Structure of Principal Component Based Neural Network Models of Dynamic Systems

Želimir Kurtanjek

Faculty of Food Technology and Biotechnology, University of Zagreb, Croatia

A new structure of neural network based systems for modeling and control of dynamic industrial processes is developed. The structure is composed of three serially connected subsystems: 1) ARMA — auto regression moving averages to account for system dynamics; 2) PCA — principal component analysis as statistical preprocessor for noise reduction and pattern compression 3) ANN — artificial neural network with static neurons and feed forward pattern propagation for nonlinear mapping of input/output interaction. Training of neural networks is performed with Ribiéra–Polack–Powell conjugate gradient method for minimization of the variance in output patterns between a real and a model system. The proposed modeling procedure is applied to data from a fed batch operation of an industrial deep jet bioreactor. Predictive power of the model is based on the analysis of responses in pseudosteady and oscillatory mode of operation with the trained and untrained patterns. The aim of the work is to develop a general neural network structure and analyze its applicability in the process control in biotechnology.

Keywords: auto regression, principal component decomposition, neural networks, process modeling, process control, biotechnology.

1. Introduction

From a systems point of view, artificial neural networks (ANN) are general models for mapping between high dimensional spaces of input and output patterns. ANN is perceived as a vast collection of simple processing units (neurons) which are highly interconnected (Rumelhart and McClelland, 1988; Zupan and Gasteiger, 1993). Data and rules are stored in ANN in the form of neuron parameters (activity) and connecting relations, which are result of learning (adaptation) process with selected training patterns to match outputs of an ANN to a real system. Their applicability in control engineering provides very attractive advantages compared to analytical methods (Ydstie, 1990; Bath and McAvoy, 1990; Chitra 1993).

This work proposes a new network structure based on decomposition of patterns into principal components applicable in the control of complex dynamic systems, such as operations with industrial bioreactors. Neural networks have very high potential for use in biology and industrial biotechnology. They can be applied in a broad spectrum of important tasks such as: computer modeling of specific structures and biological activity of proteins, on-line estimation of biological variables, adaptive control of biological reactors without available analytical models, integration of expert rules with on-line process control algorithms, and many others by which productivity and selectivity in production based on information stored in microorganism genetic funds can be increased (Karim and Riviera, 1991; Schubert et al, 1994; Kurtanjek, 1993, 1994).

2. System structure and algorithms

The work proposes a structure of network for on-line estimation and adaptive control in process engineering, especially for control of production in bioreactors. The structure is composed of three subsystems: ARMA, PCA and ANN; shown as independent modules on Fig. 1.

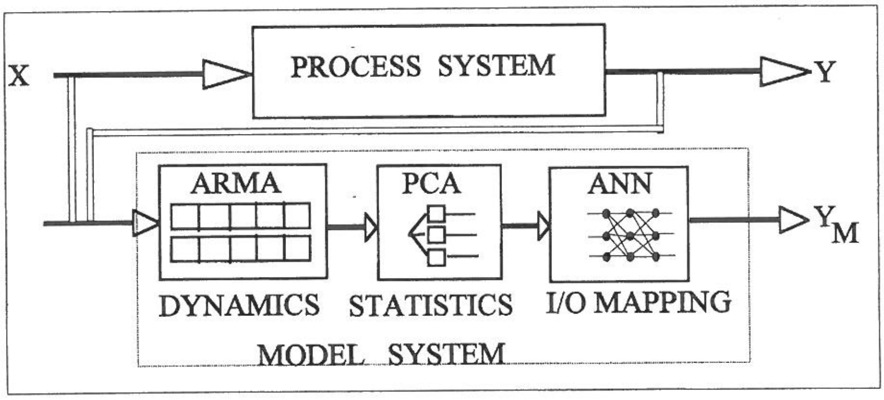

Fig. 1. General structure of neural network system composed of the submodels: 1) ARMA for approximation of process dynamics; 2) PCA principal component analysis for noise reduction and pattern compression; 3) ANN layers of neurons for nonlinear input/output mapping.

