

# Soft-sensor for Mother Liquor Purity in The Industrial Cane Sugar Crystallization Based on Extreme Learning Machine and Particle Swarm Optimization

Yanmei Meng, Haifeng Pang, Jian Chen, Zhihong Tang and Xiaochun Wang

<sup>1</sup>*Department of Mechanical Engineering, Guangxi University, China*  
*gxu\_mengyun@163.com*

## Abstract

*One of the key procedures in sugar production is the complex sugar crystallization process. Monitoring and controlling this process would pose great impact on the implantation of automatic sugar production. It has been well known that the mother liquor purity in the cane sugar crystallization determines the forming speed of sugar crystals and the mother liquor supersaturation. The lower mother liquor purity can enhance the formation of crystals. Generally, it needs to go through chemical laboratory analysis of mother liquor sample to obtain its purity because there is no instrument that can directly measure the purity. Aimed to tackle this issue, in this paper, a novel soft sensor approach based on extreme learning machine (ELM) is proposed to estimate the mother liquor purity parameter in sugar crystallization process. Also, the selection of the cost parameter and kernel parameter of ELM with Gaussian kernel is optimized by particle swarm optimization (PSO) method. Experimental results demonstrate that the proposed approach is feasible to predict the sugar mother liquor purity accurately in sugar crystallization process. Compared with other model building method like BP, RBF, SVM, LS-SVM and Primal ELM, the proposed model can perform higher prediction accuracy and faster building capacity. Implementation of soft-sensor model for cane sugar mother liquor purity makes it easier to adjust the sugar crystallization process in real conditions for higher productivity.*

**Keywords:** *Cane sugar crystallization process, Mother liquor purity, Soft-sensor, Extreme learning machine, Particle swarm optimization*

## 1. Introduction

From a sugar production field's view, the sugar productivity greatly depends on the control techniques for cane sugar crystallization process [1-2]. However, it's nearly impossible to build a precise model for sugar crystallization process due to its complex physical mechanism, nonlinearity, great inertia and strong coupling links, where lots of uncertain factors are involved [3-4]. In the sugar crystallization process monitoring field, the vast majority of researchers focus on the advanced control techniques [3-16] like nonlinear control, fuzzy predictive control, neural network with PID and robust control. However, the detection sensor devices for process parameters like supersaturation, concentration, purity, crystal content and crystal size uniformity *etc.*, are hard to be measured in the real world, which makes the application of all those advanced control techniques stay at theoretical stage but not an actual one [17].

Crystallization process consists of A crystallization, B crystallization and C crystallization these three phases in the cane sugar production process [16-17]. In order to

extract the most sugar from the syrup, it needs to obtain the largest gap of molasses purity between the A and C crystallization phases. In other words, the lower the molasses purity in C crystallization becomes, the greater the absorption of sugar in the crystallization process is. In addition, mother liquor purity impacts the crystallization rate and supersaturation at a certain scale.

As analyzed above, building the precise model for mother liquor purity is beyond consideration. Besides, the purity measurement in the traditional way is confined at the labs, which is not feasible to be applied in the industrial condition for its time-delaying and noise [1]. Aimed to solve these issues, this paper proposes a soft-sensor model for mother liquor purity to make the purity measurement online possible.

In the past decades, soft-sensor has been the research hot spot in the academic and industrial fields for its capacity to utilize the historical data to predict the data trend [18-21]. The basic concept of soft-sensor is to establish a mathematical relation between the auxiliary variables and dominant variables. The dominant variables denote the difficult-to-measure variables while the auxiliary variables denote the easy-to-measure ones which are closely related to the dominant variables [22]. Soft-sensor implements the computer technologies into the industrial field, which has avoided the necessity of building the hardware system to realize the industrial measurement [23-24].

The soft-sensor modeling methods in general are classified into the following four categories. (1) Mechanism model. (2) State estimation model. (3) Statistical regression model. (4) Artificial intelligence model. Among them, mechanism model is only feasible in the simple industrial process. However, all the industrial processes that need to be analyzed are complex ones. Kalman filter or extended Kalman filter model process often utilized in state estimation model is highly depended on the precise process knowledge and prior estimation of measurement noise [26-28]. As to statistical regression model, its regular model pattern is to analyze the data by PCA or PLS method while the data must be linearly correlated [29-31]. Obviously, this method is not able to model the nonlinearly correlated data obtained from the industrial process. Recently, soft-sensor based on artificial intelligence has received much attention, along with the development of ANN and SVM [32-39].

As a classical regression and classification algorithm, Single-hidden Layer Feedforward Neural Network (SFFN) is widely used in many fields [40-41]. The traditional learning algorithm of SFFN, like BP, usually updates the weights and bias of the new network by gradient descent approach [42-44]. However, the development of SFFN is compromised by its own defects, which is shown as follows [45-46]. (1) The gradient descent approach would slower the training process by multiple iterations. (2) The performance of network is greatly affected by the variation of learning speed. The lower learning speed would result in the lower convergence speed of network, while the higher would lead to an unstable training process. (3) The global minimum point is hardly obtained because SFFN is easily stuck at the local minimum point.

In 2004, Huang and other researchers proposed a different SFFN called ELM to overcome the issues discussed above [47-51]. ELM initializes the weights of input layer and hidden layer randomly, also the bias of hidden layer, which propels the development of Feedforward Neural Network (FNN). Compared with traditional SFFN, ELM is of better performance, like faster learning speed, better generalization and easy variable adjustment [52-53]. If you want the only optimal solution, all you need to do is set the number of hidden layers. With its excellent features, ELM draws huge attention among the academic field. In the past decade, the application of SVM has been rigorously exploited in the machine learning field. It is especially suitable for the regression and classification issues. In literature [54-55], Huang added kernel function to the ELM to improve its generalization and training speed.

This paper analyzes the easy-to-measure process variables that are closely related to the mother liquor purity. The next step is to build a soft-sensor model for the mother liquor purity. To do this, it needs to establish the nonlinear relationship between mother liquor purity and process variables first. Then it can propose the specific model based on ELM. In details, normalize the historical data sample to obtain the training set and testing set in the first place. The next step is to build the soft-sensor model through the newly training set. Meanwhile, optimize the cost parameter  $C$  and kernel parameter  $\gamma$  of the model iteratively by PSO. Once we have the optimal group  $(C, \gamma)$ , the optimal soft-sensor model would come along. At last, verify the model outputs based on the testing set and compare them with other models outputs, aiming to prove that the performance of specific model discussed here is better.

This paper is organized as follows. Section 2 introduces the methods applied in the soft-sensor model for mother liquor purity in sugar crystallization process. Section 3 details the steps of building the soft-sensor model based on ELM. Section 4 verifies the performance of the model by test and comparison with other models. Then the conclusion part is in Section 5.

## 2. Soft-sensor Methods

### 2.1. Extreme Learning Machine

The network structure of traditional SFFN consists of input layer, hidden layer and output layer. Assume  $n$  denote the number of input layer neurons, which indicates that we have  $n$ -th input variables. Similarly, we set the number of hidden layer neurons as  $l$ , while the number of output layer neurons is  $m$ . The sample set  $S = \{(x_k, y_k)\}_{k=1}^N$  is composed by  $N$ -th samples, where  $\mathbf{x}_i = [x_{k1}, x_{k2}, \dots, x_{kn}]^T \in \mathbb{R}^n$ ,  $\mathbf{y}_k = [y_{k1}, y_{k2}, \dots, y_{km}]^T \in \mathbb{R}^m$ . With respect to the principles of Feedforward Neural Network, the function relationship from input to output is described as:

$$f(\mathbf{x}_k) = \sum_{i=1}^l \beta_i G(\mathbf{w}_i \cdot \mathbf{x}_k + b_i) = \mathbf{y}_k, \quad k = 1, \dots, N \quad (1)$$

where  $\mathbf{b}$  is the connecting weight from  $i$ -th hidden layer node to the output layer node;  $G$  denotes the output matrix of hidden layer related to the  $i$ -th input variable;  $\mathbf{w}_i$  is the weight vector and  $b_i$  is the bias.

