

Research Article

D-FNN Based Modeling and BP Neural Network Decoupling Control of PVC Stripping Process

Shu-zhi Gao,¹ Jing Yang,² and Jie-sheng Wang³

¹ College of Information and Engineering, Shenyang University of Chemical Technology, Shenyang 110142, China

² Multi-Functional Design and Research Academy, Zhengzhou University, Zhengzhou 450001, China

³ School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114044, China

Correspondence should be addressed to Jie-sheng Wang; wang_jiesheng@126.com

Received 23 December 2013; Accepted 18 March 2014; Published 15 April 2014

Academic Editor: Qintao Gan

Copyright © 2014 Shu-zhi Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

PVC stripping process is a kind of complicated industrial process with characteristics of highly nonlinear and time varying. Aiming at the problem of establishing the accurate mathematics model due to the multivariable coupling and big time delay, the dynamic fuzzy neural network (D-FNN) is adopted to establish the PVC stripping process model based on the actual process operation datum. Then, the PVC stripping process is decoupled by the distributed neural network decoupling module to obtain two single-input-single-output (SISO) subsystems (slurry flow to top tower temperature and steam flow to bottom tower temperature). Finally, the PID controller based on BP neural networks is used to control the decoupled PVC stripper system. Simulation results show the effectiveness of the proposed integrated intelligent control method.

1. Introduction

With VCM as a raw material, the suspension method to produce polyvinyl chloride (PVC) resin is a kind of typical intermittent chemical production process [1]. The traditional polymerization monomer conversion ratio is controlled to the scope 85%–90%. Unreacted vinyl chloride monomer is recovered by self-pressure, and 1%–2% of the monomers are still remained in PVC slurry. In the production, vinyl chloride monomer must be getting a further removal and recovery due to its some toxicity. The removal of residual vinyl chloride in polyvinyl chloride can improve the quality of polyvinyl chloride, reduce the cost of production, and solve the problem of environmental pollution caused in plastic processing and usage. At present, the stripping technique is very effective among all the removal methods of redundant VCM monomers in PVC slurry [2]. Now the research on the control methods in PVC industry mostly is in the stage of polymerization and distillation, where many advanced control techniques, such as neural network, fuzzy control, expert system, and predictive control methods, have been widely used [3–5]. Li [3] proposed a BP neural

network decoupling controller applied to the temperature control of rectifying column. The simulation results show that the algorithm has strong robustness. Xiao et al. [4] established steady-state optimization model of expanded bed energy consumption and designed feedback control system of multivariable state based on artificial neural network for the drying process of fluidized bed, effectively removing the decoupling association between the variables and achieving the dynamic optimization control of bed temperature. The actual operation results show that this system has strong robustness and achieve the minimum energy consumption control indicators of expanded bed. Khairiyah et al. [5] proposed the predictive control system based on neural network applied to the control of the rectifying tower, which makes the system have a smaller overshoot and shorter response time.

However, the research on the advanced control methods for the PVC stripping process is lack. The stripping process has characteristics of highly nonlinear, strong coupling, and time varying. It is difficult to establish a precise mathematical model and achieve good control effect by using traditional control methods.

Dynamic fuzzy neural network (DFNN) is a hybrid model of fuzzy theory and neural network method, whose function is equivalent to the TSK fuzzy system [6–10]. Based on the problem complexity and precision demand, D-FNN model can be constructed combining the system prior knowledge. An improved fuzzy neural network modeling method [11] was proposed to realize the soft sensor of molten index in the polymerization process, in which a hybrid learning method based on LM algorithm and a steepest descend method to train the network parameters exist. Simulation results show that the model has the merits of being not sensitive to initial values, fast convergence velocity, and high forecasting precision. The particle swarm algorithm (PSO) is adopted to optimize the parameters of the fuzzy neural network to set up a soft-sensor model of acrylonitrile yield [12]. Li and Lee [13] proposed an interval type-2 fuzzy system integrated with an observer-based hierarchical fuzzy neural controller, which can greatly reduce the number of adjusted parameters. Wu et al. proposed the maximum likelihood estimators used in nonparametric maximum likelihood fuzzy neural networks (MFNNs) for nonlinear regression problems and simple weight updating rules based on gradient descent and iteratively reweighted least squares (IRLS) will be derived [14]. Nasr and Chtourou [15] proposed a self-organizing map-based initialization for hybrid training based on a two-stage learning approach for feedforward neural networks. The weights between input and hidden layers are firstly adjusted by Kohonen algorithm with fuzzy neighborhood, whereas the weights connecting hidden and output layers are adjusted using gradient descent method. Hung-Ching proposed an adaptive self-constructing fuzzy neural network (ASCFNN) controller for a real inverted pendulum system, where the Mahalanobis distance (M-distance) method in the structure learning is also employed to determine if the fuzzy rules are generated/eliminated or not [16]. D-FNN is also successfully applied in many fields, such as function approximation, permanent-magnet synchronous motor drive control [17, 18].

In this paper, the D-FNN model of the PVC stripping process has been established firstly. Then the neural network decoupling modules and BP neural network PID controller are applied in the stripping temperature control. The effectiveness of the proposed intelligent control strategy is verified by the simulation experiments. The paper is organized as follows. In Section 2, the technique flowchart of PVC stripping process is introduced. The stripping process model based on dynamic fuzzy neural networks is presented in Section 3. In Section 4, the neural network decoupling controller of stripping process is summarized. In Section 5, simulation results are introduced in detail. Finally, the conclusion illustrates the last part.

2. Technique Flowchart of PVC Stripping Process

In the polymerization reaction process of vinyl chloride, according to the different techniques and resin grades, the polymerization conversion rate of vinyl chloride is generally controlled within the scope of (80%, 90%). Even the unreacted monomer is disposed by the autoclave recovery and

precipitated groove blowing, and there still remains small amount of vinyl chloride monomer in resin products, which will influence the products quality and increase the hazards when using the produced plastic. In addition, vinyl chloride emission into atmosphere per year not only causes a huge waste of raw materials, but also causes serious environment pollution. Currently, among all the approaches to the removal of excess VCM monomer in PVC slurry, the stripping technique is very effective.

Firstly, the stripping mechanism of PVC slurry is introduced in detail. In the PVC suspension to be stripping treated in the off chute of PVC polymerization, the mass fraction of PVC is about 25%–30%, the VCM is 2%–4%, and the rest is water. The mass approximation allocation ratio of VCM in gassed is about 1 : 1000 : 100. Because of the different amount distribution of VCM in the three phases, the slurry itself has the characteristic of the diffused concentration gradient. In PVC slurry, the VCM in the solid PVC either by parsing or by spreading must pass through the film layer to a low concentration of VCM in water for spreading through the pores in PVC particles. Moreover, the solubility of VCM in water changes within the variation of temperature. In different temperature, when the VCM in water reaches to solubility, the part which diffused from PVC into water will break the static pressure of the liquid water layer and diffuse into the gas phase. With the above mechanism, the heat exchange makes the slurry's temperature increase so that the VCM will be parsed.

There are two main tower stripping devices. What we research in this paper is the sieve wear weir plate stripper. The stripping technique of the PVC slurry is shown in Figure 1.

After the polymerization reaction, the PVC slurry is threw into the top of stripper from the polymerization vessel through the inlet slurry pump. But before entering the stripper, they need to carry through the heat exchange with the slurry stripped from the bottom of stripper and then go into the top of stripper. PVC slurry sprinkles evenly on the tray by sprinklers and flow from the top to bottom along the strays from the first layer stray. The low pressure steam enters from the stripper bottom, rises through the orifice of the tray from the bottom to top, and countercurrent contacts with the declining PVC slurry at each plate for heat and mass transfer. Finally, it will resolve internal vinyl chloride and the residue in the surface of the resin particles. Steam and parsed vinyl chloride gas are discharged from the top of the stripper. After the condensation by overhead condenser, it enters into the condensate sink, then separates condensate, and controls the level of condensate sink. The excess condensate will be discharged to the centrifuge mother liquor water recovery unit, and the separate vinyl chloride gas will be recovered to vinyl chloride gas cabinets. The key parameters influencing the PVC stripping process can mainly include temperature, steam, slurry flow rate, and pressure. Table 1 gives the parameters values in the PVC stripper controller of a certain type resin.

