

## Research Article

# Optimal Control of Complex Systems Based on Improved Dual Heuristic Dynamic Programming Algorithm

Hui Li,<sup>1,2</sup> Yongsui Wen,<sup>1</sup> and Wenjie Sun<sup>1</sup>

<sup>1</sup>*School of Electrical and Electronic Engineering, Changchun University of Technology, Changchun, China*

<sup>2</sup>*Jilin University, Changchun, China*

Correspondence should be addressed to Hui Li; [lihui@mail.ccut.edu.cn](mailto:lihui@mail.ccut.edu.cn)

Received 22 May 2017; Revised 9 August 2017; Accepted 27 August 2017; Published 9 October 2017

Academic Editor: Jun M. Wang

Copyright © 2017 Hui Li 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.

When applied to solving the data modeling and optimal control problems of complex systems, the dual heuristic dynamic programming (DHP) technique, which is based on the BP neural network algorithm (BP-DHP), has difficulty in prediction accuracy, slow convergence speed, poor stability, and so forth. In this paper, a dual DHP technique based on Extreme Learning Machine (ELM) algorithm (ELM-DHP) was proposed. Through constructing three kinds of network structures, the paper gives the detailed realization process of the DHP technique in the ELM. The controller designed upon the ELM-DHP algorithm controlled a molecular distillation system with complex features, such as multivariability, strong coupling, and nonlinearity. Finally, the effectiveness of the algorithm is verified by the simulation that compares DHP and HDP algorithms based on ELM and BP neural network. The algorithm can also be applied to solve the data modeling and optimal control problems of similar complex systems.

## 1. Introduction

With the increase of the dynamic complexity of controlled objects and their widespread application, some dynamic models of controlled objects can be obtained by mathematical methods. But some of the controlled objects are too complex to establish an accurate mathematical model, such as a robot system and a chemical generation process. Even though the mathematical model of complex systems has been established, it will be usually a high-order nonlinear time-varying complex differential equation. So, it could not describe the system accurately, and it is too difficult to analyze and process the data, and hence the unknown model must be learned by the observational data [1]. This is because of the great fault tolerance, self-adaptation, self-organization, and learning and memory ability of neural networks, which provides a new method for the modeling of complex systems [2–5]. However, BP networks, RBF networks, and SVM have some defects such as slow convergence, which causes a big gap between the approximation ability and the actual demand of complex systems [6, 7]. In order to solve this problem, Huang GB proposed single hidden layer feedforward network (SLFN)

training—Extreme learning Machine (ELM)—in 2006 [8]. The ELM gives the weights and thresholds of the weights randomly and then calculates the output weights by the regularization principle, which can still approach any continuous system [6, 7]. It has been proved that the SLFN hidden layer node parameter randomly accessed does not affect the convergence ability, and it also makes the speed of learning the ELM thousands of times faster than the traditional BP network and SVM.

In recent years, dynamic programming has been used to solve the optimal control problem, the “curse of dimensionality” problem, so no optimal solution could be obtained [9]. In 1977, Werbos proposed a heuristic dynamic programming (HDP) and dual heuristic programming (DHP) concept and proposed a method of approximate dynamic programming (adaptive/approximate dynamic programming, ADP) to solve the “curse of dimensionality” problem [10]. Werbos defines “intelligence” as the brain’s ability to learn a utility function maximally in a complex, unknown, nonlinear environment [11]. ADP is the general scheme for learning approximate optimal action strategies. Therefore, ADP can be regarded as a key method which is able to design the

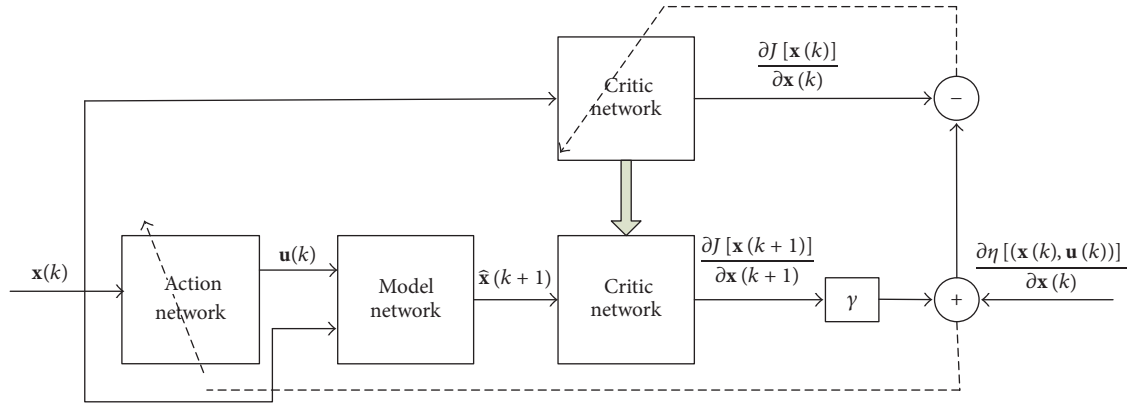


FIGURE 1: The structural diagram of the DHP algorithm.

intelligent system of a brain. According to the basic principle, realization structure, and current development of the ADP method, Lewis and others gave a summary and prospect of the research and pointed out that ADP is an effective data-driven method [12–15]. ADP can realize the optimization control of the nonlinear system by using neural networks based on online data and control information to approximate the performance index function of the optimal control law, without a mathematical model of a nonlinear control system [16]. To develop neural dynamic programming results and relax the system dynamic requirements, Zhong et al. proposed a new goal representation ADP online optimization control structure for nonlinear systems. But the network could not output the derivative function information of the cost function directly based on the implementation structure of HDP, and the control effect of the HDP structure needed improvement [17]. In fact, some studies show that DHP and GDHP can be controlled better than HDP in the structure of ADP method to some extent [18, 19].

In general, the study on the optimal control of nonlinear systems of ADP based on the traditional neural network has made great progress. But the ADP still has problems of slow response and poor stability. In this paper, ELM algorithm gives random input weights and thresholds to improve the response speed of the DHP algorithm and the stability of the DHP algorithm is improved by calculating the output weights by the regularization principle. In order to verify the validity of the algorithm, the ELM-DHP was designed to control the molecular distillation system with multivariability, nonlinearity, strong coupling, and large delay.

## 2. Algorithm Principle

**2.1. DHP Algorithm Principle.** The discrete-time nonlinear dynamic system is described as follows:

$$x(k+1) = F[x(k), u(k), k], \quad k = 0, 1, \dots \quad (1)$$

In formula (1),  $x(k) \in \mathbf{R}^n$  represents the state vector of the system,  $u(k) \in \mathbf{R}^m$  represents control variables, and  $F$  represents the system function.

The performance index function (also called the cost function) corresponding to the system is

$$J[x(k), k] = \sum_{i=k}^{\infty} \gamma^{i-k} \eta[x(i), u(i), i], \quad (2)$$

where  $\eta$  is the utility function,  $\gamma$  is the discount factor ( $0 < \gamma \leq 1$ ),  $J$  is the cost function of state  $x(k)$ , and  $J$  depends on the initial time  $k$  and the initial state  $x(k)$ . For DHP, the purpose of dynamic programming is to select a control sequence  $u(i)$ ,  $i = k, k+1, \dots, l$ , which minimizes the function  $\partial J[x(k)]/\partial x(k)$ .

The DHP structure is shown in Figure 1, which contains three neural networks: model network, critic network, and action network. The neural network has a powerful function of universal approximation, so the model network can be used to model the unknown nonlinear or complex nonlinear system and make the DHP method widely used. The input of the critic network is a state variable. The output of the critic network is approximation performance index function  $J$  on the state  $x$  derivative, which is also known as the costate. The action network, also known as “Actor,” represents the mapping between system state variables and control variables [20–22].