They are designed to perform pattern transformation for approximation of system dynamics, statistical preprocessing for measurement noise reduction and data compression to principal components, and feedforward signal propagation through layers of static neurons to provide nonlinear input-output mapping. Classical neurons are static nonlinear processors and measured system responses have to be transformed to patterns which can account for system dynamics. We have considered a general dynamic system given in the state model representation:

$$\tfrac{d}{dt}\vec{y} = \vec{f}(\vec{y}, \vec{x}, t), \qquad \vec{y}(t = 0) = \vec{y}_0 \qquad (1)$$

When system input $\vec{x}$ and output $\vec{y}$ are on-line sampled, for a linear system approximation of the model (1) based on set of discrete patterns is given in the form of auto regression moving averages, ARMA (Ljung and Söderström, 1987):

$$\vec{y}_{k+1} = \sum_{j=0}^{N} A_j \cdot \vec{y}_{k-j} + \sum_{j=0}^{M} B_j \cdot \vec{x}_{k-j} \qquad (2)$$

where A_j and B_j are the autoregression coefficients; N and M are integers determined by the duration of impuls response of a studied system, i.e. a stable system with a finite memory is considered. Since the system (1) is nonlinear, instead of estimation of in the linear model (2), nonlinearity is modeled by a neural network. The discrete patterns are stored in the ARMA subsytem buffers (Fig. 1.) and are shifted forward with propagation of input/output sampling. Output from an ARMA subsystem are sets of input and output patterns (X, Y) which contain information about system evolution:

$$X = \{\vec{y}(k), \vec{y}(k-\Delta), \ldots, \vec{x}(k-\tau), \vec{x}(k-\Delta-\tau), \ldots\}$$
$$Y = \{\vec{y}(k+\Delta)\}, \qquad \Delta = n \cdot t_s \qquad (3)$$

where t_s is sampling time, τ is a process delay, n is an integer, and Δ is a time difference. The ARMA approximation increases dimension of input vector, and also due to large number of on-line measured process variables and high frequency of sampling produces very large data logs. These data contain significant level of random noise due to imperfect measurements, systematic errors could be present because of incorrectly calibrated instruments, and as a result of process characteristics there can also be an appreciable collinearity between variables. Such data, if directly used for ANN training, would lead to very complex network with too many neurons difficult to optimize, and adaptation to random information inevitably leads to the loss of ANN predictive power. Therefore, ARMA output is fed to a statistical preprocessor for noise elimination and significant pattern compression. This task can be effectively conducted by PCA — principal composition analysis (Hutton, 1992; Geladi and Kowalski; 1986). Compression is accomplished by projections from high dimension input X and output spaces Y to corresponding low dimension target spaces T_X and T_Y. The projections are defined by the linear mappings:

$$X \cdot P_X = T_X, \qquad Y \cdot P_Y = T_Y \tag{4}$$

where P_X and P_Y are constant matrixes composed of the corresponding principal component vectors (loadings) $\vec{p}_x$ and $\vec{p}_y$. These vectors are determined from minimization of the corresponding covariances:

$$\begin{aligned} \vec{p}_x &: \min \|Y^T \cdot X - (Y^T \cdot X \cdot \vec{p}_x^T) \cdot \vec{p}_x^T\| \\ \vec{p}_y &: \min \|Y - (Y \cdot \vec{p}_y) \cdot \vec{p}_y^T\| \end{aligned} \tag{5}$$

The target space of input patterns T_X is tilted toward output patterns, i.e. minimized is the covariance of projection of X onto Y to provide maximum correlation between input and output targets. The minimization problems (5) are solved by the QR algorithm for eigenvector calculations. It is often sufficient to take only first few, r, principal components associated with the largest eigenvalues to account for almost all variance presented in patterns. Reduction of dimension can be very significant, such as in a case of a typical industrial application in biotechnology where measured data usually have 1–5 % level of relative error. Rejected information is considered to be measurement random noise and collinear data. Besides significant reduction in dimension and pattern compression, the PCA produces pattern targets which are mutually orthonormal resulting in further simplification and reduction of neural network structure. In the first stage, the outputs from PCA are applied to training of neural network, and later input targets are applied to network prediction. Original output variables Y are restored from ANN predicted output targets T_Y by the simple inverse mapping:

$$\begin{aligned} T_Y &= NN(T_X) \\ Y &= T_Y \cdot P_Y^T \end{aligned} \tag{6}$$

Input patterns to NN are outputs from the PCA module, i.e. targets which are mutually orthogonal. Due to orthogonality patterns are decoupled and the network model function NN has a simple parallel structure with r pairs of corresponding input and outputs:

