Journal of Applied Polymer Science

Quality Control in the Polypropylene Manufacturing Process: An Efficient, Data-Driven Approach

Zhong Cheng,[1,2] Xinggao Liu[2]

[1]School of Biological and Chemical Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, People's Republic of China

[2]State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, People's Republic of China

Correspondence to: X. Liu (E-mail: lxg@zjuem.zju.edu.cn)

ABSTRACT: In the propylene polymerization process, the melt index (MI), as a critical quality variable in determining the product specification, cannot be measured in real time. What we already know is that MI is influenced by a large number of process variables, such as the process temperature, pressure, and level of liquid, and a large amount of their data are routinely recorded by the distributed control system. An alternative data-driven model was explored to online predict the MI, where the least squares support vector machine was responsible for establishing the complicated nonlinear relationship between the difficult-to-measure quality variable MI and those easy-to-measure process variables, whereas the independent component analysis and particle swarm optimization technique were structurally integrated into the model to tune the best values of the model parameters. Furthermore, an online correction strategy was specially devised to update the modeling data and adjust the model configuration parameters via adaptive behavior. The effectiveness of the designed data-driven approach was illustrated by the inference of the MI in a real polypropylene manufacturing plant, and we achieved a root mean square error of 0.0320 and a standard deviation of 0.0288 on the testing dataset. This proved the good prediction accuracy and validity of the proposed data-driven approach. © 2014 Wiley Periodicals, Inc. J. Appl. Polym. Sci. **2015**, *132*, 41312.

KEYWORDS: manufacturing; synthesis and processing; theory and modeling

Received 28 May 2014; accepted 21 July 2014

DOI: 10.1002/app.41312

INTRODUCTION

The purpose of the industrial polypropylene manufacturing process is to produce qualified products, and then, the quality control of the products becomes the core problem. The melt index (MI), as a key parameter in determining the product's grade and quality control of polypropylene, is generally evaluated offline with an analytical procedure that takes almost 2 h to complete in the laboratory.[1] This causes a large time delay in the quality control system without any quality indicator during this period of time. Currently, a number of first-principle models, black box models that use neural networks, statistical data modeling techniques, and hybrid models have been developed to monitor, control, and optimize these complex processes.[2,3] Theoretically, an online analyzer can be constructed with a physical process model on the basis of the description of the polymerization behaviors; this is also called the *knowledge-driven model*.[4–9] However, it is difficult to build an adequate knowledge-driven model for predicting the MI because of the lack of understanding and/or to the ability to satisfactorily construct a mathematical model of this polymerization process.[10–14] Fortunately, many researchers have successfully inferred the MI with many kinds of soft-sensor models that relate the MI to other easily measurable process variables from the huge amount of historical data stored in the real-time database of the distributed control system (DCS). These soft sensors, also called *data-driven models*, which involve mathematical equations, are assessed from the analysis concurrent input and output data. For example, Lou et al.[15] proposed a novel, multiple-priori knowledge-based neural network inferential model for MI prediction. By embedding previous knowledge, the model ensured safety in the quality control of the MI. Zhang et al.[16] set up a novel radial basis function (RBF) prediction model under chaotic theory for MI prediction and achieved quite good performances. Jiang et al.[17] devised an optimal soft-sensing model, called *least squares support vector machines with ant colony-immune clone particle swarm optimization* (AC–ICPSO–LS-SVM), to predict the MI successfully. Ge et al.[18] developed a so-called combined local Gaussian process regression, and it showed the best MI quality prediction results. Park et al.[19] used

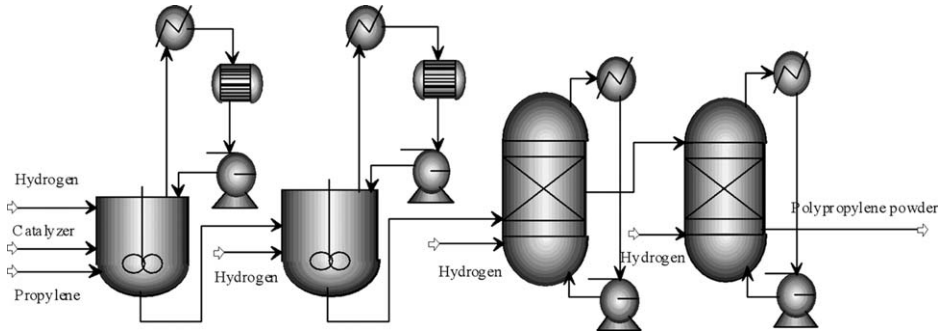

Figure 1. Schematic of the propylene polymerization process.

partial least squares (PLS) and support vector regression to predict the MI in the high-density polyethylene process. The simulation results show that both PLS and support vector regression exhibited excellent prediction performances, even for operating situations accompanying severely frequent grade changes. Han et al.[20] introduced three different approaches, including support vector machines (SVMs), PLS, and back-propagation neural networks, to estimate the MI and concluded that the standard SVM yielded the best prediction. Rallo et al.[21] provided a fuzzy adaptive resonance theory map (ARTMAP) neural system and two hybrid networks to infer the MI of six different low-density polyethylene grades produced in a tubular reactor. Although these works achieved better and better MI prediction accuracy, the greater prediction performance and the better universality of the estimation model were still the first-line goal in both the academic and industrial communities.