Through the analysis of PVC stripping process, it has the characteristics of strong nonlinear and coupling. Because of lack of effective mathematical model and strong coupling between the variables, it is difficult to achieve satisfactory

TABLE 1: Parameters of PVC string tower.

Parameter	Scope	Alarm value	Normal scope	Real value
Temperature in tower top	0–150°C	Upper limit: 110°C Lower limit: 70°C	80–105°C	85°C
Temperature in tower bottom	0–150°C	Upper limit: 130°C Lower limit: 100°C	105–113°C	107°C
Pressure in tower top	0–0.1 MPa	Upper limit: 0.08 MPa Lower limit: 0.02 MPa	0.03–0.06 MPa	0.03 MPa
Pressure difference in tower	0–60 MPa	Upper limit: 30 MPa Lower limit: 5 MPa	10–25 MPa	15 MPa
Steam flow rate	0–4000 kg/h	Upper limit: 3500 kg/h Lower limit: 1000 kg/h	1500–3000 kg/h	2200 kg/h
Inlet slurry flow rate	0–80 m ³ /h	Upper limit: 70 m ³ /h Lower limit: 15 m ³ /h	35–50 m ³ /h	40–45 m ³ /h
Steam pressure in tower	0–0.6 MPa	Upper limit: 0.5 MPa Lower limit: 0.15 MPa	0.35–0.4 MPa	0.38 MPa

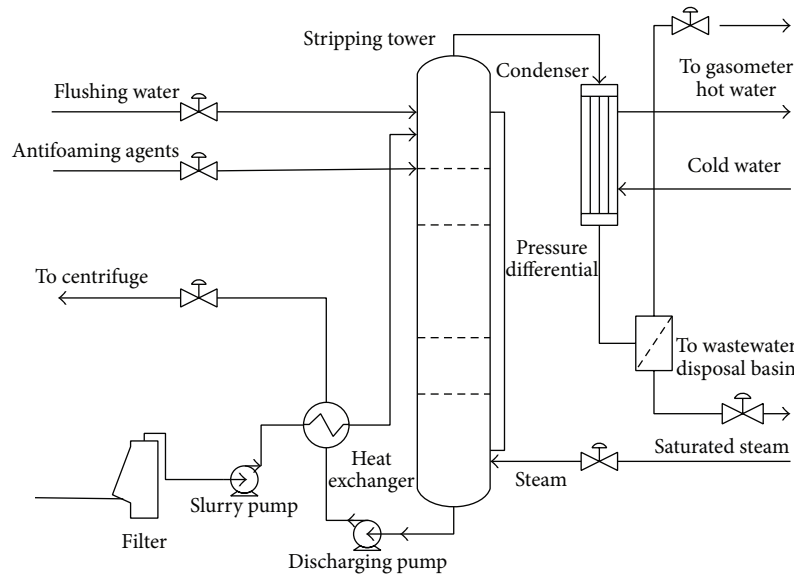


FIGURE 1: Flow diagram of PVC stripping process.

control effect. The integrated model and hybrid intelligent control strategies based on neural network technology are put forward for the PVC stripping process in this paper.

The data used for modeling all derived from the history data recorded in the operating process of PVC stripping tower. Each data sample includes the following elements: top tower temperature, bottom tower temperature, top tower pressure, tower differential pressure, slurry flow rate, and steam flow rate. According to the analysis of stripping technique, the key factors which influence vinyl chloride removed from PVC are the top tower temperature and the bottom tower temperature. So there are two mainly important control factors, which are the steam flow rate and the slurry flow rate during the process regulation. Therefore, the D-FNN model is built in which the steam flow rate and slurry flow rate are the input variables, and the top tower temperature and the bottom tower temperature are the output variables.

3. Stripping Process Modelling Based on Dynamic Fuzzy Neural Networks

According to the analysis of the stripping process, the changeable slurry flow or the steam flow rate not only influences the top tower temperature but also the bottom tower temperature. Consequently, the coupling exists in the PVC stripping process. The coupling object model needs to be built and the stripping process needs to be decoupled. Suppose the dynamic system model is described as the following nonlinear equation:

$$y(k) = f[y(k-1), \dots, y(k-n_a), u_1(k), \dots, u_1(k-n_b), u_2(k), \dots, u_2(k-n_b)], \quad (1)$$

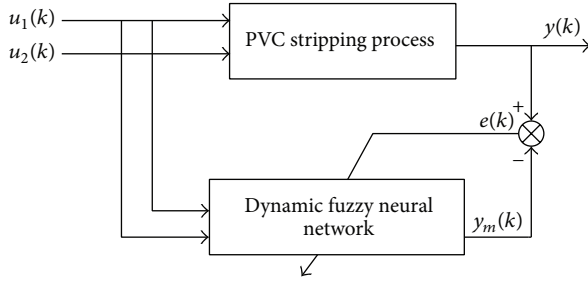


FIGURE 2: Modeling of stripping process based on D-FNN.

where y is the top tower temperature or bottom tower temperature, u_1 is the steam flow rate, u_2 is the slurry flow rate, and k is the sample time (20 s).

The D-FNN control strategy of the PVC stripping process is put forward shown in Figure 2, whose inputs variables are the steam flow rate and the slurry flow rate and output variables are the top tower temperature and the bottom tower temperature.

3.1. Structure of D-FNN. The structure diagram of D-FNN [19–23] is shown in Figure 3. In Figure 3, x_1, x_2, \dots, x_r is the input linguistic variables, y is the system output, MF_{ij} represents the j th membership function of the i th input variable, R_j is the j th fuzzy rule, N_j is the j th normalized node, w_j is the connection weight of the j th rule, and u refers to the numbers of the system rules. Then each layer of the D-FNN is described in detail.

Layer 1 is the input layer, whose each node represents an input linguistic variable.

Layer 2 is the membership function layer, whose each node represents a membership function (MF) defined in the followed Gaussian function:

$$\mu_{ij}(x_i) = \exp \left[-\frac{(x_i - c_{ij})^2}{\sigma_j^2} \right], \quad (2)$$

where $i = 1, 2, \dots, r$, $j = 1, 2, \dots, u$, μ_{ij} is the j th membership function, c_{ij} is the center of the j th Gaussian membership function, σ_j is the width of the j th Gaussian membership function of x_i , and r is the number of input variables.

Layer 3 is the T-norm layer, whose each node represents a possible IF-part of fuzzy rules. For the j th rule R_j , its output is defined as follows:

$$\phi_j = \exp \left[-\frac{\sum_{i=1}^r (x_i - c_{ij})^2}{\sigma_j^2} \right] = \exp \left[-\frac{\|X - C_j\|^2}{\sigma_j^2} \right], \quad (3)$$

where $j = 1, \dots, u$, $X = (x_1, \dots, x_r)$ and $C_j = (c_{1j}, \dots, c_{rj})$ is the center of the j th RBF unit.

