Artificial neural networks model for the prediction of steady state phenol biodegradation in a pulsed plate bioreactor

K Vidya Shetty,* Santosh Nandennavar and G Srinikethan

Department of Chemical Engineering, National Institute of Technology Karnataka, Surathkal, PO Srinivasanagar-575025, Karnataka, India

Abstract

BACKGROUND: A recent innovation in fixed film bioreactors is the pulsed plate bioreactor (PPBR) with immobilized cells. The successful development of a theoretical model for this reactor relies on the knowledge of several parameters, which may vary with the process conditions. It may also be a time-consuming and costly task because of their nonlinear nature. Artificial neural networks (ANN) offer the potential of a generic approach to the modeling of nonlinear systems.

RESULTS: A feedforward ANN based model for the prediction of steady state percentage degradation of phenol in a PPBR by immobilized cells of *Nocardia hydrocarbonoxydans* (NCIM 2386) during continuous biodegradation has been developed to correlate the steady state percentage degradation with the flow rate, influent phenol concentration and vibrational velocity (amplitude × frequency). The model used two hidden layers and 53 parameters (weights and biases). The network model was then compared with a Multiple Regression Analysis (MRA) model, derived from the same training data. Further these two models were used to predict the percentage degradation of phenol for blind test data.

CONCLUSIONS: The performance of the ANN model was superior to that of the MRA model and was found to be an efficient data-driven tool to predict the performance of a PPBR for phenol biodegradation.

Keywords: pulsed plate bioreactor; vibrational velocity; biodegradation; *Nocardia hydrocarbonoxydans*; immobilized cells; artificial neural networks

NOTATION

Symbol	Description	Units
Q	flow rate	$mL\,h^{-1}$
S_i	influent phenol concentration	$mg\,L^{-1}$
Vv	vibrational velocity cm	s^{-1}
Y	MRA predicted percentage degradation	–
α_o	constant in MRA model	–
$\alpha_1, \alpha_2, \alpha_3$	linear coefficients in MRA model	–
$\alpha_{11}, \alpha_{22}, \alpha_{33}, \alpha_{12}, \alpha_{23}, \alpha_{13}$	cross product coefficients in MRA model	–
i	number of input neurons in ANN (1to 3)	–
j	number of neurons in first hidden layer of ANN (1 to 4)	–
k	number of neurons in second hidden layer of ANN(1 to 6)	–
w_{ji}	weights for input to first hidden layer in ANN	–
v_{kj}	weights for first hidden layer to second hidden layer	–
u_{ok}	weights for second hidden layer to output layer	–
b^{I}_{j}	bias for first hidden layer neurons	–
b^{II}_{k}	bias for second hidden layer neurons	–
b^{o}	bias for output layer neuron	–
μ	momentum term	–

INTRODUCTION

Phenol is an aromatic compound that occurs naturally in the environment but arises more commonly, and detrimentally, from industrial activities such as petroleum processing, chemical, petrochemical and steel industries, plastic manufacturing and the production of resins. Wastewaters generated by these industries frequently contain high concentrations of phenolic compounds,[1] which represent a serious ecological problem owing to their widespread use, toxicity and occurrence throughout the environment.[2] A water-soluble compound, phenol is generally found

* Correspondence to: K Vidya Shetty, Department of Chemical Engineering, National Institute of Technology Karnataka, Surathkal, PO Srinivasanagar-575025, Karnataka, India
E-mail: vidyaks68@yahoo.com; vidyaks95@nitk.ac.in
(*Received 15 May 2007; revised version received 24 November 2007; accepted 15 December 2007*)
Published online 27 May 2008; DOI: 10.1002/jctb.1892

to contaminate streams, rivers and lakes that are situated near, or are receiving waters from industries. Because of their toxicity to human and marine life, increasingly stringent restrictions have been imposed on the concentrations of these compounds in the wastewater for safe discharge.[3]

Traditionally, efficient treatment of phenolic wastewater can be conducted by either conventional physical–chemical[4–6] or biological techniques.[7,8] However, these treatments are complex and expensive. This situation is triggering the development of new treatment technologies for phenolic wastewater, especially technologies claiming reduced production of additional waste sludge.[9] It has been demonstrated that various toxic organic compounds are not eliminated by conventional biological effluent treatment systems, owing to the presence of relatively high concentrations of easily biodegradable substances. Furthermore, the treatment of small volumes of concentrated toxic compounds at the site of emission, using specific microbial strains and better reactors, is preferable.[10]

Biological films formed by immobilized cells are commonly used in wastewater treatment such as the trickling filter, rotating disc contactor and fluidized bed bioreactor. Fixed film bioreactors exhibit properties that make them preferable to suspended cell systems for many bioprocess applications. These properties include extremely high cell concentration, enhanced cell retention owing to cell immobilization, and increased resistance to toxic shock loading, higher volumetric throughputs owing to the independence of cell growth rate from reactor dilution rate.[11] Immobilization eliminates the costly processes of cell recovery and cell recycle.