$\partial J[x(k)]/\partial x(k)$  is based on the iteration of the derivative for performance index function and utility function to state.

$$\frac{\partial J[x(k)]}{\partial x(k)} = \frac{\partial \eta\{x(k), u[x(k)]\}}{\partial x(k)} + \gamma \frac{\partial J[x(k+1)]}{\partial x(k)}. \quad (3)$$

In (3),  $u[x(k)]$  is a feedback control variable, and costates  $\partial J[x(k)]/\partial x(k)$  and  $\partial J[x(k+1)]/\partial x(k)$  are the outputs of the critic network. If the weight of the critic network is set to  $\omega$ , the right type of formula (1) is set to

$$e[x(k), \omega] = \frac{\partial \eta\{x(k), u[x(k)]\}}{\partial x(k)} + \gamma \frac{\partial J[x(k+1)]}{\partial x(k)}. \quad (4)$$

At the same time, the left type of formula (1) can be written as  $\partial J[x(k), \omega]/\partial x(k)$ . By adjusting the weights  $\omega$  of

the critic network, the least-mean-square-error function is as follows:

$$\omega^* = \arg \min_{\omega} \left\{ \left| \frac{\partial J[\mathbf{x}(k), \omega]}{\partial \mathbf{x}(k)} - e[\mathbf{x}(k), \omega] \right|^2 \right\}. \quad (5)$$

According to the principle of optimality, the optimal control should satisfy the first-order differential necessary condition.

$$\begin{aligned} \frac{\partial J^*[\mathbf{x}(k)]}{\partial \mathbf{u}(k)} &= \frac{\partial \eta[\mathbf{x}(k), \mathbf{u}(k)]}{\partial \mathbf{u}(k)} + \frac{\partial J^*[\mathbf{x}(k+1)]}{\partial \mathbf{u}(k)} \\ &= \frac{\partial \eta[\mathbf{x}(k), \mathbf{u}(k)]}{\partial \mathbf{u}(k)} \\ &\quad + \frac{\partial J^*[\mathbf{x}(k+1)]}{\partial \mathbf{x}(k+1)} \frac{\partial F[\mathbf{x}(k), \mathbf{u}(k)]}{\partial \mathbf{u}(k)}. \end{aligned} \quad (6)$$

So, the optimal quantity is obtained:

$$\begin{aligned} u^* &= \arg \min_u \left\{ \left| \frac{\partial J[\mathbf{x}(k)]}{\partial \mathbf{u}(k)} - \frac{\partial \eta[\mathbf{x}(k), \mathbf{u}(k)]}{\partial \mathbf{u}(k)} \right. \right. \\ &\quad \left. \left. - \frac{\partial J^*[\mathbf{x}(k+1)]}{\partial \mathbf{x}(k+1)} \frac{\partial F[\mathbf{x}(k), \mathbf{u}(k)]}{\partial \mathbf{u}(k)} \right| \right\}. \end{aligned} \quad (7)$$

In formula (7),  $\partial J^*[\mathbf{x}(k+1)]/\partial \mathbf{x}(k+1)$  is the optimal costate, satisfying formula (5).

From (1) to (7), we can conclude that the optimal control quantity of the DHP method can be obtained directly by the costate. Compared with the HDP method which obtained the optimal control by the relationship between the weights ( $\omega$ ) of the critic network and the input-output, the method of DHP has more computational efforts, but better control effect [9].

**2.2. ELM Algorithm Principle.** For a standard SLFN with  $L$  hidden layer neurons learning  $N$  arbitrary distinct samples  $(\mathbf{x}_j, \mathbf{t}_j)$ ,  $\mathbf{x}_j = [x_{j1}, x_{j2}, \dots, x_{jn}]^T \in R^n$ ,  $\mathbf{t}_j = [t_{j1}, t_{j2}, \dots, t_{jm}]^T \in R^m$  and activation function  $g(\cdot)$  are mathematically modeled as [23]

$$\sum_{i=1}^L \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{o}_j, \quad (8)$$

$$j = 1, 2, \dots, N, \quad b_i, \beta_i \in R, \quad a_i \in R,$$

where  $\mathbf{o}_j = [o_{j1}, o_{j2}, \dots, o_{jm}]^T \in R^m$  is the model output of the network,  $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  is the input weight matrix between the input layer neuron and the  $i$ th hidden layer neuron,  $\beta_i$  is the output weight matrix between the  $i$ th hidden layer neuron and the output layer neurons,  $b_i$  is the threshold of the  $i$ th neuron in the hidden layer, and  $\mathbf{w}_i \cdot \mathbf{x}_j$  is the inner product of  $\mathbf{w}_i$  and  $\mathbf{x}_j$ .

The learning objective of the SLFN is to minimize the output error. Error can be expressed as

$$\sum_{j=1}^N \|\mathbf{t}_j - \mathbf{o}_j\| = 0. \quad (9)$$

The presence of  $\mathbf{w}_i$ ,  $\mathbf{x}_j$ , and  $b_i$  makes

$$\sum_{i=1}^L \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j, \quad j = 1, 2, \dots, N. \quad (10)$$

So, (10) can be written as  $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$ , where

$$\begin{aligned} \mathbf{H}(\mathbf{w}_1, \dots, \mathbf{w}_L, b_1, \dots, b_N, \mathbf{x}_1, \dots, \mathbf{x}_N) \\ &= \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \cdots & \cdots & \cdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L}, \\ \boldsymbol{\beta} &= \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \\ \mathbf{T} &= \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}. \end{aligned} \quad (11)$$

$\mathbf{H}$  is a hidden layer output matrix of ELM. So, the training of ELM is equivalent to the least-squares solution  $\hat{\boldsymbol{\beta}}$  of linear system  $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$ .

$$\|\mathbf{H}(\hat{\mathbf{w}}_i, \hat{b}_i) \hat{\boldsymbol{\beta}} - \mathbf{T}\| = \min_{\mathbf{w}, b, \boldsymbol{\beta}} \|\mathbf{H}(\hat{\mathbf{w}}_i, \hat{b}_i) \cdot \boldsymbol{\beta} - \mathbf{T}\|. \quad (12)$$

In (12),  $i = 1, 2, \dots, L$ , (12) is equivalent to minimizing the loss function

$$E = \sum_{j=1}^N \left\| \sum_{i=1}^L \beta_i \cdot g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) - \mathbf{t}_j \right\|^2. \quad (13)$$

Huang et al. [23] proved that the minimum value of the least-squares solution of the linear system satisfies the following.

(1) *Minimum Training Error.* The special solution  $\hat{\boldsymbol{\beta}} = \mathbf{H}^{-1}\mathbf{T}$  is one of the least-squares solutions of a general linear system  $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, \mathbf{H}^{-1}$  which is a generalized inverse matrix of  $\mathbf{H}$ .

(2) *Smallest Norm of Weights and Best Generalization Capability.* Further, the special solution  $\hat{\boldsymbol{\beta}} = \mathbf{H}^{-1}\mathbf{T}$  has the smallest norm among all of the least-squares solutions of  $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$ :  $\|\hat{\boldsymbol{\beta}}\| = \|\mathbf{H}^{-1}\mathbf{T}\| \leq \|\boldsymbol{\beta}\|, \forall \boldsymbol{\beta} \in \{\boldsymbol{\beta} : \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| \leq \|\mathbf{H}\mathbf{z} - \mathbf{T}\|, \forall \mathbf{z} \in \mathbf{R}^{N \times N}\}$ . The generalization ability of SLFN with minimum weight is independent of the number of parameters [24]. The smaller the weight, the stronger the generalization ability of SLFN.