Layer 4 is the normalization layer, where the output of the N_j node is defined as

$$\psi_j = \frac{\phi_j}{\sum_{k=1}^u \phi_k} \quad j = 1, 2, \dots, u. \quad (4)$$

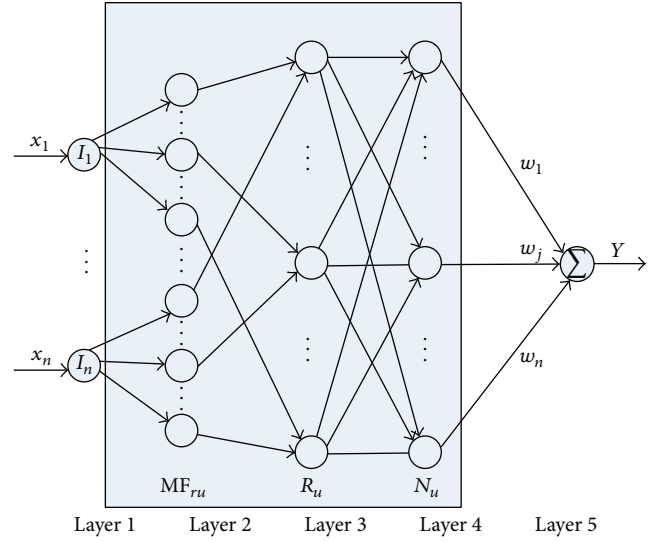


FIGURE 3: Structure diagram of D-FNN.

Layer 5 is the output layer, whose each node is an output variable, a summation of input signals described as follows:

$$y(K) = \sum_{k=1}^u w_k \cdot \psi_k, \quad (5)$$

where y is the output variable and w_k is the weight of the k th rule.

For the TSK model, it can be represented as follows:

$$w_k = a_{k0} + a_{k1}x_1 + \dots + a_{kr}x_r \quad k = 1, 2, \dots, u. \quad (6)$$

Then (2)–(5) are fed into (5) to obtain

$$y(X) = \sum_{i=1}^u \left[(a_{i0} + a_{i1}x_1 + \dots + a_{ir}x_r) \times \exp \left(-\frac{\|X - C_i\|^2}{\sigma_i^2} \right) \right] \times \left(\sum_{i=1}^u \exp \left(-\frac{\|X - C_i\|^2}{\sigma_i^2} \right) \right)^{-1}. \quad (7)$$

3.2. Learning Algorithm of D-FNN. Each node in layer 3 of the D-FNN represents an RBF unit or the IF-part for fuzzy rules. If the number of fuzzy rules is required to be identified, the D-FNN structure is not predefined. So a new learning algorithm for the D-FNN is adopted to automatically determine the fuzzy rules satisfying the required system performances.

3.2.1. Rule Generation Criteria. If there are only few rules, the system cannot fully represent the input-output state space. But if there are too many rules, the system can be unnecessarily complicated, and the generic ability of D-FNN can become worse because of the greatly increased calculation burden. So the system errors are the principal elements to determine whether the new rules can be added.

The error criterion can be described as follows. For the i th observation point (X_i, t_i) , where X_i is the input vector and t_i is the desired output, compute the overall D-FNN output y_i according to (7). Define the error as follows:

$$\|e_i\| = \|t_i - y_i\|. \quad (8)$$

If the following condition is satisfied, a new rule should be added. In (9), k_e is determined according to the desired accuracy of the D-FNN:

$$\|e_i\| > k_e. \quad (9)$$

3.2.2. Accommodation Boundary. Gaussian function has good locality characteristic because its output is monotonic decreasing along with the increasing distance between the output and the centers. When the membership function of input variables is described by Gaussian function, the whole input space is divided by a series of Gaussian function. If a new sample is located in some Gaussian function, that is to say within accommodation boundary, the new sample can be represented by the existed Gaussian function, without new Gaussian unit produced by D-FNN.

The accommodation boundary criterion is described as follows. For the i th observation point (X_i, t_i) , calculate the distance $d_i(j)$ between the observation X_i and the center C_j of the existed RBF units by

$$d_i(j) = \|X_i - C_j\|, \quad j = 1, 2, \dots, u. \quad (10)$$

Find the minimum distance by (11). If $d_{\min} > k_d$, a new fuzzy rule should be added. Otherwise, the observation X_i can be represented by the nearest existed RBF unit. Here, k_d is the effective radius of the accommodation boundary

$$d_{\min} = \arg \min (d_i(j)). \quad (11)$$

3.2.3. Hierarchical Learning Strategy. The concept of hierarchical learning is that the accommodation boundary of each RBF unit is not fixed but adjusted dynamically in the following style. Firstly, the accommodation boundaries are set larger for achieving rough but global learning. Then, they are gradually reduced for fine adjustment. Inspired by this idea, a simple method based on monotonically decreasing function is proposed to reduce both the effective radius of each RBF unit and error index gradually [22]. To be more specific, the discussed above k_e and k_d are not constant. They are changeable according to the following equations:

$$\begin{aligned} k_e &= \max [e_{\max} \times \beta^i, e_{\min}], \\ k_d &= \max [d_{\max} \times \gamma^i, d_{\min}], \end{aligned} \quad (12)$$

where e_{\max} is the predefined maximum error, e_{\min} is the desired D-FNN accuracy, β ($0 < \beta < 1$) is the convergence constant, γ ($0 < \gamma < 1$) is the decay constant, and d_{\max} and d_{\min} are the largest and the smallest length of the input space, respectively.

The key idea of the hierarchical learning is to firstly find the most troublesome positions, which have large errors

between the desired and the actual outputs but are not properly covered by existing fuzzy rules. This stage is called the coarse learning. When k_e and k_d reach e_{\min} and d_{\min} , respectively, this stage is named the fine learning.

3.2.4. Allocation of Precedent Parameters. After a fuzzy rule is obtained, the problem is how to determine its parameters. The width of the RBF unit has great significant on its generalization. If it is less than the distance between adjacent inputs, the RBF unit does not generalize well and D-FNN will not give meaningful outputs in response to inputs. However, if the width is too large, the output of the RBF nodes may always be large irrespective of inputs. Hence, the initial parameters of the new produced rule are allocated based on the following equations:

$$\begin{aligned} C_i &= X_i, \\ \sigma_i &= k \times d_{\min}, \end{aligned} \quad (13)$$

where k ($k > 1$) is an overlap factor.

What we needed to pay attention to is that when the first pattern (X_1, t_1) enters the D-FNN, it is selected as the first fuzzy rule: $C_1 = X_1$, $\sigma_1 = \sigma_0$, where σ_0 is a preset constant. It is worth highlighted that only when $\|e_i\| > k_e$, $d_{\min} > k_d$, one fuzzy rule needs to be added.

3.2.5. Determination of Decision Parameters. According to the criteria of rule generation, suppose u fuzzy rules are generated based on the n observation datum. So the output of N nodes can be obtained according to (4), which can be represented as follows:

$$\Psi = \begin{bmatrix} \psi_{11} & \cdots & \psi_{1n} \\ \vdots & & \vdots \\ \psi_{u1} & \cdots & \psi_{un} \end{bmatrix}. \quad (14)$$

Then, for any input X_j ($x_{1j}, x_{2j}, \dots, x_{rj}$), the system output y_j can be calculated by (5), which can be described in more compact form:

$$W\Psi = Y. \quad (15)$$

Suppose the ideal output is $T = (t_1, t_2, \dots, t_n) \in \mathfrak{R}^n$. The relation between $\Psi \in \mathfrak{R}^{(r+1)u \times n}$ and $T \in \mathfrak{R}^n$ is represented as follows:

$$\begin{aligned} Y &= W \times \Psi, \\ \tilde{E} &= \|T - Y\|. \end{aligned} \quad (16)$$

Find an optimal coefficient vector based on

$$W^* \times \Psi = T. \quad (17)$$

The optimal W^* is in the following form:

$$W^* = T(\Psi^T\Psi)^{-1}\Psi^T, \quad (18)$$

where Ψ^T is the transposition of Ψ and $\Psi^+ = (\Psi^T\Psi)^{-1}\Psi^T$ is the pseudoinverse of Ψ .

