

*Full Length Research Paper*

# Estimation of biochemical variables using quantum-behaved particle swarm optimization (QPSO)-trained radius basis function neural network: A case study of fermentation process of L-glutamic acid

Zhiling Liao\* and Congli Mei

School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, China.

Accepted 29 April, 2011

Due to the difficulties in the measurement of biochemical variables in fermentation process, soft-sensing model based on radius basis function neural network had been established for estimating the variables. To generate a more efficient neural network estimator, we employed the previously proposed quantum-behaved particle swarm optimization (QPSO) algorithm for neural network training. The experiment results of L-glutamic acid fermentation process showed that our established estimator could predict variables such as the concentrations of glucose, biomass and glutamic acid with higher accuracy than the estimator trained by the most widely used orthogonal least squares (OLS). According to its global convergence, QPSO generated a group of more proper network parameters than the most popular OLS. Thus, QPSO-RBF estimator was more favorable to the control and fault diagnosis of the fermentation process, and consequently, it increased the yield of fermentation.

**Key words:** Soft-sensing model, quantum-behaved particle swarm optimization algorithm, neural network.

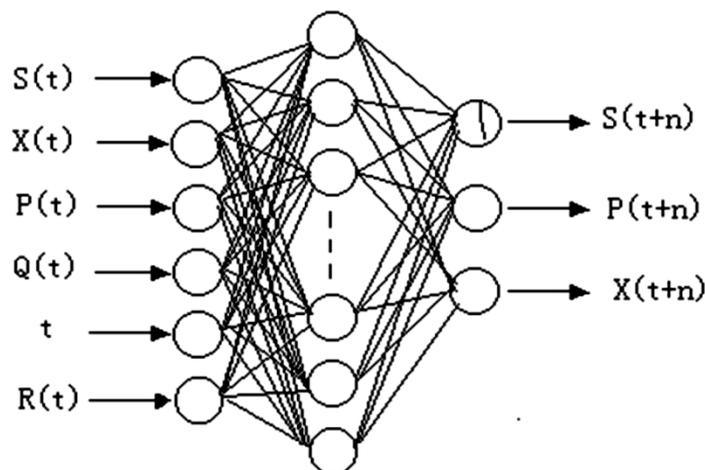
## INTRODUCTION

In industrial production through fermentation, the main effect variables include physical variables (the temperature of fermentor, pH value, mixing speed, ventilation rate, etc.) and biochemical variables (the remaining sugar concentration, the concentration of product, urea fed-batch, etc.). The physical variables could be measured online and, in turn, automatically controlled by distributed control system (DCS). The biochemical variables are closely related to metabolism and growth of microbe, and play an important role in the control or optimization fermentation. Due to the fact that there is no available commercial sensor to measure them, the only way to acquire the information about such biochemical variables is, in most practical application at present, the laboratory biochemical analyses offline (Gonzalez, 1999). However, the whole process of offline analyses generally takes a long time, so that it makes it impossible to meet the demand of

optimization control and fault diagnosis in line with the growth states of microbes, especially for real-time control.

Soft-sensing method is an efficient way by which the variables to be estimated are calculated through the available information about other variables (Tham et al., 1991). Using modeling fermentation process by soft-sensor, we can estimate the variables (the products concentration, biomass, the remaining sugar, etc.) online using the acquired information about other measurable variables (agitation rate and ventilation rate, etc.). Neural network (NN) is an important soft-sensing modeling approach for industrial fermentation process (Cinar, 2005; Desai et al., 2005; Dutta et al., 2004; Pai et al., 2008; Patnaik, 1997; Sahinkaya et al., 2007). When neural network is used for process modeling, it must be trained on training data to obtain a group of optimized network parameters. Thus, an efficient algorithm is the key to neural network design. Most existing training algorithms for neural network are based on local optimization techniques, by which the solution may be stuck into local optima or suboptima and thus, lead the network to low estimation accuracy (Marquez et al., 1992; Meireles et al., 2003).

\*Corresponding author. E-mail: [liao zhiling@ujs.edu.cn](mailto:liao zhiling@ujs.edu.cn). Tel: 86-511-88791245.