Eq. (1) can be further written as:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{y} \quad (2)$$

where  $\mathbf{H}$  is the output matrix of hidden layer.

The detailed forms of  $\mathbf{H}$ ,  $\boldsymbol{\beta}$ ,  $\mathbf{y}$  is presented in Eq. (2).

$$\mathbf{H} = \mathbf{H}(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l, b_1, b_2, \dots, b_l, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$$

$$= \begin{bmatrix} G(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & G(\mathbf{w}_2 \cdot \mathbf{x}_1 + b_2) & \dots & G(\mathbf{w}_l \cdot \mathbf{x}_1 + b_l) \\ G(\mathbf{w}_1 \cdot \mathbf{x}_2 + b_1) & G(\mathbf{w}_2 \cdot \mathbf{x}_2 + b_2) & \dots & G(\mathbf{w}_l \cdot \mathbf{x}_2 + b_l) \\ \vdots & \vdots & \dots & \vdots \\ G(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & G(\mathbf{w}_2 \cdot \mathbf{x}_N + b_2) & \dots & G(\mathbf{w}_l \cdot \mathbf{x}_N + b_l) \end{bmatrix}_{N \times l} \quad (3)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_l^T \end{bmatrix}_{l \times m} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix}_{N \times m} \quad (4)$$

Huang *et al.*, [47-53] proved that the network variables didn't need to be adjusted in the whole.  $w_i$  and  $b_i$  can randomly be assigned, at the same time the training process can remain constant. In other words, the connecting weight  $\boldsymbol{\beta}$  can be obtained by Eq. (5).

$$\text{Minimize : } \|\mathbf{H}\boldsymbol{\beta} - \mathbf{y}\|^2 \quad \text{and} \quad \|\boldsymbol{\beta}\| \quad (5)$$

where the solution of it is:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^+ \mathbf{y} \quad (6)$$

where  $\mathbf{H}^+$  is the Moore-Penrose generalized inverse matrix of hidden output matrix  $\mathbf{H}$ . There are plenty of methods available to calculate the Moore-Penrose generalized inverse matrix, like orthogonal projection method, orthogonalization method, iterative method and singular value decompositions (SVD). Here in ELM, we take orthogonal projection method to obtain the Moore-Penrose generalized inverse matrix. If  $\mathbf{H}^T \mathbf{H}$  becomes a nonsingular matrix, then  $\mathbf{H}^+$  can be calculated by  $\mathbf{H}^+ = \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1}$ . In a similar way, once  $\mathbf{x}$  is a nonsingular matrix, we have  $\mathbf{H}^+ = \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1}$ .

Detailed steps of primal ELM approach are shown as follows:

Step1: Designate the specific neurons number and the activation function of hidden layer. Randomly assign the connecting weight  $w$  between input layer and hidden layer, also the bias  $b_i$  of neurons of hidden layer.

Step2: Calculate the input matrix  $\mathbf{H}$  of hidden layer.

Step3: Then we can have  $\boldsymbol{\beta}$  by  $\hat{\boldsymbol{\beta}} = \mathbf{H}^+ \mathbf{y}$ .

## 2.2. ELM with Kernels

Similar to the principles of SVM, Huang *et al.* [54-55] add kernel function into the ELM approach to further optimize the primal ELM and improve its generalization and training speed.

Learned from Eq. (1), the output of ELM can be described as:

$$f(\mathbf{x}) = \sum_{i=1}^l \beta_i G(\mathbf{w}_i, b_i, \mathbf{x}) = \boldsymbol{\beta} \cdot \mathbf{h}(\mathbf{x}) \quad (7)$$

where  $\mathbf{h}(\mathbf{x})$  is regarded as a nonlinear mapping function,  $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \dots, h_l(\mathbf{x})] = [G(\mathbf{w}_1, b_1, \mathbf{x}), \dots, G(\mathbf{w}_l, b_l, \mathbf{x})]$ , it maps the  $n$ -th dimensional input space into the  $l$ -th high-dimensional feature space nonlinearly, which converts the nonlinear issue to linear ones.

The regression and classification of ELM can be converted to the optimization issues with constraints of equation, which is shown as follows.

$$\text{Minimize : } F(\boldsymbol{\beta}, e) = \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \frac{1}{2} C \sum_{i=1}^N e_i^2 \quad (8)$$

Subject to :  $y_i = \mathbf{h}(\mathbf{x}) \cdot \boldsymbol{\beta} + e_i \quad i = 1, \dots, N$

where  $e$  is the empirical error variance;  $C$  is the error penalty factor, which indicates the degree of emphasis of empirical error. According to the KKT conditions, construct the Lagrange function by Eq. (8), which is shown as follows.

$$L(\boldsymbol{\beta}, e, \alpha) = F(\boldsymbol{\beta}, e) - \sum_{i=1}^N \alpha_i \{ \mathbf{h}(\mathbf{x}) \cdot \boldsymbol{\beta} + e_i - y_i \} \\ = \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \frac{1}{2} C \sum_{i=1}^N e_i^2 - \sum_{i=1}^N \alpha_i \{ \mathbf{h}(\mathbf{x}) \cdot \boldsymbol{\beta} + e_i - y_i \} \quad (9)$$

Where  $\alpha_i$  denotes the Lagrange Multiplier. Strive for the partial derivative of function  $L(\boldsymbol{\beta}, e, \alpha)$  with respect to  $\boldsymbol{\beta}, e, \alpha$  respectively, then the necessary condition of the minimum  $L(\boldsymbol{\beta}, e, \alpha)$  is obtained as:

$$\begin{cases} \frac{\partial L}{\partial \boldsymbol{\beta}} = 0 \rightarrow \boldsymbol{\beta} = \sum_{i=1}^N \alpha_i \mathbf{h}(\mathbf{x}_i)^T = \mathbf{H}^T \boldsymbol{\alpha} & (a) \\ \frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = C e_i, & (b) \quad , i = 1, \dots, N \\ \frac{\partial L}{\partial \alpha_i} = 0 \rightarrow \mathbf{h}(\mathbf{x}_i) \boldsymbol{\beta} + e_i - y_i = 0, & (c) \end{cases} \quad (10)$$

where  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_N]^T$ .

In Eq. (10), substitute formula (a) and (b) into (c) to obtain the following condition:

$$\boldsymbol{\alpha} \left( \frac{1}{C} + \mathbf{H} \mathbf{H}^T \right) = \mathbf{y} \quad (11)$$

By using formula (a) and Eq. (11),  $\boldsymbol{\beta}$  is expressed as:

$$\boldsymbol{\beta} = \mathbf{H}^T \left( \frac{1}{C} + \mathbf{H} \mathbf{H}^T \right)^{-1} \mathbf{y} \quad (12)$$

According to Eq. (7) and Eq. (11), the output of ELM is described as

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \mathbf{H}^T \left( \frac{1}{C} + \mathbf{H} \mathbf{H}^T \right)^{-1} \mathbf{y} \quad (13)$$

Known from Mercer, the inner-product of vectors in high-dimensional space can be replaced by kernel function satisfied the Mercer conditions. In other words, all we need to do is calculate the kernel value of input variable in the original space, without concerning the specific form of  $\mathbf{h}(\mathbf{x}_i)$ .

