Chemical Engineering Journal 159 (2010) 195–202

Contents lists available at ScienceDirect

Chemical Engineering Journal

journal homepage: www.elsevier.com/locate/cej

A methodology for modeling batch reactors using generalized dynamic neural networks

Moein Navvab Kashani, Shahrokh Shahhosseini*

Process Simulation and Control Research Laboratory, School of Chemical Engineering, Iran University of Science and Technology (IUST), Narmak, Tehran, Iran

ARTICLE INFO

Article history:
Received 15 May 2009
Received in revised form 11 February 2010
Accepted 25 February 2010

Keywords:
Modeling
Batch reactors
Dynamic neural network
Principle component analysis (PCA)
Sequencing batch reactor (SBR)

ABSTRACT

This paper presents a methodology based on the application of dynamic artificial neural networks (DANNs) for modeling batch reactors. The network structure was designed by a specific method, called leave-one-out cross-validation. In order to reduce the number of input parameters, the multiway principal component analysis (MPCA) was employed. As a case study, sequencing batch reactor was selected to examine the suggested procedure. The results of DANN model were compared to the experimental data, extracted from the literature. Different statistical tools were used as the evaluation criteria for this comparison. The relative error of training and testing sets were 2.11% and 2.6%, respectively. The regression between the network outputs and the experimental data was more than 0.95. Therefore, the model developed in this work has an acceptable generalization capability and accuracy. In addition, it was proved that the implementation of MPCA with dynamic neural network could enhance the model performance. Furthermore, the comparison between the DANN model predictions with those of a mechanistic model revealed that the recommended model was over two and half times more accurate.

1. Introduction

The use of artificial neural networks (ANNs) has been demonstrated to be successful in a number of applications involving function approximation, due to their ability to map highly nonlinear and time depending behaviors [1]. Neural networks can be classified into dynamic and static categories. Dynamic networks are generally more powerful. Since dynamic networks have memory, they can be trained to learn sequential or time varying patterns [2].

Principle component analysis (PCA) is a tool for data compression and information extraction in order to find different combinations of the variables or factors that describe major trends in a data set [3]. The maximum variability of the original multivariate data set is represented by the first principal component (PC), and the second one represents the maximum variances of the residual data set. Then, the third one represents the most important variability of the next residual data set, and so forth. Statheropoulos et al. and Dong and McAvoy have described the algorithm of PCA in detail in their papers [4,5].

Relative to continuous processes, batch data have an added dimension of the batch number in addition to the measured variables and sampling times. Therefore, the basic method of conventional PCA is not directly applicable to batch processes. Nomikos and MacGregor presented the MPCA approach for batch processes, which is an extension of PCA for three-dimensional batch data. The objective of MPCA is to decompose a three-way matrix ($\underline{\boldsymbol{X}}$) into a large two-dimensional matrix ($\mathbf{X}$) [6].

Sequencing batch reactor (SBR) is among the most complicated batch processes. SBRs have made significant progress in the past two decades with increasing use in the wastewater treatment. In its simplest form, a SBR system consists of a single vessel in which the periods of filling, reacting, settling, drawing and idling sequentially take place [7].

A mathematical model has been developed for the SBR activated sludge system by Sun, which is based on the standard activated sludge models of Lawrence and McCarty and IAWPRC [7]. The most significant advantage of this model is that it expresses soluble microbial products (SMP) formed from substrate metabolism and released from activated sludge decay. It also classifies different basic chemical oxygen demand (COD) categories based on their biodegradability. Moreover, total COD is divided into two parts: soluble and insoluble. The soluble part is further subdivided into easy to biodegradable substrates (EBS), difficult to biodegradable substrates (DBS) and biologically inert organic materials (IOM). The classification of COD makes it possible to describe the degradation process of organic substrates more precisely. The EBS degradation followed Monod kinetics and DBS degradation followed first order kinetics. There are several variables as properties, kinetic and stoichiometric parameters in the Sun's model [7]. The modeling of bioprocesses such as SBR is traditionally based on balance equations together with rate equations for microbial growth, substrate consumption and product formation. Since microbial reactions are

* Corresponding author. Tel.: +98 21 77240540–50x2701.
E-mail address: shahrokh@iust.ac.ir (S. Shahhosseini).

1385-8947/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cej.2010.02.053

nonlinear, time-variable with a complex nature, empirical modeling of bioreactors has shown some limitations [7–9].

Aguado et al. compared different models with the use of principal component regression (PCR), partial least squares (PLS) and artificial neural networks [10–12]. Zhang combined the individual neural networks to form a multiple network, termed as a bootstrap aggregated network (BAGNET), to improve the robustness and estimation accuracy of nonlinear models built from a limited amount of data with principal components using PCR to calculate the combination weights [13,14]. The other studies indicate that consideration of statistical principles in the ANN model building process may improve the modeling performance [15]. For instance, principal components analysis can be used for pruning ANNs and improving nonlinear mapping [16,17]. The use of ANNs in combination with PCA has been shown to have some benefits [17,18].

The aim of this research is to introduce a detailed procedure for modeling the batch reactors with dynamic neural networks instead of mechanistic mathematical models. The network architecture was selected by a specific method, called leave-one-out cross-validation (LOOCV), which enhances the generalization ability of the model [2]. The delay between the inputs and outputs of the dynamic network was calculated using correlation coefficient matrix [2]. In order to combine input parameters and reduce the number of them, MPCA technique was used. As a case study, a SBR neural network model was developed using the proposed methodology. The results of this model were compared with those of a mechanistic model and the experimental data.

2. Modeling

2.1. Batch process data and unfolding methods

To clarify the nature of the available data, consider a typical batch run data as shown in matrix **X**. The multivariate data can be organized in *J* variables and *K* samples (observations) per batch:

$$\mathbf{X}=\begin{bmatrix} x_{11} & x_{12} & . & . & . & x_{1J} \\ x_{21} & x_{22} & . & . & . & x_{2J} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_{K1} & x_{K2} & . & . & . & x_{KJ} \end{bmatrix} \quad j=1,2,\ldots,J \text{ and } k=1,2,\ldots,K \tag{1}$$

Similar data exist for other batch runs, where each batch is numbered from 1 to *I* (*i* = 1, 2, ..., *I*). Batch numbers can be added to the matrix as its third dimension. Therefore, all the data can be summarized in the $\underline{\boldsymbol{X}}$ ($I\times J\times K$) array as illustrated in Fig. 1. The data of several batch runs can be organized in a three-dimensional matrix ($\underline{\boldsymbol{X}}$). A method, which can decompose the three-dimensional $\underline{\boldsymbol{X}}$ into a large two-dimensional matrix (**X**) is MPCA, which is statistically and algorithmically consistent with the principal component analysis and has the same goals and benefits [6,19,20,21,22].