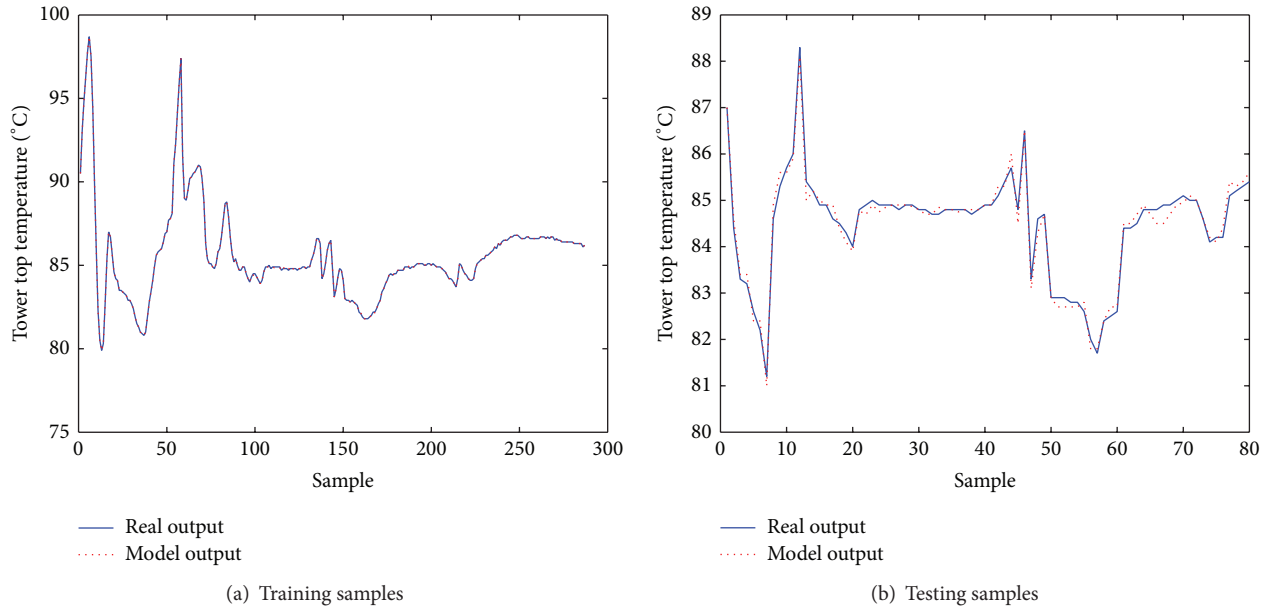


FIGURE 4: Simulation results of D-FNN modeling of tower top temperature.

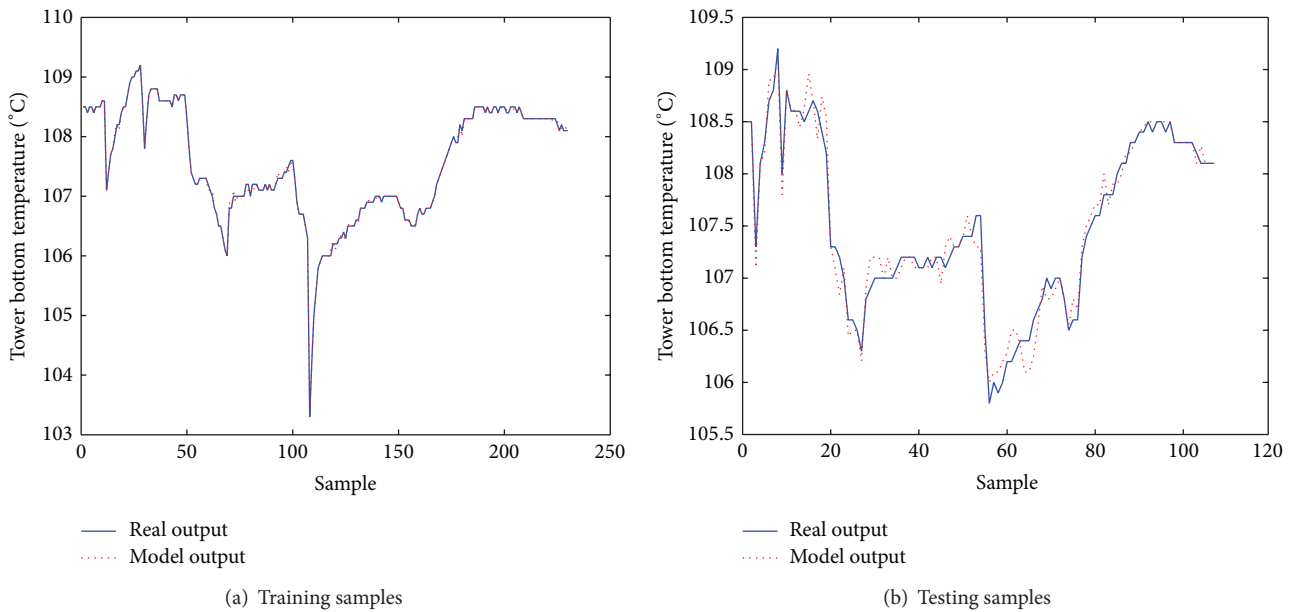


FIGURE 5: Simulation results of D-FNN modeling of tower bottom temperature.

3.3. Simulation Results. The samples are derived from the actual operation datum of the certain chemical corporation, in which 250 groups of input-output data have been selected to identify the system randomly. The other 80 groups of input-output data are used to test the identification effect. Then the D-FNN models are obtained, in which the steam flow rate and slurry flow rate are inputs variables, and the top tower temperature and the bottom tower temperature are output variables separately. The identification effect and test results are shown in Figures 4 and 5. It can be seen from Figures 4 and 5 that the output curves of the D-FNN model and the

actual curves of the tower temperature match very well, and the errors are within ± 0.1 , which can meet the requirement of the process optimization control.

4. Neural Network Decoupling Controller of Stripping Process

4.1. Structure of Intelligent Decoupling Controller. The intelligent decoupling control strategy based on neural network is put forward aiming at the above-established PVC stripping process model. The system diagram is shown in Figure 6.

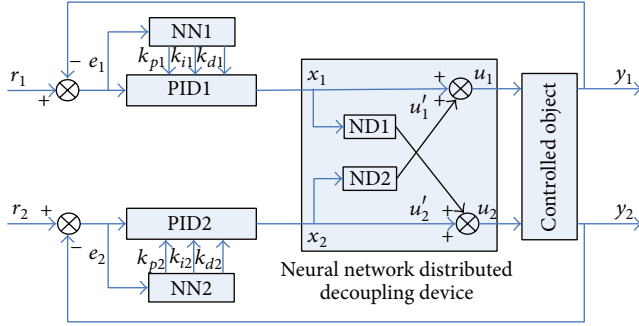


FIGURE 6: Decoupling controller structure of PVC stripping process.

The control system consists of PVC stripper object (slurry flow rate to top tower temperature and steam flow rate to tower bottom temperature), two neural network open-loop decoupling modules ND_1 and ND_2 , and two PID controllers online optimized by two BP neural networks NN_1 and NN_2 .

In Figure 6, r_i is the system input, y_i is the system output, x_i is the controller output vector, u_i is the output vector of the neural network decoupling module, and u'_i is the control input vector ($i = 1, 2$). The intelligent decoupling control system of the stripping process can be divided into two stages.

4.1.1. Training Stage. The neural network PID controller is disconnected and the system is in open-loop training state. Firstly, the neural decoupling module ND_1 is trained and ND_2 is maintained as no input. If decoupling is successful, the channel between the slurry flow rate and the tower top temperature will not have output (y_1 is 0). Then y_1 is set as the training index function of ND_1 . The connection weights of ND_1 's each layer are adjusted. Similarly, ND_2 is trained in the same way. Finally, the whole stripper controlled object is decoupled into two SISO systems (slurry flow rate, the tower top temperature, and the steam flow rate, the tower bottom temperature).