As is well known, the polypropylene manufacturing process is a highly nonlinear process evidenced by mechanistic analysis of the reactions and plants. So, nonlinear data-driven models should be considered. Now, the least squares support vector machines (LS-SVM) method is widely used in areas of nonlinear system identification, optimal control, and pattern recognition.[22–24] As a soft-sensing method, its learning and generalization ability is greatly affected by the parameter settings and the selection of input variables. Independent component analysis (ICA),[25] as a very general-purpose projection technique, can be used to extract relevant features and to concurrently obtain fewer underlying components that are maximally independent from each other. This is also beneficial for cutting down the subsequent data-driven model structure. In virtue of the global convergence characteristics, the particle swarm optimization (PSO) algorithm has proven to be very successful and efficient in identifying the optimum parameters.[26] Accordingly, on the basis of feature selection and parameter optimization techniques, an optimal LS-SVM data-driven model was developed, in which ICA was in charge of discerning the inputs of the LS-SVM model and PSO was responsible for optimizing the model parameters. Here, it is worthwhile to note that the appropriate configuration parameters of LS-SVM usually covary with the selection of input variables. Thus, in our designed procedure, the selection of feature variables and the parameter setting of LS-SVM were regarded as a combination optimal problem, and an objective function based on the root mean square error (RMSE) was constructed. Their optimum values were searched by a fine-grid-based experimental design. Moreover, when addressing the process time-varying nature, we explored online correction strategy (OCS) to update the modeling data and adjust the values of the model configuration parameters via adaptive behavior. This scheme in a fashion to minimize the prediction error was only activated whenever a model mismatch happened with the addition of new process data and the removal of older ones recursively.

The rest of this article is organized as follows. In the Experimental section, we briefly describe the polypropylene manufacturing process and the data collection. Then, we present the framework of the proposed data-driven modeling approach and demonstrate in detail the principles and techniques in building the soft-sensor model. In the Results and Discussion section, we review the research results of the previous approach on a real plant with different batches and discuss all sides. Lastly, we make some concluding remarks.

EXPERIMENTAL

Process Description and Data Collection

The process considered here was the propylene polymerization manufacturing process, which is currently used for commercial purposes in real plants. To obtain an intuitive understanding of the process, its simple schematic diagram is illustrated in Figure 1. The process consists of a chain of reactors in series, two continuous stirred-tank reactors, and two fluidized-bed reactors. Propylene and hydrogen are fed into each reactor, but the catalyst is added only to the first reactor along with the solvent. These liquids and gases supply reactants for the growing polymer particles and provide the heat-transfer media. The polymerization reaction takes place in the liquid phase in the first two reactors and is completed in the vapor phase in the third and fourth reactors to produce the powdered polymer products.

Thirty process variables were measured in our study. According to the workers' experience in operating the propylene polymerization plant and our thorough analysis of its mechanism,[27] a pool of nine process variables (T, P, H, a, $f1$, $f2$, $f3$, $f4$, and $f5$) were chosen as the major factors or the original process variables to develop the MI prediction model, where T, P, H, and a are the process temperature, pressure, level of liquid, and percentage of hydrogen in the vapor phase, respectively; $f1$, $f2$, and $f3$ are the flow rates of the three streams of propylene; and $f4$ and $f5$ are the flow rates of the catalyst and aid-catalyst, respectively.

The data used for training, testing, and generalizing the developed data-driven model were acquired from the historical logs

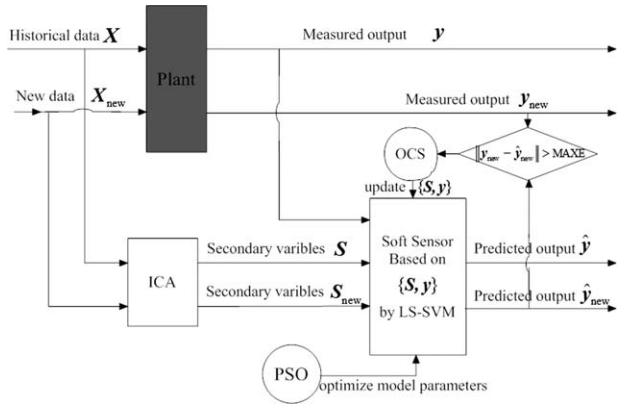

Figure 2. Structure of the proposed data-driven approach.

recorded in a real propylene polymerization plant. The sampling time of MI was 2 h, the dataset of nine process variables were from the DCS database, and their interval time of automatic recording by DCS was only several seconds. The average residence time for this real propylene polymerization process was also about 2 h and was considered in the data initialization. First, they were filtered to discard outliers, and a collection of 85 pairs of input–output data was used in this research. Fifty pairs were used as the training dataset, another 20 pairs were the testing dataset, and the rest were the generalization dataset, the latter two of which were used to test the accuracy and the universality of the models, respectively. The testing dataset was taken from the same batch as the training dataset, whereas the generalization dataset was obtained from a different one. All of these original input and output data are normalized linearly into the range [0,1]. Consequently, the direct output of the prediction model was denormalized to obtain the true MI value.

Data-Driven Modeling and Optimization

The architecture of our proposed data-driven modeling approach is depicted in Figure 2. It consisted of four modules: feature extraction, nonlinear modeling, parameter optimization, and dynamic adjustment. First, from the selected original process variables having an influence on MI, the independent components (or secondary variables) were extracted by ICA. Subsequently, LS-SVM was granted to establish a nonlinear model between these secondary variables and the response variable MI. At the same time, the optimum values of the model configuration parameters were sought through the PSO evolutionary algorithm. Because a reliable dynamic model could be useful for long-term simulation, the OCS was integrated to modify a trained LS-SVM model by means of incremental updating and decremental pruning algorithms. Henceforth, the developed data-driven model could also be called OCS–PSO–ICA–LS-SVM.

