Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Design of extended Kalman filtering neural network control system based on particle swarm identification of nonlinear U-model

Fengxia Xu, Xinyu Zhang, Zhongda Lu & Shanshan Wang

To cite this article: Fengxia Xu, Xinyu Zhang, Zhongda Lu & Shanshan Wang (2022) Design of extended Kalman filtering neural network control system based on particle swarm identification of nonlinear U-model, Automatika, 63:3, 463-473, DOI: 10.1080/00051144.2022.2052398

To link to this article: https://doi.org/10.1080/00051144.2022.2052398

Published online: 18 Mar 2022.

AUTOMATIKA
2022, VOL. 63, NO. 3, 463–473
https://doi.org/10.1080/00051144.2022.2052398

Design of extended Kalman filtering neural network control system based on particle swarm identification of nonlinear U-model

Fengxia Xu[a,b], Xinyu Zhang[c], Zhongda Lu[a,b] and Shanshan Wang[c]

[a]College of Mechanical and Electrical Engineering, Qiqihar University, Qiqihar, People's Republic of China; [b]Collaborative Innovation Center of Intelligent Manufacturing Equipment Industrialization of Heilongjiang Province, Qiqihar, People's Republic of China; [c]College of Computer and Control Engineering, Qiqihar University, Qiqihar, People's Republic of China

ABSTRACT
This paper studies the modelling of a class of nonlinear plants with known structures but unknown parameters and proposes a general nonlinear U-model expression. The particle swarm optimization algorithm is used to identify the time-varying parameters of the nonlinear U-model online, which solves the identification problem of the nonlinear U-model system. Newton iterative algorithm is used for nonlinear model transformation. Extended Kalman filter (EKF) is used as the learning algorithm of radial basis function (RBF) neural network to solve the interference problem in a nonlinear system. After determining the number of network nodes in the neural network, EKF can simultaneously determine the network threshold and weight matrix, use the online learning ability of the neural network, adjust the network parameters, make the system output track the ideal output, and improve the convergence speed and anti-noise capability of the system. Finally, simulation examples are used to verify the identification effect of the particle swarm identification algorithm based on the U-model and the effectiveness of the extended Kalman filtering neural network control system based on particle swarm identification.

ARTICLE HISTORY
Received 21 August 2021
Accepted 7 March 2022

KEYWORDS
U-model; particle swarm identification; extended Kalman filtering; neural network control

Nomenclature

$y(t)$	the actual output of the nonlinear controlled plant
M	the order of the nonlinear controlled plant
$\alpha_j(t)$	the time-varying coefficient of the controlled plant of the nonlinear system
$u(t\text{-}1)$	the control input of the nonlinear controlled plant
$e(t)$	the error caused by uncertainty factors of nonlinear system
x_i	the position vector of the i-th particle
v_i	the velocity vector of the i-th particle
p_i	the individual extreme value of the population
p_g	the global extreme value of the population
ω	the search step of the population
c_1/c_2	the learning rates of the population
r_1/r_2	mutually opposite pseudo-random numbers
m	Number of particle swarm populations
n	Search dimensions for each particle
k	the number of iterations
$U(t)$	the controller output of the nonlinear U-model system and is also the input of newton iterative algorithm
θ	the state parameter
W	the weight parameters parameters of the neural network
V	the threshold parameters of the neural network
l	the number of hidden nodes in the neural network
$\Delta(t)$	the control interference
$g(\theta(t))$	the actual output of the RBF network

1. Introduction

In the actual production process, nonlinear characteristics are ubiquitous, especially with the rapid development of large machinery informatization, intelligence, and integration; the research of nonlinear systems has become particularly important. With the deepening of the study on the nonlinear theory, researchers have achieved good results, and many nonlinear research results have been applied to actual production. At the same time, due to the continuous improvement of the production process and the higher precision of the process demands, the nonlinear characteristics also show an increasingly complex trend. However, the design of a nonlinear control system with good performance requires a precise nonlinear model. The nonlinear system's increasing complexity limits the production process's precision to a certain extent, which prompts more researchers to study the model and control problems of the nonlinear system. Due to the ubiquity of nonlinear