**Figure 1.** The topology structure of the established RBF NN estimator for biochemical variables in glutamic acid fermentation process.

In this study, radius basis function (RBF) NN was employed to soft-sense the biochemical parameters of glutamic acid fermentation. To enhance the estimation accuracy, we used our previously proposed quantum-behaved particle swarm optimization (QPSO) method (Sun et al., 2004, 2005) as the training algorithm for RBF-based soft-sensor (RBF NN estimator). The experiment results suggest that QPSO could generate a RBF NN estimator that can predict the biochemical variables accurately.

## MATERIALS AND METHODS

### Organism, media and culture conditions

*Corynebacterium glutamicum* UV18 was a stock culture of the Laboratory of Biochemistry, School of Biotechnology, Jiangnan University, Wuxi, Jiangsu province, China.

The seed culture medium contained (per liter): 50 g glucose, 6 g urea, 3 g  $\text{Na}_2\text{HPO}_4$  and 3 ml corn steep liquor. The initial pH was adjusted to 6.8 with NaOH after autoclaving at 120 for 15 min, whereas urea was sterilized by membrane filter (0.2  $\mu\text{m}$ ).

The production medium contained (per liter): 50 g glucose, 6 g urea, 3 g  $\text{Na}_2\text{HPO}_4$ , 2 g KCl, 1 g  $\text{MgSO}_4\cdot\text{H}_2\text{O}$ , 0.01 g  $\text{MnSO}_4$ , 100  $\mu\text{g}$  thiamine-HCl and 5  $\mu\text{g}$  biotin. Biotin, thiamine-HCl and urea were sterilized by membrane filter (0.2  $\mu\text{m}$ ), whereas glucose and minerals were sterilized separately by autoclaving at 15 psi (121°C) for 15 min. Fermentation was carried out in 30 L fermenter equipped with a dissolved oxygen (DO) analyzer and a pH controller.

### Analytical methods

The pH value and dissolved oxygen concentration were measured online by pH electrode and  $\text{O}_2$  electrode, respectively. The growth was monitored by measuring the optical density (OD) at 600 nm with a UV-Vis spectrophotometer. The biomass concentration was transformed from OD value according to the standard curve. The glucose concentration was measured by dinitrosalicylic colorimetric

method (Miller, 1959). The glutamic acid concentration was measured by the automatic amino acid analyzer (Hitachi L-8900).

The acquired data was filtered to 21 sets of representative data divided into 294 groups, 224 of which were employed as training data (training set) for RBF NN estimator training, and the other 70 were used as testing data to test the estimation accuracy of the trained estimator. Before the training and testing, the data was preprocessed by linear transformation to fall into the allowable scope of the process unit of RBF NN. The final results were anti-transformed after training and testing.

### Soft-sensing model with RBF

In this study, the glutamic acid fermentation process was modelled with RBF NN. RBF NN was structured by embedding radial basis function in a two-layer feed-forward neural network. Such a network was characterized by a set of inputs and a set of outputs. In between the inputs and outputs, there is a layer of processing units called hidden units. A distinguished feature of RBF NN was that it could be trained directly through sample data without prior knowledge about the model of the process. It got the relationship between input and output by treating the objective as a "black box". Theoretically, given sufficient input-output sample data, RBF network can approximate infinitely the actual mapping relationship between input and output sample (Schilling et al., 2001).

Figure 1 shows the topology structure of the RBF NN estimator established for estimation of the biochemical variables. The inputs of the network included the glucose concentration  $S(t)$ , biomass concentration  $X(t)$ , glutamic acid concentration  $P(t)$ , fermentation time  $t$  and DO  $Q(t)$ . The outputs of the network were  $S(t+n)$ ,  $X(t+n)$  and  $P(t+n)$  at time  $t+n$ . In order to use the RBF NN estimator in estimation, we needed to specify the hidden unit activation function, the number of processing units, a criterion for modeling a given task and in turn, a training algorithm for finding the parameters of network. In our RBF NN estimator, Gaussian basis function was adopted. Thus, if we had at hand a set of input-output pairs, we can optimize the network parameters in order to fit the network outputs to the given inputs. The fit was evaluated by means of a cost function, usually assumed to be the mean square error. The process of optimizing the network parameters was called network training.