Feature Extraction: ICA

Included in a prediction model, *highly correlated variables,* or variables that are unrelated to the outcome of interest, can lead to overfitting, and the accuracy and reliability can suffer. The aim of ICA is to recover independent sources given only sensor observations. In contrast to correlation-based transformations, such as principal component analysis (PCA), ICA not only decorrelates the signals (second-order statistics) but also reduces higher order statistical dependencies.[25] In this context, we first used ICA for data reduction. We computed a compact and optimal description of the model inputs.

Consider a process data matrix $\mathbf{X}$ ($n \times p$) composed of n samples and p process variables. In the ICA algorithm, the measured process variables $x \in \Re^p$ (where x the measured process vector in the p-dimensional Euclidean space) can be expressed as a linear combination of unknown independent components ($s = [s_1, s_2, \ldots, s_d] \in \Re^d$ (where s the independent vector in the d-dimensional Euclidean space), where d is the number of independent components), that is

$$x = \boldsymbol{A}\boldsymbol{s} \tag{1}$$

where $\mathbf{A} \in \Re^{p \times d}$ is the mixing matrix. ICA tries to estimate $\mathbf{A}$ and s only from the known x. Therefore, it is necessary to find a demixing matrix ($\boldsymbol{P}$), which is given as

$$\hat{\boldsymbol{s}} = \boldsymbol{P}\boldsymbol{x} \tag{2}$$

such that the reconstructed vector ($\hat{\boldsymbol{s}}$) becomes as independent as possible. For convenience, we assumed $d = p$ and $E(ss^T) = \boldsymbol{I}$. where s^T is the transpose of s, $\boldsymbol{I}$ is an identity matrix. The whitening transformation (t) is expressed as

$$\boldsymbol{t} = \boldsymbol{Q}x = \boldsymbol{Q}\boldsymbol{A}s = \boldsymbol{B}s \tag{3}$$

where the whitening matrix $\boldsymbol{Q} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{U}^{\mathrm{T}}$, $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalues of the data covariance matrix, $\boldsymbol{U}$ is a matrix with the corresponding eigenvectors as its columns, and $\boldsymbol{B}$ is an orthogonal matrix. The relationship between $\mathbf{P}$ and $\boldsymbol{B}$ is as follows:

$$\boldsymbol{P} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{Q} \tag{4}$$

Hence, eq. (2) can be rewritten as

$$\hat{\boldsymbol{s}} = \boldsymbol{P}x = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{Q}x = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{U}^{\mathrm{T}}x \tag{5}$$

According to eq. (5), the ICA problem can be reduced to find $\boldsymbol{B}$.

To calculate $\boldsymbol{B}$, Hyvärinen[28] introduced a fast, fixed-point algorithm for ICA. For large amounts of data, the data-reducing algorithm for estimating independent components[29] can be substituted here.

Nonlinear Modeling: LS-SVM

LS-SVM is a regularized supervised approximator.[30] Given a training dataset of n data points $\{(s_i, y_i)\}_{i=1}^{n}$ with the ith input $s_i \in \Re^d$ (where s_i is the i-th data point in the d-dimensional Euclidean space) and its corresponding output $y_i \in \Re$, the following regression model was used:

$$y(s) = \boldsymbol{w}^{\mathrm{T}}\varphi(s) + b \tag{6}$$

where $\varphi(\cdot)$ is a mostly nonlinear function that maps the input variable space into a higher dimensional embedding space, $\boldsymbol{w}$ is a weight vector, b is the bias term, and y is the response variable. Next in the LS-SVM for function estimation, the objective function of the optimization problem is defined as follows:

$$\min_{w,b,\xi} J(\boldsymbol{w}, b, \xi) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{1}{2}\gamma\sum_{i=1}^{n}\xi_i^2 \tag{7}$$

and is subject to the equality constraints

$$y_i = \boldsymbol{w}^T \varphi(s_i) + b + \xi_i,\ i = 1, 2, \ldots, n \tag{8}$$

where γ is the user-defined regularization factor, also known as the hyper parameter, which balances the model complexity and approximation accuracy, and ξ_i is the approximation error. To solve this constrained optimization problem, the corresponding Lagrange function $[L(w,b,\xi,\alpha)]$ is constructed as follows:

$$L(\boldsymbol{w}, b, \xi, \alpha) = J(\boldsymbol{w}, b, \xi) - \sum_{i=1}^{n} \alpha_i \left[\boldsymbol{w}^T \varphi(s_i) + b + \xi_i - y_i\right] \tag{9}$$

where α_i is the Lagrange multiplier. The solution of the previous equation can be obtained by partially differentiating with respect to each variable:

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \phi(s_i) \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} a_i = 0 \\ \dfrac{\partial L}{\partial \xi_i} = 0 \rightarrow \alpha_i = \gamma \xi_i,\ i = 1, 2, \ldots, n \\ \dfrac{\partial L}{\partial \alpha_i} = 0 \rightarrow \boldsymbol{w}^T \phi(s_i) + b + \xi_i - y_i = 0,\ i = 1, 2, \ldots, n \end{cases} \tag{10}$$