A recent innovation in fixed film bioreactors is the pulsed plate bioreactor (PPBR) with immobilized cells.[12,13] A pulsed plate column with the space between the plates packed with glass particles immobilized with the cells of *Nocardia hydrocarbonoxydans* has been used as a bioreactor for the biodegradation of phenol.[12,13] The potential for this column to be used as a bioreactor and the advantages of this bioreactor over other kinds of bioreactor were reported earlier.[12]

Conventionally, analysis of the performance of bioprocesses is based on deterministic mathematical models, which are usually described by a set of differential equations derived from mass balances. The successful development of a theoretical model relies on the availability of good process information.[14] However, to obtain a sufficiently accurate and robust mathematical model for a bioprocess is a time-consuming and costly task owing to the complexity of its physiology and performance. The major challenge is the nonlinear and time-varying nature of such processes.[15] The modeling of nonlinear systems is not simple, and success has been limited to restrictive classes of nonlinear systems.[16] A deterministic mathematical model consisting of mass balance equations at steady state, with complete mixing liquid phase conditions and accounting for simultaneous external film mass transfer, internal diffusion and reaction may be developed to describe the steady state degradation of phenol in a PPBR by *Nocardia hydrocarbonoxydans* immobilized onto glass beads. The use of a theoretical model requires certain parameters, such as biofilm thickness and biofilm density, and these parameters vary with the operating conditions of the reactor. So a correlation may have to be developed relating these parameters with the operating conditions of the bioreactor. Another parameter required is the phenol diffusivity through the biofilm. This may also be a function of operating conditions of the reactor. Estimation of this parameter, although not impossible, may be difficult. Thus, the application of an artificial neural network (ANN) was tried to predict the performance of the bioreactor. The ANN model serves as a black box model and does not require the knowledge of any parameters.

The ANN modeling technique is well suited to fuzzy, noisy, incomplete data and it is a true multiple input multiple output (MIMO) algorithm that has the ability to mimic the human learning process and can store large amounts of information through knowledge indexing. Prediction with ANN is made by learning experimentally generated data or using validated models.[17] Bioprocesses are typical MIMO, unstructured and nonlinear systems, which are not easy to model. The main appeal of ANNs is that they offer the potential of a generic approach to the modeling of nonlinear systems. Several papers deal with the use of ANN models to predict the performance of bioprocesses.[18–23] Balan *et al*.[24] applied an ANN model to predict the extent of degradation of phenol by *Pseudomonas pictorum* at different nutrient compositions under batch conditions. They compared the ANN model prediction with a Multiple Regression Analysis (MRA) model and found that ANN was better at predicting the degradation. Annadurai and Lee[25] used an ANN model for the development of an optimized complex medium for phenol degradation using *P. pictorum* under batch conditions. They tested and compared the efficiency of the model generated by the ANN with results obtained from an established second-order polynomial MRA. Further, they used the two models (ANN and MRA) to predict the percentage degradation of phenol for blind test data. They recommended ANN, based on the performance of both models. In both studies, the ANN was applied to model the biodegradation of phenol under batch conditions. Reports on the application of ANNs, for modeling the biodegradation of phenol in continuous bioreactors or for predicting continuous bioreactor phenol biodegradation performance are scarce. The novelty of the present study lies in the development and application of an ANN based model, to predict the steady state performance of an innovative bioreactor,

J Chem Technol Biotechnol **83**:1181–1189 (2008)
DOI: 10.1002/jctb

such as the PPBR with immobilized cells, for the continuous biodegradation of phenol.

In the present investigation, an attempt has been made to apply an ANN to predict the performance of a PPBR with immobilized cells for continuous biodegradation of phenol in synthetic wastewater under different operating conditions, such as flow rate, influent phenol concentration and vibrational velocity (amplitude×frequency). The efficiency of the ANN was demonstrated by comparing it with MRA model.

MATERIALS AND METHODS

Nocardia hydrocarbonoxydans (NCIM 2386) which was chosen for the present study by virtue of its effectiveness to degrade phenolic waste,[12,13,26] was obtained from NCIM, a division of National Chemical Laboratories, Pune, India. The strains were periodically sub- cultured and acclimatized to required phenol concentrations and then immobilized onto glass beads.

Growth of organisms, nutrient media used, procedures for acclimatization, cell immobilization on glass beads and phenol analysis are described elsewhere.[12]

Experimental bioreactor

The bioreactor is a pulsed plate column with the space between the plates forming a stage, packed with glass particles immobilized with the cells of *Nocardia hydrocarbonoxydans*. The entire plate stack containing five plates and hence four stages, can be pulsed at the required frequency and amplitude. The schematic diagram of the experimental PPBR and the detailed description of the bioreactor and the experimental procedure are given elsewhere.[12] Experiments were performed at different flow rates, phenol concentrations and vibrational velocities indicated in Table 1. Steady state conditions were considered when the phenol concentration in the effluent remained constant for a period of 12 h.

MATHEMATICAL MODEL

Artificial neural networks

The ability of artificial neural networks[27–31] to represent nonlinear systems make them a powerful tool for process modeling and much work has been reported over the last decade. The term 'artificial neural network' originates from research, which attempted to understand, and proposed simple models of, the operation of the human brain.