The neuron number of the hidden layer, that is, the cluster number of training set, must be determined before the parameter selection of RBF NN estimator. In this study, we adopted an efficient method; rival penalized competitive learning (RPCL) (Xu et al., 1993), to decide the cluster number. If the neuron numbers of the hidden layer has been decided, the performance of RBF depends on the selection of the network parameters. There were three types of parameters in the RBF NN estimator with Gaussian basis functions: RBF centers (hidden layer neurons), widths of RBFs (standard deviations in the case of a Gaussian RBF) and output layer weights. Different strategies exist for training of RBF neural network models. Orthogonal least squares (OLS) using Gram-Schmidt algorithm is the most popular for RBF NN training (Chen et al., 1991; Gomm and Yu, 2000). However, together with other training algorithm, OLS cannot guarantee the search of the global optimal solution of the training problem.

### QPSO-RBF algorithm

To obtain a more efficient RBF NN estimator, we used our previously proposed QPSO as the training algorithm for the RBF NN estimator. QPSO algorithm, like particle swarm optimization (PSO), which was originally introduced by Kennedy and Eberhart in 1995 (Clerc, 1999; Kennedy et al., 1995; Shi and Eberhart, 1999), simulates the knowledge evolution of a social organism. In

QPSO, each individual was treated as an infinitesimal particle in the  $n$ -dimensional space. The position vector of the particle  $X_i(k) = [X_{i1}(k), X_{i2}(k), \dots, X_{in}(k)]$  represents a candidate solution for the optimization problem. The update equation for the position is as follows:

$$p_{ij}(k) = \varphi \cdot P_{ij}(k) + (1 - \varphi) \cdot P_{gj}(k) \quad (1a)$$

$$X_{ij}(k+1) = p_{ij}(k) \pm \alpha \cdot |C_j(k) - X_{ij}(k)| \cdot \ln(1/u) \quad (1b)$$

Where,  $\varphi$  and  $u$  are two different random numbers uniformly distributed on  $[0, 1]$  and  $k$  is the iteration number. Vector  $P_i(k) = [P_{i1}(k), P_{i2}(k), \dots, P_{in}(k)]$  is the best previous position (the position giving the best objective function value) of particle  $i$  known as the personal best position (pbest); vector  $P_g(k) = [P_{g1}(k), P_{g2}(k), \dots, P_{gn}(k)]$  is the position of the best particle among all the particles in the population and is known as the global best position (gbest).  $C_j$  is defined as the mean of the personal

$$C_j(t) = \frac{1}{M} \sum_{i=1}^M P_{ij}(t)$$

best positions among all particles, that is,

Parameter  $\alpha$  is called contraction-expansion (CE) coefficient, which could be tuned to control the convergence speed of the algorithms. For more detailed information about QPSO, one may refer to literatures such as Sun et al. (2004, 2005).

When training the RBF NN estimator by QPSO, a decision vector represents a particular group of network parameters. Therefore, a component of the position corresponds to a network parameter and a RBF NN was structured according to the particle's position vector. Training the corresponding network by inputting the training samples, we could obtain an error value as the objective function computed by the following formula:

$$E = \frac{1}{2Q} \sum_{j=1}^Q \sum_{s=0}^c [y_{s,j}(x_j) - g_{s,j}(x_j)]^2 \quad (2)$$

Where,  $y_{s,j}(x_j)$  and  $g_{s,j}(x_j)$  are the actual response (output) and network's predicted response (output) at output unit  $s$  on  $x_j$ , respectively.  $Q$  is the number of the training sample and  $c$  is the number of output units. The particle was evaluated by the obtained error value, by which it could be determined whether  $P_i$  and  $P_g$  need to be updated. The goal of training RBF NN with QPSO was to minimize the error function (2) to generate a group of optimal network parameters.