Where L is the Lagrange function defined in the above equation (9). which can be written as the solution to the following set of linear equations after the elimination of **w** and ξ_i:

$$\begin{bmatrix} 0 & 1_v^T \\ 1_v & \boldsymbol{K} + \gamma^{-1} \boldsymbol{I} \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{11}$$

with $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^{\mathrm{T}}$, $1_v = [1, 1, \ldots, 1_n]^{\mathrm{T}}$, $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^{\mathrm{T}}$, $\boldsymbol{I}$ is an $n \times n$ identity matrix and the subscript v notes that 1_v is a vector. $\boldsymbol{K}$ is known as the Gram matrix and its element K_{ij} $(s_i s_j) = \phi(s_i)^T \phi(s_j)$ is the kernel function and must follow Mercer's theory.[31] The classical examples of kernel functions are linear, polynomial, and Gaussian kernels. In this study, we considered the most common case of the LS-SVM model for regression, namely, the LS-SVM model with a Gaussian kernel,[32] the equation of which is

$$K(s, s_i; \sigma) = \exp\left(-\frac{\|s - s_i\|}{\sigma^2}\right) \tag{12}$$

Hereto, the resulting LS-SVM model for function estimation can be expressed as

$$y(s) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{s}, \boldsymbol{s}_i) + b \tag{13}$$

where α_i and b are the solution to eq. (11).

Model Optimization: PSO

Compared with genetic algorithm,[33] the PSO algorithm is easily implemented with fewer arguments; thus, it is very suitable and can help one obtain the global optimal solution.[26] PSO is initialized with a group of random particles (solutions), and then searches for optima are done by the updating of the generations.

Now, with the assumption that $\boldsymbol{s}_m = (\boldsymbol{s}_{m,1}, \boldsymbol{s}_{m,2}, \ldots, \boldsymbol{s}_{m,d})$ is the position of the mth particle in d-dimension space, $v_m = (v_{m,1}, v_{m,2}, \ldots v_{m,d})$ is its velocity, which represents its direction of searching. In the iteration process, each particle keeps the best position found by itself (L_{best}); it also knows the best position searched by the group particles (G_{best}) and changes its velocity according to these two best positions. After the two best values are determined, the particle updates its velocity and positions with the following two equations:[34]

$$\begin{aligned} v_{m,q}(t+1) = \eta v_{m,q}(t) &+ \phi_1 c_1 \left[L_{\mathrm{best}_{m,q}} - s_{m,q}(t)\right] \\ &+ \phi_2 c_2 \left[G_{\mathrm{best}_q} - s_{m,q}(t)\right] \end{aligned} \tag{14}$$

$$s_{m,q}(t+1) = s_{m,q}(t) + v_{m,q}(t+1) \tag{15}$$

where $v_{m,q}$ and $s_{m,q}$ are the velocity and its responding solution of the mth particle in the qth dimension ($q = 1,2,\ldots d$), respectively. Here, t describes current state, $t+1$ describes the next state, c_1 and c_2 are acceleration constants, ϕ_1 and ϕ_2 are random numbers uniformly distributed in [0,1], and η is the inertia factor for controlling the learning rate. Because the velocity of the particle has been determined, the particle's solution will be modified at the next time step $(t+1)$. According to eqs. (14) and (15), the direction of each particle will alter its trajectory and gradually move toward the direction of G_{best}.

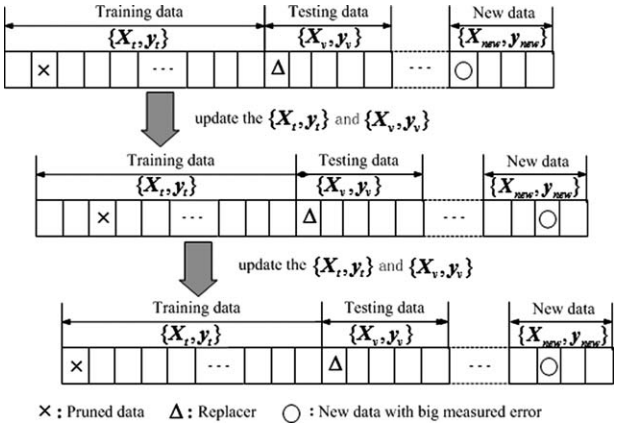

Figure 3. Schematic diagram of the OCS.

Dynamic Modification: OCS

OCS is used to deal with prediction control problem. An LS-SVM model with fixed parameters cannot be adapted to a dynamic system. The main idea of OCS is the timely updating of the modeling data and then the rebuilding of a new LS-SVM prediction model. The flow chart of OCS is shown in Figure 3.

According to the time sequence, the historical data are divided into two parts, the training dataset ($\{\boldsymbol{X}_t, \boldsymbol{y}_t\}$) and the testing dataset ($\{\boldsymbol{X}_v, \boldsymbol{y}_v\}$), to construct an LS-SVM model. When the newly measured variables ($\boldsymbol{X}_{\mathrm{new}}$) are collected online, their independent components (S_{new}) are extracted and fed into this established LS-SVM model, their corresponding prediction results ($\hat{y}_{\mathrm{new}}$) will be obtained. Because of the delay of analysis, the melt index analytic value ($\boldsymbol{y}_{\mathrm{new}}$) is available for a limited period of time. Now, OCS calculates the prediction error between $\boldsymbol{y}_{\mathrm{new}}$ and $\hat{y}_{\mathrm{new}}$. Provided that the prediction error of some individual from the new dataset exceeds the given limit, such as the maximum absolute error (MAXE) on the training dataset, this new individual should be added to the testing

dataset in time order. At the same time, the first sample from the testing data record will substitute for that with the biggest fitting error from the training dataset. Otherwise, with the purpose of reducing the expense of model modification, those new data with small prediction errors will not be entered into the testing dataset to rebuild the soft-sensor model.