Fig. 1 illustrates how a three-way data matrix ($\underline{\boldsymbol{X}}$) can be decomposed to a two-way data matrix **X**. Different batch runs are organized along the vertical axis, the measured variables along the horizontal axis, and their time evolution occupies the third dimension. Each horizontal slice of this array is a ($J\times K$) data matrix representing the time histories or trajectories for all variables of a single batch (i). Each vertical slice is a ($I\times J$) matrix representing the values of all the variables for all of the batches at a time (*k*) [6,23,24].

Six possible methods of unfolding a three-way data array $\underline{\boldsymbol{X}}$ are described in Table 1 [25]. $\underline{\boldsymbol{X}}$ contains vertical slices (I) side by side to the right, starting with one, corresponding to the first time interval. Each of the six possible rearrangements of the data array $\underline{\boldsymbol{X}}$ corresponds to looking at a different type of variability.

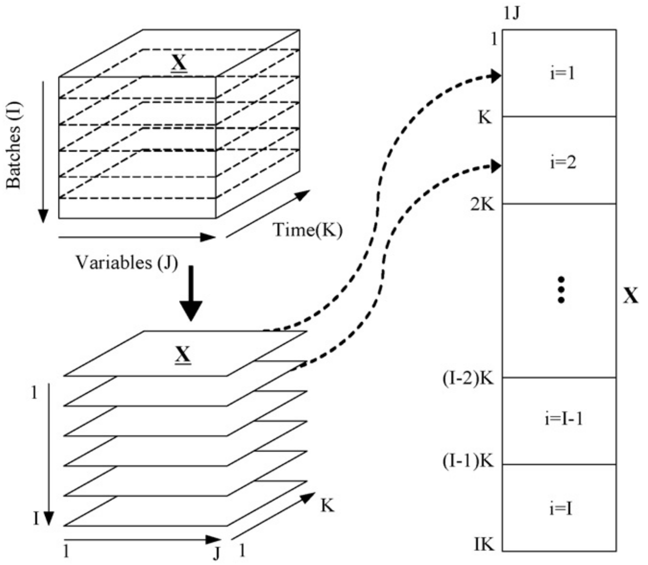

Fig. 1. Decomposing three-way batch process data to two-way array with variable wise unfolding method type A.

Type A unfolding is a method to change the three-way array into a two-way matrix of size ($IK\times J$) by preserving the variable direction, as shown in Fig. 1 [26,27]. It was used in this research. The batch and time wise unfolding were not considered in order to avoid model over fitting [28].

2.2. Dynamic neural network architecture

In this study, an itemized methodology has been applied to select the optimized structure of the dynamic neural networks with generalization ability in order to model batch reactors.

2.2.1. Topology of dynamic neural network

The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network, with feedback connections including particular layers of the network. Since the true output is attainable during the training of the network, the true output was fed to the network instead of feeding back the estimated output, as shown in Fig. 2. Where, $y(t)$ is the true output or the target of the network and $u(t)$ and $\hat{y}(t)$ are the independent input and predicted output vector of the network, respectively. Applying $y(t)$ increases the accuracy of the network. In addition, the resulting network has an entirely feed forward architecture, and static back propagation can be utilized for training [2].

2.2.2. Neural network design with the aid of cross-validation

When designing a neural network model, one of the main purposes is to attain acceptable generalization ability. To obtain this goal, selecting the right number of layers and neurons in the net-

Table 1
Types of unfolding the three-way data array.

Type	Direction of unfolding	Structure of matrix **X**	Dimension of matrix **X** in SBR process
A	Variable	$IK\times J$	154 × 12
B	Time	$JI\times K$	168 × 11
C	Time	$IJ\times K$	168 × 11
D	Batch	$I\times KJ$	14 × 132
E	Batch	$I\times JK$	14 × 132
F	Variable	$J\times IK$	12 × 154

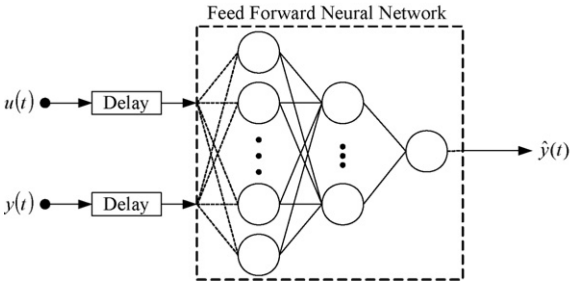

Fig. 2. Series-parallel architecture of NARX network.

work structure is crucial. Since it has been reported in several references that any complex function can be approximated by only one layer, under the circumstances that, the adequate degrees of freedom can be supplied by changing the number of hidden neurons, the model was built with one hidden layer [2,29].

At this point, different network structures created by varying the number of neurons in the hidden layer, and leave-one-out cross-validation technique were used to find the optimal number of the hidden neurons. Fig. 3 shows how the experimental data of 14 batches are divided to training, validation and testing subsets according to the leave-one-out cross-validation approach.

Therefore, in each trial two values of RMSE were calculated. The first was $\overline{RMSE\_t}$ that is the average of root mean squared error for training parts of one trial and the other is *RMSE_loocv*, that is the root mean squared error for one batch that was left out for validation according to *leave-one-out cross-validation method*. After this method is repeated for all of the *N* batches, $\overline{\overline{RMSE\_t}}$ and $\overline{RMSE\_loocv}$ are calculated that represent the average of $\overline{RMSE\_t}$ for training parts and the average of *RMSE_loocv* for all of the batches that were left out for validation of all trials, respectively. Fig. 4 illustrates a flowchart of this procedure. It shows early stopping concept was used to stop the training. In this manner, the network training is stopped, upon the error of validation batches increases for a specified number of consecutive iterations (here 6 iterations).