(3) *Special Solution.* The least-squares solution of  $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$  is unique.

**2.3. Proof the Stability of ELM-DHP.** The stability of the ELM-DHP algorithm is proved (i.e., the output error of the system is 0). The discrete nonlinear system is controlled

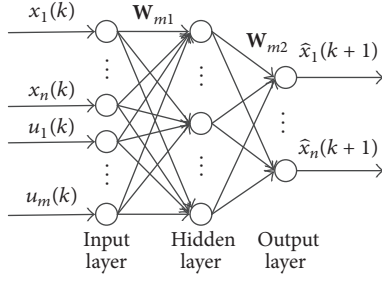


FIGURE 2: The structure of the model network.

by the ELM-DHP algorithm, and the three networks of the ELM-DHP algorithm are all based on the fixed ELM implementation. Therefore, it just needs to be proved that ELM can approximate the discrete nonlinear system by 0 error.

The ELM learning algorithm is chosen as a SLFN with  $L$  hidden layer neurons. The  $N$  arbitrary distinct samples  $(\mathbf{x}_j, \mathbf{t}_j)$ , where  $L \leq N$ ,  $\mathbf{x}_j = [x_{j1}, x_{j2}, \dots, x_{jn}]^T \in R^n$ , and  $\mathbf{t}_j = [t_{j1}, t_{j2}, \dots, t_{jm}]^T \in R^m$  of the nonlinear discrete system and nonlinear activation function  $g(\cdot)$ , are mathematically modeled as formula (8).

ELM learns a large number of samples generally, and the number of neurons in the hidden layer is far less than the number of samples,  $L \ll N$ . So, we only need to prove that the learning error of ELM was 0 when  $L \leq N$ . Huang et al. [7, 23, 25] proved in detail that the SLFN with  $L$  neurons can approximate any arbitrary sample  $(\mathbf{x}_j, \mathbf{t}_j)$  at any small error; that is,

$$e = \sum_{j=1}^N \|\mathbf{t}_j - \mathbf{o}_j\| = \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| = 0. \quad (14)$$

The work above proves that the learning error of ELM is 0 (i.e., the stability of the ELM-DHP algorithm).

### 3. Implementation of the ELM-DHP Algorithm

The ELM-DHP algorithm includes three networks: model network, critic network, and action network. The hidden layer of the three networks is a sigmoidal bipolar function and the output layer is a pure linear function. The realization process of the ELM-DHP algorithm is studied by using the discrete-time nonlinear dynamic programming of  $n$ -dimensional state vector and  $m$ -dimensional control vector as the research object.

**3.1. Network Model.** The model network adopts  $(m+n) - l_m - n$  structure. The  $m+n$  inputs are the  $n$  components of the state vector  $\mathbf{x}(k)$  in the  $k$  moments and the  $m$  components of the predicted output  $\mathbf{u}(k)$  of the action network to state  $\mathbf{x}(k)$  in the system of  $(k-1)$  moments. The  $n$  output is the  $n$  components of the prediction vector  $\hat{\mathbf{x}}(k+1)$  to the state vector  $\mathbf{x}(k+1)$  in the system of  $(k+1)$  moments. The model network has  $l_m$  hidden layer neurons. The structure of the model network is shown in Figure 2.

The model network is trained offline, and the calculation process is as follows.

The input layer to the hidden layer weight matrix  $\mathbf{W}_{m1}$  and the hidden layer threshold matrix  $\mathbf{B} = [b_1, b_2, \dots, b_{l_m}]$  are randomly generated. Define the input vector  $\mathbf{M}(k)$  and the expected output vector  $\hat{\mathbf{x}}(k)$  of the model network in  $k$  moments:

$$\mathbf{M}(k) = [u_1(k), u_2(k), \dots, u_m(k), x_1(k), x_2(k), \dots, x_n(k)], \quad (15)$$

$$\hat{\mathbf{x}}(k) = [\hat{x}_1(k), \hat{x}_2(k), \dots, \hat{x}_n(k)].$$

Calculate the output matrix  $\mathbf{m}_{h2}(k)$  of the hidden layer in the model network

$$m_{h1j}(k) = \sum_{i=1}^{n+m} M_i(k) \cdot W_{m1ij}(k) + b_j(k), \quad j = 1, 2, \dots, l_m, \quad (16)$$

$$m_{h2j}(k) = \frac{1 - e^{-m_{h1j}(k)}}{1 + e^{-m_{h1j}(k)}}, \quad j = 1, 2, \dots, l_m,$$

where  $m_{h1j}(k)$  is the input of the  $j$ th node in the model network hidden layer,  $m_{h2j}(k)$  is the output of the  $j$ th node in the model network hidden layer, and  $\mathbf{m}_{h2} = [m_{h21}, m_{h22}, \dots, m_{h2l_m}] \in R^m$ .

Calculate the weights  $\mathbf{W}_{m2}(k)$  from the hidden layer to the output layer:

$$\hat{\mathbf{x}}(k+1) = \mathbf{m}_{h2}(k) \times \mathbf{W}_{m2}(k). \quad (17)$$

According to the idea of the ELM, the error is minimized as

$$E_m(k+1) = \sum_{i=1}^n \|x_i(k+1) - \hat{x}_i(k+1)\|^2 = 0. \quad (18)$$

In equality (18),  $x_i(k+1)$  is the expected output  $i$ th output layer neurons of the model network.

$\mathbf{W}_{m2}(k)$  is equivalent to solving the least-squares solution  $\widehat{\mathbf{W}}_{m2}(k)$  of the linear system  $\mathbf{m}_{h2}(k) \times \mathbf{W}_{m2}(k) = \hat{\mathbf{x}}(k+1)$ :

$$\begin{aligned} & \|\mathbf{m}_{h2}(k) \times \widehat{\mathbf{W}}_{m2} - \hat{\mathbf{x}}(k+1)\| \\ &= \min_{\mathbf{W}_{m2}} \|\mathbf{m}_{h2}(k) \times \mathbf{W}_{m2} - \hat{\mathbf{x}}(k+1)\|. \end{aligned} \quad (19)$$

The special solution  $\widehat{\mathbf{W}}_{m2}(k)$  of the weight matrix of the hidden layer and output layer in the model network is as follows:

$$\widehat{\mathbf{W}}_{m2}(k) = \mathbf{m}_{h2}^{-1}(k) \times \hat{\mathbf{x}}(k+1), \quad (20)$$

where  $\mathbf{m}_{h2}^{-1}(k)$  is a generalized inverse matrix of  $\mathbf{m}_{h2}(k)$  in  $k$  moments.

**3.2. Critic Network.** The critic network is composed of  $n - l_c - n$ . The  $n$  inputs are the  $n$  components of the state vector  $\hat{\mathbf{x}}(k)$ , and the output is the estimation of the state

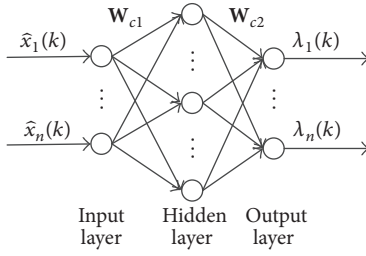


FIGURE 3: The structure of the critic network.

$\lambda(k) = \partial J(k)/\partial \mathbf{x}(k)$ ,  $J(k) = \gamma J(k+1) + \eta(k)$ .  $l_c$  is the number of hidden layer neurons in the critic network. In the critic network, the weight matrix from the input layer to the hidden layer, the weight matrix from the hidden layer to the output layer, and the hidden layer threshold matrix of  $k$  time are, respectively, defined as  $\mathbf{W}_{c1}$ ,  $\mathbf{W}_{c2}$ ,  $\mathbf{C}(k) = [c_1(k), \dots, c_{l_c}(k)]$ . Figure 3 shows the structure of the critic network.