CONTACT Fengxia Xu xufengxia_hit@163.com Qiqihar University, No.42 wenhua Street, Jianhua District, Qiqihar City, Heilongjiang Province, China

characteristics and the inherent complexity of nonlinear time-varying systems, it isn't easy to establish a general and high-precision mathematical model, which is also the premise and foundation for designing a nonlinear control system. A nonlinear U-model is proposed based on the nonlinear autoregressive moving average model [1,2]. In [3], a dynamic inversion algorithm of the U model was designed based on the U model of continuous-time (CT) system described by polynomials and state space. When using U-model to represent the nonlinear model, there is no loss of model accuracy, which improves the accuracy of nonlinear plant modelling. For complex nonlinear systems, there are many factors that affect the modelling, which leads to a large difference between the ideal model of the system and the actual model. Therefore, model identification becomes a method to improve the modelling accuracy. In [4], the least square method was used to identify the U-model coefficients of stochastic nonlinear plant, and the radial basis neural network was used as the controller. Considering that the number of U-model coefficients of random plant is known, a minimum performance index was proposed, and a variable learning operator was constructed by using the least mean square deviation method to update the coefficients of U-model and the weights of radial basis neural network. Assuming that all parameters of the U-model were completely unknown, the unknown plant was identified by the weighted iterative least square method using the U-model framework, the time-varying coefficients of the proposed U-model expression were identified, and the convergence of online identification of time-varying parameters was proved [5]. In [6], the U-model was used to express the unknown MIMO bilinear system model, and the radial basis function neural network was used to identify the MIMO bilinear plant model online. In [7], a coupling multivariable underactuated nonlinear adaptive U-model control method is designed. In [8–17], for the nonlinear plant with an unknown model structure, the neural network model was used to approximate the nonlinear plant, and the structure of the plant is identified online using the adaptive filter. In [18], a U-model-based two-DOF internal model control (UTDF-IMC) structure is proposed to accommodate modelling errors and disturbances while eliminating linearization techniques widely used in nonlinear models. Other identification methods of nonlinear plant based on U model are still worth exploring.

The neural network has good generalization ability, and the network structure is simple, which can avoid complex mathematical calculations. Research on the function approximation capability of RBF neural networks shows that RBF neural networks can approach any nonlinear function with any precision. Therefore, RBF neural network has attracted much attention in nonlinear control research, and some research results have been obtained. With the known order of nonlinear U-model plant, Chang et al. [19] used the least-squares algorithm to identify unknown parameters of nonlinear U-model plant and designed a radial basis function neural network (RBFNN) online controller. In [20], based on a pole placement PID controller designed, a composite control method of RBF and PD was proposed for the tracking control of nonlinear dynamic systems. The proposed scheme combined the stability of PD and the ability of the RBF to approximate any function with any precision and combined the control-oriented nature of the U-model to achieve accurate tracking of nonlinear plants. In order to control the pitch angle of wind turbine blades in the rated power area, the controller of the RBF network was proposed in [21]. Aiming at the stability problem of nonlinear systems, a neural fuzzy hybrid controller based on radial basis function network (HNFRBFN) was proposed in [22]. The paper [23] proposed a model reference adaptive speed controller based on an artificial neural network for induction motor drives, where RBF is utilized to compensate for the unknown nonlinearity in the control system adaptively. In [24], an adaptive algorithm of radial basis function neural network is designed. By automatically updating the weights and network parameters and adjusting the adaptive controller, the output of the controlled nonlinear object can be completely tracked to the ideal output. These control methods used traditional gradient descent backpropagation learning algorithms for network training. In the learning algorithm based on the gradient descent method, the neural network weights were updated in the negative gradient of the error cost function to achieve the global minimum. Although the design process of this method was simple, it made the network easy to fall into local minimum value and slow convergence speed [25]. In order to solve this problem, some improved methods were proposed, including modified learning rates and momentum factors [26]. In [27], a universal U neural network structure is proposed to facilitate the design and control of all dynamic systems modelled by nonlinear polynomial equations. But in many cases, these improved methods add computational complexity. When there is noise interference in the control system, it will reduce the network's learning ability. Therefore, this paper uses the extended Kalman filtering algorithm to improve the neural network learning algorithm.

In the case of slow time variation, due to the multi-objective optimization ability of particle swarm optimization algorithm and its fast convergence ability, for the nonlinear plants with known structure and unknown parameters, this paper uses the particle swarm identification algorithm to identify the time-varying parameters of the U-model online and solve the identification problem of nonlinear U-model system. Considering the influence of control interference on the system, in order to improve the learning ability of the neural network, an extended Kalman filter neural network control system is proposed. The extended Kalman filtering algorithm is used as the learning algorithm of the RBF neural network, and the weights and parameters of the neural network algorithm are adjusted online to complete the design of the nonlinear control system.