Consequently, ANNs possess characteristics, in common with the biological system – they consist of numerous simple processing elements (neurons) joined together by variable strength connections (synapses) to form a massively parallel and highly interconnected, information processing system.[27] This gives the ANN several characteristics that are appealing for the modeling of nonlinear systems.

Table 1. Output of ANN and MRA model for the training set

Experimental no.	Q (mL h^{-1})	S_i (mg L^{-1})	V_v (cm s^{-1})	Phenol degradation (%)	ANN output (%)	MRA output (%)
1	800	900	2.35	94.56	94.34	94.15
2	1000	100	4.7	100	100.00	99.09
3	400	800	2.35	97.04	97.04	97.95
4	800	800	2.35	96.62	96.64	95.9
5	400	900	2.35	96.6	96.61	96.6
6	400	300	2.35	99.62	99.69	99.51
7	800	500	2.35	99.16	99.07	99.51
8	400	500	2.35	99.53	100.00	99.08
9	400	500	4.7	100	99.92	97.89
10	1000	900	2.35	93.23	93.46	93.12
11	600	500	2.35	99.2	99.29	99.44
12	400	200	2.35	99.83	99.76	98.78
13	400	200	4.7	99.5	99.65	100.11
14	200	100	4.7	100	100.08	100.78
15	400	300	4.7	99.65	99.64	99.72
16	200	500	4.7	100	99.99	99.321
17	400	800	4.7	91.85	91.84	92.55
18	400	900	2.35	96.6	96.61	96.6
19	1000	300	2.35	99	99.04	99.66
20	800	300	2.35	99.2	99.2	99.47
21	400	100	4.7	100	99.74	100.16
22	600	500	4.7	96.2	96.46	96.59
23	800	500	4.7	94.9	94.78	95.43
24	1000	500	4.7	93.94	93.83	94.4
	RMS error (%)				0.154	0.699
	Average absolute error (%)				0.102	0.538
	Coefficient of correlation (R)				0.9989	0.9605

The artificial neuron

The basic unit of neural networks, the artificial neuron, is much simpler than the biological neuron. Figure 1 shows the basics of an artificial neuron. A neuron with a single n-element input vector is shown in Fig. 1. Here the individual element inputs are multiplied by weights and the weighted values are fed to the summing junction. The neuron has a bias b, which is summed with the weighted inputs (w) to form the net input z. This sum, z, is the argument of the transfer function f. The behavior of an ANN depends on both the weights and the input–output function (transfer function, activation function) that is specified for the units.

Feedforward ANNs with back propagation algorithm

A large number of ANN have been proposed and used in recent years. However, most of the commonly used ANNs for process modeling are layered feedforward neural networks with a back propagation algorithm.[24] The network is operated in two distinct phases called training and recall.[16] When the network is trained, it can be used in the recall mode where the network weights are fixed and it is tested with blind data sets. This testing or validation of a network is a very important step in the development cycle of a nonlinear neural network model. The overall goal is to develop a network, which can emulate the underlying system, which produced the training data.[16]

Feedforward networks mean that, when operating in the recall mode, the flow of information through the network is solely in the forward direction from inputs to outputs as shown in Fig. 2.

Development of ANN model

Training phase

The back propagation algorithm is used in the present study and is the most widely investigated supervised learning algorithm. All modeling studies were carried out using the Levenberg–Marquardt learning rule. In total, 31 data sets were used, and are shown in Table 1 and Table 2, which were obtained from the continuous biodegradation experiments. Among them, 24 data sets were used for training the ANN. The rest were used for the blind test. During segregation of data sets, it was ensured that the training set contained data sets that included the extreme values of all the inputs used, to facilitate applicability of the ANN model within those ranges. Similarly, the validation set may contain input values that have not been seen by the network during training: in this study, some data sets with a vibrational velocity value of 3.525 cm s^{-1} were chosen as part of validation data set, which were not used in any training set.

The output of the bioprocess is used only for training the network. The model predicts the process output using the same input as the process after training. Transformation of inputs or scaling was not required. The goal of network training is to minimize the mean

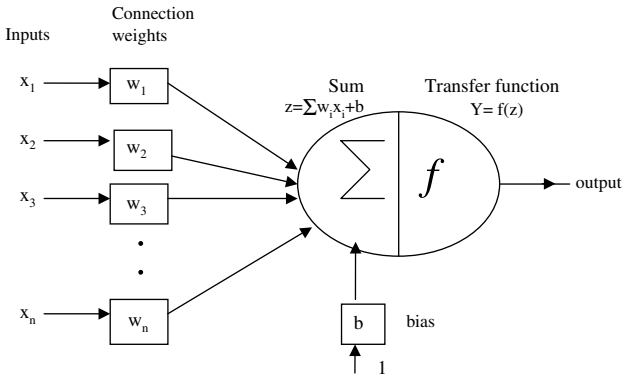

Figure 1. Processing element (Artificial neuron).