The critic network uses the least-squares method of ELM, whose forward calculation process is

$$\begin{aligned}
 c_{h1j}(k) &= \sum_{i=1}^n \hat{x}_i(k) W_{c1ij}(k) + c_j(k), \quad j = 1, 2, \dots, l_c, \\
 c_{h2j}(k) &= \frac{1 - e^{-c_{h1j}(k)}}{1 + e^{-c_{h1j}(k)}}, \quad j = 1, 2, \dots, l_c,
 \end{aligned} \quad (21)$$

$$\lambda(k) = \mathbf{c}_{h2}(k) \times \mathbf{W}_{c2}(k),$$

where  $c_{h1j}$  is the input of the  $j$ th node in the critic network hidden layer,  $c_{h2j}$  is the output of the  $j$ th node in the critic network hidden layer,  $\mathbf{c}_{h2}(k) = [c_{h21}, c_{h22}, \dots, c_{h2l_c}]$ , and  $\lambda(k)$  is the output of the critic network output layer. The inputs  $\hat{x}_i(k)$  of the critic network come from the output of the model network and the outputs of the critic network are costate function  $\partial J(k)/\partial \mathbf{x}(k)$  in the DHP.  $\lambda_a(k)$  is expressed to the expected output of the critic network, which can be written as

$$\lambda_a(k) = \frac{\partial J(k)}{\partial \mathbf{x}(k)}. \quad (22)$$

The training error of DHP critic network is minimized based on the idea of ELM.

$$\begin{aligned}
 \|E_c\| &= \sum_k E_c(k) = \frac{1}{2} \sum_k \sum_{j=1}^n E_j^2(k) = 0, \\
 E_c(k) &= \frac{1}{2} \sum_{j=1}^n E_j^2(k) = \frac{1}{2} \mathbf{E}(k) \times \mathbf{E}^T(k) = 0,
 \end{aligned} \quad (23)$$

where  $E_c(k)$  is the error of the critic network in  $k$  moments and  $\|E_c\|$  is the error of all the time points in the critic network.

According to the DHP structure and the definition of the expected outputs  $\lambda_a(k)$  of the critic network, we can obtain

$$E(k) = \lambda(k) - \frac{\partial^+ \eta(k)}{\partial \mathbf{x}(k)} - \gamma \frac{\partial^+ J(k+1)}{\partial \mathbf{x}(k)}. \quad (24)$$

In formula (24),  $\partial^+ \eta(k)/\partial \mathbf{x}(k)$  and  $\partial^+ J(k+1)/\partial \mathbf{x}(k)$  represent the notion that  $\eta(k)$  and  $J(k+1)$  take the derivative of composite function  $\mathbf{x}(k)$ .

Based on (23) and (24), we can acquire

$$\lambda(k) = \frac{\partial^+ \eta(k)}{\partial \mathbf{x}(k)} + \gamma \frac{\partial^+ J(k+1)}{\partial \mathbf{x}(k)}. \quad (25)$$

According to  $\mathbf{C}_{h2}(k) \times \mathbf{W}_{c2} = \lambda(k)$ , the weight  $\mathbf{W}_{c2}$  from the hidden layer to the output layer is equal to the least-squares solution  $\widehat{\mathbf{W}}_{c2}$  of the linear system  $\mathbf{C}_{h2}(k) \times \mathbf{W}_{c2} = \lambda(k)$ , and hence we get  $\widehat{\mathbf{W}}_{c2}$ :

$$\begin{aligned}
 &\|\mathbf{C}_{h2}(k) \times \widehat{\mathbf{W}}_{c2} - \lambda(k)\| \\
 &= \min_{\mathbf{W}_{c2}} \|\mathbf{C}_{h2}(k) \times \mathbf{W}_{c2} - \lambda(k)\|,
 \end{aligned} \quad (26)$$

$$\widehat{\mathbf{W}}_{c2} = \mathbf{C}_{h2}^{-1}(k) \times \lambda(k).$$

In formula (26),  $\mathbf{C}_{h2}^{-1}(k)$  is a generalized inverse matrix of  $\mathbf{C}_{h2}(k)$ . Based on the DHP structure and the chain rule [9], we can obtain

$$\begin{aligned}
 \frac{\partial^+ J(k+1)}{\partial \mathbf{x}(k)} &= \frac{1}{2} \cdot \lambda(k+1) \times \mathbf{W}_{m2}^T \times \left\{ \mathbf{W}_{m1x}(k) \right. \\
 &\quad \left. \otimes \underbrace{[1 - \mathbf{m}_{h2}(k) \otimes \mathbf{m}_{h2}(k); 1 - \mathbf{m}_{h2}(k) \otimes \mathbf{m}_{hh2}(k)]}_{\text{total } n} \right\}^T \\
 &\quad + \left( \frac{1}{4} \lambda(k+1) \times \left\{ \mathbf{W}_{m2}^T(k) \right. \right. \\
 &\quad \left. \left. \otimes \underbrace{[1 - \mathbf{m}_{h2}(k) \otimes \mathbf{m}_{h2}(k); 1 - \mathbf{m}_{h2}(k) \otimes \mathbf{m}_{h2}(k)]}_{\text{total } n} \right\} \right. \\
 &\quad \left. \times \mathbf{W}_{m1u}^T(k) \times \mathbf{W}_{a2}^T(k) \times \left\{ \mathbf{W}_{a1}(k) \right. \right. \\
 &\quad \left. \left. \otimes \underbrace{[1 - \mathbf{a}_{h2}(k) \otimes \mathbf{a}_{h2}(k); 1 - \mathbf{a}_{h2}(k) \otimes \mathbf{a}_{h2}(k)]}_{\text{total } n} \right\} \right)^T.
 \end{aligned} \quad (27)$$

In (27),  $\mathbf{W}_{m1u} = \mathbf{W}_{m1}(1:m, :)$  represents front  $m$  lines of the weight matrix  $\mathbf{W}_{m1}$ , and  $\mathbf{W}_{m1x} = \mathbf{W}_{m1}(m+1:m+n, :)$  represents from  $(m+1)$ th to  $(m+n)$ th line of the weight matrix  $\mathbf{W}_{m1}$ .

$$\begin{aligned}
 \frac{\partial^+ \eta(k)}{\partial \mathbf{x}(k)} &= 2\mathbf{x}(k) + \mathbf{u}(k) \times \mathbf{W}_{a2}^T(k) \times \left\{ \mathbf{W}_{a1}(k) \right. \\
 &\quad \left. \otimes \underbrace{[1 - \mathbf{a}_{h2}(k) \otimes \mathbf{a}_{h2}(k); 1 - \mathbf{a}_{h2}(k) \otimes \mathbf{a}_{h2}(k)]}_{\text{total } n} \right\}^T.
 \end{aligned} \quad (28)$$

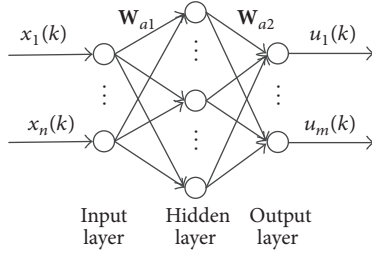


FIGURE 4: The structure of the action network.

**3.3. Action Network.** The action network uses the structure of  $n-l_d-m$ .  $n$  inputs are the  $n$  components of the state vector  $\mathbf{x}(k)$  of the system at  $k$  moments.  $m$  outputs are the  $m$  components of the control vector  $\mathbf{u}(k)$  corresponding to the input state vector  $\mathbf{x}(k)$ .  $l_d$  represents the number of neurons in the action network hidden layer.  $\mathbf{W}_{a1}$  and  $\mathbf{W}_{a2}$  are, respectively, the weight matrix from the input layer to the hidden layer and the weight matrix from the hidden layer to the output layer in the action network.  $\mathbf{d}(k) = [d_1(k), \dots, d_{l_d}(k)]$  is the hidden layer threshold matrix of the action network. Figure 4 is the structure of the action network.