The rest of this article is organized as follows. The second section mainly discusses the establishment of the nonlinear U model and the basic knowledge of particle swarm optimization algorithm. In Section 3, the overall structure of the nonlinear U model control system is studied. In Section 4, the online identification of time-varying parameters of the U model is presented. In Section 5, the U model is transformed. In Section 6, we discuss using EKFRBF to deal with models that do not match Newtonian iterations. In Section 7, we simulate the optimized U model. Finally, the thesis is summarized.

2. Basic knowledge

2.1. Nonlinear U-model

In 2002, the concept of nonlinear U-model was proposed in [2]. It establishes a general mapping, which can represent a large class of nonlinear controlled plants. Its expression is as follows:

$$y(t) = \sum_{j=0}^{M} \alpha_j(t) u^j(t-1) + e(t) \tag{1}$$

In Eqaution (1), $y(t)$ represents the actual output of the nonlinear controlled plant, M represents the order of the nonlinear controlled plant, $\alpha_j(t)$ represents the time-varying coefficient of the controlled plant of the nonlinear system, which is a function of $y(t-1)$, $y(t-2),\cdots,y(t-n)$, $u(t-2),\cdots,u(t-n)$ and $e(t-1)$, $e(t-2),\cdots,e(t-n)$. $u(t-1)$ represents the control input of the nonlinear controlled plant. $e(t)$ represents the error caused by uncertainty factors of nonlinear system, such as modelling error and external interference.

2.2. Particle swarm optimization algorithm

Particle swarm optimization is a swarm intelligence optimization algorithm based on biological population simulation. It has the characteristics of simple structure and easy implementation. The particle swarm optimization algorithm expresses each possible solution as a particle in the population. Each particle has a corresponding position vector and velocity vector. The fitness of each particle is determined according to the plantive function. The optimal global value is found by updating the current optimal value. Assume in an n-dimensional search space. In a population of m particles of an n-dimensional search

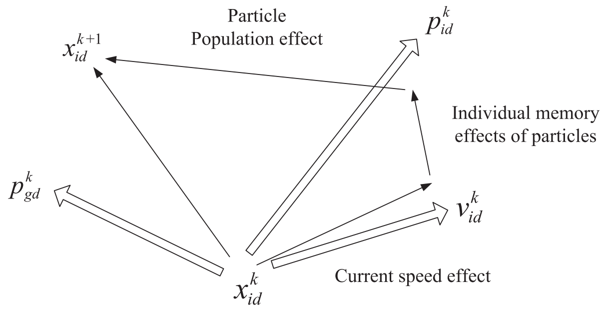

Figure 1. Schematic of the particle positions updated.

space, where the position vector of the i-th particle is $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})$, its velocity vector is $\upsilon_i = (\upsilon_{i1}, \upsilon_{i2}, \cdots, \upsilon_{in})$, the individual extreme value of the population is $p_i = (p_{i1}, p_{i2}, \cdots, p_{in})$, and the global extreme value is $p_g = (p_{g1}, p_{g2}, \cdots, p_{gn})$, the updated formula of the individual position vector and velocity vector of the population is as follows:

$$\begin{cases} \upsilon_{id}^{k+1} = \omega\upsilon_{id}^{k} + c_1 r_1 \left(p_{id}^{k} - x_{id}^{k}\right) \\ \qquad + c_2 r_2 \left(p_{gd}^{k} - x_{id}^{k}\right) \\ x_{id}^{k+1} = x_{id}^{k} + \upsilon_{id}^{k} \end{cases} , d = 1, \ldots, n \tag{2}$$

where, ω represents the search step of the population, c_1 and c_2 represent the learning rates of the population, r_1 and r_2 are mutually opposite pseudo-random numbers, subject to the uniform distribution on the interval [0,1], $\upsilon \in [-\upsilon_{\max}, \upsilon_{\max}]$, and $\upsilon_{\max}$ is the normal number.

The initial position vector x_{id}^{k} and velocity vector υ_{id}^{k} of each particle is updated according to the current individual velocity, memory, and population influence of each particle, the current optimal position and global optimal position of each particle, and the position vector and velocity vector of each particle at the next moment are updated.

In summary, the updating method of particle position is shown in Figure 1.