Finally, by plotting $\overline{\overline{RMSE\_t}}$ and $\overline{RMSE\_loocv}$ versus the number of hidden neurons, the best number of hidden neurons was found. The plots indicated a gradual reduction of $\overline{\overline{RMSE\_t}}$ with the increase in the number of hidden neurons. However, the plot of $\overline{RMSE\_loocv}$ had a minimum, which corresponded to an optimal number of hidden neurons.

2.3. Selection of input variables and implementation of MPCA

Table 2 demonstrates the characteristic, kinetic and stoichiometric variables used in Sun's model. More details could be found in the reference published by Sun [7]. It is clear from this table that many parameters are needed to be estimated to apply Sun's model for SBR. Sometimes, it is arduous to utilize these kind of models since they are highly mathematical or complicated in their expressions that their parameters are difficult to estimate.

In this research, in order to reduce the number of these variables and introduce compressed newer ones as input vectors of DANN, a preprocessing procedure was implemented on the variables of SBR model presented by Sun [7] for which a flowchart is presented in Fig. 5. In agreement with Fig. 5, after removing the constant variables during an operational cycle, the correlation analysis was carried out. By this analysis, the input variables with a remarkable linkage between them and output variables are selected to estimate the output parameters in DANN.

In accordance to the series-parallel architecture of NARX network (Fig. 2), there are two different categories of input vectors for this structure. $y(t)$ as dependent input (regressed output) variable vector, is the first category. This vector contains S and VSS concentrations in this work. The input vector, $y(t)$, is produced when the three-dimensional matrix of ($\underline{\boldsymbol{X}}$) with the size of (14 × 2 × 11) unfolds to a two-dimensional array with the size of (154 × 2).

$u(t)$ in the position of independent input (exogenous input) variable vector, is the second category. The MPCA algorithm was applied to a SBR three-way data array for this kind of variables. The dimensions of the array were 14 × 12 × 11. In this array, $K = 11$ that is the number of time instants throughout the batch (sample), $J = 12$ that is the number of process variables, and $I = 14$ that is the historical data set. The three-way matrix ($\underline{\boldsymbol{X}}$) of the batch process data was unfolded variable wise (type A of Table 1) to produce a two-dimensional data matrix (**X**) with the size of (154 × 12) as demonstrated in Fig. 1. Afterwards, PCA was performed on the matrix (**X**) to form the new compressed variables entitled the principle component (PC) variables.

Fig. 6 depicts the variance represented by each PC and the total variance. The model with 6 PCs, represent 96.92% of the total variability of all previous 12 variables. Therefore, the model was built using only 6 principal components as independent input vector. Consequently, the resulting array was a matrix with the size of (154 × 6). From Table 3 it is evident that using one principal component can represent 39.33% of the total variance, while employing

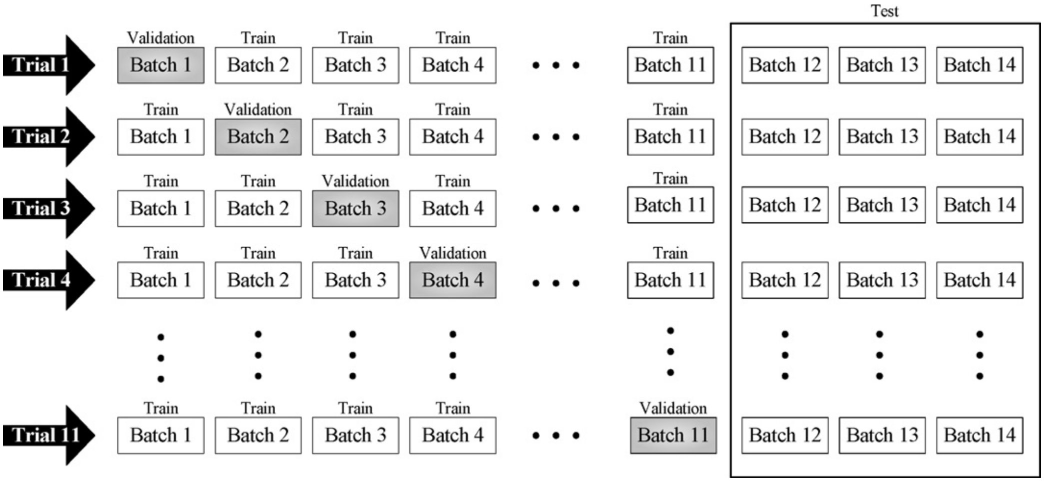

Fig. 3. Demonstration of leave-one-out cross-validation method and divisions of the 14 batches for training, validation and testing subsets.

Table 2
Variables of SBR model presented by Sun [7].

No	Parameters	Typical values	Units
1	Soluble COD concentration in influent: S_0	14.13–65.9	mg/l
2	Concentration of VSS in influent: X_{V0}	657–10,502	mg/l
3	Influent flow rate: Q	5.3	h^{-1}
4	Volume of the reactor: V	4	l
5	Fraction of EBS: f_1	0.31 ± 0.011	Dimensionless
6	Fraction of DBS: f_2	0.65 ± 0.021	Dimensionless
7	Fraction of IOM: f_3	0.02 ± 0.006	Dimensionless
8	Rate constants for EBS: k_1	11–13	d^{-1}
9	Rate constants for DBS: k_2	0.07–0.09	d^{-1}
10	Half velocity constant for EBS degradation: K_{s1}	87–100	mg/l
11	Half velocity constant for DBS degradation: K_{s2}	0.68–1.76	mg/l
12	Fractions of EBS released from sludge endogenous decay: f_{x1}	0.93–0.96	Dimensionless
13	Fractions of DBS released from sludge endogenous decay: f_{x2}	0.0	Dimensionless
14	Fractions of IOM released from sludge endogenous decay: f_{x3}	0.04–0.07	Dimensionless
15	Conversion coefficient for soluble COD from VSS: γ_X	0.01–0.05	Dimensionless
16	Conversion coefficient for soluble COD from UAP: γ_S	0.3–1.8	Dimensionless
17	Rate constant of endogenous decay: k_e	0.051 ± 0.01	d^{-1}
18	Yield factor of mg biomass/mg COD: Y	0.33 ± 0.025	Dimensionless
19	Constant for switch function to limit the degradation of DBS: K_0	20–40	mg/l

two or three PCs can represent 61.65% or 77.78% of the variance and so on.