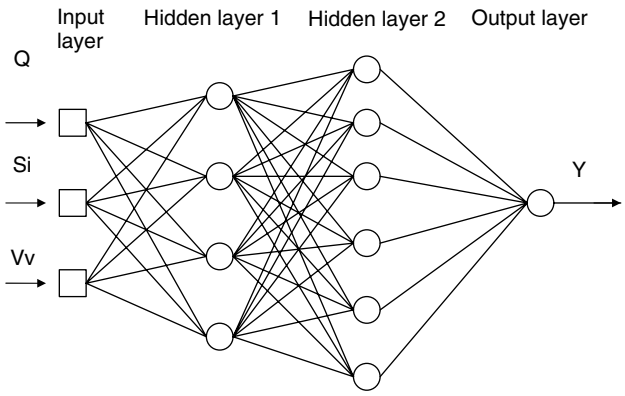

Figure 2. Neural architecture of phenol biodegradation in the pulsed plate bioreactor.

Table 2. Output of ANN and MRA model for blind test data

Experiment no.	Q (mL h^{-1})	S_i (mg L^{-1})	V_v (cm s^{-1})	Phenol degradation (%)	ANN output (%)	MRA output (%)
Data used for network selection and testing						
1	600	300	2.35	99.23	99.41	99.42
2	800	100	4.7	100	99.11	99.31
3	1000	800	2.35	94	94.04	95.07
Data used for validation						
4	400	300	3.525	99.62	99.64	100.48
5	400	800	3.525	91.5	92.58	96.12
6	600	100	4.7	100	99.44	99.67
7	400	200	3.525	99.66	99.72	100.31
		Average absolute error (%)			0.411	1.273

J Chem Technol Biotechnol **83**:1181–1189 (2008)
DOI: 10.1002/jctb

square error (MSE) between the measured value and the neural network output by adjusting its weights and biases. This performance goal was set at 10^{-10} in this study to ensure adequate ANN model training.

A computer program has been written using the Neural Networks Tool Box of MATLAB version 7.0.1 for training and testing the ANN. In the MATLAB code, a standard network was created by calling a MATLAB function that uses a layer initialization function, which initializes a layer's weights and biases according to the Nguyen–Widrow initialization algorithm. This algorithm chooses initial weights and bias values for a layer, in order to distribute the active region of each neuron in the layer approximately evenly across the layer's input space. Advantages over purely random weights and biases are that few neurons are wasted (since all the neurons are in the input space) and training proceeds faster (since each area of the input space has neurons).

The Levenberg–Marquardt Back Propagation (LMBP) training algorithm was adopted to train the neural networks owing to its fast convergence and memory efficiency.[32] The basic error back propagation algorithm (EBP),[16,33] which was used in earlier days, adjusts the weights in the steepest descent direction (negative of the gradient). This is the direction in which the performance function is decreasing most rapidly. In this training algorithm, a learning rate is used to determine the length of the weight update (step size). If the objective function has many local and global optima, the optimal learning rate often changes dramatically during the training process. Trying to train a NN using a constant learning rate is usually a tedious process requiring much trial and error. Although the EBP was a significant milestone in NN research, it has a very poor convergence rate. Many attempts have been made to speed up the EBP algorithm. Commonly known heuristic approaches,[34–37] such as momentum[16] and variable learning rate,[38] lead to only a slight improvement. With batch training, there is no need to use a constant learning rate. In fact, there is no reason to use standard back propagation at all, since vastly more efficient, reliable, and convenient batch training algorithms exist. A significant improvement in performance can be observed by using various second-order approaches, namely Newton's method, conjugate gradient or the Levenberg–Marquardt (LM) optimization technique.[39–42] Among these methods, the LM algorithm is widely accepted as the most efficient in the sense of realization accuracy.[42] It gives a good compromise between the speed of the Newton algorithm and the stability of the steepest descent method. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, the scalar momentum term, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm. This algorithm appears to be the fastest method for training moderate-sized feedforward neural networks (up to several hundred weights). It also has a very efficient MATLAB implementation, since the solution of the matrix equation is a built-in function; thus, its attributes become even more pronounced in a MATLAB setting. MATLAB neural networks toolbox function TRAINLM implements this algorithm. Initially the set of parameters required to train the network are selected. The main parameters selected are maximum number of epochs, performance goal with mean squared error (MSE), minimum performance gradient, initial value for momentum (μ), μ decrease factor, μ increase factor and maximum μ. Momentum μ, allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low-pass filter, momentum allows the network to ignore small features in the error surface. This term ensures that general trends are reinforced and oscillatory behaviour is dampened. Without momentum a network may become stuck in a shallow local minimum. With momentum a network can slide through such a minimum.[16] MATLAB implementation of the algorithm used an automatic adjustment of the Marquardt μ parameter. The initial value for μ is multiplied by the μ decrease factor whenever the performance function is reduced by a step. It is multiplied by the μ increase factor whenever a step would increase the performance function. If μ becomes larger than maximum μ, the algorithm is stopped. The selected parameters are shown in Table 3. The maximum number of epochs (training cycles) was chosen by a trial and error approach. Performance goal and minimum performance gradient were set so as to ensure a model with good performance. All the other parameters were set at their default values given by the MATLAB function TRAINLM. Numbers of training samples are presented to the network before each weight update is carried out. This number of training samples is called the epoch size and is selected randomly from the training set. The epoch size used can be equal to the entire training set or a subset of the training set. In the present study epoch size was taken as 24, which is equal to the entire training set. The number of layers and the number of processing element per layer are important decisions. There is no quantifiable, best answer to the layout of the network for any particular application. In the present study a trial and error method was followed to obtain the network architecture. Network architecture, i.e. the number of hidden layers, number of neurons in each hidden layer and the activation functions, were varied. Each, resultant network architecture was trained with 24 training data sets. Training stops when any of the following conditions occur: the maximum number of epochs is reached, performance has been minimized to the goal, the performance gradient falls below minimum gradient, μ exceeds maximum μ. The trained network