Data-Driven Soft Sensor: Experimental Procedure

From what has been discussed previously, the whole procedure of the explored data-driven soft sensor can be summarized in the following steps:

Step 1. First, d is initialized to 1. The independent component, accompanied by **P,** is derived from the selected original process variables by ICA on the basis of the training dataset.

Step 2. The kernel parameter (σ) and γ are set with two random numbers, an initial LS-SVM model is established on this independent components and the corresponding response variable MI.

Step 3. Then, for the testing dataset, the original process variables are projected to **P** and produce their independent components.

Step 4. These independent components are fed into the initial LS-SVM model, and their MI prediction values are obtained.

Step 5. The RMSE indicator is counted on the testing dataset.

Step 6. The PSO algorithm is carried out, and this procedure is repeated from steps 2 to 5 to obtain the local optimal values of σ and γ.

Step 7. All of the previous six steps are repeated with alteration of σ, γ, and d in a grid search manner because d is integral and its optimal value will be in the range 1–9. In the end, the global optimum values of d, σ, and γ are searched in accordance with the minimum RMSE.

Step 8. It is determined whether the prediction errors of some new data points exceed their given limit, such as the MAXE. If they do, a new LS-SVM model should be set up by the substitution of these new data points for the older ones. Otherwise, the LS-SVM model, brought forth in the step 7, will be used as an effective soft sensor to predict MI all along, as shown in Figure 2.

RESULTS AND DISCUSSION

Model Performance Criteria

Models are commonly evaluated on the basis of comparisons against observations. This comparison is generally achieved with a set of statistical indicators to analyze the model performance. The difference between the model output and the desired output is referred to as the *error,* and it can be measured in different ways. Here, RMSE, MAXE, mean absolute error (MAE), mean relative error (MRE), standard deviation (STD), and Theil's inequality coefficient (TIC)[35] were used as derivation measurements between the measured and predicted values of MI. These error indicators were defined as follows:

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{16}$$

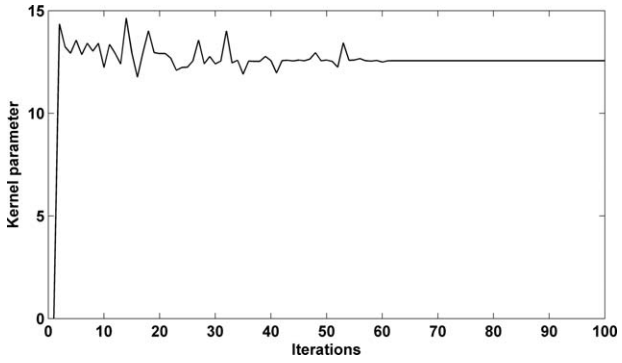

Figure 4. σ optimization curve with PSO.

$$\mathrm{MAXE}=\max_{i\in\{1,2,\ldots,n\}}\{|y_i-\hat{y}_i|\} \tag{17}$$

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{18}$$

$$\mathrm{MRE}=\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i-\hat{y}_i|}{y_i} \tag{19}$$

$$\mathrm{STD}=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(e_i-\bar{e})^2} \tag{20}$$

$$\mathrm{TIC}=\frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{n}y_i^2}+\sqrt{\sum_{i=1}^{n}\hat{y}_i^2}} \tag{21}$$

where $e_i=y_i-\hat{y}_i$, $\bar{e}=\frac{1}{n}\sum_{i=1}^{n}e_i$, and y_i and $\hat{y}_i$ denote the measured and predicted values, respectively. Among eqs. (16–21), MAE, MRE, and RMSE indicate the prediction accuracy of the proposed data-driven approach, whereas TIC and STD reveal the predictive stability of the approach. As for MAXE, it can reflect the model's resistance to those samples that show abrupt changes. Because the prediction accuracy is most important for the MI prediction models, these indicators can well evaluate the performance of the models. In the following text, different data-driven models are compared by their performances on the testing dataset and the generalization dataset.

Model Parameter Tuning

Suppose that we have a data-driven model with several unknown parameters and a training dataset to which the model can be fit. The learning process optimizes the model parameters to make the model fit the training data as well as possible. If we then take an independent sample of testing data from the same population as the training data, it will generally turn out that the model does not fit the testing data as well as it fits the training data. This is called *overfitting.* Thus, to prevent overfitting and assess how a model will generalize to the independent dataset, a testing dataset is used to construe the data-driven model and decide the model parameters.

In our proposed data-driven modeling approach, the optimal values of all of its parameters were determined by the previous RMSE indicator. That is, eq. (16) was now used as the objective function and fitness function to be minimized. There were too many parameters that had to be chosen carefully so as to ensure good performance of the data-driven model. First, d in ICA was fixed according to the minimal prediction error on the testing dataset by the cross-validation technique.[36] For example, four

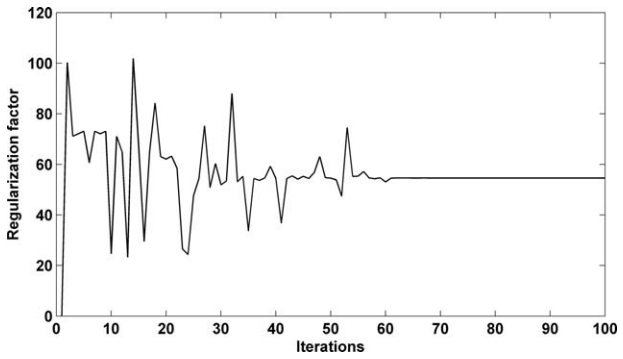

Figure 5. γ optimization curve with PSO.