2.4. Dynamic neural network architecture for modeling the SBR

Six steps were carried out to design a dynamic neural network for SBR modeling as described below.

2.4.1. Selection of DANN structure

The network contains three layers as shown in Fig. 7. The training data set consists of 14 batch data; each DANN was trained with 10 batches and tested with three other batches. The remained batch was used for validation.

In the training step, the ANN can converge to a poor local minimum. For this reason, the training was started with 30 different and random initial weights for each ANN. The initial weights were chosen in the ranges from −0.6 to 0.6 in accordance with the recommendations of Al-Shayji [30]. By testing several different initial conditions, the robust network performance was verified.

Another important factor in ANN design is the type of transfer (activation) functions. To select the most suitable transfer function for the system, different kinds of activation functions were examined including linear, sigmoid and hyperbolic tangent functions. At the end, the hyperbolic tangent function gave the best performance compared to other activation functions in the learning process and was chosen to be the activation function as given below:

$$f(x) = \tan\, h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (2)$$

2.4.2. Delay of dynamic network

One of the main steps in the procedure of dynamic neural network design is finding lag time between the output vector $y(t)$ and input vector $u(t)$. This is carried out with the aid of correlation coefficient matrix. The correlation coefficient matrix represents the normalized measure of the strength of linear relationship between the variables in an input matrix, in which rows are observations and columns are variables (each column represents a separate quantity) [2]. In this study, the input matrix for this analysis was 11 × 7. Its rows were observations in different sampling times and its columns were the variables. Correlation analysis was performed, whereas the first column was either S or VSS and the rest were PC1 to PC6. Eventually, the inspections of the obtained correlation coefficient matrix (7 × 7) revealed that $y(t)$ was one time step behind of $u(t)$.

Table 3
Principle component extraction.

Principal component number	Variance	Variance captured by this PC (%)	Accumulative variance (%)
PC1	5.899	39.331	39.331
PC2	3.348	22.320	61.651
PC3	2.419	16.131	77.783
PC4	1.434	9.564	87.348
PC5	0.974	6.499	93.487
PC6	0.461	3.079	96.927

2.4.3. Training algorithm and network performance criteria

The performance of neural network model was evaluated in terms of root mean square (RMSE) criterion. The RMSE performance index was defined by the following equation:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{p=1}^{P}(y_i - \hat{y}_i)^2}{P}} \qquad (3)$$

where P is the number of input patterns, $\hat{y}_i$ is the predicted network output and y_i is the measured (desired) output values. During the training process, the root mean squared error (RMSE) in the function (Eq. (3)) was minimized by adjusting the network parameters.

Another widely used criterion is the coefficient of determination (R^2) or regression analysis. R^2 provides a measure of the strength of the correlation. The best known formula to calculate this index is the Pearson product–moment correlation coefficient as follows:

$$R^2 = \left[\frac{\sum_p(\hat{y}_i - \hat{y}_{i,ave})(y_i - y_{i,ave})}{\sqrt{\sum_p(\hat{y}_i - \hat{y}_{i,ave})}\sqrt{\sum_p(y_i - y_{i,ave})}}\right]^2 \qquad (4)$$

Levenberg–Marquardt training method was applied in order to improve convergence speed and performance of the network [2,31–34].

The DANNs were trained in an on-line mode. That is, the weights were updated after each training pattern was presented. The order of presenting training patterns to the DANNs was randomized for each epoch to reduce the probability of converging in a local minimum as recommended by Haykin [2].

2.4.4. Training stop criterion for generalization improvement

Over fitting is a major problem that occurs during neural network training. The error on the training set can be driven to a very small value. However, when new data are presented to the network, the error is larger. One of the effective methods to ensure generalization and avoid over fitting is early stopping. The available data in this method are divided into three subsets. The initial subset is the training set, which is employed for calculating the gradient and updating the network weights and biases. The validation subset is the second. The error on the validation set is observed while the training of the network. The validation error reduces through the early phase of training, similar to the training set error. However, the error on the validation set usually begins to move upward, when the network begins to over fit the data [2].

Fig. 8 displays the over fitting region. When the validation error increased for a specified number of iterations (10 iterations), the training was stopped and the weights and the biases at the point of minimum validation error were recorded.