was tested with three of the seven blind data sets, which were not used during training. The network, which gave a coefficient of correlation (R) between the model prediction and experimental results near to 1, for both training data sets (24) and test data sets (3), was considered to be suitable and hence selected. A value of R greater than 0.99 for both cases was selected as a measure of accuracy. Based on this analysis, the neural network model proposed in this study has the architecture shown in Fig. 2.

It contains a back propagation network with an input layer, two hidden layers and an output layer. The NN architecture has three input neurons, four sigmoidal neurons in the first hidden layer, six sigmoidal neurons in the second hidden layer and one linear neuron as output layer. Each connection has a weight associated with it. The input, hidden and output units carry out two calculations shown in Fig. 1. Sigmoid activation functions are used for the hidden layers and a pure linear function is used for the output layer. The input variables for the networks were flow rate (Q), influent phenol concentration (Si) and vibrational velocity (V_v). Output variable is the steady state percentage degradation.

The network shown in Fig. 2, learns by making changes in the weights associated with each connection and bias values associated with each neuron. In the training phase the correlations between various inputs and the corresponding outputs for different input data were identified and learnt by the ANN. Three of the blind data sets, which are not seen by the neural network during the training period, are used in examining the trained network. A smaller error on the test data set or coefficient of correlation between the model prediction and experimental results near to one means the trained network has achieved better generalization.

Prediction phase

The training set (24 data points) and blind set (seven data points) were then fed to the ANN model and all the output values were calculated. Four data points of the seven blind data sets were not used earlier in the test set for network selection. Validity of the ANN model was tested with these four data points.

Table 3. Neural network structure, activation functions and parameters used during training

Network architecture	3-4-6-1
Activation functions (MATLAB function)	Sigmoidal(logsig) – sigmoidal (logsig)-linear(purelin)
Maximum number of epochs	300
Epoch size	24 (=training set data)
Performance goal with MSE	10^{-10}
Minimum Performance gradient	10^{-1000}
Initial value of momentum,μ	0.001
μ decrease factor	0.1
μ increase factor	10
Maximum value of μ	$1 \times 10^{+10}$

Multiple regression analysis

The steady state performance of a PPBR is approximated to the following second-degree polynomial equation:

$$Y = \alpha_o + \alpha_1 Q + \alpha_2 S_i + \alpha_3 V_v + \alpha_{11} Q^2 + \alpha_{22} {S_i}^2 + \alpha_{33} {V_v}^2 + \alpha_{12} Q S_i + \alpha_{23} S_i V_v + \alpha_{13} Q V_v \quad (1)$$

where Y = predicted percentage degradation, α_o = constant, α_1, α_2, α_3 = linear coefficients, α_{11}, α_{22}, α_{33}, α_{12}, α_{23}, α_{13} = cross product coefficients. LABFIT software[43] is used for the estimation of coefficients in the model equation. The software uses the Levenberg–Marquardt algorithm to estimate the coefficients in the model equation. %root mean square error (%RMSE), average absolute error (%) and correlation coefficient were calculated to study the performance of the model. Statistical significance of the model is evaluated using results of the Analysis of Variance (ANOVA) performed in the LABFIT software. The 24 data sets used for training the ANN model were used to fit the MRA model. MRA model validation was done with the seven blind data sets.

RESULTS AND DISCUSSION

Continuous biodegradation experiments were carried out to study the effect of flow rate, influent phenol concentration and vibrational velocity on the performance of a PPBR for biodegradation of phenol. The number of continuous experimental runs was 31 and are shown in Table 1 and Table 2. The 24 experimental data sets were used to train the network and to fit the MRA model. The remaining seven data points unused during training were used to predict the percentage degradation of phenol. For the ANN model, three of the seven data points were also used for selection of the best network architecture. The structure of the NN model was selected using training and testing methods. The accuracy of network prediction was studied by varying the transfer function, number of hidden layers and number of processing elements in each hidden layer. Two hidden layer networks, with three neurons in the input layer, four sigmoidal neurons in the first hidden layer, six sigmoidal neurons in the second hidden layer and one linear neuron in the output layer was found to be suitable for forecasting purposes. The weights and bias associated with the ANN model are shown in Table 4 and Table 5, respectively. The coefficients associated with MRA are shown in Table 6. The ANOVA for evaluation of the second-order MRA model is shown in Table 7. The test statistic has an F value of 18.53. Since the test statistic is much larger than the critical