The calculation process of the action network is as follows:

$$a_{h1j}(k) = \sum_{i=1}^n x_i(k) \cdot W_{a1ij}(k) + d_j(k), \quad j = 1, 2, \dots, l_d, \quad (29)$$

$$a_{h2j}(k) = \frac{1 - e^{-a_{h1j}(k)}}{1 + e^{-a_{h1j}(k)}}, \quad j = 1, 2, \dots, l_d,$$

$$\mathbf{u}(k) = \mathbf{a}_{h2}(k) \cdot \mathbf{W}_{a2}(k),$$

where  $a_{h1j}(k)$  is the input of the  $j$ th node and  $a_{h2j}(k)$  is the output of the  $j$ th node in the action network hidden layer and  $\mathbf{a}_{h2}(k) = [a_{h21}, a_{h22}, \dots, a_{h2l_d}]$ . According to the idea of weight adjustment of ELM, the weight matrix  $\mathbf{W}_{a2}$  from the hidden layer to the output layer is obtained:

$$\mathbf{W}_{a2} = \mathbf{a}_{h2}^{-1}(k) \times \mathbf{u}(k). \quad (30)$$

In (30),  $\mathbf{a}_{h2}^{-1}(k)$  is a generalized inverse matrix of  $\mathbf{a}_{h2}(k)$  and  $\mathbf{u}(k)$  is the expected output of the action network. The weights of the network will be corrected if  $\mathbf{u}(k)$  can be got. The inverse sigmoidal function is defined as  $g(\cdot)$ . The calculation process of  $\mathbf{u}(k)$  is as follows:

$$\mathbf{A} = \left[ \frac{\partial^+ U(k)}{\partial \mathbf{x}(k)} + \gamma \frac{\partial^+ J(k+1)}{\partial \mathbf{x}(k)} \right] \times \mathbf{W}_{c2}^{-1}, \quad (31)$$

$$\mathbf{B} = g(\mathbf{A}) \times \mathbf{W}_{c1}^{-1} \times \mathbf{W}_{m2}^{-1}, \quad (32)$$

$$\begin{bmatrix} \mathbf{u}(k) \\ \mathbf{x}(k) \end{bmatrix} = g(\mathbf{B}) \times \mathbf{W}_{m1}^{-1}. \quad (33)$$

In (33),  $\mathbf{u}(k)$  is the first  $m$  rows of matrix  $g(\mathbf{B}) \times \mathbf{W}_{m1}^{-1}$ .

Define  $\mathbf{u}_x = g(\mathbf{B}) * \mathbf{W}_{m1}^{-1}$ ; we have

$$\mathbf{u}(k) = \mathbf{u}_x(1 : m, :). \quad (34)$$

So,  $\mathbf{W}_{a2}$  can be got:

$$\mathbf{W}_{a2} = \mathbf{a}_{h2}^{-1}(k) \times \mathbf{u}(k). \quad (35)$$

**3.4. Training Strategy.** In this paper, the model network of the DHP algorithm is trained by an offline method at first to obtain the weight matrix of the model network. Then, the action network and the critic network are trained simultaneously. Training strategies are as follows:

(1) First, the model is trained by an offline method and the weight matrix of the model network is obtained.

(2) Taking  $\mathbf{x}(k)$  into the action network,  $\mathbf{u}(k)$  can be obtained.

(3) Taking  $\mathbf{u}(k)$  and  $\mathbf{x}(k)$  into the model network,  $\hat{\mathbf{x}}(k+1)$  will be obtained.

(4) Taking  $\hat{\mathbf{x}}(k+1)$  into the critic network,  $\hat{\lambda}[\mathbf{x}(k+1)]$  can be obtained.

(5) Calculate the expected output value of the critic network  $\partial^+ \eta(k) / \partial \mathbf{x}(k)$ .

(6) Next, calculate the value of  $\partial^+ \eta(k) / \partial \mathbf{x}(k)$ .

(7) Next, calculate and update the weights of the critic network.

(8) Last, make  $k = k + 1$  and go back to the second step until  $\lambda(k) = \gamma * \hat{\lambda}(k+1) + \partial^+ \eta(k) / \partial \mathbf{x}(k)$ .

## 4. Simulation Analysis

**4.1. Simulation Example Analysis.** The molecular distillation technology was also called short films. When enough energy is obtained, the average free path that escapes from the surface of a liquid of light molecules differs from that of heavy molecules, which achieve the nonequilibrium liquid-liquid separation process under high vacuum conditions [26]. The molecular distillation technology has advantages of low temperature distillation, short heating time, and high separation efficiency, and it is conducive to separate the material, that is, high boiling point, heat sensitivity, and high viscosity material separation. This technology is widely used in food, medicine, oil processing, and petrochemical industry [27–29]. Molecular distillation equipment can be divided into four types: stationary, falling film, scraped film, and centrifugal type [30]. At present, wiped film molecular distillation is the most widely used technology in scientific research and industrial production. The evaporation effect of the molecular distillation system is not only related to the size and shape of the evaporator and space, the distance to the surface evaporation condensation, the manufacturing process, and other types of equipment, but also connected with the pressure within the parameters of the feed flow rate, temperature of the evaporator, scraping, and other devices running the motor speed film process parameters [31]. In order to enhance the purification effect of molecular distillation, Wang et al. found that the head wave has an effect on the separation efficiency of molecular distillation by the study of the head wave [32]. Micov et al. studied the separation factors of the wiped film molecular distillation process and established a one-dimensional mathematical model [30]. Cvengros and Tkac established a mathematical equation which can be used to calculate the one-dimensional

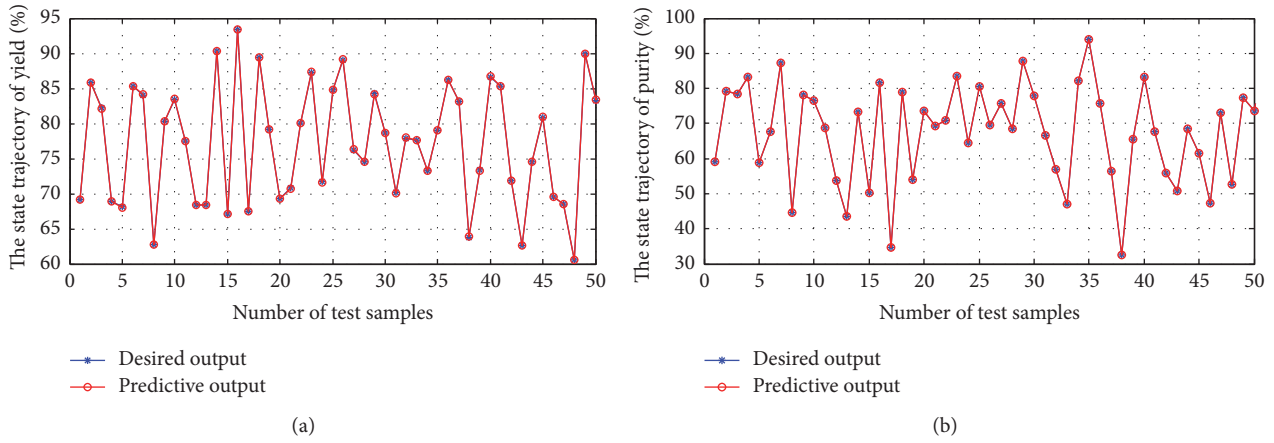


FIGURE 5: BP network model fitting ability test.