3. Control structure of nonlinear U-model

The nonlinear control system based on U-model is designed. Particle swarm optimization algorithm is used to identify time varying parameters of nonlinear U-model. The model transformation is completed by newton iteration algorithm. The neural network is used as the controller of the nonlinear U-model system. The extended Kalman filtering algorithm is used to optimize the neural network controller to ensure the stability of the system and improve the disturbance rejection performance of the system. The extended Kalman filtering neural network control system based on particle swarm identification is shown in Figure 2.

When the input error exists due to various factors, a particle swarm optimization algorithm is used to find the optimal time-varying parameters to reduce

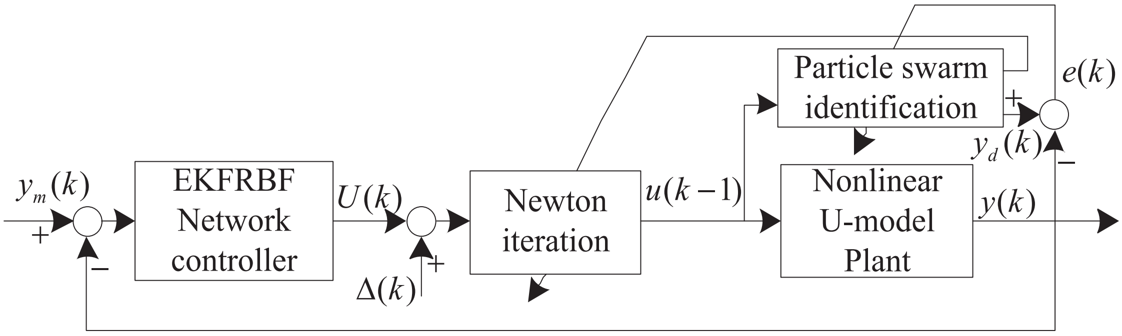

Figure 2. Diagram of extended Kalman filtering neural network control system based on particle swarm identification.

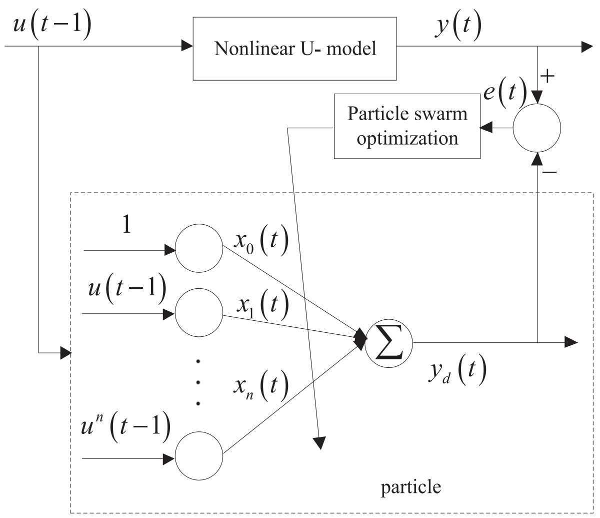

Figure 3. Structure diagram of identification of U-model by particle swarm optimization algorithm.

the error. The particle swarm optimization algorithm needs to update the nonlinear object model to complete the system control, so it has certain limitations on the tracking speed of the nonlinear system based on the U-model, which leads to a certain tracking error of the system. Therefore, Newton iterative algorithm is used to transform the nonlinear U-model. There may be a deviation between U-model and Newton iterative algorithm. The output of the controlled object is adaptively tracked to the expected output of the nonlinear U-model by EKFRBF neural network.

4. Particle swarm algorithm identification

Since the particle swarm optimization algorithm has multi-plantive optimization and fast convergence, the particle swarm optimization algorithm can be used to identify the time-varying parameters of the nonlinear U-model, and the i-th time-varying parameter in the U-model is regarded as the i-th particle in the population. The structure is shown in Figure 3.

The specific steps of identifying the time-varying parameters of U-model by particle swarm optimization algorithm are as follows:

(1) Establish a population of m particles, initialize the position vector and velocity vector of each particle;
(2) The position vector of each particle is taken as the time-varying parameter of the U-model, and then the fitness value of each particle is calculated according to the fitness formula $fitness = \frac{1}{2}e^2(t)$;
(3) Update the current optimal position p_i and global optimal position p_g of each particle;
(4) Update the position vector and velocity vector of each particle according to (2);
(5) Determine whether the number of iterations reaches the set value, and if the termination condition is met, the algorithm is terminated; Otherwise, return step (2).