Let's assume the kernel matrix of ELM is  $\mathbf{K}_{ELM} = \mathbf{H} \mathbf{H}^T$ , then we have  $K(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i) h(\mathbf{x}_j)$ . The output of ELM in Eq. (13) is expressed as

$$f(\mathbf{x}) = \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \dots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^T \left( \frac{1}{C} + K(\mathbf{x}_i, \mathbf{x}_j) \right)^{-1} \mathbf{y} \quad (14)$$

### 2.3. Particle Swarm Optimization (PSO)

PSO is originally attributed to Eberhart and Kennedy [56-57] and was first intended for simulating social behavior, as a stylized representation of the movement of organisms in a bird flock or fish school [58]. At present, PSO is widely applied in lots of optimization fields, like multi-objective optimization, pattern identification, neural network training and function optimization.

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulas. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, that a satisfactory solution will eventually be discovered [58-59].

Let a swarm consisted of  $N$  particle,  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \dots, \mathbf{x}_N)$ , in a  $D$ -th dimensional searching space; the velocity of  $i$ -th particle is  $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ , its position is  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ . In each swarm iteration optimization process, every particle would update its position by the measurement of two extreme values. The first extreme value is the optimal solution with regard to the  $i$ -th particle present position, which is called the individual extreme value  $\mathbf{P}_i = (P_{i1}, P_{i2}, \dots, P_{iD})$ . The second extreme value is the optimal solution for the whole swarm, called global extreme value. The movements of the particles are guided both by their  $\mathbf{P}_i$  and  $\mathbf{P}_g$ . Their velocities and positions are constantly updated by the following equations.

$$v_{id}^{k+1} = w v_{id}^k + c_1 r_1 (p_{id}^k - x_{id}^k) + c_2 r_2 (p_{gd}^k - x_{id}^k) \quad (15)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (16)$$

where  $1 \leq i \leq N$ ,  $1 \leq d \leq D$ ;  $k$  is the number of iteration;  $w$  is the inertia weighted factor (non-negative) which has great impact on the optimization process;  $c_1$  and  $c_2$  denote the acceleration coefficient, whose value is among (0-2);  $r_1$  and  $r_2$  are two independent random numbers which range from 0 to 1.

The bigger  $w$  is conducive to the global search while the smaller would be in favor of local search.  $w$  demonstrates how proximate the present velocity is to the previous one. For the sake of balancing the global and local searching capacity, the inertia weight is selected by the following equation.

$$w(k) = w_{start} - (w_{start} - w_{end}) \cdot k / k_{max} \quad (17)$$

Where  $w_{start}$  is the initial weight;  $w_{end}$  is the inertia weight with respect to the end of iteration;  $k$  denotes the iteration times at present while  $k_{max}$  is the max iteration time. In general, we usually set the inertia weight as  $w_{start}=0.9$ ,  $w_{end}=0.4$  because it is when the algorithm is on its best performance condition. Proceeded with the iteration, the  $w$  will be decreased from 0.9 to 0.4 linearly, which strengthens the global searching capacity in the

initial stage and ensures a more precise local searching capacity in the late of iteration.

Algorithm steps of PSO are shown as follows:

**Step 1:** Set the searching range for solutions space, the population size of swarm and the maximum times of iteration.

**Step 2:** Initialize the position and velocity of swarm.

**Step 3:** Calculate the fitness of every particle,  $fitness(x)$ .

**Step 4:** Update the individual and group to their optimal states using the following equations.

$$P_{id}^{k+1} = \begin{cases} x_{id}^{k+1} & fitness(x_{id}^{k+1}) < fitness(p_{id}^k) \\ p_{id}^k & fitness(x_{id}^{k+1}) \geq fitness(p_{id}^k) \end{cases} \quad (18)$$

$$P_g = \min \{ p_1^{k+1}, p_2^{k+1}, \dots, p_{ND}^{k+1} \} \quad (19)$$

**Step 5:** Update the particle position and velocity with Eq. (15) and Eq. (16).

**Step 6:** If the iteration's requirement is meted, then the output  $P_g$  and its relevant fitness  $fitness(x)$  are determined. Otherwise jump to **Step 2**.

### 3. Modeling of Soft-sensor Using ELM and PSO

#### 3.1. Auxiliary Variable Selection

It's fair to say that the auxiliary variable selection can be classified into the four following categories, selection based on variable selection principle, selection based on the industrial process, selection based on the technological process and selection based on expertise's experience. In order to enhance the capacity of soft-sensor for better refecction of the dynamic process and more flexible data processing, it needs to determine the amount of auxiliary variables in accordance with degree of freedom of system and features of the specific process.

In fact, the sucrose process is too complicate to be modeled from the perspective of process mechanism, where a variety of modals of qualitative and quantitative and mutual interaction of them are involved. However, there exists a certain link between the mother liquor purity and the state of sucrose process, which sheds a light for us to dig into the purity. As we know, the process variables, vacuum degree, massecuite temperature, vapor pressure and temperature, vacuum level, material inflow rate and massecuite brix etc., are easy-to-measure and closely related to the mother liquor purity. Therefore, we define those process variables as the inputs of soft-sensor model, the purity as the output, which is detailed in Table 1.

**Table 1. The Inputs Variables of Soft-sensor Model**

No.	Description	Unit
X1	Vacuum degree in crystallization process	-MPa
X2	Massecuite Temperature in crystallization process	°C
X3	Vapor pressure needed for heating the crystallization process	MPa
X4	Vapor temperature needed for heating the	°C

	crystallization	
X5	Vacuum level in crystallization process	cm
X6	Inflow rate of material in crystallization process	m <sup>3</sup> /h
X7	Masseccuite brix in crystallization process	Brix

### 3.2. Modeling

Here is the implementation of this soft-sensor model based on ELM for mother liquor purity. Gaussian kernel is selected as the kernel function of this model. Similar to SVM and LSSVM, the performance of ELM is sensitive to the cost parameter  $C$  and kernel parameter  $\gamma$ , as shown in Figure 1. In order to achieve good generalization performance, the cost parameter  $C$  and kernel parameter  $\gamma$  of ELM need to be chosen appropriately. Since parameters ( $C$ ,  $\gamma$ ) determined manually are not getting us the optimal model, it is necessary to optimize the key model parameters by algorithms. PSO may be an excellent choice to do this task. The essence of PSO is to go through user-defined times of iteration in a certain space for the optimal group of parameters. It proceeds in a continuous range of data.

Steps of implanting the soft-sensor model are described as follow:

**Block1:** To obtain the sample data set from sucrose crystallization process, including auxiliary data and dominant data.

**Block2:** Randomly arrange the sample set and divide them into training sample set and testing sample set.

**Block3:** Normalize the training sample set and testing sample set.

**Block4:** Initialize the size of swarm, acceleration coefficient and, particles initial position, initial velocity, maximum times of iteration and inertia weight factor, which is called the initialization of PSO. Kernel function parameter and regularization parameter are indexes for the two dimensional space of particles.