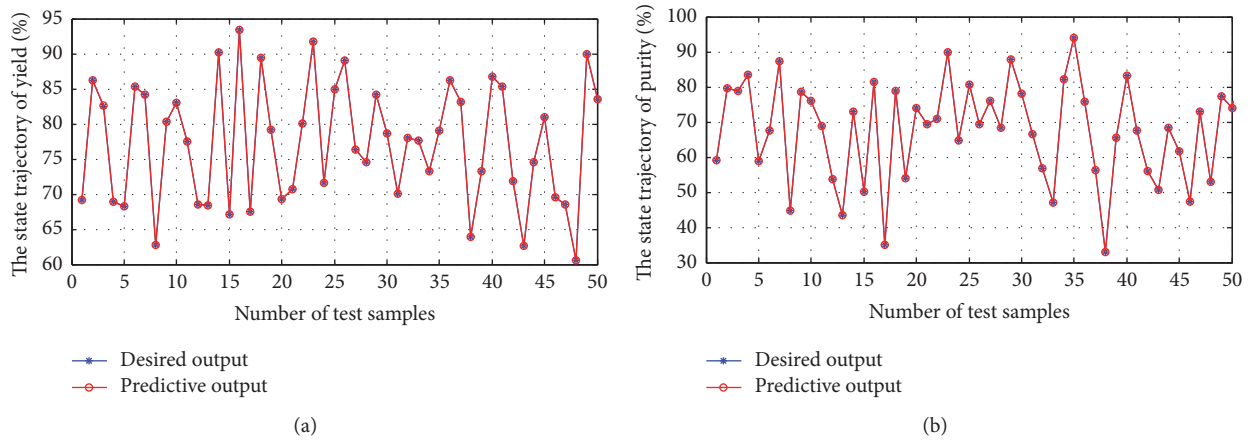


FIGURE 6: ELM network model fitting ability test.

analysis mathematical equation of micro unit movement velocity in distillation equipment through the DSMC method and summarized the effects of evaporation temperature, distance, and vacuum degree and other related factors on the separation results [33]. Wu studied the simulation of the temperature, pressure, and reflux ratio on yield and purity by using the central response surface method combined with thin film evaporation and rectification coupling technology [34]. Although much research has been made, there are still many problems in molecular distillation system with multivariability, nonlinearity, strong coupling, and large delay. Therefore, the effectiveness of the ELM-DHP algorithm was verified by controlling the scraping film molecular distillation system.

The current state variables of the molecular distillation system are determined by the amount of state variables in the preceding section of the system and the control variables in the previous stage. So, distillation temperature, evaporation pressure, wiper motor speed, feeding speed, and *Schisandra* yield and purity of the front section were used as the input of the ELM-DHP controller, and the current *Schisandra* yield and purity were used as the output of the ELM-DHP controller.

4.2. *Simulation Comparison.* The structures of the model network, critic network, and action network were set as 6-20-2, 2-14-2, and 2-5-4 through experiment, respectively. In the process of system identification, the weight values of the three networks between the input layer and the hidden layer are selected in the range  $[-0.1, 01]$ . 600 groups of data were used to study, and 150 groups of data were used as the test set. Firstly, we need to train the model network offline; the least-squares solutions were calculated as the weight matrix between the hidden layer and the output layer. Then, we complete the training of the model network and keep its weight unchanged. The 50 time steps of the model network are shown in Figures 5 and 6.

Figures 5 and 6 show that the predicted values of the BP network and the ELM algorithm are in good agreement with the expected values. Figures 7 and 8 show that the maximum error of the BP network in the prediction of the state is 0.4, but the maximum error of ELM for the state prediction is about 0.06. Thus, it can be concluded that ELM has higher prediction accuracy and better generalization ability.

Parameter setting will affect the convergence speed of the algorithm to a certain extent. After the experiment, the discount factor was chosen as  $\gamma = 0.9$ . Next, the weights of

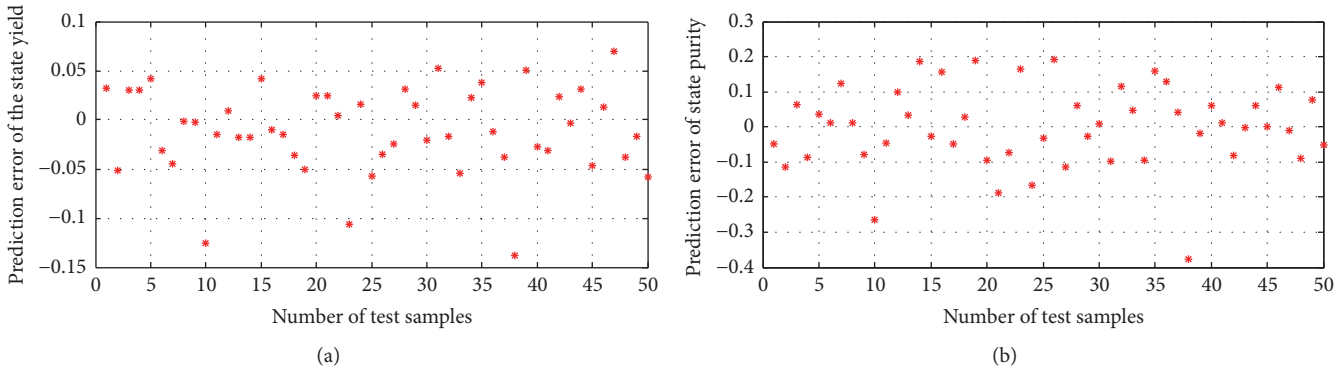


FIGURE 7: BP network model prediction error.

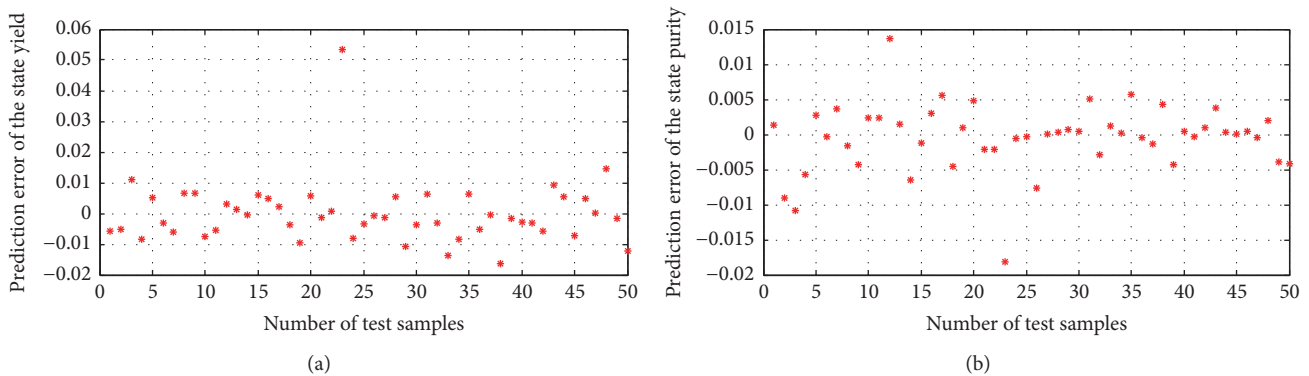


FIGURE 8: ELM network model prediction error.

the critic network and the action network from the hidden layer to the output layer are calculated. Then, the training of the critic network and action network is set to 150 steps with 100 training epochs for each step.