In summary, the process of using the particle swarm optimization algorithm to identify the time-varying parameters of the U-model is shown in Figure 4.

5. Nonlinear model transformation

Newton iterative algorithm can be used to solve the polynomial (1), which provides a transformation method for the nonlinear U-model, and can improve the response speed and control accuracy of the nonlinear system. The output of the newton iteration formula is $u(t-1)$. The newton iteration formula can be described as:

$$u_{k+1}(t-1) = u_k(t-1) - \left. \frac{\sum_{j=0}^{M} \alpha_j(t) u_k^j(t-1) - U(t)}{d\left[\sum_{j=0}^{M} \alpha_j(t) u_k^j(t-1)\right] / du(t-1)} \right|_{u^j(t-1)=u_k^j(t-1)} \tag{3}$$

In Equation (3), k is the number of iterations, the iteration of $k+1$ times is obtained from k iterations, $k \geq 0$. $U(t)$ represents the controller output of the nonlinear U-model system and is also the input of newton iterative algorithm. The input of nonlinear U-model plant can be calculated by newton iterative algorithm (3).

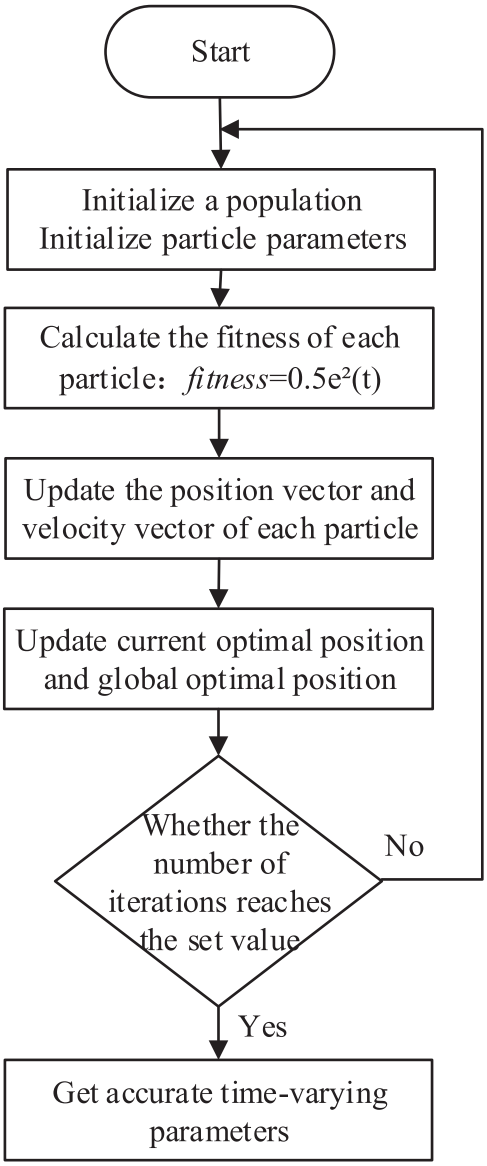

Figure 4. Flow chart of particle swarm optimization algorithm for identifying U-mode.

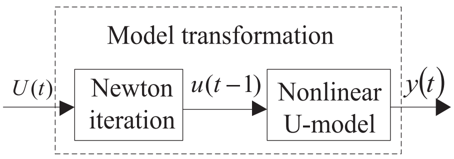

Figure 5. Diagram of nonlinear model transformation.

Newton iterative algorithm is used to transform the nonlinear U-model, and its transformation structure is shown in Figure 5.

It can be seen from Equations (1) and (3) that the newton iteration formula and the U-model are inverse functions of each other. When the iterative algorithm is completely inversed with the U-model, the model transformation of system is matched, which is equivalent to cancelling the nonlinear part, $y(t) = U(t)$, and the system output can completely track the input; When the iterative algorithm is not completely inversed with the U-model or the calculation order of the iterative algorithm is limited, the system model transformation has deviation. In the actual nonlinear system, the U-model cannot completely describe the nonlinear plant completely. And it is difficult to completely match the U-model with the newton iteration formula. Therefore, it is necessary to design the controller to complete the control requirements.