**Block5:** Set the searching range for parameters and select the MSE function of soft-sensor model as fitness function.

**Block6:** According to the output of fitness function, set the optimal position in the light of comparison between individual optimal position and swarm optimal position.

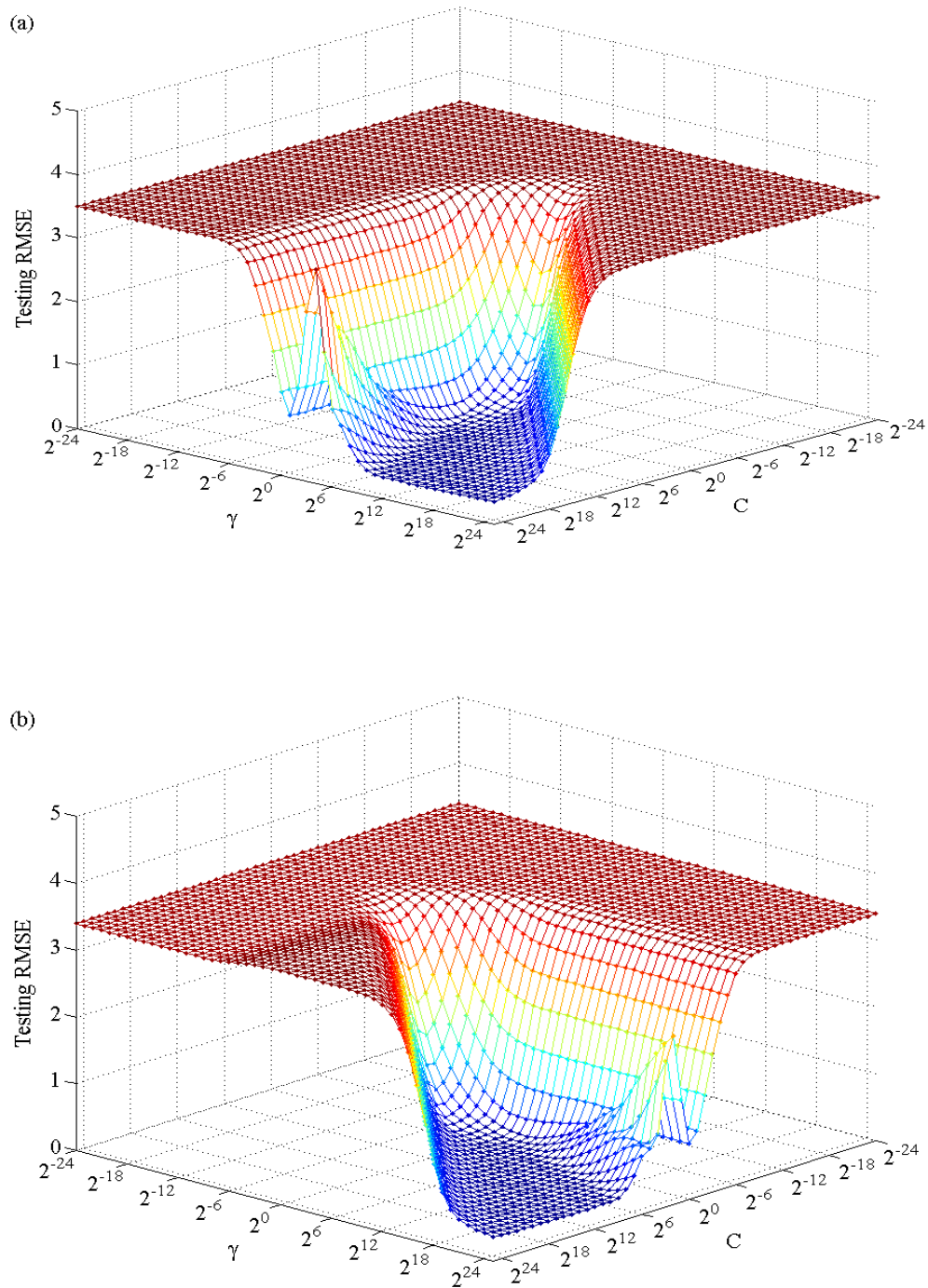
**Block7:** Update velocity and position for particles, the size of swarm and also the individual and swarm optimal position.

**Block8:** If the maximum times of iteration are counted, output the optimal parameters and penalty factors of kernel function. In the other hand, if iteration is not over, jump to Block5 for a new searching till the maximum iteration times are met.

**Block9:** Once we have the optimal parameters and penalty factors, rebuild the soft-sensor model based on the training set.

**Block10:** Test the newly obtained model in Block9 with testing sample set. Analyze the results and make an assessment on the performance of this model.





**Figure 1. The Effect of Tunable Parameters ( $C$ ,  $\gamma$ ) for Different Soft-sensors on the Generalization Performance. (a) LS-SVM, (b) ELM with Gaussian Kernel**

### 3.3. Performance Criteria

Generally, the following approaches are exploited to distinguish if a model is excellent, like RMSE, MRE and MAE. Here let's see the formula definitions of those approaches in Table 2. In the table,  $y_i$  denotes the actual value of  $i$ -th sample;  $\hat{y}_i$  denotes the predicted value of  $i$ -th sample;  $\bar{y}$  represents the sample mean and  $n$  is the number of samples.

Another performance index for soft-sensor is the decision coefficient  $R^2$ , whose formula is shown at Table 2.  $R^2$  is defined as the square of two correlation coefficients. One is the coefficient of predicted output and the other is the coefficient of actual output of soft-sensor model. Decision coefficient  $R^2$  determines the feasibility of the model. In other words, a bigger  $R^2$  demonstrates that the model is capable to identify a more complex multivariable samples, which can get us a model of better regression performance and higher feasibility.

**Table 2. Basic Definitions of Performance Criteria of Different Soft-sensor Models**

Performance criteria	Formula definition
RMSE	$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}$
MRE	$MRE = \frac{\sum_{i=1}^n \left  \frac{y_i - \hat{y}_i}{y_i} \right }{n} \times 100\%$
MAE	$MAE = \frac{\sum_{i=1}^n  y_i - \hat{y}_i }{n}$
$R^2$	$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$

## 4. Experiment Results and Discussion

### 4.1. Dataset Acquisition

As shown in Figure 2, it was a novel comprehensive experimental platform for cane sugar crystallization process, which covered all the data needed in this paper. This platform consisted of a small vacuum pan, peripheral components and a set of SCADA system. The main function of SCADA system was to acquire and control the data came from all monitoring equipment of this experimental platform. To a further extent, we could save all experimental data on the computer via SCADA system, for a more sophisticated research on intelligent control in the future. The acquisition of soft-sensor auxiliary variables was

implanted by SCADA, meanwhile the dominant variable was acquired a specific laboratory instrument.