J Chem Technol Biotechnol **83**:1181–1189 (2008)
DOI: 10.1002/jctb

value of 1.00, the null hypothesis of equal population means was rejected and it is concluded that there is a (statistically) significant difference among the population means. The probability of obtaining an F as large, or larger, than the one calculated from the data is the probability value (P-value). If it is lower than the significance level, then the null hypothesis can be rejected and the outcome is said to be statistically significant. Traditionally, experimenters have used either the 0.05 level (sometimes called the 5% level) or the 0.01 level (1% level) for the significance level, although the choice of levels is largely subjective. The P-value for an F of 18.55, obtained from ANOVA, in this case is less than 0.001, which is less than even the lower significance level of 0.01. So the test statistic is significant at that level. It confirms the adequacy of the MRA model.

Figure 3 shows the plot of experimental and predicted percentage degradation values of the ANN model for the 24 data sets used for training. For all the training data sets, the data points of experimental and predicted percentage degradation values coincide with each other. This indicates perfect training of the network. Table 1 compares the experimental percentage degradation with a multiple nonlinear regression model output and a back propagation NN model output for the 24 data sets used to train the network and to fit the MRA model. The ANN has low %RMSE and average absolute error (%). Correlation coefficient (R) for the ANN model is very near to 1 and is higher than the R-value for the MRA model for 24 training data sets. Analysis of the training data set predictions reveals that the ANN is better and is recommended.

Table 2 shows the comparison of experimental values of percentage degradation with those predicted by the ANN model and the MRA model for the seven test and validation data sets. The error associated with these blind sets is characteristic of the network's ability to generalize. Although 53 parameters are used in the ANN model and 10 coefficients used in the MRA model, the computation time for each model is less than 10^{-5} s, and is insignificant. The ANN model gives lower average absolute error than the MRA model. The ANN model and MRA model predicted values of percentage degradation were compared with the experimental values for all 31 data sets. The coefficient of correlation estimated using these 31 data sets for the ANN

Table 4. ANN weights

w_{ji} (j = 1 to 4; i = 1 to 3)	v_{kj} (k = 1 to 6; j = 1 to 4)	u_{ok} (k = 1 to 6)
	$v_{11} = 1.2910$	$u_{o1} = 14.2166$
$w_{11} = 0.0082$	$v_{12} = 7.4204$	$u_{o2} = -9.0608$
$w_{12} = 0.0218$	$v_{13} = -4.6207$	$u_{o3} = 44.9336$
$w_{13} = 4.2609$	$v_{14} = 2.1271$	$u_{o4} = -12.8629$
$w_{21} = 0.0204$	$v_{21} = -7.5565$	$u_{o5} = 6.9246$
$w_{22} = 0.0242$	$v_{22} = -12.0573$	$u_{o6} = 17.3692$
$w_{23} = 4.0937$	$v_{23} = -8.7869$	
$w_{31} = 0.0023$	$v_{24} = 6.3559$	
$w_{32} = 0.0012$	$v_{31} = -3.3919$	
$w_{33} = 0.0601$	$v_{32} = 10.5125$	
$w_{41} = -2.2134$	$v_{33} = -14.9096$	
$w_{42} = 1.9171$	$v_{34} = -0.4414$	
$w_{43} = -21.3853$	$v_{41} = -18.8981$	
	$v_{42} = 1.5365$	
	$v_{43} = -0.5441$	
	$v_{44} = 7.0610$	
	$v_{51} = -10.1322$	
	$v_{52} = 8.3560$	
	$v_{53} = 8.3916$	
	$v_{54} = 3.7285$	
	$v_{61} = 2.8173$	
	$v_{62} = -10.6528$	
	$v_{63} = -6.3517$	
	$v_{64} = 3.7172$	

Table 5. Bias values for neurons in ANN

b^{I}_{j} (j = 1 to4)	b^{II}_{k} (k = 1 to 6)	b°
$b^{I}_{1} = -31.3661$	$b^{II}_{1} = -0.4193$	48.0682
$b^{I}_{2} = -0.0944$	$b^{II}_{2} = -10.4478$	
$b^{I}_{3} = 1.5054$	$b^{II}_{3} = 7.8764$	
$b^{I}_{4} = 7.5275$	$b^{II}_{4} = 5.7315$	
	$b^{II}_{5} = 8.7528$	
	$b^{II}_{6} = -6.0708$	

Table 6. MRA coefficients

MRA coefficients
$\alpha_{o} = 83.54$
$\alpha_{1} = 4.951 \times 10^{-3}$
$\alpha_{2} = 3.115 \times 10^{-2}$
$\alpha_{3} = 6.632$
$\alpha_{11} = 1.692 \times 10^{-6}$
$\alpha_{22} = -1.729 \times 10^{-5}$
$\alpha_{33} = -0.6277$
$\alpha_{12} = -1.009 \times 10^{-5}$
$\alpha_{23} = -4.7699 \times 10^{-3}$
$\alpha_{13} = -1.72 \times 10^{-3}$