6. Design of EKFRBF network control system

The extended Kalman filtering algorithm is used as the learning algorithm of neural network to optimize the neural weights and threshold parameters. The state space equation obtained by combining RBF neural network with extended Kalman filtering expression is as follows:

$$\begin{aligned}\theta(t+1) &= \theta(t)\\ U(t) &= g(\theta(t)) + \Delta(t)\end{aligned} \tag{4}$$

In Equation (4), $\theta = \begin{bmatrix} W & V \end{bmatrix}^T$ is the state parameter, W and V are the weight parameters and threshold parameters of the neural network, respectively. $W = [w_1\ w_2\ \cdots\ w_l]$, $V = \begin{bmatrix} v_1(t) & \cdots & v_j(t) & \cdots & v_l(t) \end{bmatrix}$. l is the number of hidden nodes in the neural network. $\Delta(t)$ is the control interference, which is white noise with constant variance. $g(\theta(t))$ is the actual output of the RBF network. $U(t)$ is the output of the RBF neural network controller with disturbance, and is also the input of newton iterative algorithm.

The hidden layer output of the RBF network is:

$$h_j(t) = [||y_m(t) - v_j(t)||^2 + \gamma^2]^{1/1-p} \tag{5}$$

In Equation (5), γ and p are constants, $H = \begin{bmatrix} h_1(t) & \cdots & h_j(t) & \cdots & h_l(t) \end{bmatrix}$.

The output of the RBF neural network controller, ie the input of the newton iteration algorithm, is:

$$U(t) = W * H^T + \Delta(t) \tag{6}$$

The partial derivative of $U(t)$ is:

$$G = \frac{\partial U(t)}{\partial w(t)} + \frac{\partial U(t)}{\partial v(t)} = \begin{bmatrix} G_w \\ G_v \end{bmatrix} \tag{7}$$

where, $\begin{cases} G_w = \begin{bmatrix} h_1(t) & \cdots & h_j(t) & \cdots & h_l(t) \end{bmatrix}^T \\ G_v = \begin{bmatrix} -2\frac{p}{1-p} w_1(t) h_j(y_m(t) - v_1(t)) \cdots \\ -2\frac{p}{1-p} w_l(t) h_l(y_m(t) - v_l(t)) \end{bmatrix}^T \end{cases}$

The Kalman gain is:

$$K = P * G * (G^T * P * G)^{-1} \tag{8}$$

The error covariance matrix:

$$P = P - K * G^T * P + Q \tag{9}$$

The update algorithm of neural network weights and threshold parameters is:

$$\theta = \theta + K * (y_m(t) - U(t)) \tag{10}$$

The neural network is trained by the extended Kalman filtering algorithm, and the network parameters are adjusted. The controlled plant output is adaptively tracked on the desired output of the nonlinear U-model system.

7. Simulation

Through two nonlinear plants, the identification effect of particle swarm identification algorithm on nonlinear U-model plants is verified first, and then the validity of extended Kalman filtering neural network control system based on particle swarm identification is verified when the nonlinear U-model system has control interference.

7.1. Simulation of particle swarm identification

The laboratory liquid level system and continuous stirred tank reactor were selected as nonlinear plants to verify the effectiveness of particle swarm optimization algorithm in identifying nonlinear U-model.

Particle swarm optimization algorithm parameters [28]: $m = 20$, $n = 1$, $\omega = 0.5$, $c_1 = c_2 = 1$, The initialization values of position vector and velocity vector are random numbers, both of which follow normal distribution, and the maximum number of cycles is 30.

Simulation 1: the U-model of the laboratory level system is as follows:

$$y(t) = \alpha_0(t) + \alpha_1(t)u(t-1)$$

where,

$$\begin{aligned}
\alpha_0(t) = {} & 0.9722y(t-1) - 0.04288y^2(t-2) \\
& + 0.1663y(t-2)u(t-2) \\
& + 0.2573y(t-2)e(t-1) \\
& - 0.03259y^2(t-1)y(t-2) \\
& - 0.3513y^2(t-1)u(t-2) \\
& + 0.3084y(t-1)y(t-2)u(t-2) \\
& + 0.2939y^2(t-2)e(t-1) - 0.1295u(t-2) \\
& + 0.6389u^2(t-2)e(t-1) \\
\alpha_1(t) = {} & 0.3578 - 0.3103y(t-1) \\
& + 0.1087y(t-2)u(t-2) \\
& + 0.4770y(t-2)e(t-1)
\end{aligned}$$

The triangular and sinusoidal waves are selected as input signals for simulation, and the simulation results of identifying unknown nonlinear U-model plant using particle swarm optimization algorithm are shown in Figures 6–9.