4.1.2. Control Stage. NN_1 and NN_2 are used to online optimize the PID controller parameters for realizing the closed-loop control for the two SISO systems. After training of ND_1 and ND_2 , the weights are maintained unchanged. At this time, the entire stripper controlled object is decoupled into two SISO systems. Then two BPNN PID controllers (NN_1 and NN_2) are, respectively, adopted to realize the closed-loop control of two SISO systems. NN_1 and NN_2 will be used to online adjust the PID controller's parameters K_p , K_i , and K_d .

4.2. Structure of Neural Network Decoupling Module. The neural network decoupling module consists of $n \times (n - 1)$ SISO neural network (n is the number of the object inputs), whose structure is shown in Figure 7. Each neural network decoupling module is single-input-single-output (SISO) subsystem, which is responsible for only one channel decoupling. So its structure and indicator function are relatively simple because the total objective function has been decomposed into single objective functions and the decoupling modules'

interconnection problem needs not be considered. At the same time, the dispersion of the objective function is also easy to achieve parallel distributed processing in real time. In addition, the decoupling structure does not change the dynamic characteristics of the controlled process main channel. It will make the design problem easy in engineering. As long as the reasonable structure and parameters of controller c_i are chosen, the required dynamic characteristics can be obtained.

The PVC stripping process is double input and double output coupling system. The structure of the neural decoupling module is shown in Figure 7.

The decoupling module is composed of two different neural networks ND_1 and ND_2 , which are dynamic time delay TDNN networks. Each network has an input terminal and an output terminal. The decoupling module outputs can be expressed as

$$\begin{aligned} u'_2(k) &= f_1 [x_1(k), x_1(k-1), \wedge, x_1(k-n_1), u'_2(k-1), \\ &\quad u'_2(k-2), \wedge, u'_2(k-m_2)], \\ u'_1(k) &= f_2 [x_2(k), x_2(k-1), \wedge, x_2(k-n_2), u'_1(k-1), \\ &\quad u'_1(k-2), \wedge, u'_1(k-m_1)], \end{aligned} \quad (19)$$

where u is the output of neural network decoupling module, k is sampling frequency, and m_1 , m_2 , n_1 , and n_2 are the positive integers, which are determined according to the controlled process' order.

The decoupling module in this paper adopts a three-layer neural network. The neural network's excitation function is the hyperbolic tangent function $f(x) = (1 - e^{-x}) / (1 + e^{-x})$. Its derivative is $f'(x) = 1 / (2 \times (1 - f^2(x)))$. $u_1(k)$ and $u_2(k)$ are the outputs of the decoupling module. After superimposing the controller outputs x_1 and x_2 , the system control output is obtained by $u_1 = u'_1 + x_1$ and $u_2 = u'_2 + x_2$.

4.3. Training Algorithm of Neural Network Decoupling Module. For the neural decoupling module shown in Figure 8, theoretically speaking, the expected output of the neural decoupling module should be viewed as the target function. But the NN expected output is unknown. In order to avoid this contradiction, the neural decoupling module and the controlled object are viewed as a generalized object. Thus the neural network training can adopt the system output as the performance function. For neural decoupling modules ND_1 and ND_2 , the performance index functions are defined as follows:

$$\begin{aligned} J_1 &= \frac{1}{2} \sum_{K=0}^N [y_2(k)]^2, \\ J_2 &= \frac{1}{2} \sum_{K=0}^N [y_1(k)]^2. \end{aligned} \quad (20)$$

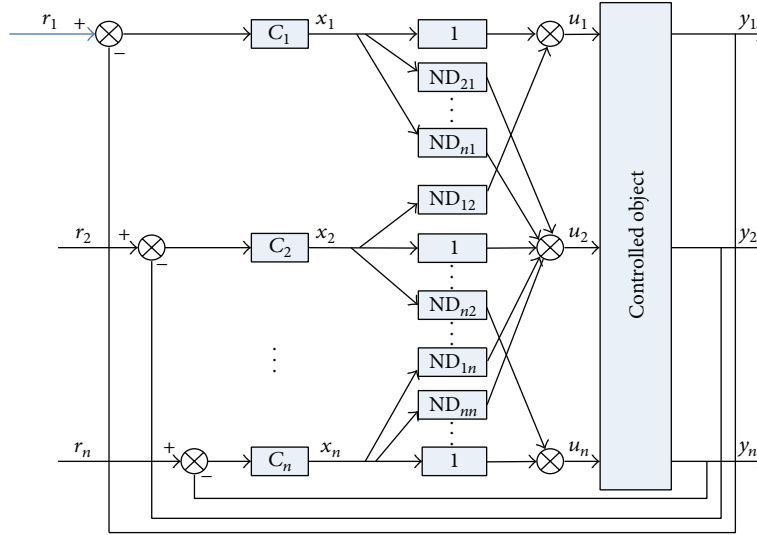


FIGURE 7: Structure diagram of decoupling module.

According to the BP learning algorithm [24, 25], the network weights are updated based on the following equation:

$$\Delta w(k) = -\eta \frac{\partial J}{\partial w} + \alpha * \Delta w(k-1), \quad (21)$$

where η is the learning rate, α is the momentum coefficient, w is the weights of the corresponding neural network, and N is the number for training samples.

For ND_1 and ND_2 , differentiate the performance index function to obtain

$$\Delta w(k) = -\eta \sum_{y_2}^N \frac{\partial y_2}{\partial u_2'} \frac{\partial u_2'}{\partial w} + \alpha * \Delta w(k-1), \quad (22)$$

$$\Delta w(k) = -\eta \sum_{y_1}^N \frac{\partial y_1}{\partial u_1'} \frac{\partial u_1'}{\partial w} + \alpha * \Delta w(k-1).$$

For the convenience of discussion, the subscripts are omitted to get the universal formula:

$$\Delta w(k) = -\eta \sum_{y}^N \frac{\partial y}{\partial u'} \frac{\partial u'}{\partial w} + \alpha * \Delta w(k-1), \quad (23)$$

where Δw is the change of weight.

In order to adopt BP learning algorithm, the partial derivative $\partial y / \partial u$ needs to be calculated. However, the learning rate is an adjustable parameter. The partial derivatives multiply the learning rate's product deciding the change of weights. Therefore, the absolute value of the partial derivatives is not very important. If the partial derivative's symbols are known, it is recorded as $\text{sign}(\partial y / \partial u)$. Adjust the learning rate to obtain

$$\eta' = \eta \left| \frac{\partial y}{\partial u'} \right|. \quad (24)$$

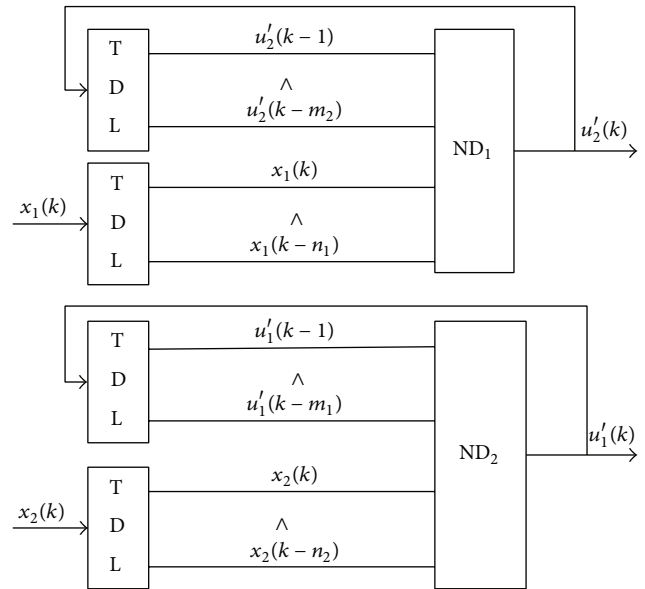


FIGURE 8: Schematic diagram of distributed decoupling.