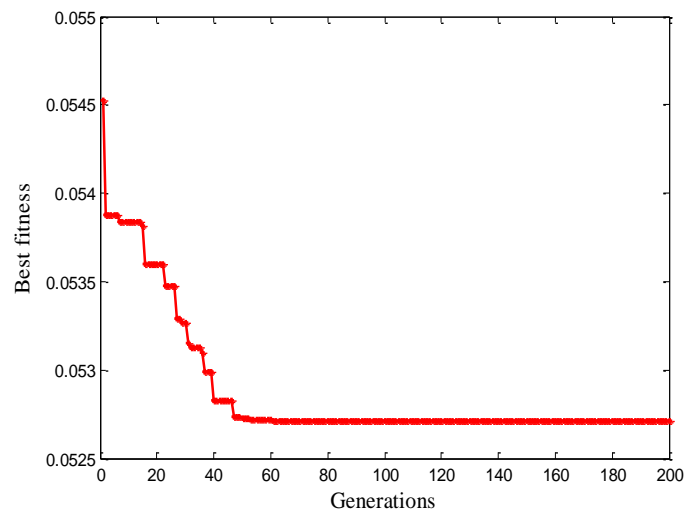
**Figure 2. The Novel Comprehensive Experimental Platform for Cane Sugar Crystallization Process**

#### 4.2. Model Validation

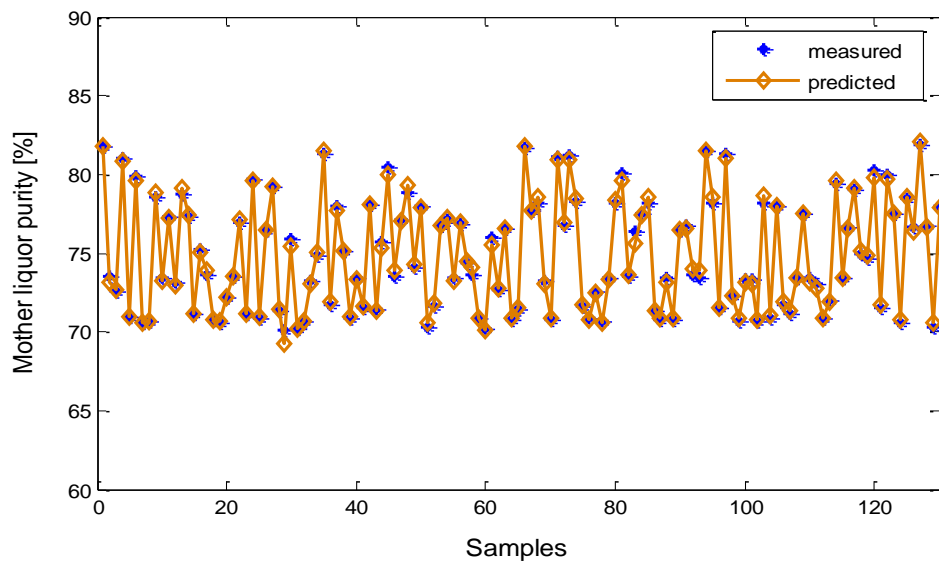
The sample set of 210 samples was randomly arranged to training set and testing set, to ensure that the model had better generalization for new data. At this point, the training set of 130 was exploited to build the soft-sensor model and to select the model parameters while the testing set was used to evaluate the prediction performance of accuracy of this model.

As discussed above, the key parameters ( $C$ ,  $\gamma$ ) of ELM model were optimized by PSO approach. The iteration time was set as 200, the optimization range of kernel function parameters and penalty factor was within the range of  $[2^{-24}, 2^{25}]$ , MSE was chose to be the fitness function. After 200 times of iteration, the fitness curve in the optimization process was shown in Figure 3, from which we knew that the optimization process was fast enough to find all the optimal parameters, to a point that the soft-sensor model conducted was of the best performance.

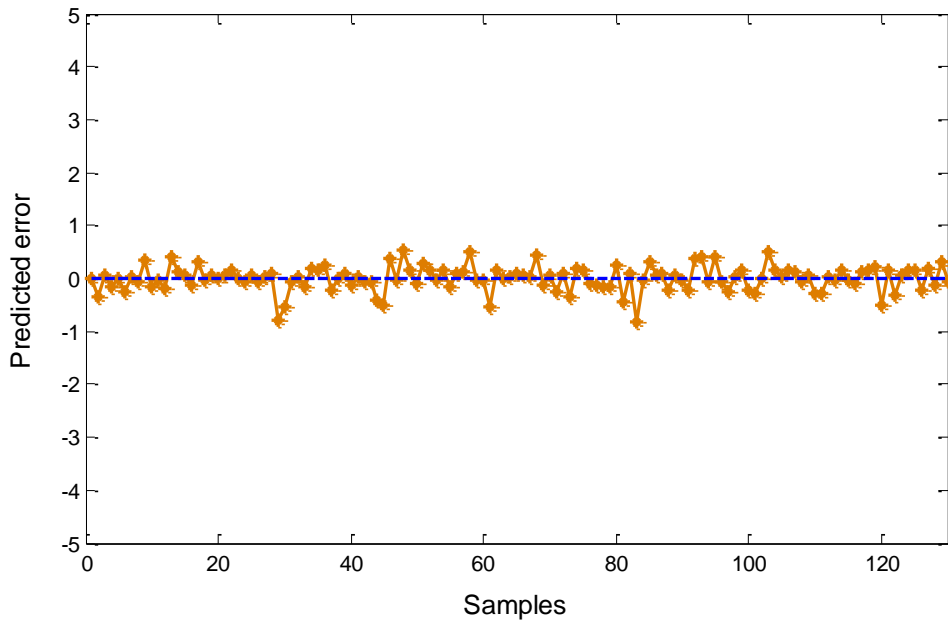
Soft-sensor model optimized by PSO was trained using training set, and a final model was obtained. The prediction outputs of training data were shown in Figure 4 and Figure 5 showed the error between the predicted outputs and actual samples based on training data. The next step was to test the final model with testing set, whose prediction outputs were shown in Figure 6. Figure 7 showed the error between the predicted outputs and actual samples based on testing set. The testing results gave us a solid conclusion that soft-sensor model based on ELM and PSO was of excellent generalization performance. Also, the fluctuation of predicted outputs with respect to actual samples was rather small, which proved that the prediction accuracy of the model was very high. In addition, it was feasible to apply this model online since the total amount of time was less than 20s.



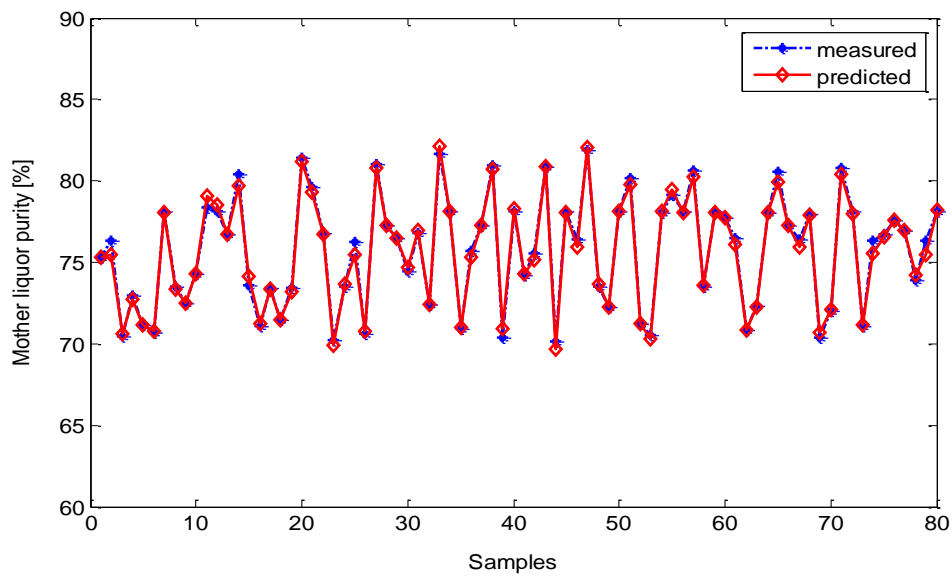
**Figure 3 The Optimal Fitness Curve for the Soft-sensor Model Optimization**