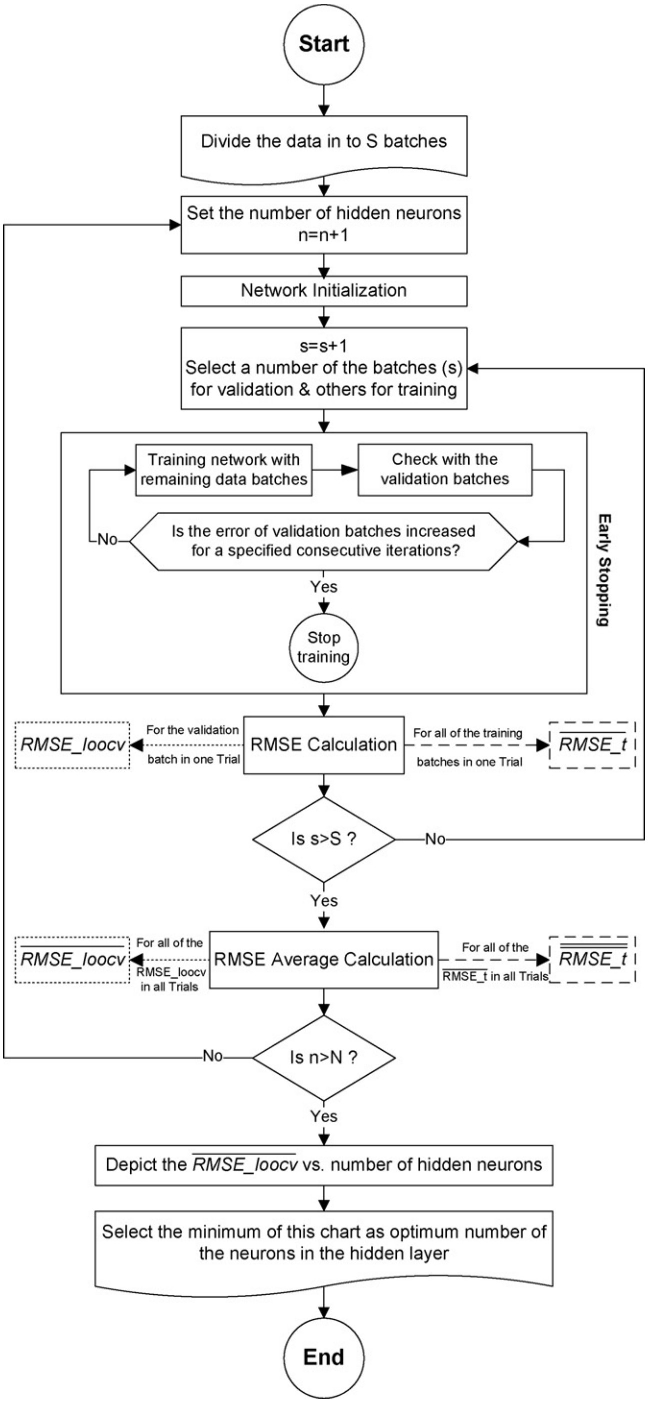

Fig. 4. Illustration of the procedure to find an optimal number of hidden neurons according to leave-one-out cross-validation method.

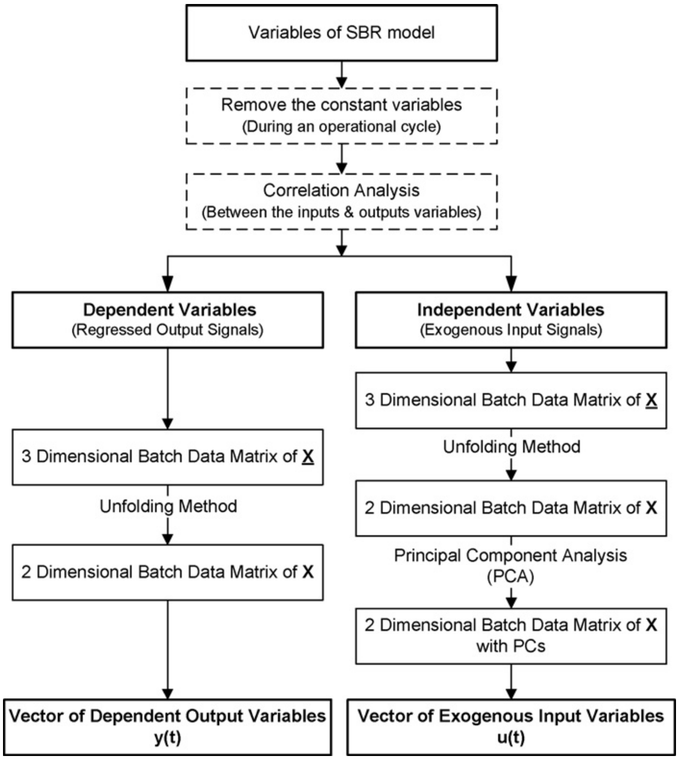
Variables of SBR model
Remove the constant variables
(During an operational cycle)
Correlation Analysis
(Between the inputs & outputs variables)
Dependent Variables
(Regressed Output Signals)
Independent Variables
(Exogenous Input Signals)
3 Dimensional Batch Data Matrix of X
Unfolding Method
2 Dimensional Batch Data Matrix of X
3 Dimensional Batch Data Matrix of X
Unfolding Method
2 Dimensional Batch Data Matrix of X
Principal Component Analysis
(PCA)
2 Dimensional Batch Data Matrix of X
with PCs
Vector of Dependent Output Variables
y(t)
Vector of Exogenous Input Variables
u(t)

Fig. 5. The details of preprocessing procedure for converting SBR model variables to input vectors of DANN model.

3. Results and discussion

Two values of RMSE were obtained from Section 2.2.2 and Fig. 4. The $\overline{\overline{RMSE\_t}}$ describes the fitting to the training data and $\overline{RMSE\_loocv}$ depicts the generalization capability acquired by leave-one-out cross-validation technique. Fig. 9(a) and (b) displays these two RMSE versus the number of neurons, respectively. In the selection of the number of hidden neurons, priority was given to that num-

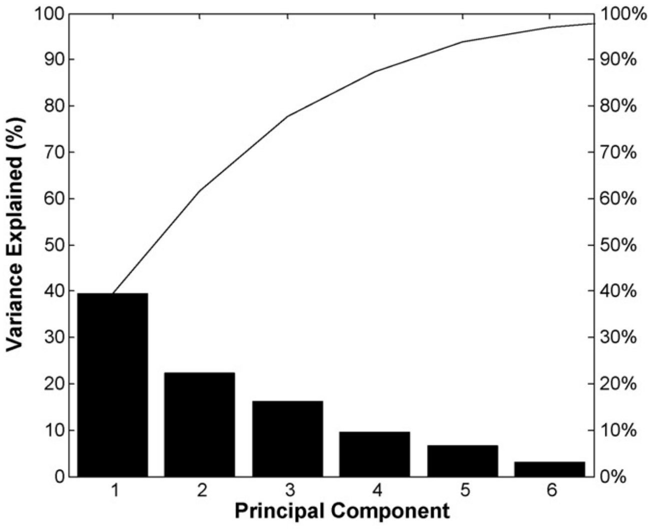

Fig. 6. The proportion of each PC to represent the total variance.