Therefore, (23) can be written as

$$\Delta w(k) = -\eta' \text{sign} \left(\frac{\partial y}{\partial u'} \right) \sum_{y}^N y \frac{\partial u'}{\partial w} + \alpha * \Delta w(k-1). \quad (25)$$

In the actual industrial production, there are many controlled processes which have monotone increasing or decreasing characteristics. That is to say, y is the monotonically increasing or decreasing function of u' . When y is the monotone increasing of u' , $\text{sign}(\partial y / \partial u) = 1$, or it is -1 .

The relationship between network input and output can be expressed as

$$u' = w^{(2)} u^{(1)}, \quad (26)$$

$$u^{(1)} = f(w^{(1)} u^{(0)}),$$

$$u^{(0)} = [x(k), x(k-1), \wedge, x(k-n), u'(k-1), u'(k-2), \wedge, u'(k-m)]^T, \quad (27)$$

where u' is the output signal of the network output layer, $u^{(1)}$ is the output signal of the network middle layer, $u^{(1)} = [u_1^{(1)}, u_2^{(1)}, u_3^{(1)}, \wedge]$ and $u^{(0)}$ is the input signal of the network input layer, and $w^{(2)} = [w_0^{(2)}, w_1^{(2)}, w_2^{(2)}, \dots]$ is the connection weight vector between network middle layer and output layer.

$w^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \dots \\ w_{21}^{(1)} & w_{22}^{(1)} & \dots \\ \dots & \dots & \dots \end{bmatrix}$ is the connection weight vector between network input layer and middle layer.

For the output layer,

$$\frac{\partial u'}{\partial w_i^{(2)}} = \frac{1}{2} (1 - u'^2) u_i^{(1)}. \quad (28)$$

For the middle layer,

$$\frac{\partial u'}{\partial w_{ji}^{(1)}} = \frac{\partial u'}{\partial u_j^{(1)}} \frac{\partial u_j^{(1)}}{\partial w_{ji}^{(1)}} = \frac{1}{2} (1 - u'^2) w_j^{(2)} \frac{1}{2} (1 - u_j^{(1)2}) u_i^{(0)}. \quad (29)$$

Put (28) and (29) into (23); the weight of the output layer and middle layer is updated by the following equations.

For the output layer,

$$\Delta w_i(k) = -\frac{\eta'}{2} \text{sign} \left(\frac{\partial y}{\partial u'} \right) \sum y (1 - u'^2) u_i^{(1)} + \alpha * \Delta w_i(k-1). \quad (30)$$

For the middle layer,

$$\Delta w_{ji}(k) = -\frac{\eta}{4} \text{sign} \left(\frac{\partial y}{\partial u'} \right) \sum y (1 - u'^2) w_j^{(2)} (1 - u_j^{(1)2}) u_i^{(0)} + \alpha * \Delta w_{ji}(k-1). \quad (31)$$

The algorithm procedure is described as follows.

- (1) Connect the neural network decoupling module ND₁ and ND₂ with the controlled process.
- (2) Initialize the neural decoupling module weights with the smaller random values.
- (3) Give the input signals and make the controller output signal correspond to the input signal of the neural decoupling module.

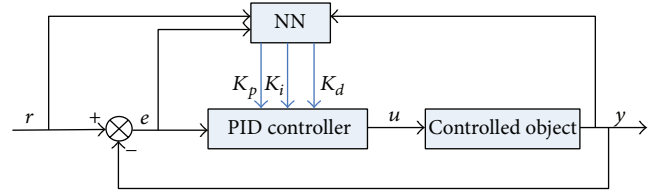


FIGURE 9: PID controller based on BP neural network.

- (4) According to the selected samples and time interval, adopt (26) to calculate the output of the neural network u' , $u^{(1)}$, $u^{(0)}$, and N values of the process variables changed with time (set $u' = 0$ at the begin of calculation).
- (5) According to (30) and (31), calculate the weight variances of the output layer and middle layer and update weights $w^{(1)}$ and $w^{(2)}$.
- (6) Replete steps (4) and (5) until reaching the specified iterations to determine $w^{(2)}$ and $w^{(1)}$ at the end.

The above process does not involve each variable's subscript. In practice, the training of network weights can be completed individually according to the above procedure. When ND₁, ND₂, ..., ND_n are determined, the whole system can be viewed as n single variable nonlinear systems. Then each loop can be designed the corresponded controller.

4.4. PID Controller Based on BP Neural Network. The structure of PID controller based on BP neural network is shown in Figure 9. According to the system's operation states, the PID controller parameters are adjusted to achieve the optimization performance. The neural units in the output layer are in correspondence with three PID controller adjustable parameters k_p , k_i , and k_d . Through self-learning of neural network, the weighted coefficients are adjusted to make the neural network output corresponding to the PID controller parameters with a certain optimal control law.

Three layers BP neural network [26] is used for the PVC stripping process. The input layer have 3 neurons: input r , output y , and error $error$:

$$O_i = x_{il}(i), \quad i = 1, 2, 3, \quad (32)$$

where the subscripts i , j , and l , respectively, represent the input layer, the hidden layer, and the output layer.

The hidden layer has 5 neurons. The input of the hidden layer is calculated by

$$net_j(k) = \sum_{i=1}^3 w_{ji} O_i. \quad (33)$$

The output of the hidden layer is calculated by

$$O_j(k) = f(net_j(k)) \quad j = 1, 2, \dots, 5. \quad (34)$$

The activation function of the hidden layer neuron adopts the symmetrical sigmoid function $f(x) = \tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$.

The output layer has 3 neurons. The input of the output layer is calculated by

$$net_l(k) = \sum_{j=1}^5 w_{lj} O_j(k). \quad (35)$$

The output of the output layer is calculated by

$$O_l(k) = g(net_l(k)), \quad l = 1, 2, 3. \quad (36)$$

The output layer nodes are in accordance with three adjustable parameters k_p , k_i , and k_d . Since k_p , k_i , and k_d cannot be negative, the output layer neuron's activation function is not negative sigmoid function: $g(x) = 1/(2 \times (1 + \tanh(x))) = e^x/(e^x + e^{-x})$.

The performance index function is selected as $E(k) = 1/(2 \times (r(k) - y(k))^2)$. The network weights w_{ji} and w_{lj} are revised according to the gradient descent method. That is to say, according to $E(k)$, the weighted coefficients are searched and adjusted in the negative gradient direction with an inertia which make search rapidly converge to global minima.

From the output layer to the hidden layer,

$$\Delta w_{ij}(k) = -\eta \frac{\partial E(k)}{\partial w_{ij}} + \alpha \Delta w_{ij}(k-1), \quad (37)$$

where η is the learning rate and α is the inertial coefficient:

$$\begin{aligned} \frac{\partial E(k)}{\partial w_{ij}(k)} &= \frac{\partial E(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial O_l(k)} \cdot \frac{\partial O_l(k)}{\partial net_l(k)} \cdot \frac{\partial net_l(k)}{\partial w_{ij}(k)}, \\ \frac{\partial net_l(k)}{\partial w_{ij}(k)} &= O_j(k). \end{aligned} \quad (38)$$

Because $\partial y(k)/\partial u(k)$ is unknown, it is approximated by the symbol function $\text{sign}(\partial y(k)/\partial u(k))$. The problem of calculation accuracy can be solved by adjusting the learning rate η to compensate.