The stop criterion of the algorithm was terminated after executing a pre-specified number of iterations. The following is the procedure of training RBF NN by QPSO algorithm:

#### QPSO-RBF procedure

Step 1: Initialize the population by randomly generating the position

vector  $X_i$  of each particle and set  $P_i = X_i$ ;

Step 2: Structure a RBF neural network by treating the position vector of each particle as a group of network parameters;

Step 3: Training each RBF network on the training set;

Step 4: Evaluate the fitness value of each particle by formula (2), update the personal best position  $P_i$  and obtain the global best

position  $P_g$  across the population;

Step 5: If the stop criterion is met, go to step 7; or else go to step 6;

Step 6: Update the position vector of each particle according to Equation (1);

Step 7: Output the  $P_g$  as a group of optimized parameters.

The algorithm was achieved by using the Matlab.

## RESULTS AND DISCUSSION

The QPSO and OLS algorithms were used for RBF NN estimator training and testing. When QPSO was used, 20 particles and 50 iterations were employed for the training. The trained RBF NN estimators were called QPSO-RBF and OLS-RBF, respectively. The estimation curves and relative estimation errors for the biochemical variables are recorded in Figures 2, 3 and 4. The relative estimation errors averaged over the time horizon are listed in Table 1.

The results showed that QPSO-RBF estimator estimated the biochemical variables with higher accuracy (or lower relative error) than OLS-RBF. The advantage of QPSO over OLS algorithm is that QPSO algorithm is a global convergent search technique (Sun et al., 2005). Due to the multimodality of the RBF NN training problem, QPSO can find out the global optimal of the problem more easily than OLS algorithm. It has also been indicated that QPSO has fast convergence speed and lower computational time consumption than PSO and the genetic algorithm (GA) (Sun et al., 2005).

The results from this study are valuable in fermentation production of glutamic acid. The difficulties in online measurement of many variables, particularly biochemical variables, were unfavorable to the operations on the variables, fault diagnosis and process control. The soft-sensing model based on RBF NN is an efficient tool for this problem, because it could monitor the process efficiently and effectively, and thus make fault diagnosis and control possible. However, given the RBF NN model, to find out a group of optimal parameters for the neural network was of much significance to the estimation accuracy which the RBF NN estimator could reach. This study showed that QPSO algorithm is a promising tool for this task.

## Conclusion

We used the QPSO to design the RBF NN estimator for biochemical variables in glutamic acid fermentation process. Thanks to its global convergence, QPSO could generate a group of more proper network parameters than the most popular OLS. Thus, QPSO-RBF estimator could estimate the biochemical variables with higher accuracy than OLS-RBF estimator and thus, was more favorable to the control, fault diagnosis of the fermentation process, and consequently increased the output of

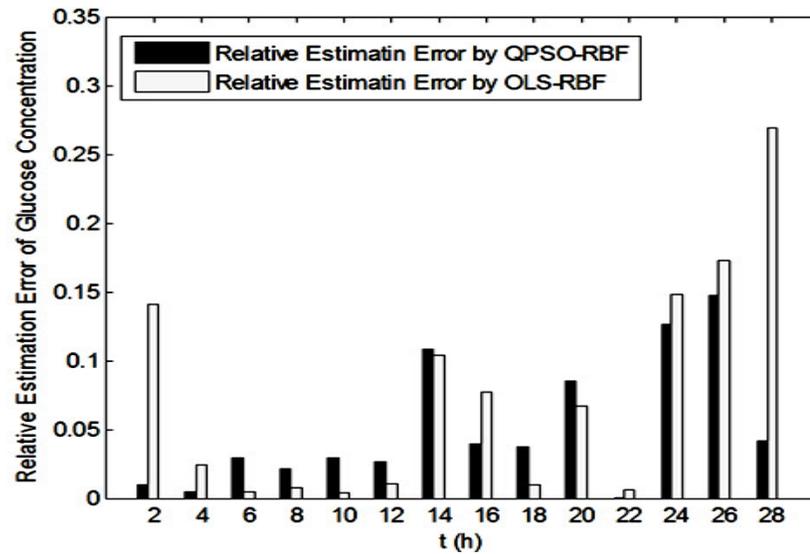
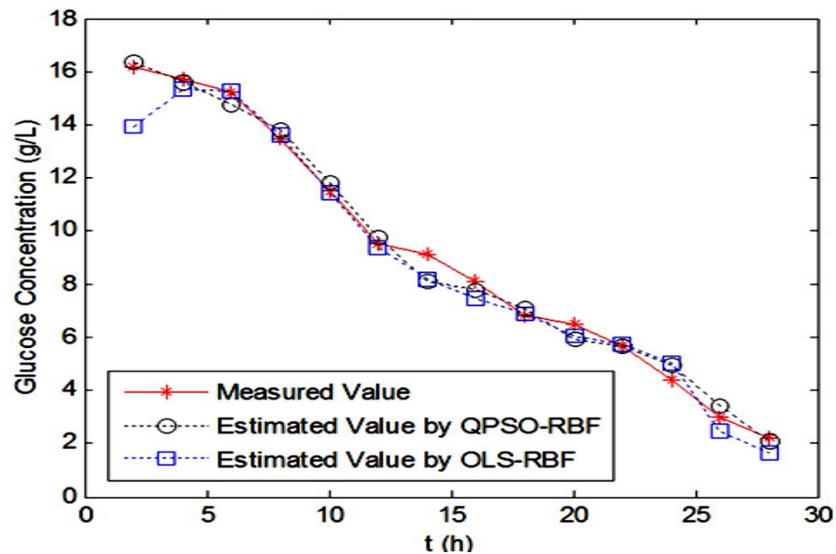