$$NN = \sum_{i=1}^{i=r} NN_i, \qquad \vec{t}_{yi} = NN_i(\vec{t}_{xi}) \tag{7}$$

Each network subunit NN_i is a single input — single otput system (SISO) constructed by three layers (input, single hidden, and output) with static neurons defined by the sigmoid activation function:

$$\begin{aligned} y &= \frac{1}{1 + \exp[-(\vec{w}^T \cdot \vec{x} + \theta)]} \\ \vec{\beta} &= (\vec{w}_j, \theta_j), \qquad j = 1, \ldots, n_n \end{aligned} \tag{8}$$

where $\vec{w}_j$ are weighing coefficients, θ_j are sensitivity thresholds, n_n is the total number of neurons in each subunit NN_i, and $\vec{\beta}$ is the vector of all parameters in each subunit, which need to be adapted in training. Decoupling of patterns results in separated training, i.e. each i-th subunit is trained separately by minimization of the sum of squares (variance) of differences between predicted and measured test patterns:

$$\sigma_i^2(\vec{\beta}) = \frac{1}{2} \cdot \|NN_i(\vec{t}_{xi}) - \vec{t}_{yi}\|^2 \tag{9}$$

For the purpose of minimization Ribiére–Polack conjugate gradient algorithm with Powell modification is applied. It is applied in a batch type of data processing suitable for relatively small scale problems which can be implemented on a standard PC computer. The gradient is evaluated by the back propagation δ algorithm (Rumelhart and McClelland, 1989). At each iteration neuron parameters are calculated by the one dimensional search procedure:

$$\vec{\beta}_{l+1} = \min_g \sigma^2(\vec{\beta}_l + g \cdot \vec{d}_l) \tag{10}$$

where $\vec{d}$ is the vector of search direction, and g is the gain factor. The search direction is a linear combination of the current gradient and the direction from the previous iteration:

$$\vec{d}_l = -\frac{\partial}{\partial \vec{\beta}_l} \sigma^2(\vec{\beta}_l) + \gamma_l \cdot \vec{d}_{l-1} \tag{11}$$

$$\begin{aligned} \gamma_l &= \max\left\{0, \chi^{-1} \cdot \left(\frac{\partial}{\partial \vec{\beta}} \sigma^2(\vec{\beta}_l)^T - \frac{\partial}{\partial \vec{\beta}} \sigma^2(\vec{\beta}_{l-1})\right)\right\}, \\ \chi &= \left\| \frac{\partial}{\partial \vec{\beta}} \sigma^2(\vec{\beta}_{l-1}) \right\| \end{aligned} \tag{12}$$

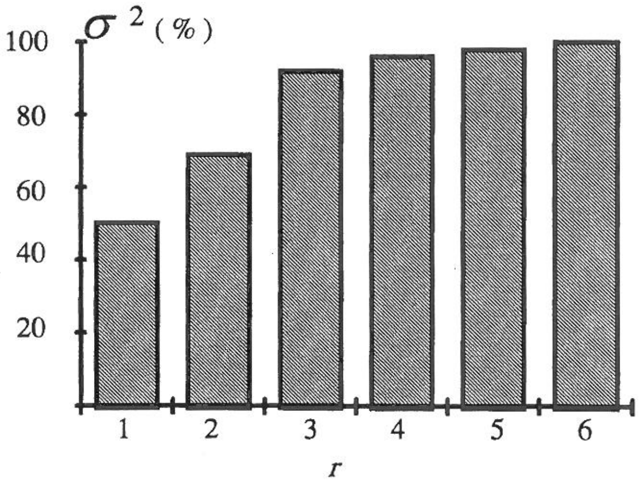

Fig. 2. Fractions of variance σ^2 in % accounted by the number r of principal components.