Table 7. ANOVA of the second-order multiple regression model

Source	Sum of squares	Degrees of freedom	Mean square	F-value	P-value
Regression	134.3	9	14.92	18.55	<0.001
Error	11.27	14	0.805		
correlation total	145.57	23			
	R-squared = 0.9226		Adjusted R-squared = 0.8729		

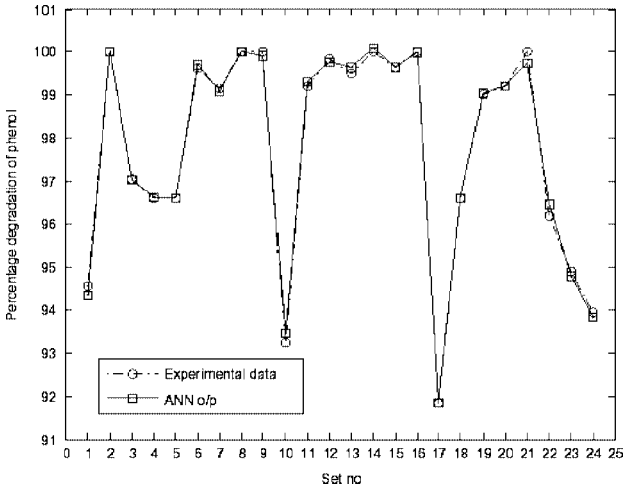

Figure 3. Plot of experimental and ANN predicted values of percentage degradation for training set data.

model was 0.9929 and that for the MRA model was 0.8881.

This indicates that the NN model is far superior to the MRA model in predicting the steady state percentage degradation of phenol in a PPBR. However, as with any black box model, caution is needed in applying this model beyond the range of inputs with which training has been done. As process modeling is a starting point for process optimization, this model can be adopted for the optimization of process operating conditions.

CONCLUSIONS

A back propagation ANN based model for predicting the steady state performance of a PPBR with immobilized cells, has been developed to correlate the steady state percentage degradation with the flow rate, influent phenol concentration and vibrational velocity (amplitude×frequency). The structure of the NN model has been selected using training and testing methods. A two hidden layer network, with three neurons in the input layer, four neurons in the first hidden layer, six neurons in the second hidden layer and one in the output layer was found to be suitable for forecasting purposes. The optimum network for accurate performance prediction was obtained by varying the transfer function, number of hidden layers and number of processing elements in the hidden layer. The accuracy of prediction was studied, and the performance of the NN was compared with that of a nonlinear MRA model and the former was found to be far superior to the latter, as indicated by the higher coefficient of correlation, lower %RMSE and lower average absolute error (%) associated with the predicted values. The NN-based model was found to be an efficient data-driven tool to predict the performance of a PPBR.

REFERENCES

1 Chang YH, Li CT, Chang MC and Shieh WK, Batch phenol degradation by *Candida tropicalis* and its fusant. *Biotechnol. Bioeng* **60**:391–395 (1998).
2 Fava F, Armenante PM and Kafkewitz D, Aerobic degradation and dechlorination of 2-chorophenol, 3-chlorophenol and 4-chlorophenol by a *Pseudomonas pickettii* strain. *Lett Appl Microbiol* **21**:307–312 (1995).
3 Hsien TY and Lin YH, Biodegradation of phenolic wastewater in a fixed biofilm reactor. *Biochem Eng J* **27**:95–103 (2005).
4 Reddy KA, Anand PS and Dasare BD, Sorption of phenolic compounds by porous polymeric adsorbents. *Indian J Environ Health* **31**:197–206 (1989).
5 Throop WM, Alternative methods of phenol wastewater control. *J Hazard Mater* **1**:319–329 (1975–1976).
6 Eisenhower HR, The ozonation of phenolic wastes. *J Water Pollut Control Fed* **40**:1887–1899 (1968).
7 Klein JA and Lee DD, Biological treatment of aqueous water from usual conversion processes. *Biotechnol Bioeng Symp* **8**:379–390 (1978).
8 Kobayshi H and Rittman BE, Microbial removal of hazardous organic compound. *Environ Sci Technol* **19**:A 470–481 (1982).
9 Gonzalez G, Herrera MG, Garcia MT and Pena MM, Biodegradation of phenol in a continuous process. Comparative study of stirred tank and fluidized bed bioreactors. *Bioresour. Technol* **76**:245–251 (2001).
10 Schröder MC, Müller C, Posten WD, Deckwer D and Hecht V, Inhibition kinetics of phenol degradation from unstable steady-state data. *Biotechnol Bioeng* **54**:567–576 (1997).
11 Ehrhardt HM and Rehm HJ, Phenol degradation by microorganisms adsorbed on activated carbon. *Appl Microbial Biotechnol* **21**:32–36 (1985).
12 Shetty KV, Kalifathulla I and Srinikethan G, Performance of pulsed plate bioreactor for biodegradation of phenol. *J Hazard Mater* **140**:346–352 (2007).
13 Shetty KV, Ramanjaneyulu R and Srinikethan G, Biological phenol removal using immobilized cells in a pulsed plate bioreactor: Effect of dilution rate and influent phenol concentration. *J Hazard Mater* **149**(2):452–459 (2007).
14 Schügerl K, Progress in monitoring, modeling and control of bioprocesses during the last 20 years. *J Biotechnol* **85**:149–173 (2001).
15 Rani KY and Rao VR, Control of fermenters – a review. *Bioprocess Eng* **21**:77–88 (1999).
16 Doherty SK, Control of pH in chemical processes using artificial neural networks. PhD thesis, School of Engineering, Liverpool John Moores University (1999).
17 Fagundes-Klen MR, Ferri P, Martins TD, Tavares CRG and Silva EA, Equilibrium study of the binary mixture of cadmium–zinc ions biosorption by the *Sargassum lipendula* species using adsorption isotherm models and neural network. *Biochem Eng J* **34**:136–146 (2007).
18 Baughman DR and Liu YA, *Neural Networks in Bioprocessing and Chemical Engineering*. Academic Press (1995).
19 Syu MJ and Hou CL, A neural network study on dynamic identification of a fermentation system. *Bioprocess Eng* **17**:203–213 (1996).
20 Teissier P, Perret B and Latrile E, Yeast concentration estimation and prediction with static and dynamic neural network models in batch cultures. *Bioprocess Eng* **14**:231–235 (1996).
21 Massimo DI, Willis MJ and Monteague GA, Bioprocess model building using artificial neural networks. *Bioprocess Eng* **7**:77–82 (1991).
22 Chen LZ, Nguang SK, Chen XD and Li XM, Modeling and optimization of fed-batch fermentation processes using dynamic neural networks and genetic algorithms. *Biochem Eng J* **22**:51–61 (2004).
23 Boareto AJM, De Souza MB Jr, Valero F and Valdman B, A hybrid neural model for the online monitoring of lipase production by *Candida rugosa*. *J Chem Technol Biotechnol* **82**:319–327 (2007).