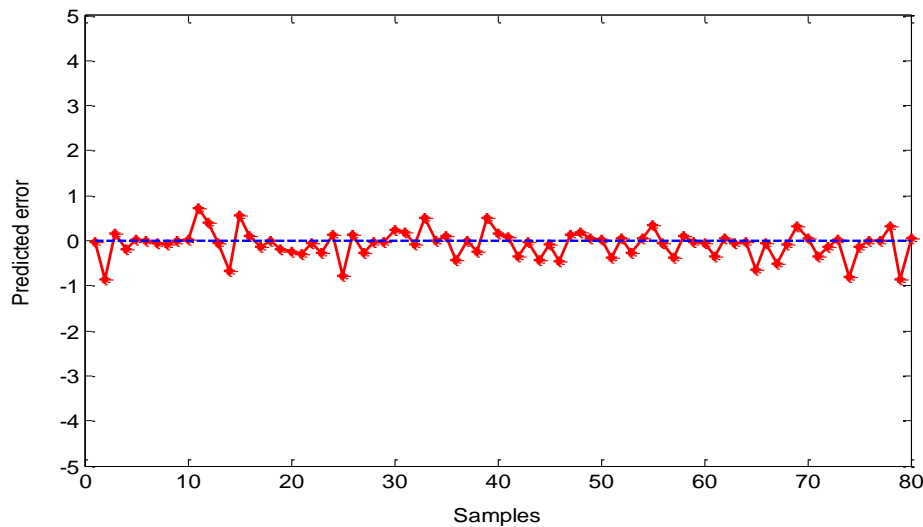
**Figure 4. The Results of Soft-sensor Model Trained by Training Set**



**Figure 5. Error between Predicted Outputs and Actual Samples based on Training Set**



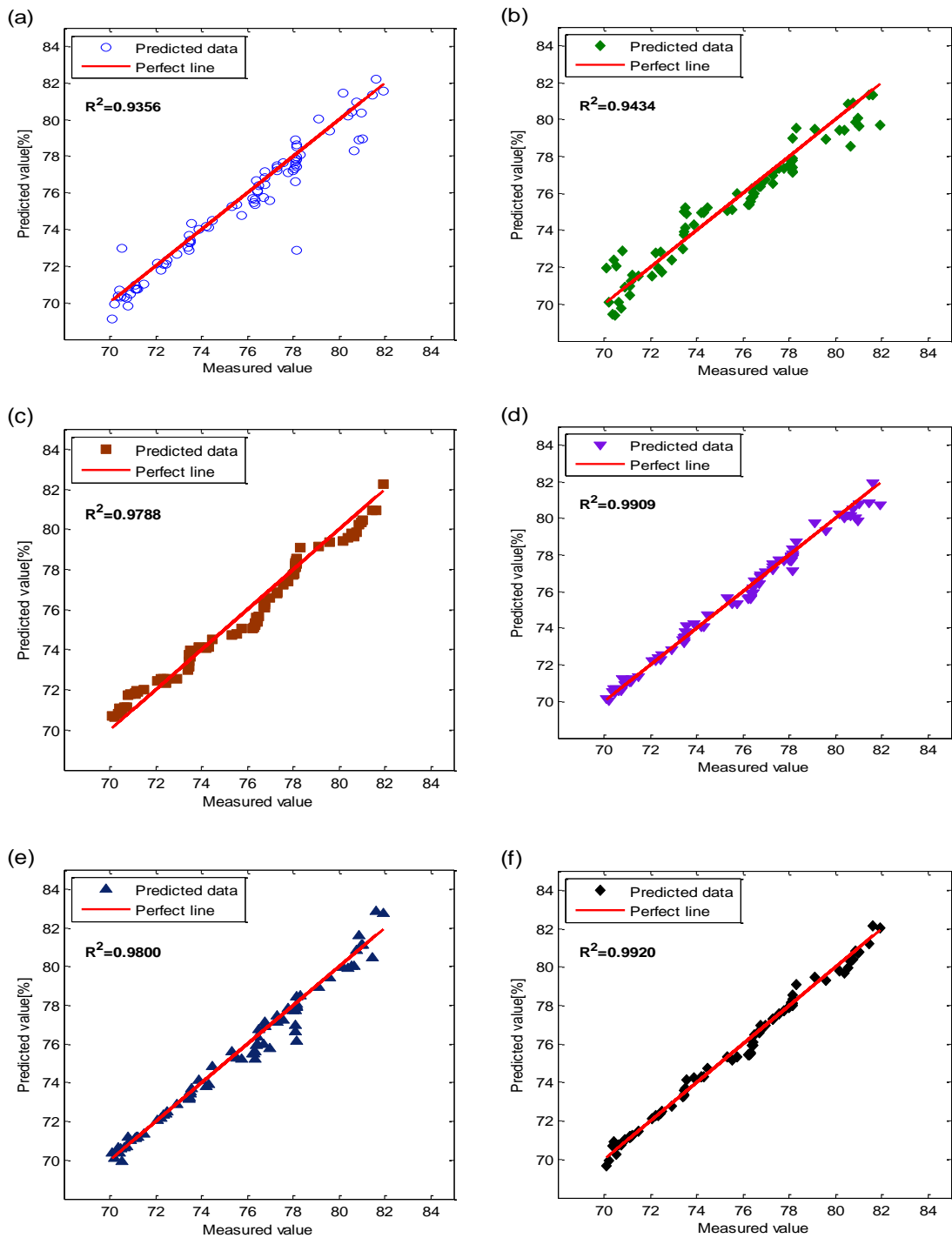
**Figure 6 The Prediction of Soft-sensor Model Regarding to Testing Set**



**Figure 7. Error between Predicted Outputs and Actual Samples based on Testing Set**

#### 4.3. Comparison

To validate the methods we proposed above, other model methods, like BP, RBF, SVM, LS-SVM and Original ELM were put into comparison. The details of model initialization were described as follows. ① BP neural work: Its network structure was of  $7 \times 3 \times 1$  layers. The transfer function of hidden layers was defined by 'tansig', while 'purelin' represented the transfer function of output layers. Similarly, the BP network training function was 'trainlm', and 'learnqdm' stood for the learning function of weight and bias. The learning rate and the mean squared error goal were set to 0.1 and 0.0001 respectively. ② RBF network: An ordinary RBF network structure was chosen. The mean squared error goal was set as 0.0, while the spread of radial basis functions was 1.0. ③ SVM and LS-SVM: Gaussian kernel function; the cost parameter and kernel parameter were optimized by PSO. ④ Original ELM: To find out what the number of hidden layer neurons would result in an optimal ELM and exploit this ELM to test the samples involved in this paper. Fig.8 showed the comparisons between predicted output and actual samples of these models. Obviously, decision coefficient varied with each model method. The method proposed in this paper yielded the decision coefficient  $R^2$  as much as 0.9920, which demonstrated the soft-sensor model based on ELM and PSO had better fitting precision and higher prediction accuracy of all these models. Table 3 gave the RMSE, MAE and MRE of these models built on different methods. As you can see, the performance of the objective model in this paper was acceptable, where RMSE=0.3121, MAE=0.2167%, MAPE=0.2843%. In the other hand, BP and RBF were the worst models among them, while PSO-SVM, PSO-LSSVM and PSO-ELM took a longer training time than BP and RBF due to their application of PSO. Even if model methods with PSO were time-consuming for model building, they also offer the better model performance. Compared with BP, RBF and methods with PSO, Original ELM exceeded at model development time and model performance. To sum up, soft-sensor model based on ELM and PSO fused the advantages of ELM and PSO to make itself stand out among these model methods.