Figure 2. Estimation curves and relative estimation errors for the glucose concentration generated by QPSO-RBF and OLS-RBF.

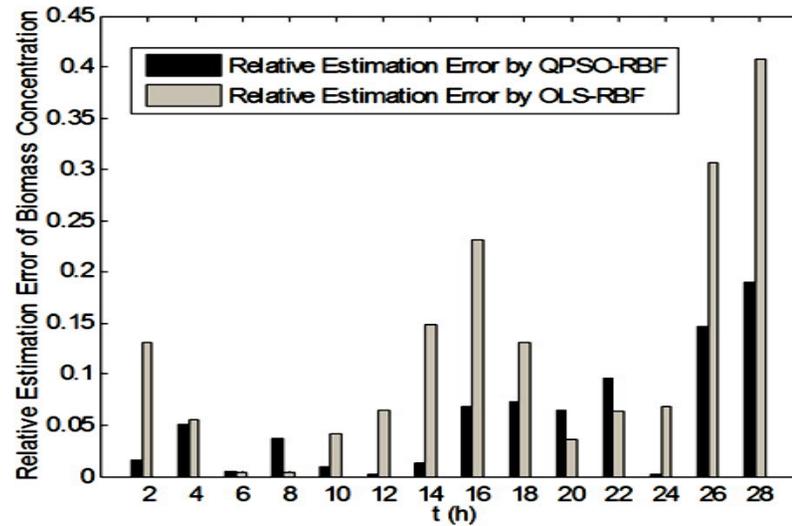
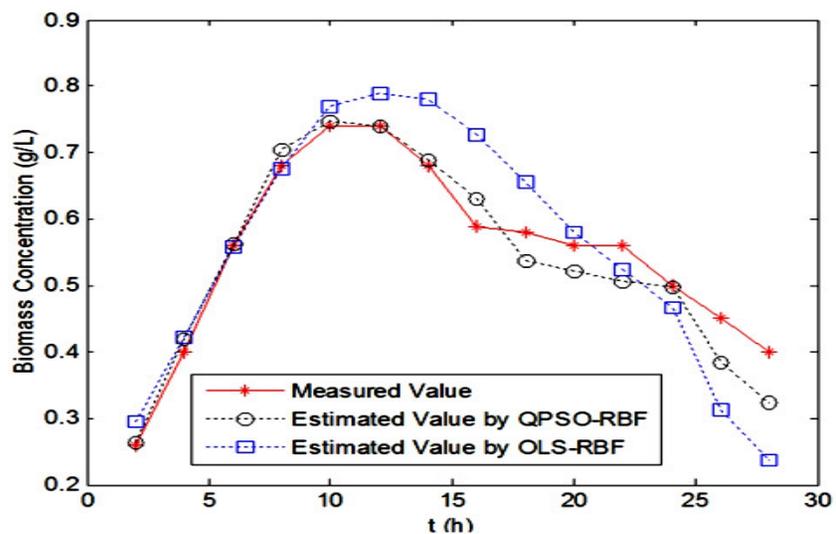


Figure 3. Estimation curves and relative estimation errors for the biomass concentration generated by QPSO-RBF and OLS-RBF.