24 Balan SM, Annadurai G, Sheeja RY, Srinivasamoorthy VR and Murugesan T, Modeling of phenol degradation system using artificial neural networks. *Bioprocess Eng* **21**:129–134 (1999).
25 Annadurai G and Lee JF, Application of artificial neural network model for the development of optimized complex medium for phenol degradation using *Pseudomonas pictorum* (NICM 2074). *Biodegradation* **18**:383–392 (2007).
26 Vidyavathi N, Bioremediation of industrial and domestic effluents by microorganisms PhD thesis, Department of Chemical Engineering, KREC Surathkal, Mangalore University, India (1998).
27 Simpson PK, *Artificial Neural Systems*. Pergamon Press (1990).
28 Lipmann RP, An introduction to computing with neural nets. *IEEE ASSP Mag* **4**:4–22 (1987).
29 Harvey RL, *Neural Network Principles*. Prentice-Hall (1994).
30 Turing AM, *Computing Machinery and Inteligence*. Reprinted in: *The Mind's I*. Harvester Press (1981).
31 Aleksander I, Why neural computing? A personal view. in *Neural Computing Architectures – The Design of Brain-Like Machines*, ed. by Aleksander J. North Oxford Academic Publishers, Chapter 1, pp. 1–7 (1989).
32 Martin T and Howard B, *Neural Network Design*, PWB Publishing Company (1996).
33 Rumelhart DE, Hinton GE and Wiliams RJ, Learning representations by back-propagating errors. *Nature* **323**:533–536 (1986).
34 Bello MG, Enhanced training algorithms, and integrated training/architecture selection for multiplayer perceptron networks. *IEEE Trans Neural Netw* **3**:864–875 (1992).
35 Solla SA, Levin E and Fleisher M, Accelerated learning in layered neural networks. *Complex Systems* **2**:625–639 (1988).
36 Sperduti A and Starita A, Speed up learning and network optimization with extended back-propagation. *Neural Networks* **6**:365–383 (1993).
37 Van Ooten A and Nienhuis B, Improving the convergence of the back-propagation algorithm. *Neural Netw* **5**:465–471 (1992).
38 Jacobs RA, Increased rates of convergence through learning rate adaptation. *Neural Netw* **1**:295–308 (1988).
39 Andersen TJ and Wilamowski BM, A modified regression algorithm for fast one layer neural network training, *World Congress of Neural Networks*. Washington DC, USA, **1**: 687–690 (1995).
40 Battiti R, First and second-order methods for learning: between steepest descent and Newton's method *Neural Computat* **4**:141–166 (1992).
41 Charalambous C, Conjugate gradient algorithm for efficient training of artificial neural networks. *IEE Proc* **139**:301–310 (1992).
42 Hagan MT and Menhaj MB, Training feed forward networks with the Marquardt algorithm. *IEEE Trans Neural Netw* **5**:989–993 (1994).
43 Silva WP and Cleide MDPS, LAB Fit curves fitting software (nonlinear regression and treatment of data program). V.7.2.34, 1999–(2006). www.labfit.net.