In addition, in order to compare with the HDP and DHP technology based on BP neural network, controllers designed by BP-HDP, BP-DHP, and ELM-DHP were proposed. Four controllers are used to control the wiped film molecular distillation system, respectively, and the 50 time steps of the simulation results are shown in Figures 9–14. Figures 9–12 show that the control quantities of BP-HDP, BP-DHP, ELM-HDP, and ELM-DHP controllers achieve stable control in 45 steps, 35 steps, 18 steps, and 7 steps individually. Thus, it can be concluded that the HDP and DHP algorithms based on ELM can achieve faster response speed. There will be a larger fluctuation when the controlled variables of the HDP controller achieve stability. So, it can be concluded that the DHP algorithm has a higher stability. The results of Figures 13 and 14 are shown in Table 1. The purification effect increases with yield and purity and the best purification effect is 100%, but it is impossible to achieve. It can be seen in Table 1 that the optimal state quantities derived by ELM-HDP and ELM-DHP were 5% higher than BP-HDP and BP-DHP, and the optimal state of ELM-DHP is slightly higher than that of ELM-HDP. In the above analysis, the superiority and effectiveness of the ELM algorithm can be demonstrated clearly.

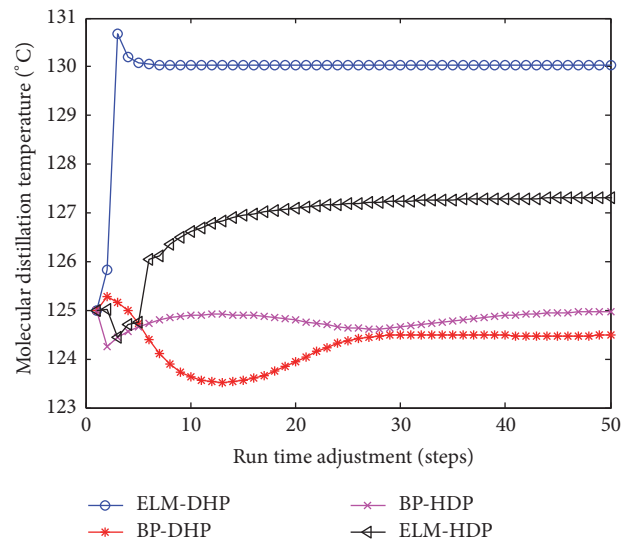


FIGURE 9: The molecular distillation temperature of optimum control quantity.

## 5. Summary

For those problems which the BP-DHP algorithm has, such as poor prediction accuracy, slow convergence speed, and poor



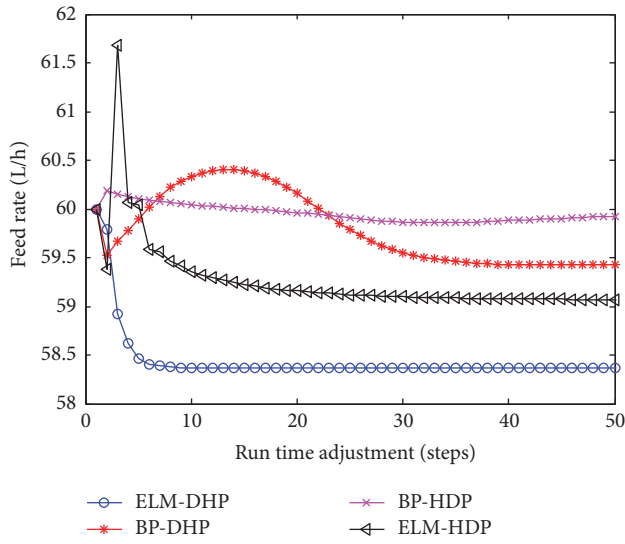


FIGURE 10: The variable feed rate of optimum control quantity.

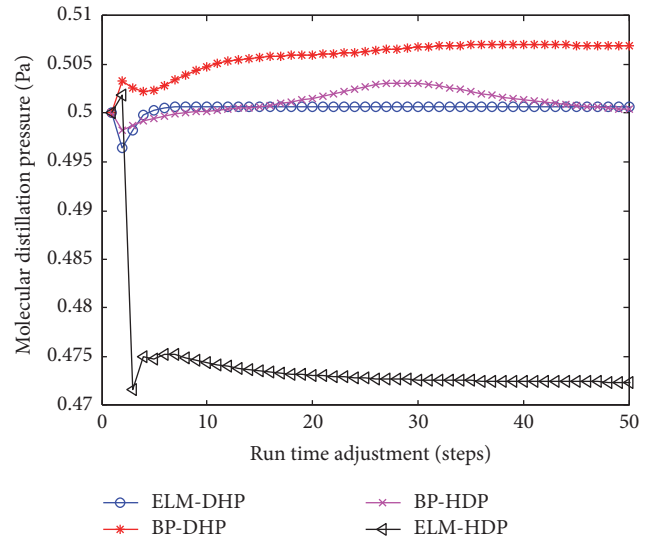


FIGURE 12: The molecular distillation pressure of optimum control quantity.

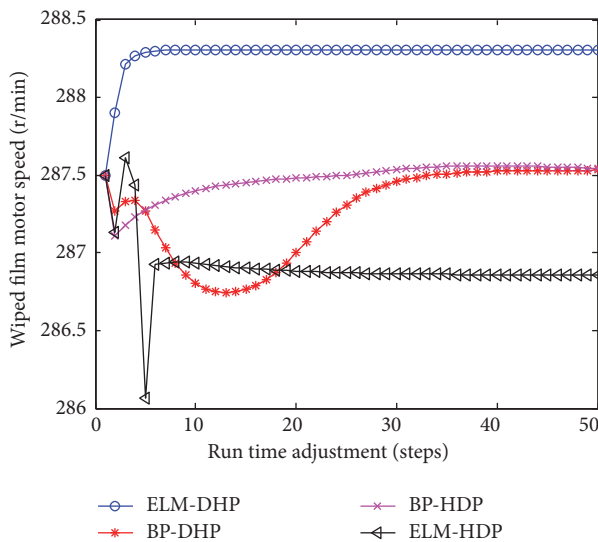


FIGURE 11: The wiped film motor speed of optimum control quantity.

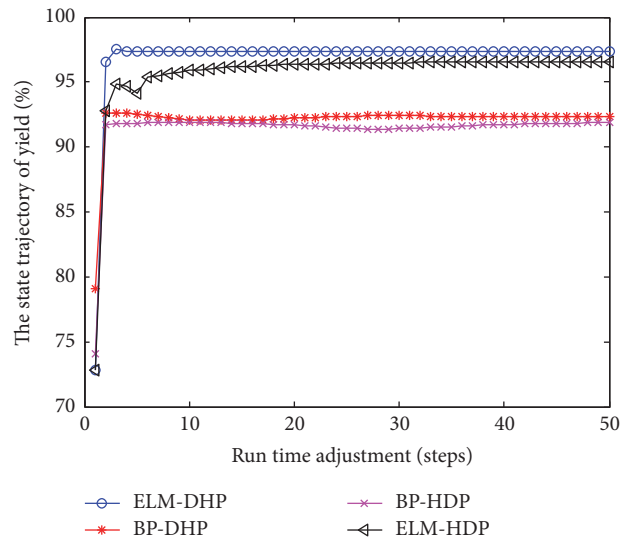


FIGURE 13: The yield of optimum state.

TABLE 1: Optimal state of controller.