**Figure 8. The decision coefficient values of different soft-sensor models. (a) BP , (b) RBF , (c) PSO-SVM , (d) PSO-LS-SVM , (e) Primal ELM , (f) ELM with Gaussian Kernel**

**Table 3. Performance of BP, RBF, PSO-SVM, PSO-LS-SVM, Primal ELM and ELM with Gaussian Kernel for Soft-Sensor Model**

Algorithms	Training time(s)	Training dataset			Testing dataset		
		RMSE	MRE(%)	MAE	RMSE	MRE(%)	MAE
BP	2.2112	0.4808	0.4719	0.3520	0.9274	0.7473	0.5712
RBF	0.7534	0.5991	0.6299	0.4686	0.8446	0.8971	0.6770
PSO-SVM	30.8739	0.2366	0.2295	0.1732	0.5347	0.4895	0.3760
PSO-LS-SVM	18.3275	0.1787	0.1741	0.1311	0.3706	0.3524	0.2719
Primal ELM	0.0146	0.2078	0.2235	0.1675	0.5066	0.4418	0.3407
Proposed	14.5497	<b>0.1489</b>	<b>0.1497</b>	<b>0.1122</b>	<b>0.3121</b>	<b>0.2843</b>	<b>0.2167</b>

## 5. Conclusion

This paper proposed a new soft-sensor model based on ELM for mother liquor purity in the cane sugar crystallization process. The optimization process of ELM with Gaussian kernel is carried on by PSO for a purpose of optimal model. As usual, model is trained and tested with samples data acquire from a novel experimental plant. Also, comparison with other model methods like BP, RBF, SVM and LS-SVM is conducted to prove that the soft-sensor model based on ELM with Gaussian kernel is of higher prediction accuracy and better performance. The feasibility of purity measurement propels the development of controlling the sugar crystallization process, so as to improve the sugar productivity. The future work will focus on introducing mechanism knowledge of sugar crystallization process into model development, thereby to address the multi-phase and inter-coupling issues. Meanwhile, the whole sugar process would be categorized into several states, and models for each state are trained based on their respective state data, in a way to improve the liability and accuracy of models.

## Acknowledgements

This research is supported by National Natural Science Foundation of China (No. 51065004), Guangxi Natural Science Foundation (No. 2011GXNSFA018168) and Nanning Scientific and Technological Project (No. 20131079).

## References

- [1] B. Grondin-Perez, M. Benne, C. Bonnacaze and J. P. Chabriat, "Industrial multi-steo forward predictor of mother liquor purity of the final stage of a cane sugar crystallization plant", *Journal of Food Engineering*, vol. 66, (2005), pp. 361-367.
- [2] B. Grondin-Perez, M. Benne and J. P. Chabriat, "Supervision of C crystallization in Bois Rouge sugar mill using on-line crystal content estimation using synchronous microwave and refractometric Brix measurement", *Journal of Food Engineering*, vol. 76, (2006), pp. 639-645.
- [3] C. Damour, M. Benne, L. Boillereaus, B. Grondin-Perez and J. P. Chabriat, "NMPC of an industrial crystallization process using model-based observers", *Journal of Industrial and Engineering Chemistry*, vol. 16, (2010), pp. 708-716.
- [4] C. Damour, M. Benne, B. Grondin-Perez and J. P. Chabriat, "Nonlinear predictive control based on artificial neural network model for industrial crystallization", *Journal of Food Engineering*, 99, (2010), pp. 225-231.
- [5] R. D. Braatz, "Advanced control of crystallization processes", *Annual Reviews in Control*, vol. 26, (2002), pp. 87-99.
- [6] D. Devogelaere, M. Rijckaert, O. G. Leon and G.C. Lemus, "Application of feed forward neural networks for soft sensors in the sugar industry", *Proceedings of the VIIth Brazilian Symposium on Neural Networks*, (2002), pp. 2-6.



- [7] P. Georgieva and SF. de. Azevedo, "Neural network-based control strategies applied to a Fed-batch crystallization process", *Computational Intelligence*, vol. 3, (2006), pp. 224-233.
- [8] Q. Hu, S. Rohani, DX. Wang and A. Jutan, "Optimal control of a batch cooling seeding crystallizer", *Powder Technology*, vol. 156, (2005), pp. 170-176.
- [9] N. Moldovanyi, B. Lakatos and F. Szeifert, "Model predictive control of MSMR crystallizers", *Journal of Crystal Growth*, vol. 275, (2005), pp. 1349-1354.
- [10] Z. K. Nagy, "Model based robust control approach for batch crystallization product design, *Computers & Chemical Engineering*", vol. 33, (2009), pp. 1685-1691.
- [11] W. Paengjuntuek, A. Arpornwichanop and P. Kittisapakorn, "Product quality improvement of batch crystallizer by a batch-to-batch optimization and nonlinear control approach", *Chemical Engineering Journal*, vol. 139, (2008), pp. 344-350.
- [12] S. Aouragh, N. Hassani, K. Saidi, and T. Bounahmidi, "Steady state modeling and simulation of an industrial sugar continuous crystallizer", *Computers & Chemical Engineering*, vol. 25, (2001), pp.1351-1370.
- [13] M. Sheikhzadeh, M. Trifkovic, and S. Rohani, "Real-time optimal control of an anti-solvent isothermal semi-batch crystallization process", *Chemical Engineering Science*, vol. 63, (2008), pp. 829-839.
- [14] A. Simoglou, P. Georgieva, EB. Martin, AJ. Morris, and SF. de Azevedo, "On-line monitoring of a sugar crystallization process", *Computers & Chemical Engineering*, vol. 29, (2005), pp. 1411-1422.
- [15] U. Vollmer and J. Raisch, "HN-Control of a continuous crystallizer, *Control Engineering Practice*", vol. 9, (2001), pp. 837-845.
- [16] C. Damour, M. Benne, L. Boillereaux, B. Grondin-Perez and J. P. Chabriat, "Multivariable linearizing control of an industrial sugar crystallization process", *Journal of Process Control*, vol. 21, (2011), pp. 46-54.
- [17] C. Damour, M. Benne, L. Boillereaux, B. Grondin-Perez and J. P. Chabriat, "Soft-sensor for industrial sugar crystallization: On-line mass of crystals, concentration and purity measurement", *Control Engineering Practice*, vol. 18, (2010), pp. 839-844.
- [18] J. Yu, "A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses", *Computers and Chemical Engineering*, vol. 41, (2012), pp. 134-144.
- [19] P. Kadlec, B. Gabrys and S. Strandt, "Data driven soft sensor in the process industry", *Computers and Chemical Engineering*, vol. 33, (2009), pp. 795-814.
- [20] L. Fortuna, S. Graziani, A. Rizzo and M. G. Xibilia, "Soft sensors for monitoring and control of industrial processes", (2007), London: Springer Verlag.
- [21] M. Kano and Y. Nakagawa, "Data-based process monitoring, process control, and quality improvement: Recent developments and applications in steel industry", *Computers and Chemical Engineering*, vol. 32, (2008), pp. 12-24.
- [22] G. H. Liu, D. W. Zhou, H. X. Xu and C. L. Mei, "Model optimization of SVM for a fermentation soft sensor", *Model optimization of SVM for a fermentation soft sensor*, vol. 37, (2010), pp. 2708-2713.
- [23] P. Kadlec, B. Gabrys and S. Strandt, "Data-driven soft sensors in the process industry, *Computers and Chemical Engineering*", vol. 33, no. 4, (2009), pp. 795-814.
- [24] M. Kano and M. Ogawa, "The state of the art in chemical process control in Japan: Good practice and questionnaire survey, *Journal of Process Control*", vol. 20, no. 9, (2010), pp. 969-982.
- [25] R. Grbic, D. Sliskovic, P. Kadlec, "Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models", *Computers and Chemical Engineering*, vol. 58, (2013), pp. 84-97.
- [26] Oscar A. Z. Sotomayor, W. P. Song and C. Garcias, "Software sensor for on-line estimation of the microbial activity in activated sludge systems", *ISA Transactions*, vol. 41, (2002), pp. 127-143.
- [27] D. Simon and D. L. Simon, Kalman Filter Constraint Switching for Turbofan Engine Health Estimation, *European Journal of Control*, vol. 12, (2006), pp. 331-343.
- [28] S. Bosca and D. Fissore, "Design and validation of an innovative soft-sensor for pharmaceuticals freeze-drying monitoring", *Chemical Engineering Science*, vol. 66, (2011), pp. 5127-5136.
- [29] H. Zhang and B. Lennox, "Integrated condition monitoring and control of fedbatch fermentation processes", *Journal of Process Control*, vol. 14, (2004), pp. 41-50.
- [30] B. Lin, B. Recke, J. K. H. Knudsen and S. B. Jorgensen, "A systematic approach for soft sensor development", *Computers and Chemical Engineering*, vol. 31, (2007), pp. 419-425.
- [31] E. Zamproga, M. Barolo and D. E. Seborg, "Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis", *Journal of Process Control*, vol. 15, (2005), pp. 39-52.
- [32] A. Bucinski, M. Karamac, R. Amarowicz and R. Pegg, "Modeling the tryptic hydrolysis of pea protein using an artificial neural network", *LWT-Food Science and Technology*, vol. 41, no. 5, (2008), pp. 942-945.
- [33] A. Imenez, G. Beltran, M. P. Auilera and M. Uceda, "A sensor-software based on artificial neural network for the optimization of olive oil elaboration process", *Sensors and Actuators B: Chemical*, vol. 129, no. 2, (2008), pp. 985-990.