Critical step is the one dimensional minimization in (10). Here the second order approximation algorithm is applied. At each iteration step l there are such consecutive values $g_1 < g_2 < g_3$ that inequality $\sigma^2(g_1) > \sigma^2(g_2) < \sigma^2(g_3)$ is satisfied. The optimal value of the gain factor is calculated from:

$$g_1 = -\frac{1}{2} \cdot \{(\sigma(g_1)-\sigma(g_2))\cdot(g_1^2-g_3^2) -(\sigma(g_1)-\sigma(g_3))\cdot(g_1^2-g_2^2)\} / \{(\sigma(g_1)-\sigma(g_3))\cdot(g_1-g_2) -(\sigma(g_1)-\sigma(g_2))\cdot(g_1-g_3)\} \tag{13}$$

Each training procedure is evaluated by determination of the predicted residual sum of squares PRESS (Hutton, 1992) with untrained patterns, or by cross-validation with full available set of data. Selection of optimal structure is found by reiteration of training procedure with variable number of neurons on a hidden layer until minimal PRESS is found.

3. Case study: Neural network model of a bioreactor

Application of the proposed neural network structure to modeling and control of an industrial deep jet bioreactor for production of baker's yeast is studied in this work. The process is monitored by on-line measurement of the following process input variables: feeding rates of sources of chemical elements (q–C, q–N, q–P); and the state variables: broth level (H), temperature (T), pH, dissolved oxygen concentration (DO_2) and partial pressure of ethanol (p–EtOH). The biomass (x) is measured off-line. During the fermentation, time span of 15 h and on-line data were sampled and recorded each minute. Each data file contained approximately about 10 000 data points. For each fermentation file principal component analysis was performed. A typical result is presented on Fig. 2. It shows that the complex fermentation profiles can be very effectively compressed by projection to a low dimensional (r) space of principal directions. When fermentation data for 10 on-line measured variables are p rojected to a onedimensional space (line, $r = 1$) it accounts for 52 %, projection on two principal directions (plain, $r = 2$) improves to 70 %, and projection to the three dimensional space ($r = 3$) accounts for 95 % of total variance present in the data set. Presented result indicates that a compression factor of 2/3 can be achieved, i.e. three orthogonal target variables can replace original 10 process variables with an 5 % error of approximation, which is the assumed level of random measur ement error. The significant compression results from a strong collinearity between process data, due to an almost pseudo steady state operation. In the example of feeding rate of carbon source (q–C , molasses) presented in Fig. 3. the quality of approximation is demonstrated by the principal components. With one component ($r = 1$) a result which is close to the original variable is obtained after the transient period of from 3 to 5 h of fermentation. In the

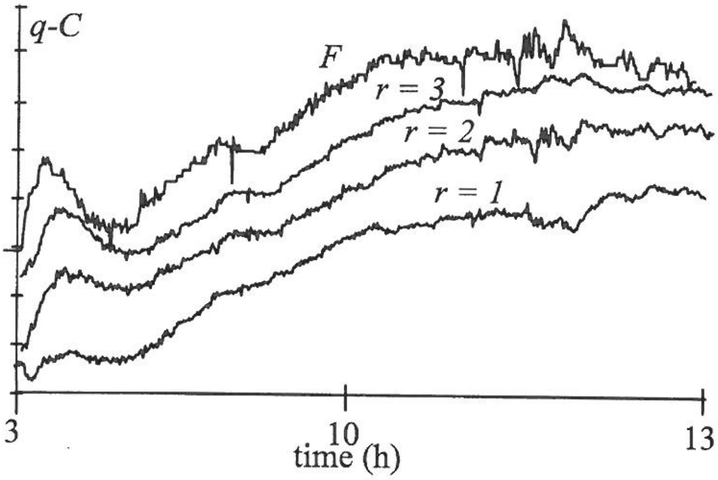

Fig. 3. Feeding profile q–C of carbon source (molasses) measured during fermentation (F), and reconstructed from the first three principal components ($r = 1, 2, 3$). The curves are shifted for a constant value to enable better comparison of dynamic features between experimental and model data.