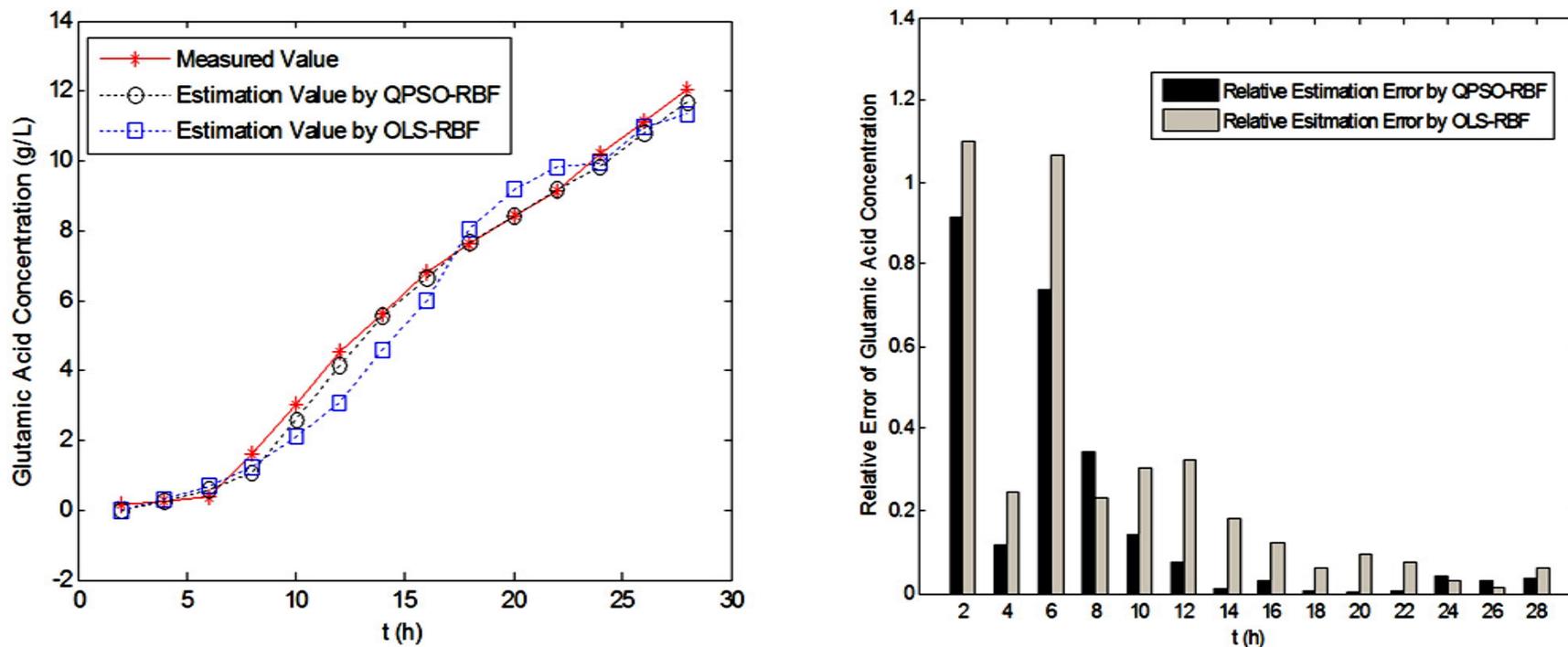


Figure 4. Estimation curves and relative estimation errors for the glutamic acid concentration generated by QPSO-RBF and OLS-RBF.

Table 1. Relative estimation error average over time.

