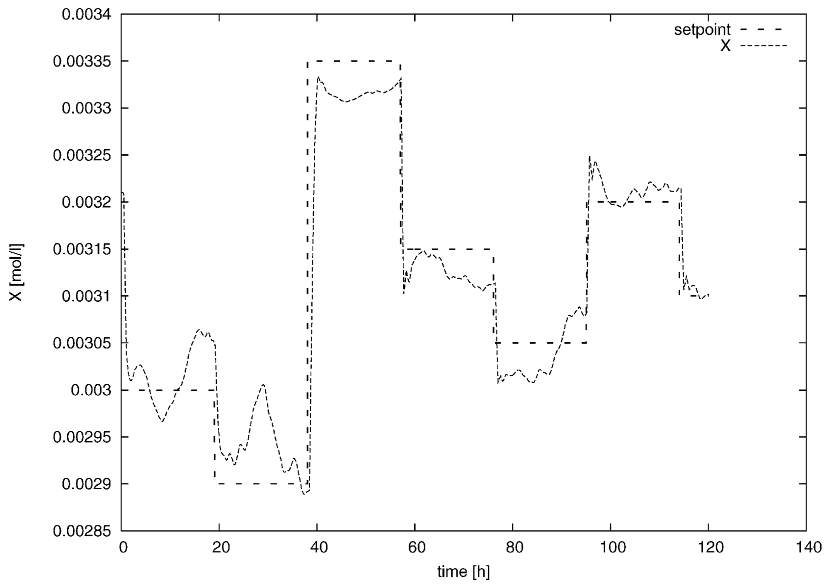

Fig. 8. Direct inverse control performance in presence of perturbation of $S_{\mathrm{in}}$.

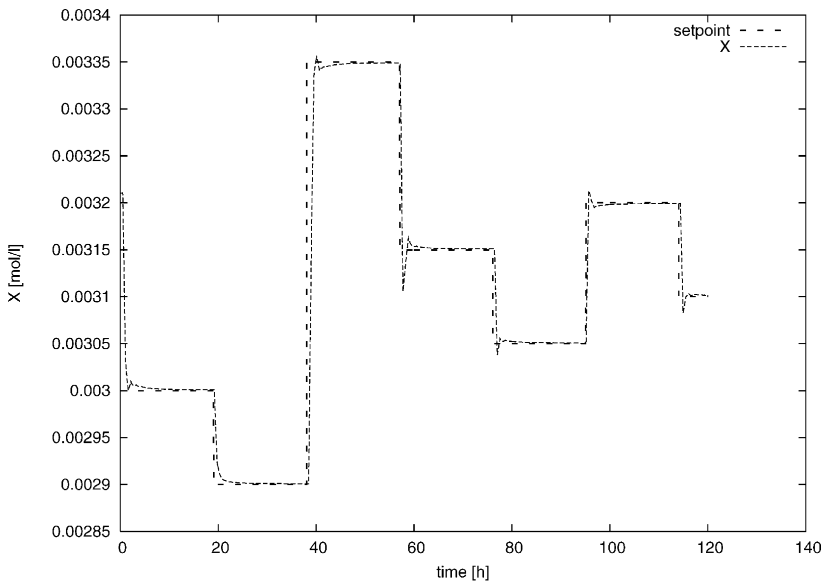

Fig. 9. Performance of adaptive neural PID control—deterministic case.

Behavior of the adaptive neural controller proposed in Section 4 was further demonstrated on the same biomass concentration control, as another case study. The weights of the neural controller were pre-trained off-line using the structure introduced in Fig. 3 with three inputs and five hidden neurons. For training the network, a modified back-propagation algorithm using conjugate gradients was applied. During on-line control, the modified structure shown in Fig. 4 was used, where one of inputs to the network, instead of unknown $X_{k+1}$, was the setpoint. Bias PID weight coefficients have been adapted only, while the other weight coefficients remained constant, set on the values obtained by off-line pretraining. New control action was optimized every 30 min on the basis of the proposed algorithm.

Results of control for both, deterministic and stochastic cases are depicted in Figs. 9 and 10, respectively. For easy comparison, setpoints applied were chosen the same as those of the case of direct inverse controller. When comparing the two cases adopted, one can easily notice the significant improvement in performance of the control system, augmented by adaptive PID terms, in both, regulatory and tracking aspects.