For the output layer,

$$\begin{aligned} \frac{\partial u(k)}{\partial O_1(k)} &= error(k) - error(k-1), \\ \frac{\partial u(k)}{\partial O_2(k)} &= error(k), \\ \frac{\partial u(k)}{\partial O_3(k)} &= error(k) - 2error(k-1) + error(k-2). \end{aligned} \quad (39)$$

Thus the learning algorithm of the output layer weights is described as

$$\begin{aligned} \Delta w_{lj}(k) &= \alpha \Delta w_{lj}(k-1) + \eta \delta_l O_j(k), \\ \delta_l &= error(k) \text{sign}\left(\frac{\partial y(k)}{\partial u(k)}\right) \frac{\partial u(k)}{\partial O_l(k)} g'(net_l(k)) \\ & \quad l = 1, 2, 3, \\ g'(x) &= g(x)(1 - g(x)). \end{aligned} \quad (40)$$

Similarly, the learning algorithm of the hidden layer weights is described as

$$\begin{aligned} \Delta w_{ji}(k) &= \alpha \Delta w_{ji}(k-1) + \eta \delta_j O_i(k), \\ \delta_j &= f'(net_j(k)) \sum_{l=1}^3 \delta_l w_{lj}(k) \quad j = 1, 2, \dots, 5, \\ f'(x) &= \frac{(1 - f^2(x))}{2}. \end{aligned} \quad (41)$$

The algorithm procedure is described as follows.

- (1) Determine the structure of BP neural network. That is to say, determine the number of the input layer nodes and the hidden layer nodes. Given all initial weighting coefficients, select the learning rate η and the inertial coefficient α .
- (2) Sample and obtain $r(k)$ and $y(k)$. Then calculate $error(k) = r(k) - y(k)$.
- (3) Calculate each layer's input and output of the neural network. The output in BPNN is the three adjustable parameters of the PID controller.
- (4) Calculate the PID output according to the incremental algorithm.
- (5) Train BP neural network to online adjust the weights for realizing the self-tuning of the PID controller parameters.
- (6) Set $k = k + 1$ and repeat from step (2) to step (6).

5. Simulation Experiments

In order to verify the effectiveness of the D-FNN model and the proposed intelligent decoupling control strategy based on neural network, the D-FNN model is used in neural network decoupling control strategy. Based on the tower top temperature D-FNN model and tower bottom temperature D-FNN model, the neural network decoupling modules ND₁ and ND₂ and neural network controller NN₁ and NN₂ are designed. The structure and the parameters of all neural networks are set as follows. The structure of ND₁ and ND₂ is 5-4-1, the momentum factor $\eta_{ND} = 0.005$, and the learning rate $\alpha_{ND} = 0.9$. The structure of NN₁ and NN₂ is 3-5-3, the inertia $\eta_{NN} = 0.25$, and learning rate $\alpha_{NN} = 0.05$.

According to the technique of the PVC stripping process, the optimum of the stripper tower top temperature is 85°C. Thus in our simulation, the tower top temperature is 85°C. The temperature should not exceed 110°C, otherwise it will make the PVC resin decomposed and metamorphic. Therefore the upper limit is 110°C. The optimum of the stripper tower bottom temperature is 107°C and its upper limit is 113°C. According to the actual possible working conditions

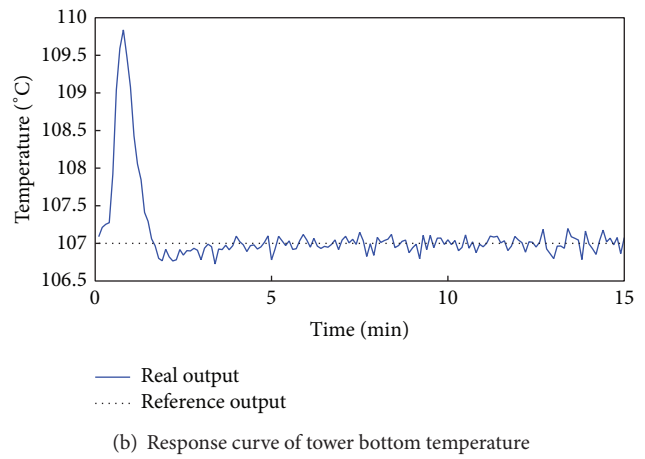
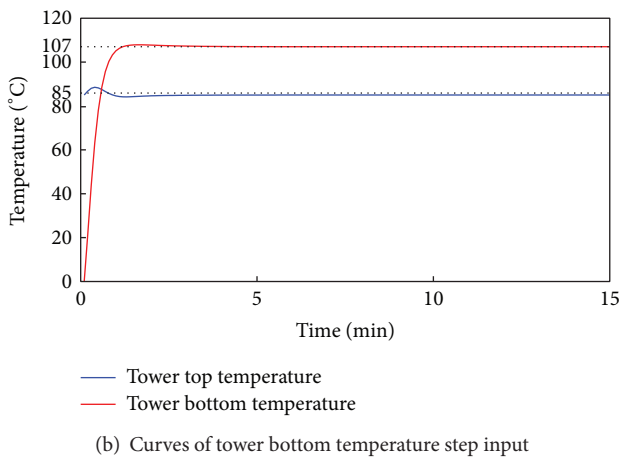
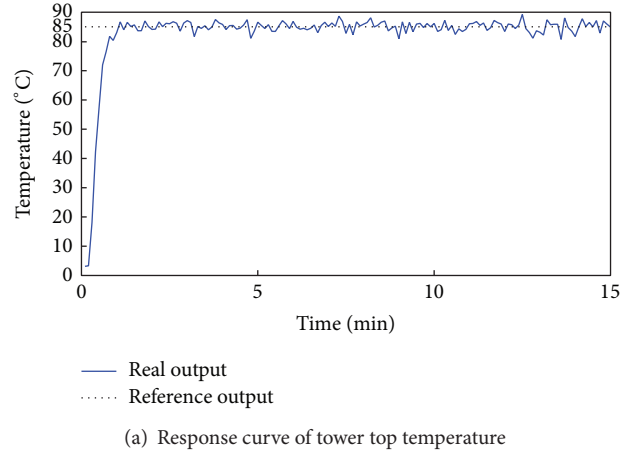
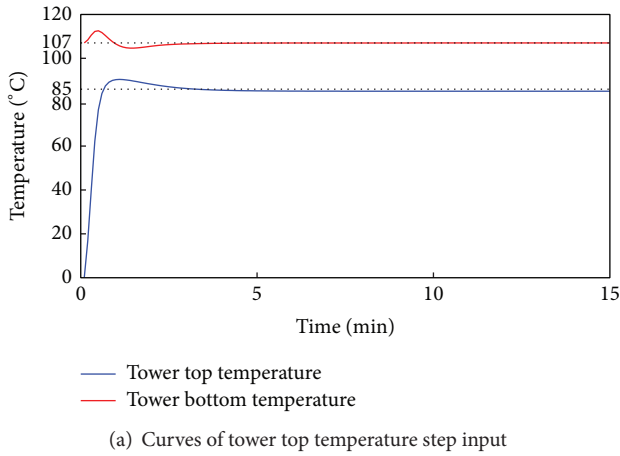


FIGURE 10: Simulation results of step input without disturbance.

FIGURE 11: Simulation results with tower top temperature disturbance.

of the stripping process, this paper considers the following situations in the simulation.

- (1) Without any disturbance, the tracking performance of the tower top temperature and tower bottom temperature is verified, which is shown in Figure 10. It can be seen from Figure 10 that the control system can track the input changes and has a high response performance.
- (2) The random disturbance $\Delta y_1 = 0.01$ is added on the tower top temperature and the simulation results are shown in Figure 11. The random disturbance $\Delta y_2 = 0.01$ is added on the tower bottom temperature and the simulation results are shown in Figure 12. By adding the disturbance signals on the tower top temperature and tower bottom temperature, the system decoupling effects are verified. It can be seen from Figures 10 and 11 that the neural network decoupling modules improve the performance of the control system and the transaction between two loops can be controlled effectively.

Based on the above simulation results, the following conclusions can be made. (1) The tracking performance of the system is better and is able to adapt to the change of set points. The response time and other parameters are satisfactory. (2) The coupling degree between the slurry flow rate, the tower top temperature, and steam flow rate, the tower bottom temperature, has been significantly reduced and the decoupling is achieved. (3) The system has certain robustness for the external disturbances. It can overcome the temperature disturbance of the tower top temperature and the tower bottom temperature.