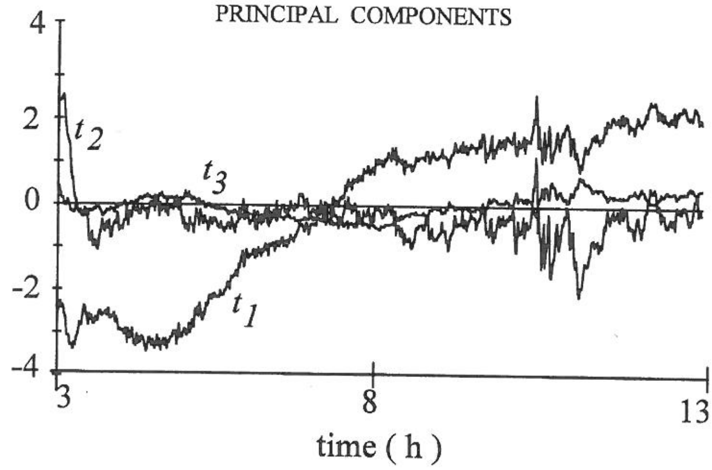

Fig. 4. Time profiles of the first three principal components or targets: t_1 , t_2 and t_3. All process data are scaled for zero mean and unity variance.

almost stationary phase, from 5 to 13 h, approximation is very close. With the approximation of 3 principal components the whole feeding profile is closely reproduced and the high frequency noise is mostly suppressed. Time courses of the individual first three target variables (t_1, t_2, t_3) are given in Fig. 4. The values are spread within the range $(-4, 4)$ around 0, which corresponds to $\pm 4\sigma$ range of the standard normal distribution $N(0, 1)$. The first target t_1 variable covers the whole range, the second t_2 variable has appreciable deviation only in the range of transient state (3–5 h), while the third t_3 variable already acts like a random function with small amplitude around zero.

Feed forward parallel ANN systems are developed with separate modules for the control variables: pH, p–EtOH, and x. For training data sets with several thousand points selected among fermentations with different feeding profiles are used. The first ordered ARMA approximation of process dynamics (almost pseudo stationary process) is applied. It is important to point out that the training set included the fermentation with the unstable oscillatory time course, shown in Fig. 5. for the output variable p–EtOH. Each ANN module has a single hidden layer with four neurons and the first three target variables are fed as inputs. The total number of neurons in the network is relatively low, only 24, which is

			AVERAGE RELATIVE ERROR%	
RUN	TRAINED	BEHAVIOR	ITERATIONS	
			155	1000
76	YES	Oscillatory	0.98	0.82
74	NO	Oscillatory	2.7	3.96
90	NO	Stationary	3.3	3.1

Table 1: ANN model accuracy for trained and untrained patterns.

the result of an effective pattern compression. All neural networks were tested on fermentation data which were not used for training. A typical summary of the results is given in the Table 1. Predictions of the neural network for the key contro l variable, q–EtOH, is presented in Fig. 5. and in Fig. 6. High predictivity of the developed neural network is demonstrated by the abilty that trained in oscillatory regime, Fig. 5., it can also predict behavior during pseudo stationary state presented in Fig. 6.

Average relative errors calculated for single fermentations are in the range from 1–5 % (Table. 1). Efficacy of the used structure can be also inferred from the fast convergence in adaptation. Number of required iterations, given by the equations (10–13), is also relatively low. Required time for training of a single neural module on a standard PC 486 is about 1 h. Increase of number of iterations could bring small improvement, but it may also deteriorate prediction with untrained data, for example as Run 74. in Table 1.

Fig. 7. presents the result of neural network for the inverse model. A network is trained for prediction of forward prediction of feeding rate of nitrogen q–N (ammonia) which supplies yeast for synthesis of biomass and is used for regulation of pH. An inverse model is developed for the use of adaptive pH regulation.

Significant improvements of the proposed structure are illustrated by a detailed exposition of an analytical example (Kurtanjek 1995). It is proven that a model derived from orthogonal decomposition and dimension reduction eliminates variables with high noise to information ratio and avoids biased parameter estimation.