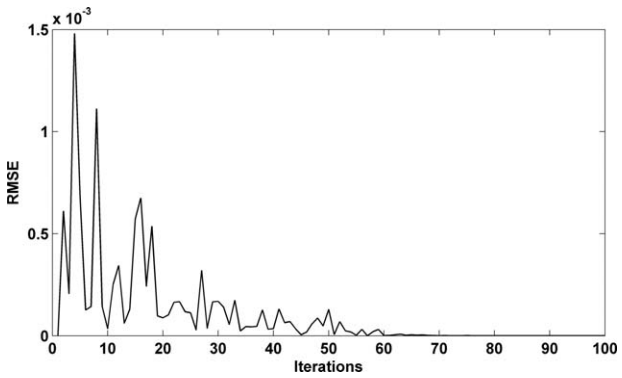

Figure 6. RMSE tendency curve with different σs and γs.

independent components ($d = 4$) were used in our proposed OCS-PSO-ICA-LS-SVM model. What about the σ and γ in the LS-SVM nonlinear modeling? It was clear that σ and γ were related to each other and had an impact on the model nonlinearity in company. γ was in charge of the trade-off between the model smoothness and accuracy. If γ became larger, less regularization was applied, and this led to a more nonlinear model. The σ value was related to the distance between the training data points and to the smooth interpolation of the resulting LS-SVM model.[37] If σ became larger, more neighbors were included in the model; this led to a more linear model. Therefore, it was most important to give their proper values, and the PSO algorithm was used to locate the global optimal solution by the fine tuning of the search process. As a result of the developed data-driven model and with increasing number of iterations, the convergence characteristics of σ, γ, and model precision RMSE are shown in Figures 4, 5, and 6, respectively.

In line with the σ and γ shift, the tendency curves of their common objective function converged gradually, and the searched results were RMSE $= 7.7617 \times 10^{-7}$ with $\sigma = 12.564$ and $\gamma = 54.597$ at the 69th iteration time. Other parameters of the PSO algorithm were chosen as follows: the particle number was set to 30, and the constants η, c_1, and c_2 were initialized to 0.75, 2.0, and 2.0, respectively. The initial value of the velocity was constrained into $[-1.0,1.0]$. The algorithm stopped when the indicator RMSE was 10^{-6} or the iteration times exceeded 100.

Results and Analysis

To evaluate the capability of our developed data-driven model, various model performance indicators on the testing dataset and on the generalization dataset were calculated and are separately listed in Tables I and II. In addition, several other data-driven modeling methods, such as simple LS-SVM, LS-SVM with ICA (called ICA-LS-SVM), and LS-SVM with PSO and ICA (called PSO-ICA-LS-SVM), were used to serve as comparative methods, and their model structures and parameters were also tuned perfectly.

As far as the model prediction ability was concerned, there are six indicators in Table I, and all of their results indicate that our proposed OCS-PSO-ICA-LS-SVM model comprehensively exceeded the other models. Specifically, it had an MAE of 0.0551, which was significantly lower than those of 0.0685, 0.0775, and 0.0827 from the PSO-ICA-LS-SVM, ICA-LS-SVM, and LS-SVM models, respectively. As a supplement, the MAE was 0.0842 from the LS-SVM reported by Shi and Liu.[38] Similarly, the RMSE also confirmed that the proposed method was superior for the model prediction accuracy. Then, the OCS-PSO-ICA-LS-SVM yielded the smallest STD among the four methods, which revealed the predictive stability of the method. Moreover, TIC of the proposed data-driven model was quite acceptable when compared with those of the other three different models; this indicated a good level of agreement between the proposed model and the studied propylene polymerization process.

To further explore the effectiveness of our proposed data-driven approach, models were also evaluated on the generalization dataset, and the results are presented in Table II. It was noted that the performances were consistent with the previous testing data results, with a slightly decrease in the predictive precision. The MAE of OCS-PSO-ICA-LS-SVM was 0.0294, compared with 0.0607 of LS-SVM; it showed a decrease of approximately 52%. Similar behaviors were observed in terms of MAXE, MRE, RMSE, STD, and TIC. The high accurate prediction of MI on this dataset gave strong support for the fact that the OCS-PSO-ICA-LS-SVM model had good universality.

Table I. Model Performance Comparison with the Testing Dataset

Method	MAXE	MAE	MRE	RMSE	STD	TIC
LS-SVM	0.3343	0.0827	0.0354	0.1146	0.1042	0.0234
ICA-LS-SVM	0.4211	0.0775	0.0331	0.1089	0.0972	0.0223
PSO-ICA-LS-SVM	0.3625	0.0685	0.0296	0.0987	0.0989	0.0203
OCS-PSO-ICA-LS-SVM	0.3607	0.0551	0.0240	0.0894	0.0896	0.0184

Table II. Model Performance Comparison with the Generalization Dataset

Method	MAXE	MAE	MRE	RMSE	STD	TIC
LS-SVM	0.1113	0.0607	0.0238	0.0722	0.0775	0.0142
ICA-LS-SVM	0.0948	0.0549	0.0216	0.0687	0.0747	0.0135
PSO-ICA-LS-SVM	0.0818	0.0465	0.0182	0.0503	0.0526	0.0099
OCS-PSO-ICA-LS-SVM	0.0415	0.0294	0.0116	0.0320	0.0288	0.0062