	BP-HDP	BP-DHP	ELM-HDP	ELM-DHP
Yield (%)	91.95	92.39	96.58	97.35
Purity (%)	89.25	91.11	97.12	97.18

stability, the ELM-DHP algorithm was studied in this paper to solve the data modeling and optimal control problem of the wiped film molecular distillation system with complex features such as multivariability, strong coupling, nonlinearity, and large time delay as an example. The ELM-DHP controller was designed to control the molecular distillation system and a simulation verification was carried out. When compared with the ELM-HDP, BP-HDP, and BP-DHP algorithms, the

prediction accuracy of ELM is higher than that of the BP neural network, and the response speed and stability of the ELM- HDP and ELM-DHP algorithms are higher than those achieved by the BP network, which shows the superiority of ELM. Compared with other algorithms, the response speed of ELM-DHP is more than two times that of the other algorithms, and the optimal state achieved by ELM-DHP is closer to the ideal result. Thus, the ELM-DHP algorithm is better than BP-HDP, BP-DHP, and ELM-HDP algorithms. The ELM-DHP algorithm does not depend on the specific mechanism model and is only in accordance with the relevant experimental data, so the algorithm can also solve the optimal control problem of similar complex

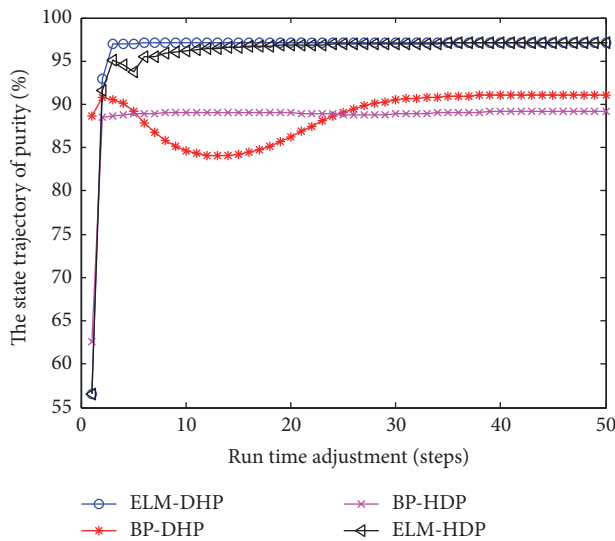


FIGURE 14: The purity of optimum state.

systems which have features such as multiple variables, strong coupling, nonlinearity, and large time delay.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61374138 (“Research on Fault Prediction and Optimal Maintenance of Complex Electromechanical System Based on Virtual Reality Technology”) by Changchun University of Technology.

## References

- [1] Y. Yang, *Research on Extreme Learning Theory for System Identification and Applications*, Hunan University, 2013.
- [2] Zh. Zhang, *Research on the Application of Neural Dynamic Programming in Temperature Control of Cement Decomposing Furnace*, Guangxi University, 2007.
- [3] T. Liu, *Neuro-controller of Cement Rotary Kiln Based on Adaptive Critic Designs*, Guangxi University, 2008.
- [4] B. Yang, *Application of Adaptive Dynamic Programming in Optimization Control of Precalciner Kiln*, Guangxi University, 2009.
- [5] Sh. Duan, *Control System of Cement Precalciner Kiln Based on Adaptive Dynamic Programming*, Guangxi University, 2010.
- [6] J. Hopfield and D. W. Tank, “Computing with neural circuits: a model,” *Science*, vol. 233, no. 4764, pp. 625–633, 1986.
- [7] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [8] G. Huang, D. H. Wang, and Y. Lan, “Extreme learning machines: a survey,” *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.
- [9] X. F. Lin, Sh. J. Song, and Ch. N. Song, *Intelligent Optimal Control based on Adaptive Dynamic Programming*, Science Press Engineering Branch, Beijing, China, 2013.
- [10] B. G. Ma, *New Dry Process Cement Production Process*, Chemical Industry Press, Beijing, China, 2007.
- [11] P. J. Werbos, “Intelligence in the brain: A theory of how it works and how to build it,” *Neural Networks*, vol. 22, no. 3, pp. 200–212, 2009.
- [12] F. L. Lewis, D. Vrabie, and K. . Vamvoudakis, “Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers,” *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.
- [13] H. G. Zhang, X. Zhang, Y. H. Luo, and J. Yang, “An overview of researches on adaptive dynamic programming,” *Acta Automatica Sinica*, vol. 39, no. 4, pp. 303–311, 2013.
- [14] D. R. Liu, H. L. Li, and D. Wang, “Data-based self-learning optimal control: research progress and prospects,” *Acta Automatica Sinica*, vol. 39, no. 11, pp. 1858–1870, 2013.
- [15] Z.-S. Hou and Z. Wang, “From model-based control to data-driven control: survey, classification and perspective,” *Information Sciences. An International Journal*, vol. 235, pp. 3–35, 2013.
- [16] J. Si and Y.-T. Wang, “On-line learning control by association and reinforcement,” *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 264–276, 2001.
- [17] X. Zhong, Z. Ni, and H. He, “A theoretical foundation of goal representation heuristic dynamic programming,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2513–2525, 2015.
- [18] H. Zhang, Y. Luo, and D. Liu, “Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints,” *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1490–1503, 2009.
- [19] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, “Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming,” *Automatica*, vol. 48, no. 8, pp. 1825–1832, 2012.
- [20] P. J. Werbos, “ADP: goals, opportunities and principles,” in *Handbook of Learning and Approximate Dynamic Programming*, J. Si, A. G. Barto, W. B. Powell, and, and D. Wunisch, Eds., pp. 3–44, Wiley-IEEE Press, Hoboken, NJ, USA, 2004.
- [21] W.-S. Lin and J.-W. Sheu, “Automatic train regulation for metro lines using dual heuristic dynamic programming,” *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 224, no. 1, pp. 15–23, 2010.
- [22] J.-W. Sheu and W.-S. Lin, “Energy-saving automatic train regulation using dual heuristic programming,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1503–1514, 2012.
- [23] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks,” in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, July 2004.
- [24] P. L. Bartlett, “The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network,” *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.
- [25] G.-B. Huang, “Learning capability and storage capacity of two-hidden-layer feedforward networks,” *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 274–281, 2003.
- [26] Zh. Tao, *Study on Purifying Three Natural Products by Molecular Distillation and Theoretical Model*, Tianjin University, 2004.

- [27] M. Bhandarkar and J. R. Ferron, "Transport processes in thin liquid films during high-vacuum distillation," *Industrial and Engineering Chemistry Research*, vol. 27, no. 6, pp. 1016–1024, 1988.
- [28] S. Komori, K. Takata, and Y. Murakami, "Flow structure and mixing mechanism in an agitated thin-film evaporator," *Journal of Chemical Engineering of Japan*, vol. 21, no. 6, pp. 639–644, 1988.
- [29] S. Komori, K. Takata, and Y. Murakami, "Flow and mixing characteristics in an agitated thin-film evaporator with vertically aligned blades," *Journal of Chemical Engineering of Japan*, vol. 22, no. 4, pp. 346–351, 1989.
- [30] M. Micov, J. Lutišan, and J. Cvengroš, "Balance equations for molecular distillation," *Separation Science and Technology*, vol. 32, no. 18, pp. 3051–3066, 1997.
- [31] M. James, G. Makelvey, and V. S. Jr, "Fluid transport in thin film polymer processors," *Polymer Engineering & Science*, vol. 19, no. 9, pp. 652–659, 1979.
- [32] H. Wang, J. P. Liu, and Ch. Ma, "Molecular distillation technology and its application," *Chemical Technology and Development*, vol. 26, no. 13, pp. 23–25, 2008.
- [33] J. Cvengros and A. Tkac, "Continuous processes in wiped films," *Industrial & Engineering Chemistry Process Design and Development*, vol. 17, no. 3, pp. 246–251, 1978.
- [34] H. B. Wu, "Study on the RTD in the wiped film molecular distillation equipment by CFD simulation method," *Journal of Anhui Normal University*, vol. 36, no. 2, pp. 141–145, 2013.



# Hindawi

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