- [34] B. G. Ankti, B. J. Jyeshtharaj, K. J. Valadi and D. K. Bhaskar, "Development of support vector regression (SVR)-based correlation for prediction of overall gashold-up in bubble column reactors for various gas-liquid systems", *Chemical Engineering Science*, vol. 62, (2007), pp. 7078-7089.
- [35] W. Yan and H. Shao, "Soft sensing modeling based on support vector machine and Bayesian model selection", *Computer and Chemical Engineering*, vol. 28, no. 8, (2004), pp. 1489-1498.
- [36] A. Chen, Z. Song and P. Li, "Soft sensor modeling based on dica-svr", *Lecture Notes in Computer Science*, 3644, (2005), pp. 868-877.
- [37] H. Kaneko, M. Arakawa, and K. Funatsu, (2009), "Development of a new soft sensor method using independent component analysis and partial least squares", *AIChE Journal*, vol. 55, pp. 87-98.
- [38] J. C. B. Gonzaga, L. A. C. Meleiro, C. Kiang and R. Maciel Filho, "ANN-based soft-sensor for real-time process monitoring and control of an industrial poly-merization process", *Computers and Chemical Engineering*, vol. 33, no. 1, (2009), pp. 43-49.
- [39] K. Desai, Y. Badhe, S. S. Tambe and B. D. Kulkarni, "Soft-sensor development for fed-batch bioreactors using support vector regression", *Biochemical Engineering Journal*, vol. 27, no. 3, (2006), pp. 225-239.
- [40] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: Optimally Pruned Extreme Learning Machine, *IEEE Trans. Neural Networks*, vol. 21, no.1, (2010), pp. 158-162.
- [41] T. Chow and S.Y. Cho, "Neural networks and computing-learning algorithms and applications", Imperial College Press, (2007).
- [42] D.S. Huang and Ji-Xiang Du, "A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks", *IEEE Trans. Neural Networks*, vol. 19, no. 12, (2008), pp. 2099-2115.
- [43] R. Zhang, Z. B. Xu, G. B. Huang and D. H. Wang, "Global convergence of online BP training with dynamic learning rate", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, (2012), pp. 330-341.
- [44] M. S. Al-Batah, N. A. M. Isa, K. Z. Zamli and K. A. Azizli, "Modified recursive least squares algorithm to train the hybrid multilayered perceptron (HMLP) network", *Applied Soft Computing*, vol. 10, no. 1, (2010), pp. 236-244.
- [45] Y. Kayaa and M. Uyar, "A hybrid decision support system based on rough set and extreme learning machine for diagnosis of hepatitis disease", *Applied Soft Computing*, 13, (2013), pp. 3429-3438.
- [46] A.A. Mohammed, R. Minhas, Q.M. Jonathan Wu and M. A. Sid-Ahmed, "Human face recognition based on multidimensional PCA and extreme learning machine, *Pattern Recognition*", vol. 44, (2011), pp. 2588-2597.
- [47] G. B. Huang, Q. Y. Zhu and C. K. Siew, "Extreme learning machine: theory and applications", *Neurocomputing*, vol. 70, (2006), pp. 489-501.
- [48] G. B. Huang, Q. Y. Zhu and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks", *Proceedings of International Joint Conference on Neural Networks, IJCNN'04*, vol. 2, Budapest, Hungary, 25-29, July (2004), pp. 985-990.
- [49] G. B. Huang, L. Chen and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes", *IEEE Transactions on Neural Networks*, vol. 17, no. 4, (2006), pp. 879-892.
- [50] G. B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine, *Neurocomputing*", vol. 71, (2008), pp. 3060-3068.
- [51] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine, *Neurocomputing*", vol. 70, (2007), pp. 3056-3062.
- [52] R. Zhang, Y. Lan, G. B. Huang and Z. B. Xu, "Universal approximation of extreme learning machine with adaptive growth of hidden nodes", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 2, (2012), pp. 365-371.
- [53] H.G. Han, L.-D. Wang and J.-F. Qiao, "Hierarchical extreme learning machine for feedforward neural network", *Neurocomputing*.
- [54] G. B. Huang, X. J. Ding and H. M. Zhou, "Optimization method based extreme learning machine for classification", *Neurocomputing*, vol. 74, (2010), pp. 155-163.
- [55] G. B. Huang, H. M. Zhou, X. J. Ding and R. Zhang, "Extreme learning machine for regression and multiclass classification", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 42, no. 2, (2012), pp. 513-529.
- [56] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*", Nagoya, Japan, (1995), pp. 39-43.
- [57] J. Kennedy and R.C. Eberhart, Particle swarm optimization, "Proceedings of the IEEE International Conference on Neural Networks", (1995), pp. 1942-1948.
- [58] F. Han, H. F. Yao and Q. H. Ling, "An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing*", vol. 116, (2013), pp. 87-93.
- [59] X. C. Guo, J. H. Yang, C. G. Wu, C. Y. Wang and Y. C. Liang, "A novel LS-SVMs hyper-parameter selection based on particle swarm optimization", *Neurocomputing*, vol. 71, (2008), pp. 3211-3215.