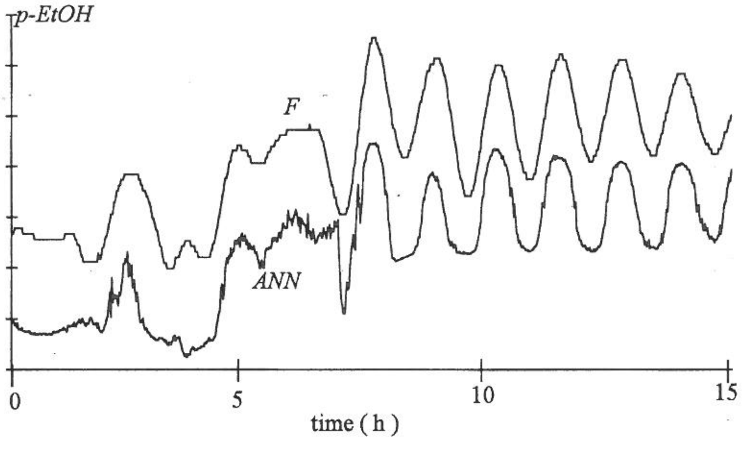

Fig. 5. Oscillatory profiles of ethanol pressure p-EtOH measured during fermentation F and predicted by the artificial neural network ANN. Shown data are used for the network training set. The curves are shifted for a constant value to enable better comparison of dynamic features between experimental and model data.

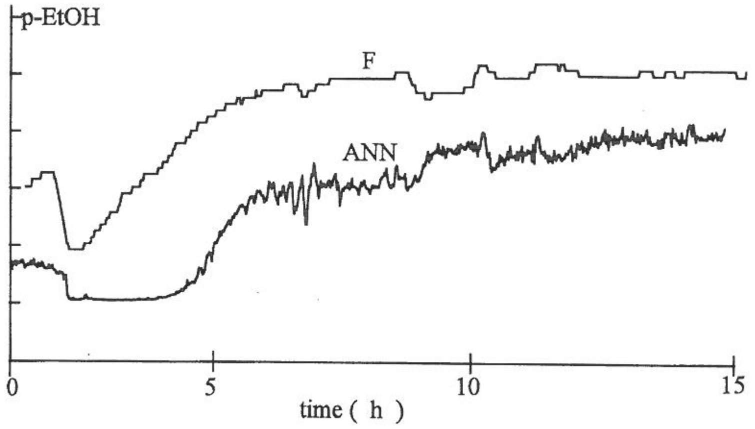

Fig. 6. Pseudo steady state profiles of ethanol pressure p–EtOH measured during fermentation F and predicted by the artificial neural network ANN. Shown data are not present in the network training set. The curves are shifted for a constant value to enable better comparison of dynamic features between experimental and model data.

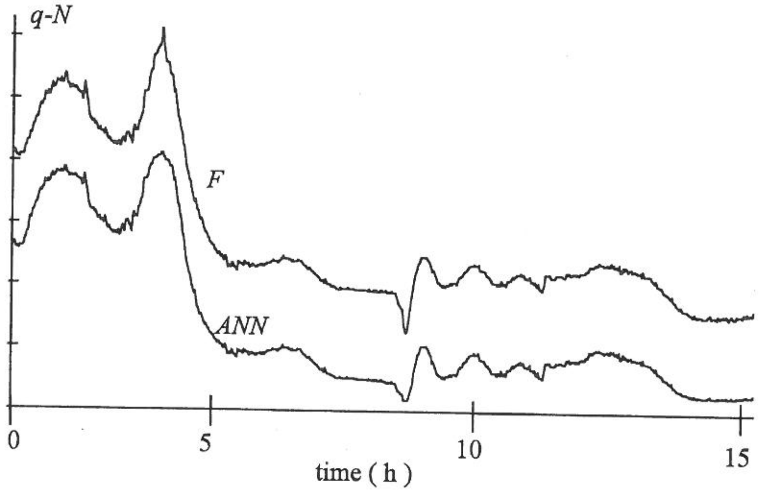

Fig. 7. Feeding profiles of nitrogen source q–N measured during fermentation F and predicted by the artificial neural network ANN. Shown data are not present in the network training set. The curves are shifted for a constant value to enable better comparison of dynamic features between experimental and model data.

4. Conclusion

The work proposes a structure of neural network system dedicated for modeling of dynamic systems applicable in process engineering, such as monitoring and control of chemical and biochemical reactors. It has a modular structure composed of the following modules: ARMA for approximation of process dynamics, PCA for pattern compression and noise reduction, and ANN for nonlinear mapping of input and output patterns.

The model is tested on data obtained in industrial production of baker's yeast in 40 m^3 deep jet reactor. Due to almost pseudo stationary character of the process, it is shown that a first order ARMA approximation is sufficient to approximate dynamics, and that projection from the space of dimension ten of process variables to space of dimesion three of principal directions can account for 95 % of variance.