For both the testing and generalization dataset, their MI measured values and the predicted values from the models are directly exhibited in Figures 7 and 8; this speaks even more powerfully than the results in Tables I and II do. Here, the curves marked with diamonds, circles, squares, and downward triangles are the MI values predicted by LS-SVM, ICA-LS-SVM, PSO-ICA-LS-SVM, and OCS-PSO-ICA-LS-SVM, respectively. Clearly, our proposed data-driven OCS–PSO-ICA-LS-SVM model gave nearly real MI value prediction, with much more accuracy than the other three models. Thus, we proved that the explored data-driven model held excellent universality in MI prediction both statistically and graphically. With the aid of an online adjustment mechanism, the proposed data-driven model accurately forecasted the testing and generalization data.

Ultimately, to further reveal the explored data-driven model's superiority, we compared it with several other soft-senor models for MI prediction reported in the open literature. In the first instance, nine different data-driven models were put into practice, and the PLS model won the best RMSE of 0.6082 (the original value was 2.72 by a different formula in the reference) on the basis of 40 testing data points.[17] In our proposed OCS-PSO-ICA-LS-SVM model, the RMSE was 0.0894, with an obviously huge percentage decrease. Next, the standard SVM model was recommended by Han et al.[20] from its compared PLS and ANN models. This was quantitatively supported by the smallest RMSE value of 1.51 on the testing dataset. However, when compared here with a value of 0.0894, it exceeds this value a lot as well. The last one, the AC–ICPSO–LS-SVM model, was proposed and verified to be superior to the LS-SVM model by Jiang et al.[17] We obtained an MAE of 0.0411 on the generalization dataset, which was the best predictive result so far reported in the open literature. However, our OCS-PSO-ICA-LS-SVM model achieved an MAE of 0.0294 on the same generalization dataset as Jiang et al.,[17] a decrease of 28.5%. In terms of MRE, STD, and TIC, similar tendencies were observed, and this further revealed the advantages of the proposed OCS-PSO-ICA-LS-SVM data-driven model.

CONCLUSIONS

MI is the dominant quality index for polypropylene. The effective monitoring of the MI and the realization of automatic quality control of the production process is an urgent problem in the plastics industry. In this study, an efficient data-driven model was developed for predicting MI through merging with feature extraction and model parameter optimization. With the help of an ICA algorithm, independent components were extracted from the selected original process variables and used as inputs of the successive LS-SVM model. So, in that sense, the feature extraction was regarded as an optimization problem for searching the model optimum inputs. Thanks to PSO the intelligent computational method, the configuration parameters of the LS-SVM model were optimized automatically. This could ensure a global convergence and high accuracy of the data-driven model. In addition, OCS was cultivated to guard the model real-time characteristics and good generalization ability. It could efficiently modify a trained LS-SVM model by means of incremental updating and decremental pruning algorithms whenever new data points are added to replace the pruned data from the training dataset. All of these theoretical conclusions were strongly supported by the practical MI prediction results in our research. Although its application focused on the

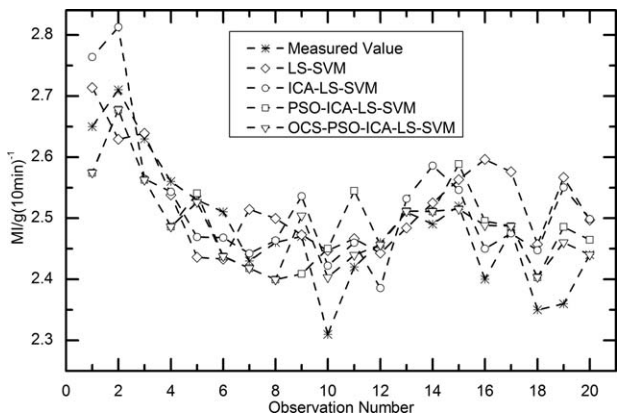

Figure 7. Prediction results of the MI for the testing dataset.

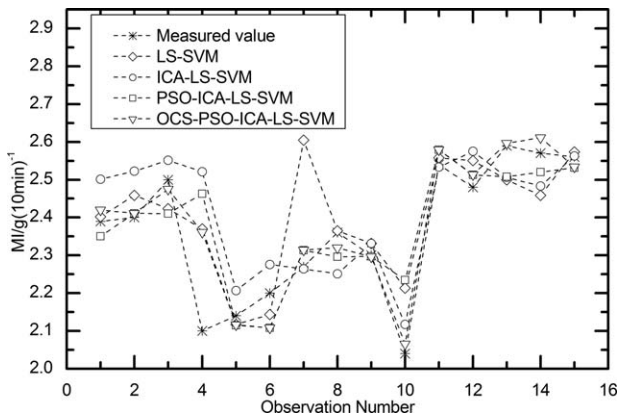

Figure 8. Prediction results of the MI for the generalization dataset.

propylene polymerization process, the proposed OCS-PSO-ICA-LS-SVM data-driven modeling approach was general and could be applied to similar industry processes in practice.

ACKNOWLEDGMENTS

This study was supported by the Joint Funds of NSFC-CNPC (National Natural Science Foundation of China and China National Petroleum Corporation) of China (contract grant number U1162130), the National High Technology Research and Development Program (863, contract grant number 2006AA05Z226), and the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scientists (contract grant number R4100133), and their support is hereby acknowledged.