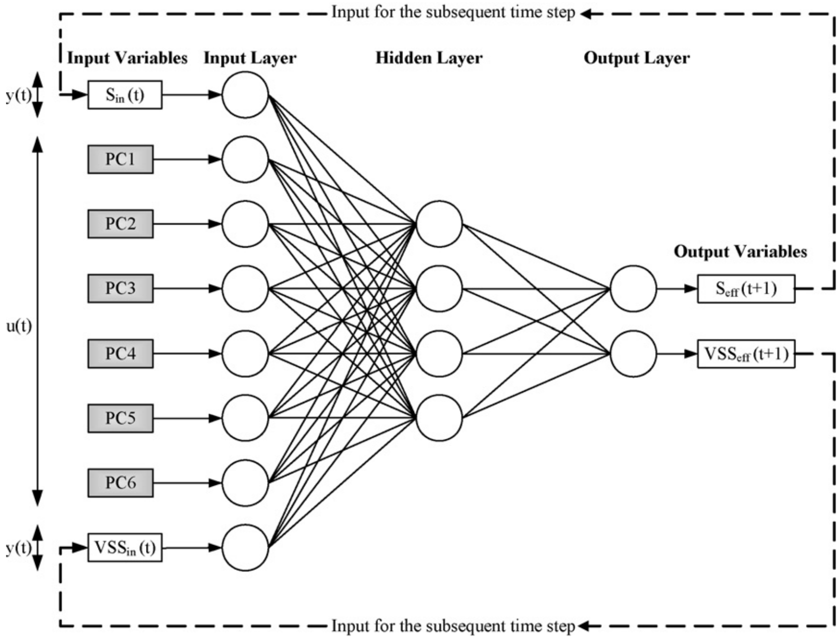

Fig. 7. Dynamic artificial neural network topology for modeling the SBR (bias connections to the neurons are omitted to simplify the presentation).

ber of neurons in which $\overline{RMSE\_loocv}$ met its minimum as shown in Fig. 9(b). This figure indicates that the minimum occurs when there are 4 hidden neurons.

In Fig. 10(a) and (b), a regression analysis between the predicted outputs and the experimental values was carried out. As it is evident from these figures, for both S and VSS concentrations the slope and the correlation coefficient are relatively close to 1, pointing out an acceptable fit. Moreover, these results illustrate that this DANN model is generalized and accurate model to anticipate the behavior of an operational cycle of SBR.

In Figs. 11 and 12, the predicted values of dynamic neural networks and Sun's SBR model are compared to the experimental data. Overall observation of these figures reveals that the DANN predicts the behavior of the SBR better than Sun's model.

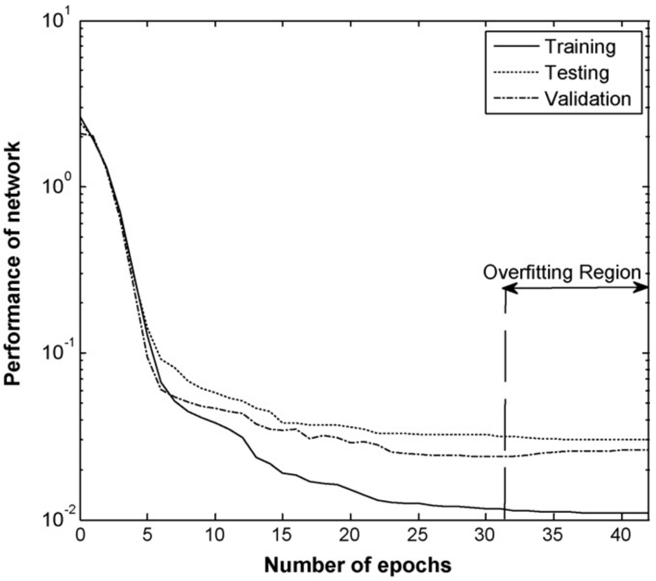

Fig. 8. Early stopping criteria and the trends of training, validation and testing subsets.

The first 60 min of the experiments was filling period, during which the substrates were fed into the reactor. Fig. 11 displays that the soluble COD curves quickly ascend at the beginning of the process. This happened since the rate of substrate degradation was less than its feed rate and a period of time was needed for the

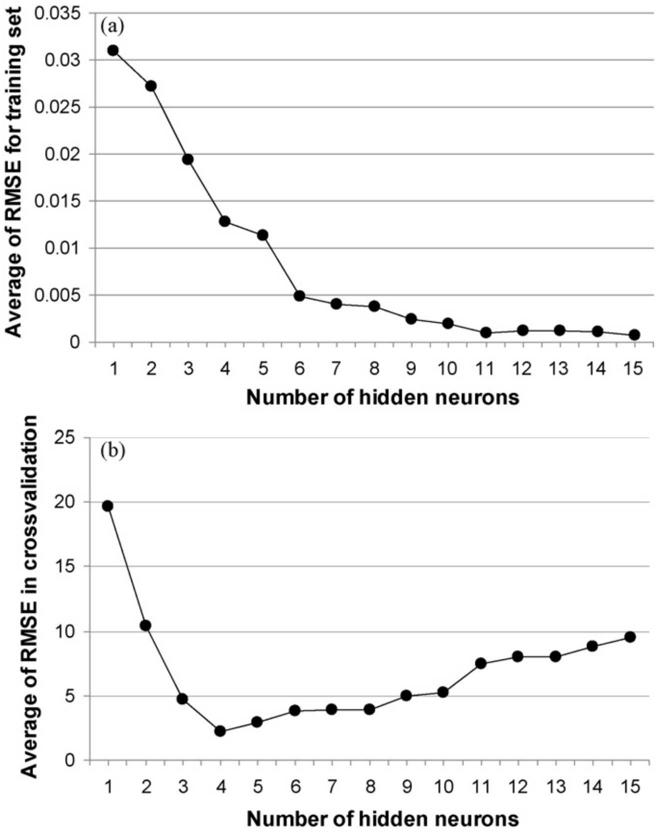

Fig. 9. The effects of the number of hidden neurons on the average RMSE (a) $\overline{\overline{RMSE\_t}}$, fitting to the training data (b) $\overline{RMSE\_loocv}$, generalization capability acquired by leave-one-out cross-validation technique.

Table 4
Error analysis obtained for training and testing sets data.

Data subset	Mean absolute error		RMSE		R^2		Relative error (%)		
	S	VSS	S	VSS	S	VSS	S	VSS	Overall
Training set	1.851	75.66	0.182	0.112	0.975	0.983	8.77	2.032	2.11
Testing set	1.26	84	0.318	0.217	0.951	0.962	4.289	2.256	2.6

microorganisms to acclimate to the substrates. Afterwards, substrate degradation was quickened reducing the feed accumulation rate, leading to a stationary phase of substrate concentration curve. This tendency insisted until filling stage was ended. In Fig. 11, there is a sharp decline in the curve after the termination of substrate feeding.

Fig. 12 depicts the experimental data as well as neural network and Sun's model predictions for VSS concentration. It shows the trends of neural network predictions and experimental data are close, indicating the model ability to predict VSS concentration changes within the operational cycle.