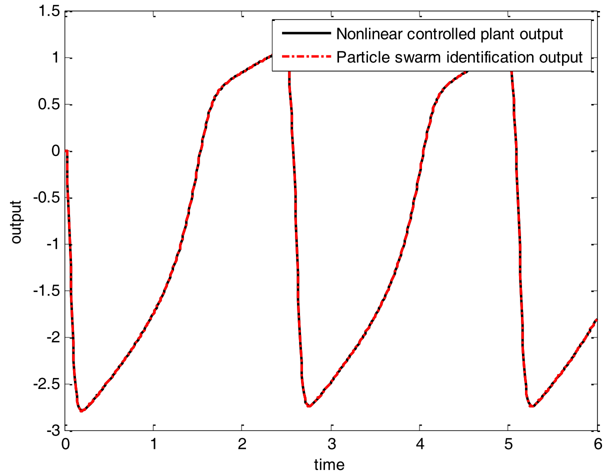

Figure 6. Output graph of particle group identification.

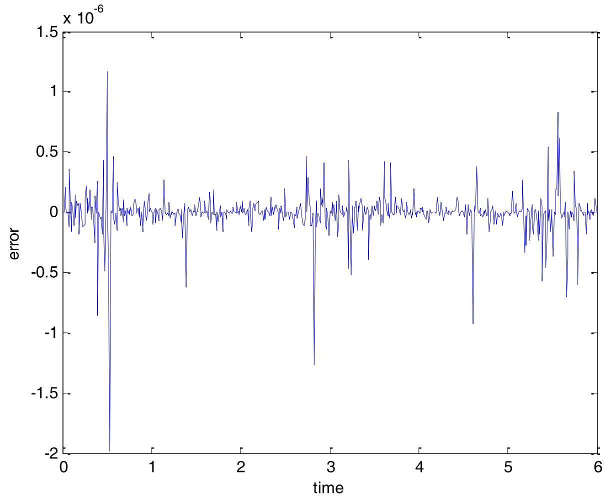

Figure 7. Error graph of particle swarm identification.

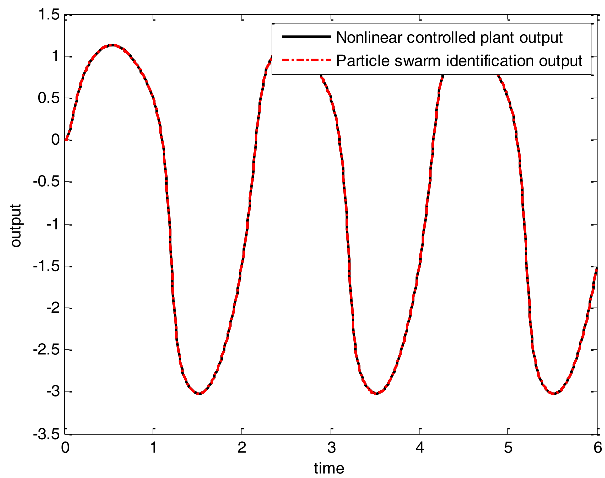

Figure 8. Output graph of particle group identification.

Simulation 2: the U-model of the continuous stirred tank reactor can be expressed as:

$$\begin{aligned}
y(t) = {} & \alpha_0(t) + \alpha_1(t)u(t-1) \\
& + \alpha_2(t)u^2(t-1) + \alpha_3(t)u^3(t-1)
\end{aligned}$$

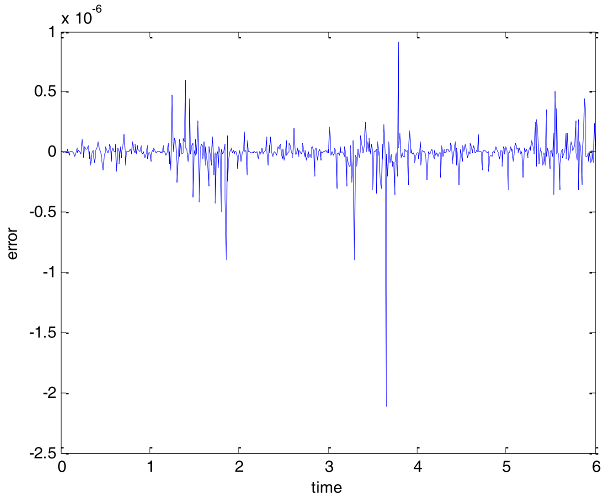

Figure 9. Error graph of particle swarm identification.