Item	QPSO-RBF (%)	OLS-RBF (%)
Glucose concentration	5.05	7.48
Biomass	5.56	12.09
Glutamic acid concentration	17.75	27.85

glutamic acid.

**ACKNOWLEDGEMENT**

This is a project funded by the Priority Academic

Program Development of Jiangsu Higher Education Institutions.

**REFERENCES**

Chen S, Cowan CFN, Grant PM (1991). Orthogonal least

squares learning algorithm for radial basis function networks. *Neural Networks, IEEE Transactions on*, 2: 302-309.

Cinar O (2005). New tool for evaluation of performance of wastewater treatment plant: Artificial neural network. *Process Biochem.* 40: 2980-2984.

Clerc M (1999). The swarm and the queen: towards a deter-

- ministic and adaptive particle swarm optimization. *Evolutionary Computation*, 1999. CEC 99. Proceedings of the 1999 Congress on. 1953: 1957.
- Desai KM, Vaidya BK, Singhal RS, Bhagwat SS (2005). Use of an artificial neural network in modeling yeast biomass and yield of beta-glucan. *Process Biochem.* 40: 1617-1626.
- Dutta JR, Dutta PK, Banerjee R (2004). Optimization of culture parameters for extracellular protease production from a newly isolated *Pseudomonas* sp using response surface and artificial neural network models. *Process Biochem.* 39: 2193-2198.
- Gomm JB, Yu DL (2000). Selecting radial basis function network centers with recursive orthogonal least squares training. *Neural Networks, IEEE Transactions on*, 11: 306-314.
- Gonzalez GD (1999). Soft sensors for processing plants. *Intelligent Processing and Manufacturing of Materials*, 1999. IPMM '99. Proceedings of the Second International Conference on. 51. 59-69
- Kennedy J, Eberhart R, leee (1995). Particle swarm optimization. 1995 IEEE International Conference on Neural Networks (ICNN 95). Perth, Australia, pp. 1942-1948.
- Marquez L, Hill T, O'Connor M, Remus W (1992). Neural network models for forecast: a review. *System Sciences*, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on. 494: 494-498
- Meireles MRG, Almeida PEM, Simoes MG (2003). A comprehensive review for industrial applicability of artificial neural networks. *Industrial Electronics, IEEE Transactions on*, 50: 585-601.
- Miller GL (1959). Use of dinitrosalicylic acid reagent for determination of reducing sugar. *Anal. Chem.* 31: 426-428.
- Pai TY, Chuang SH, Ho HH, Yu LF, Su HC, Hu HC (2008). Predicting performance of grey and neural network in industrial effluent using online monitoring parameters. *Process Biochem.* 43: 199-205.
- Patnaik PR (1997). A recurrent neural network for fed-batch fermentation with recombinant *Escherichia coli* subject to inflow disturbances. *Process Biochem.* 32: 391-400.
- Sahinkaya E, Ozkaya B, Kaksonen AH, Puhakka JA (2007). Neural network prediction of thermophilic (65 degrees C) sulfidogenic fluidized-bed reactor performance for the treatment of metal-containing wastewater. *Biotechnol. Bioeng.* 97: 780-787.
- Schilling RJ, Carroll JJ Jr, Al-Ajlouni AF (2001). Approximation of nonlinear systems with radial basis function neural networks. *Neural Networks, IEEE Transactions on*, 12: 1-15.
- Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. *Evolutionary Computation. CEC 99. Proceedings of the 1999 Congress on. 1953: 1950*
- Sun J, Xu WB, Feng B (2005) Adaptive parameter control for quantum-behaved particle swarm optimization on individual level. *Systems, Man and Cybernetics, 2005 IEEE International Conference on.* pp. 3049-3054.
- Sun J, Xu WB, Feng B, leee (2004) A global search strategy of quantum-behaved particle swarm optimization. *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, pp. 111-116.
- Tham MT, Montague GA, Julian Morris A, Lant PA (1991). Soft-sensors for process estimation and inferential control. *J. Proc. Cont.* 1: 3-14.
- Xu L, Krzyzak A, Oja E (1993). Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. *Neural Networks, IEEE Transactions on*, 4: 636-649.