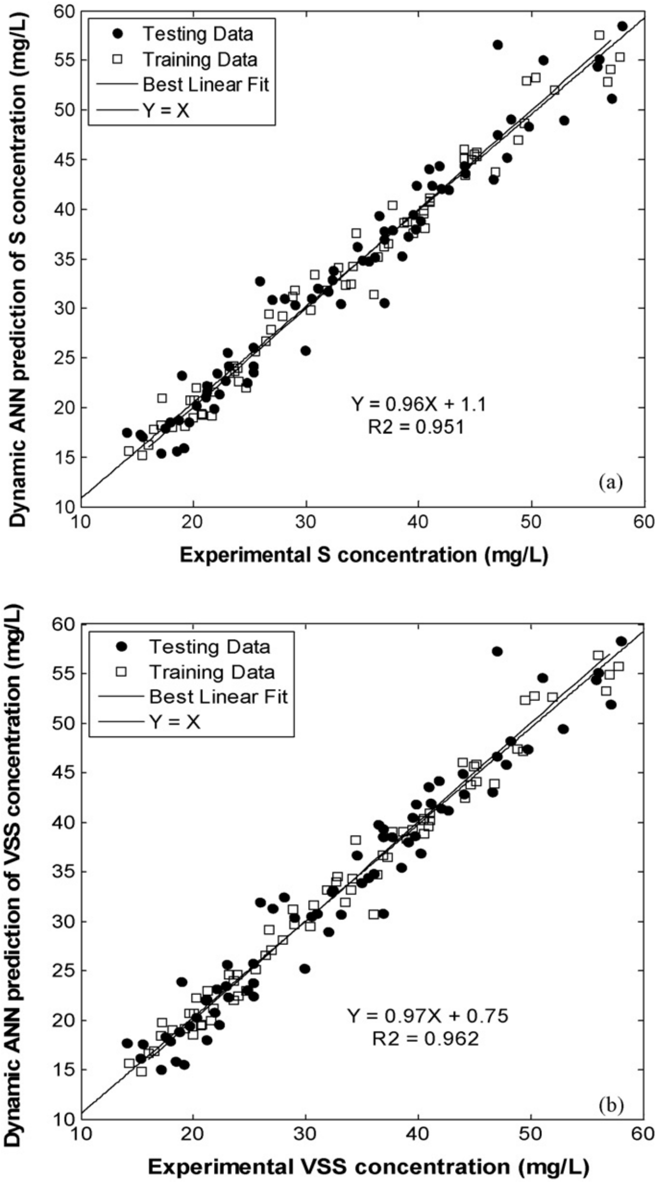

Fig. 10. Regression analysis between predicted outputs and the experimental values for: (a) S concentration and (b) VSS concentration.

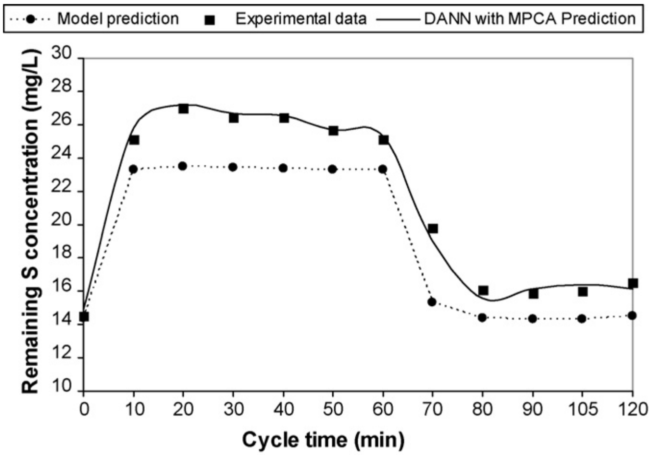

Fig. 11. Comparison between the DANN predicted values for remaining soluble COD, with Sun's SBR model and experimental data during an operational cycle.

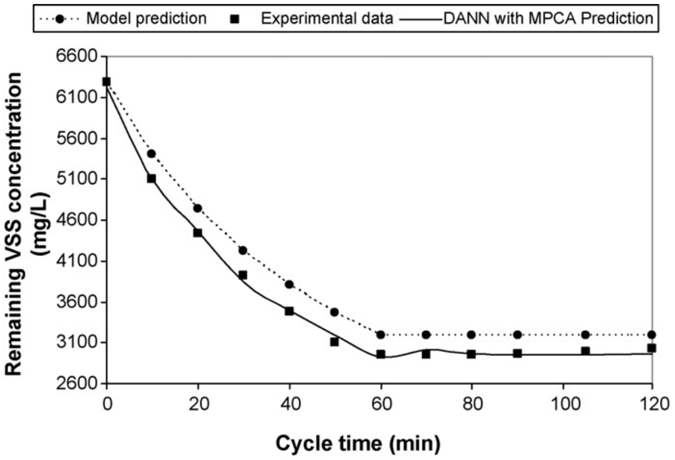

Fig. 12. Comparison between the DANN predicted values for VSS concentration, with Sun's model and experimental data during an operational cycle.

Table 4 shows the relative errors of the DANN model for training and testing sets were 2.11% and 2.6%, respectively, indicating the generalization ability of the model. This table also shows the amounts of regression analysis between the predicted output and the experimental data are reasonable for both training and testing data sets, proving the accuracy of the suggested SBR model. Table 5 displays the overall relative error for DANN model is over two and half times lower than that of the Sun's model.

Table 5
Comparison between DANN and Sun's model.

SBR models	Mean absolute error		Relative error (%)		
	S	VSS	S	VSS	Overall
Generalized DANN model based on MPCA	1.26	84	4.289	2.256	2.6
Sun's model	2.2	256.08	10.46	6.87	6.89

4. Conclusions

In this study, a methodology based on the application of dynamic artificial neural networks for modeling the batch rectors has been presented. The data of 14 SBR batches were organized in a three-way array and unfolded to a two-dimensional matrix. The matrix was then treated using MPCA method.

The results of training and testing data sets were determined and compared in terms of the RMSE and performing a regression analysis that confirmed the accuracy and generalization ability of the network. Simulation results of DANN model and Sun's model were compared to the experimental data of a SBR, extracted from the literature. The comparison indicated that the DANN model could predict the process far better than Sun's model, despite it is much more complicated than DANN model.