Due to pattern compression and modular structure, neural network has simple structure and modules can be trained separately. Effecient Ribiéra–Polack–Powell's method of conjugate gradient technique is applied. As the noise has been reduced in PCA module, objective function has better properties, i.e. local minimuma are smoothed out, and network optimization can be effectively performed on a standard PC 486 computer.

Developed neural network can have important use in process engineering applications for online monitoring of unmeasured variables, adaptive control, fault diagnosis and many others. Such neural network modules can be included in a complex software integrated with analytical algorithms, expert systems and fuzzy logic algorithms for control of industrial processes.

Acknowledgment

This work was financially supported by the Croatian Ministry of Science and Technology, project no: 4–07–017.

5. References

N. BATH and T. J. MCAVOY, (1990), "Use of neural nets for dynamic modelling and control of chemical process systems", *Computers and Chemical Engineering*, 14, 573–583.

S. P. CHITRA, (1993), "Use neural networks for problem solving", *Chemical Engineering Progress*, 88 (4) 44.

P. GELADI and B. R. KOWALSKI, (1986), "Partial least squares regression: A tutorial", *Analytica ChemicalActa*, 185, 1–17.

L. V. HUTTON', (1992), "Using statistics to assess the performance of neural network classifiers", *Johns HopkinsAPL Technical Digest*, 13 (2) 291.

M. N. KARIM and S. L. RIVIERA, (1991), "Comparison of feedforward and recurrent neural networks for bioprocess state estimation", *Computers and Chemical Engineering*, 13, p 369.

Ž. KURTANJEK, (1993), "Modelling and control by artificial neural networks in biotechnology", *Computers and Chemical Engineering*, 18, p 627.

Ž. KURTANJEK, (1994), "Mathematical modeling of bioprocesses by neural networks", *Food Technology and Biotechnology Revue*, 32 (1) 27–33.

Ž. KURTANJEK, (1995), "Singular value and principal component decomposition for process data modeling", Proceedings of KOREMA 1995 (accepted for publication).

L. LJUNG and T. SÖDERSTRÖM, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA., 1987.

D. C. PSICHOGIOS and L. H. UNGAR, (1992), "A hybrid neural network first principles approach to process modelling", *American Institute of Chemical Engineering Journal*, 38, 1449–1511.

E. D. RUMELHART, J. MCCLELLAND and THE PDP RESEARCH GROUP, *Parallel Distributed Processing*, The MIT Press, Cambridge, MA., 1989.

J. SCHUBERT, R. SIMUTIS, M. DORS, I. HAVLIK and A. LUBERT, (1994), "Bioprocess optimization and control: Application of hybrid modelling", *Journal of biotechnology*, 35, 51–68.

B. E. YDSTIE, (1990), "Forecasting and control using adaptive connectionist networks", *Computers and Chemical Engineering*, 14, 583–599.

J. ZUPAN and J. GASTEIGER, "Neural Networks for Chemists", VCH Verlagsgesellschaft mbH, Weinheim, 1993.

Received: November, 1994
Accepted: March, 1995

Contact address:

Želimir Kurtanjek
Faculty of Food Technology and Biotechnology
University of Zagreb
Pierottijeva 6
Zagreb, Croatia
Phone: +385/ 1 440 463
FAX: +385/ 1 418 230
E-mail (Internet): zkurt@mapbf.pbfrng.hr

ŽELIMIR KURTANJEK is associate professor of chemical engineering at the Faculty of Food Technology and Biotechnology, University of Zagreb, Croatia. He holds a Dipl. Eng. degree in physics from The Faculty of Natural Sciences at the University of Zagreb, and a Ph.D. degree in chemical engineering from the Department of Chemical Engineering, University of Houston, USA. At the University of Zagreb he teaches mathematical modeling and process control to students of biotechnology and chemical engineering. His main interest has been the study of limit cycle behavior and related surface structures on catalyst surface, modeling of processes in food engineering, modeling and on-line identification of biochemical reactors, optimization of fed-batch operations, and application of artificial neural networks in adaptive control of bioreactors. He is a member of AIChE (American Institute of Chemical Engineers) and of the European Federation on Biotechnology, WP for Measurement and Control.