REFERENCES

1. Bafna, S. S.; Beall, A. M. *J. Appl. Polym. Sci.* **1997**, *65*, 277.
2. Choi, D. J.; Park, H. *Water Res.* **2001**, *35*, 3959.
3. Luo, Z. H.; Su, P. L.; Shi, D. P.; Zheng, Z. W. *Chem. Eng. J.* **2009**, *149*, 370.
4. McAuley, K. B.; MacGregor, J. F. *AIChE J.* **1991**, *37*, 825.
5. McKenna, T. F.; Soares, J. B. P. *Chem. Eng. Sci.* **2001**, *56*, 3931.
6. Chen, Y.; Liu, X. *Polymer* **2005**, *46*, 9434.
7. Lee, E. H.; Kim, T. Y.; Yeo, Y. K. *Korean J. Chem. Eng.* **2008**, *25*, 613.
8. Luo, Z. H.; Su, P. L.; Wu, W. *Ind. Eng. Chem. Res.* **2010**, *49*, 11232.
9. Yan, W. C.; Luo, Z. H.; Liu, Y. H.; Chen, X. D. *AIChE J.* **2012**, *58*, 1717.
10. Castoldi, M. T.; Pinto, J. C.; Melo, P. A. *Ind. Eng. Chem. Res.* **2007**, *46*, 1259.
11. Azizi, H.; Ghasemi, I.; Karrabi, Q. *Polym. Test.* **2008**, *27*, 548.
12. Ghasemi, S. M.; Sadeghi, G. M. M. *J. Appl. Polym. Sci.* **2008**, *108*, 2988.
13. Nele, M.; Latado, A.; Pinto, J. C. *Macromol. Mater. Eng.* **2006**, *291*, 272.
14. Sharmin, R.; Sundararaj, U.; Shah, S. L.; Griend, L. V.; Sun, Y. *J. Chem. Eng. Sci.* **2006**, *61*, 6372.
15. Lou, H.; Su, H.; Xie, L.; Gu, Y.; Rong, G. *Ind. Eng. Chem. Res.* **2012**, *51*, 8510.
16. Zhang, Z.; Wang, T.; Liu, X. *Neurocomputing* **2014**, *131*, 368.
17. Jiang, H.; Yan, Z.; Liu, X. *Neurocomputing* **2013**, *119*, 469.
18. Ge, Z. Q.; Chen, T.; Song, Z. H. *Control Eng. Pract.* **2011**, *19*, 423.
19. Park, T. C.; Kim, T. Y.; Yeo, Y. K. *Korean J. Chem. Eng.* **2010**, *27*, 1662.
20. Han, I. S.; Han, C. H.; Chung, C. B. *J. Appl. Polym. Sci.* **2005**, *95*, 967.
21. Rallo, R.; Ferre-Giné, J.; Arenas, A.; Giralt, F. *Comput. Chem. Eng.* **2002**, *26*, 1735.
22. Liang, H.; Song, J. Y.; Wang, B. L. *Int. Conf. Nat. Comput.* **2007**, *1*, 610.
23. Suykens, J. A. K.; Vandewalle, J.; De Moor, B. *Neural Networks* **2001**, *14*, 23.
24. Chua, K. S. *Pattern Recognit. Lett.* **2003**, *24*, 75.
25. Hyvärinen, A.; Karhunen, J.; Oja, E. Independent Component Analysis; Wiley: Hoboken, NJ, **2001**.
26. Kennedy, J. *The IEEE International Conference on Evolutionary Computation;* **1997**; p 303. IEEE service center, Piscataway, NJ, Indianapolis.
27. Shi, J.; Liu, X. G. *Chin. J. Chem. Eng.* **2005**, *13*, 849.
28. Hyvärinen, A. *IEEE Trans. Neural Networks* **1999**, *10*, 626.
29. Chiu, S. H.; Lu, C. P.; Wu, D. C.; Wen, C. Y. *Pattern Recognit. Lett.* **2008**, *29*, 370.
30. Suykens, J. A. K.; Van Gestel, T.; De Brabanter, J.; De Moor, B.; Vandewalle, J. Least Squares Support Vector Machines; World Scientific: Singapore, **2002**.
31. Vapnik, V. N. The Nature of Statistical Learning Theory; Springer-Verlag: New York, **1995**.
32. Li, Q.; Racine, J. S. Nonparametric Econometrics: Theory and Practice; Princeton University Press: Princeton, NJ, **2007**.
33. Mitchell, M. An Introduction to Genetic Algorithms; Cambridge, MA: MIT Press, **1996**.
34. Reddy, M. J.; Kumar, D. N. *J. Hydroinform.* **2009**, *11*, 79.
35. Theil, H. Economic Forecasts and Policy, 2nd ed.; North Holland: Amsterdam, **1961**.
36. Kohavi, R. *Proc. Int. Joint Conf. Artif. Intell.* **1995**, *2*, 1137.
37. Rubio, G.; Pomares, H.; Rojas, I.; Herrera, L. J.; Guillén, A. In Artificial Neural Networks-(ICANN 2009), 19th International Conference, Limassol, Cyprus, September 2009, Part II; p 406, Springer-Verlag, Berlin Heidelberg, **2009**.
38. Shi, J.; Liu, X. *J. Appl. Polym. Sci.* **2006**, *101*, 285.