References

[1] S. Al-Asheh, F.S. Mjalli, H.E. Alfadala, Forecasting influent–effluent wastewater treatment plants using time series analysis and artificial neural network, Chem. Prod. Process Model. 2 (2007), Article 3.

[2] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd edition, Prentice Hall Co., NJ, USA, 1999.

[3] B. Wise, N. Gallagher, S. Watts, D. White, G. Barna, A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process, J. Chemometr. 13 (1999) 379–396.

[4] M. Statheropoulos, A. Pappa, P. Karamertzanis, H.L.C. Meuzelaar, Noise reduction of fast, repetitive GC/MS measurements using principal component analysis (PCA), Anal. Chim. Acta 401 (1999) 35–43.

[5] D. Dong, T.J. McAvoy, Nonlinear principal component analysis-based on principal curves and neural networks, Comput. Chem. Eng. 20 (1996) 65–67.

[6] P. Nomikos, J.F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE J. 40 (1994) 1361–1375.

[7] T. Sun, Modeling a SBR waste treatment system, PhD Thesis, University of Ottawa, Ottawa, Canada, 1996.

[8] D.S. Lee, J.M. Park, Neural network modeling for on-line estimation of nutrient dynamics in a sequentially-operated batch reactor, J. Biotechnol. 75 (1999) 229–239.

[9] M. Cote, B.P.A. Grandijean, P. Lessard, J. Yhibault, Dynamic modeling of the activated sludge process: improving prediction using neural networks, Water Res. 29 (1995) 995–1004.

[10] D. Aguado, A. Ferrer, A. Seco, J. Ferrer, Comparison of different predictive models for nutrient estimation in sequencing batch reactor for wastewater treatment, Chemometr. Intell. Lab. Syst. 84 (2006) 75–81.

[11] D. Aguado, A. Ferrer, A. Seco, J. Ferrer, Using unfolding-PCA for batch-to-batch start-up process understanding and steady-state identification in a sequencing batch reactor, J. Chemometr. 22 (2007) 81–90.

[12] D. Aguado, M. Zarzo, A. Seco, A. Ferrer, Process understanding of a wastewater batch reactor with block-wise PLS, Environmetrics 18 (2007) 551–560.

[13] J. Zhang, Developing robust non-linear models through bootstrap aggregated neural networks, Neurocomputing 25 (1999) 93–113.

[14] J. Zhang, Inferential estimation of polymer quality using bootstrap aggregated neural networks, Neural Netw. 12 (1999) 927–938.

[15] H.R. Maier, G.C. Dandy, Neural networks for prediction and forecasting of water resources variables: a review of modeling issues and applications, Environ. Model. Softw. 15 (2000) 101–124.

[16] P.P. Kanjilal, On the application of orthogonal transformation for design and analysis of feedforward networks, IEEE Trans. Neural Netw. 6 (1995) 1061–1070.

[17] M. Kompany-Zared, A. Massoumi, Sh. Pezeshk-Zadeh, Simultaneous spectrophotometric determination of Fe and Ni with xylenol orange using principal component analysis and artificial neural networks in some industrial samples, Talanta 48 (1999) 283–292.

[18] T.R. Holcomb, M. Morari, PLS/neural networks, Comput. Chem. Eng. 16 (1992) 393–411.

[19] P. Nomikos, J. Mac Gregor, Multivariate SPC charts for monitoring batch processes, Technometrics 37 (1995) 41–59.

[20] K. Kosanovich, M. Piovoso, K. Dahl, J. MacGregor, P. Nomikos, Multiway, PCA applied to an industrial batch process, in: Proceeding of the American Control Conference, 1994.

[21] C. Duchesne, T. Kourti, J.F. MacGregor, Multivariate Monitoring of startups, restarts and grade transitions using projection methods, in: Proceeding of the American Control Conference, 2003.

[22] R. Henrion, N-way principal component analysis: theory, algorithms and applications, Chemometr. Intell. Lab. Syst. 25 (1994) 1–23.

[23] S. Wold, N. Kettaneh, H. Friden, A. Holmberg, Modeling and diagnostics of batch processes and analogous kinetic experiments, Chemometr. Intell. Lab. Syst. 44 (1998) 331–340.

[24] J.A. Westerhuis, P.M. Coenegracht, Multivariate modeling of the pharmaceutical two-step process of wet granulation and tableting with multiblock PLS, J. Chemometr. 11 (1997) 379–392.

[25] J.A. Westerhuis, T. Kourti, J.F. Mac Gregor, Comparing alternative approaches for multivariate statistical analysis of batch process data, J. Chemometr. 13 (1999) 397–413.

[26] C. Undey, A. Cinar, Statistical monitoring of multistage, multiphase batch processes, IEEE Control Syst. Mag. 22 (2002) 40–52.

[27] C Undey, S. Ertunc, A. Cinar, Online batch/fed-batch process performance monitoring, quality prediction, and variable-contribution analysis for diagnosis, Am. Chem. Soc. (2003).

[28] T. Masters, Practical Neural Network Recipes in C++, Academic Press, San Diego, 1993.

[29] D. Aguado, J. Ferrer, A. Seco, T. Montoya, A methodology for Sequencing Batch Reactor identification with artificial neural networks: a case study, Comput. Chem. Eng. 33 (2009) 465–472.

[30] K.A. Al-Shayji, Modeling, simulation and optimization of large-scale commercial desalination plant, PhD Thesis, Virginia Polytechnic Institute and State University, VA, USA, 1998.

[31] D.R. Baughman, Y.A. Liu, Neural Network in Bioprocessing and Chemical Engineering, Academic Press, San Diego, CA, USA, 1995.

[32] F.S. Mjalli, S. Al-Asheh, H.E. Alfadaja, Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance, J. Environ. Manage. 83 (2007) 329–338.

[33] J.A.D. Rodrigues, A.G. Pinto, S.M. Ratusznei, M. Zaiat, R. Gedraite, Enhancement of the performance of an anaerobic sequencing batch reactor treatment low-strength wastewater through implementation of a variable stirring rate program, Braz. J. Chem. Eng. 21 (2004) 423–434.

[34] D. Baetens, enhanced biological phosphorus removal: modeling and experimental design, PhD Thesis, Ghent University, Belgium, 2000.