6. Conclusions

In the paper, the D-FNN was used to set up the PVC stripping process model. Then the strategy of combination of the neural network decoupling module and the BP neural network PID controller is put forward for controlling the stripping process temperature. The simulation results verify the effectiveness of the proposed PVC stripping intelligent control strategy.

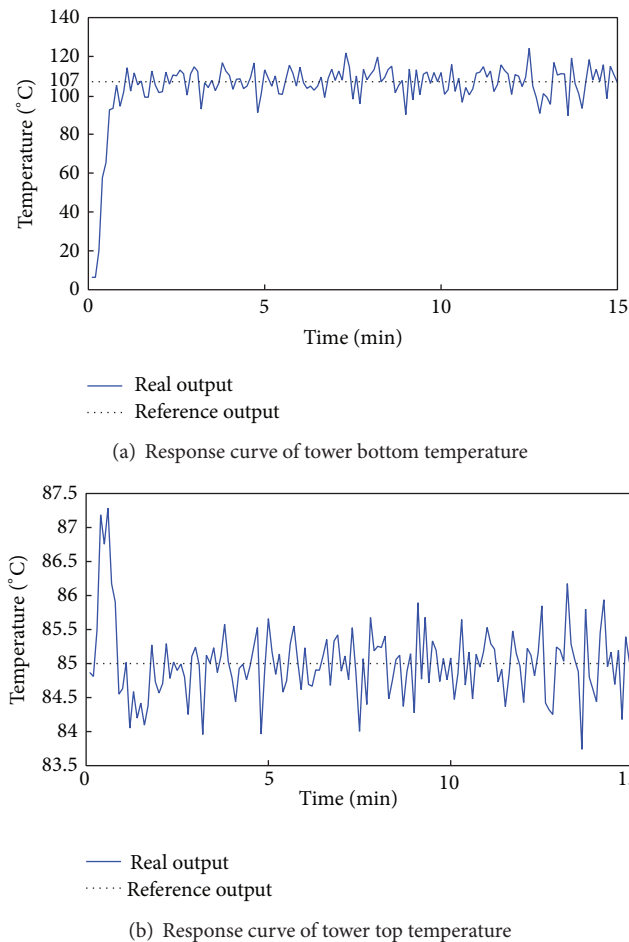


FIGURE 12: Simulation results with tower bottom temperature disturbance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the Program for China Postdoctoral Science Foundation (Grant no. 20110491510), the Program for Liaoning Excellent Talents in University (Grant no. LJQ2011027), and the Program for Research Special Foundation of University of Science and Technology of Liaoning (Grant no. 2011ZX10).

References

- [1] J. S. Wang, S. Han, and Q. P. Guo, "Echo state networks based predictive model of vinyl chloride monomer convention velocity optimized by artificial fish swarm algorithm," *Soft Computing*, vol. 18, no. 3, pp. 457–468, 2014.
- [2] Z. G. Hui, "The application of stripping technology in the production of PVC," *Polyvinyl Chloride*, vol. 7, no. 2, pp. 8–12, 2007.
- [3] H. Li, "Design of Multivariable Fuzzy-neural Network Decoupling Controller," *Control and Decision*, vol. 21, no. 5, pp. 593–596, 2006.
- [4] W. G. Xiao, Z. M. Wang, Y. F. Guo et al., "PVC boiling bed drying process optimization control system," *Automation and Instruments in Chemical Industry*, vol. 21, no. 5, pp. 12–15, 1994.
- [5] M. Y. Khairiyah, C. E. Boo, and T. S. Amy, *Formulation of Model Predictive Control Algorithm For Nonlinear Processes*, University of Technology Malaysia, 2006.
- [6] S. Q. Wu and J. Xu, *Dynamic Fuzzy Neural Network-Design and Application*, Tsinghua University Press, Beijing, China, 2008.
- [7] J. S. Wang and Q. P. Guo, "D-FNN based soft-sensor modeling and migration reconfiguration of polymerizing process," *Applied Soft Computing*, vol. 13, no. 4, pp. 1892–1901, 2013.
- [8] C. K. Kwong, T. C. Wong, and K. Y. Chan, "A methodology of generating customer satisfaction models for new product development using a neuro-fuzzy approach," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11262–11270, 2009.
- [9] K. Y. Chan, C. K. Kwong, and Y. C. Tsim, "Modelling and optimization of fluid dispensing for electronic packaging using neural fuzzy networks and genetic algorithms," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 1, pp. 18–26, 2010.
- [10] C. K. Kwong, K. Y. Chan, and Y. C. Tsim, "A genetic algorithm based knowledge discovery system for the design of fluid dispensing processes for electronic packaging," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3829–3838, 2009.
- [11] R. L. Liu, H. Y. Su, and J. Chu, "A soft sensor modeling algorithm based on modified fuzzy neural network," *Information and Control*, vol. 32, no. 4, pp. 367–370, 2003.
- [12] G.-C. Chen, Y.-F. Xu, and J.-S. Yu, "Soft-sensor modelling of acrylonitrile yield based on particle swarm optimization fuzzy neural networks," *Journal of System Simulation*, vol. 19, no. 23, pp. 5370–5372, 2007.
- [13] I. Li and L. W. Lee, "Interval type 2 hierarchical FNN with the H-infinity condition for MIMO non-affine systems," *Applied Soft Computing*, vol. 12, no. 8, pp. 1996–2011, 2012.
- [14] H.-K. Wu, J.-H. Hsieh, Y.-L. Lin, and J.-H. Jeng, "On maximum likelihood fuzzy neural networks," *Fuzzy Sets and Systems*, vol. 161, no. 21, pp. 2795–2807, 2010.
- [15] M. B. Nasr and M. Chtourou, "A self-organizing map-based initialization for hybrid training of feedforward neural networks," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 4458–4464, 2011.
- [16] H. C. Lu, M. H. Chang, and C. H. Tsai, "Adaptive self-constructing fuzzy neural network controller for hardware implementation of an inverted pendulum system," *Applied Soft Computing*, vol. 11, no. 5, pp. 3962–3975, 2011.
- [17] S. Wu and M. J. Er, "Dynamic fuzzy neural networks—a novel approach to function approximation," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 30, no. 2, pp. 358–364, 2000.
- [18] F. J. Lin, C. H. Lin, and P. H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive. Fuzzy Systems," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 5, pp. 751–759, 2001.
- [19] X. Deng and X. Wang, "Incremental learning of dynamic fuzzy neural networks for accurate system modeling," *Fuzzy Sets and Systems*, vol. 160, no. 7, pp. 972–987, 2009.
- [20] C.-F. Juang and C.-D. Hsieh, "A locally recurrent fuzzy neural network with support vector regression for dynamic-system

- modeling," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 261–273, 2010.
- [21] A. Subasi, "Automatic detection of epileptic seizure using dynamic fuzzy neural networks," *Expert Systems with Applications*, vol. 31, no. 2, pp. 320–328, 2006.
- [22] H. Adeli and X. Jiang, "Dynamic fuzzy wavelet neural network model for structural system identification," *Journal of Structural Engineering*, vol. 132, no. 1, pp. 102–111, 2006.
- [23] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, 2000.
- [24] B. H. M. Sadeghi, "BP-neural network predictor model for plastic injection molding process," *Journal of Materials Processing Technology*, vol. 103, no. 3, pp. 411–416, 2000.
- [25] Z. Xiao, S.-J. Ye, B. Zhong, and C.-X. Sun, "BP neural network with rough set for short term load forecasting," *Expert Systems with Applications*, vol. 36, no. 1, pp. 273–279, 2009.
- [26] J. Yi, Q. Wang, D. Zhao, and J. T. Wen, "BP neural network prediction-based variable-period sampling approach for networked control systems," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 976–988, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