The evolution profile of learning rate, in Fig. 11, as well as adaptation courses of the tuned $w_P$, $w_I$ and $w_D$ weight

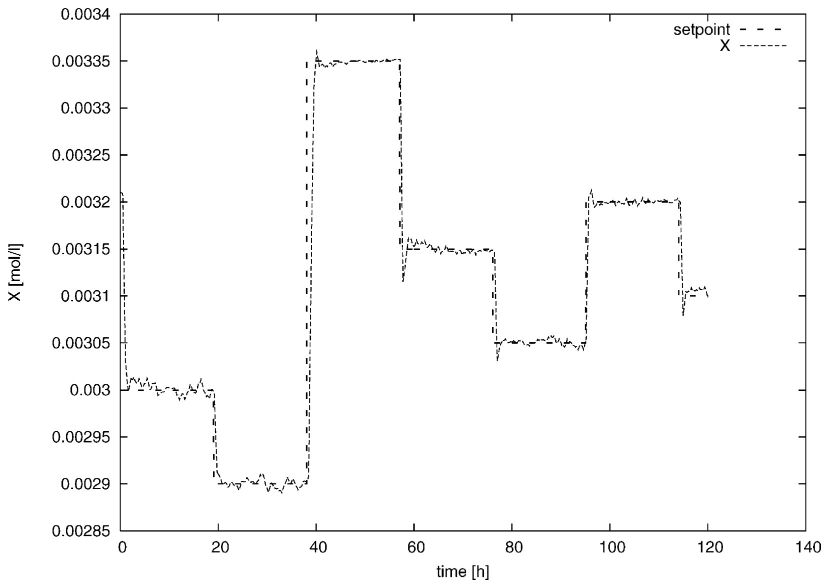

Fig. 10. Performance of adaptive neural PID control—perturbation of $S_{in}$.

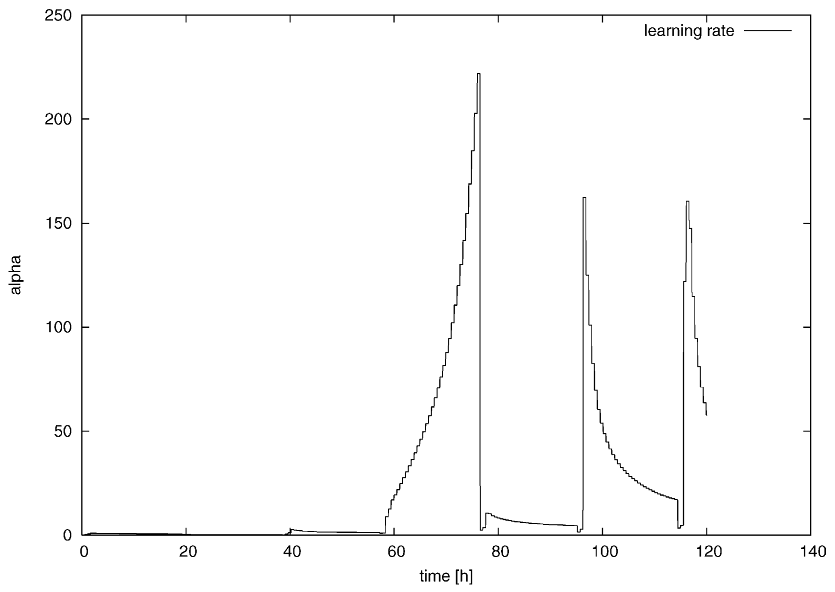

Fig. 11. Evolution of learning rate coefficient.

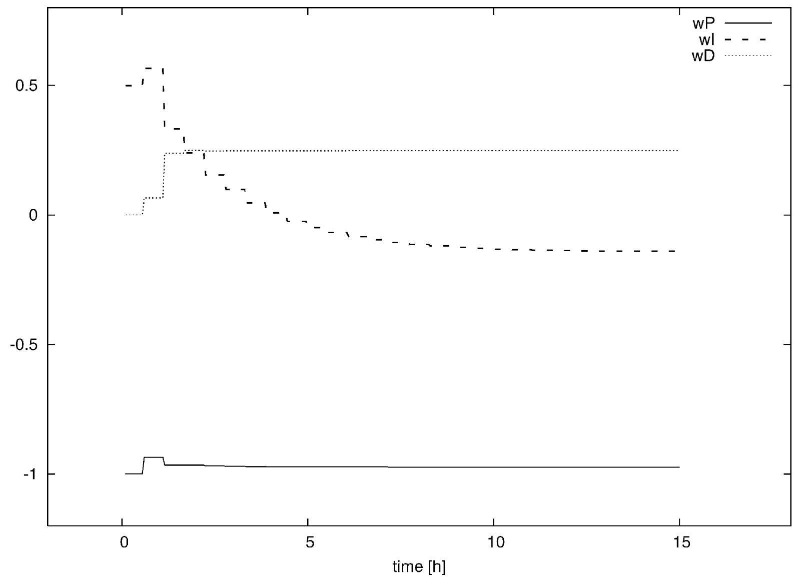

Fig. 12. Evolution of $w_P$, $w_I$ and $w_D$ weights during adaptive control.

coefficients, depicted in Fig. 12, show good convergence properties of the optimization and adaptation procedures involved.

## 6. Conclusions

A new method for adaptive nonlinear control based on artificial neural networks has been presented in this contribution. The new control law incorporates the ability for adaptation through adjustment of bias neurons and ensures offset-free performance in the presence of unmeasured disturbances. A special control structure, incorporating the algorithm for on-line tuning of PID weight coefficients has been developed, where the convergence was guaranteed by proper selection of the learning rate.

The performance of the proposed adaptive neural control system has been assessed through the simulation of the controlled operation of a nonlinear continuous biochemical process. For prediction of process output, a hybrid model, consisting of a mechanistic mass balance complemented by neural black-box pretrained off-line for kinetic prediction, has been utilized. The results of simulation experiments have confirmed good regulatory and tracking performance of the augmented adaptive neural controller implemented. The advantage is obvious especially in the presence of disturbances or parameter uncertainties when it is not possible to obtain a perfect inverse mapping of the controlled plant.

## Acknowledgements

The authors highly acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under grant nos. 1/8108/01 and 1/135/03. As this work was partly carried out at the Department of Chemical Engineering, Faculty of Engineering, University of Porto, the authors wish to thank also the Portuguese National Foundation for Scientific Research, for its support.

## References

Castrillo, J., & Ugalde, U. (1994). A general model of yeast energy metabolism in aerobic chemostat culture. *Yeast, 10*, 185–197.

Chan, H., & Rad, A. (2000). Real-time flow control using neural networks. *ISA Transactions, 39*, 93–101.

de Azevedo, S., Dahm, B., & Oliveira, F. (1997). Hybrid modelling of biochemical processes: A comparison with the conventional approach. *Computers and Chemical Engineering, 21*, S751–S756.

Enfors, S., Hedenberg, J., & Olsson, K. (1990). Simulation of the dynamics in the Baker's yeast process. *Bioprocess Engineering, 5*, 191–198.

Hoo, K., Sinzinger, E., & Piovoso, M. (2002). Improvements in the predictive capability of neural networks. *Journal of Process Control, 12*, 193–202.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359–366.

Hunt, K., Sbarbaro, D., Zbikowski, R., & Gawthrop, P. (1992). Neural networks for control systems—A survey. *Automatica, 28*, 1083–1112.

Kushner, H., & Sanvicente, E. (1974). Stochastic approximation methods for constrained systems with observation noise on the systems and constraints. In *IFAC symposium on stochastic control*.

Lizarraga, I., & Etxebarria, V. (1999). Real-time adaptive neural control of a class of nonlinear systems. *Engineering Applications of Artificial Intelligence, 12*, 3–19.

Mészáros, A., Andrášik, A., & Rusnák, A. (2000). Application of artificial neural network strategies in process control. In *Advances in soft computing. Proceedings of the European symposium on computational intelligence*. Košice: Slovakia.

Mészáros, A., Brdys', M., Tatjewski, P., & Lednický, P. (1995). Multilayer adaptive control of continuous bioprocesses using optimising control technique. Case study: Baker's yeast culture. *Bioprocess Engineering, 12*, 1–9.

Najim, K., Rusnák, A., Mészáros, A., & Fikar, M. (1997). Constrained long-range predictive control based on artificial neural networks. *International Journal of Systems Science, 28*(12), 1211–1226.

Ordónez, R., & Passino, K. (2001). Adaptive control for a class of nonlinear systems with a time-varying structure. *IEEE Transactions on Automatic Control, 46*, 152–155.

Pham, H., Larsson, G., & Enfors, S. (1998). Growth and energy metabolism in aerobic fed-batch cultures of saccaromyces cerevisiae: Simulation and model verification. *Biotechnology and Bioengineering, 60*, 474–482.

Pham, D., & Oh, S. (1999). Identification of plant inverse dynamics using neural networks. *Artificial Intelligence in Engineering, 13*, 309–320.

Psaltis, D., Sideris, A., & Yamamura, A. (1989). A multilayered neural network controller. In *IEEE Control Systems Magazine, 17–21*.

Psichogios, D., & Ungar, L. (1992). A hybrid neural network-first principles approach to process modelling. *American Institute of Chemical Engineering Journal, 10*(38), 1499–1511.

Schubert, J., Simutis, R., Dors, M., Havlik, I., & Lübbert, A. (1994). Bioprocess optimization and control: Application of hybrid modelling. *Journal of Biotechnology, 35*, 51–68.

Sweere, A. (1988). Response of Baker's yeast to transient environmental conditions relevant to largescale fermentation processes. Ph.D. thesis, Drukkerij Elinkurijk Utrecht, Utrecht.

Thibault, J., van Breusegem, V., & Cheruy, A. (1990). On-line prediction of fermentation variables using neural networks. *Biotechnology and Bioengineering, 36*, 1041–1048.

Thompson, M., & Kramer, A. (1994). Modeling chemical processes using prior knowledge and neural networks. *American Institute of Chemical Engineering Journal, 8*(40), 1328–1340.

Wang, H., Oh, Y., & Yoon, E. (1998). Strategies of modeling and control of nonlinear chemical processes using neural networks. *Computers and Chemical Engineering, 22*, S823–S826.