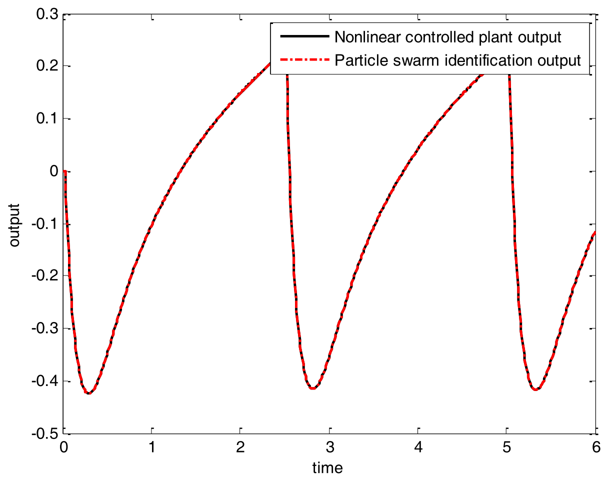

Figure 10. Output graph of particle group identification.

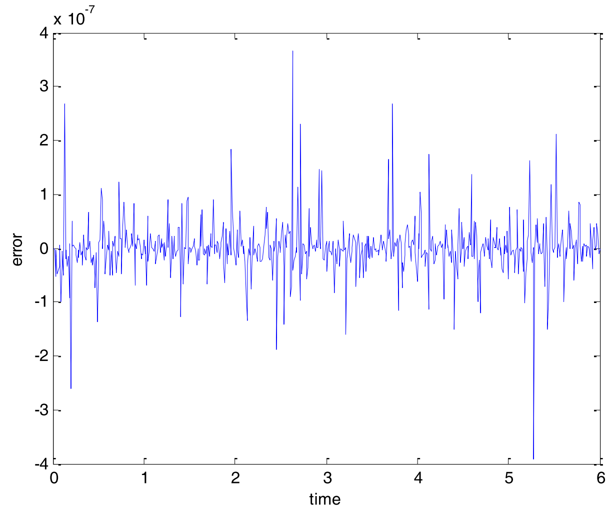

Figure 11. Error graph of particle swarm identification.

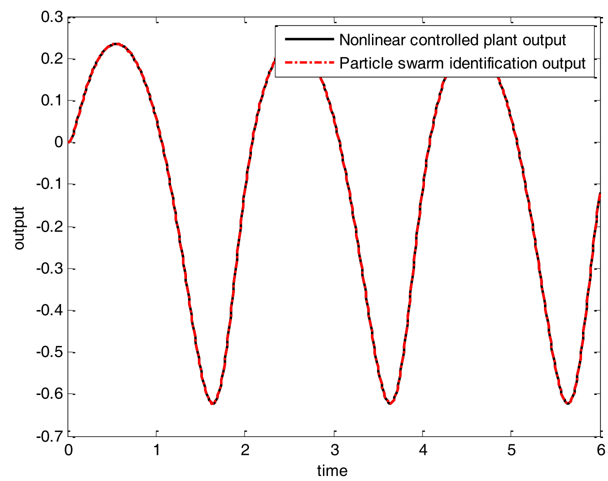

Figure 12. Output graph of particle group identification.

where,

$$\begin{aligned}\alpha_0(t) &= 0.8606y(t-1) - 0.0401y^2(t-1)\\ &\quad + 0.0017y^3(t-1) - 0.000125y^4(t-1)\\ \alpha_1(t) &= 0.0464 - 0.045y(t-1)\\ &\quad + 0.0034y^2(t-1) - 0.00025y^3(t-1)\\ \alpha_2(t) &= -0.0012 + 0.0013y(t-1)\\ &\quad - 0.0001458y^2(t-1)\\ \alpha_3(t) &= 0.00002083 - 0.00002083y(t-1)\end{aligned}$$

The triangular and sinusoidal waves are selected as input signals for simulation, and the simulation results of identifying unknown nonlinear U-model plant using particle swarm optimization algorithm are shown in Figures 10–13.

The relation between the iterative relation of particle swarm and fitness function is shown in Figure 14:

It can be seen from the output graph of particle group identification that when the triangular wave and the sinusoidal wave are used as inputs, the output of particle swarm identification algorithm can track the output of the nonlinear U-model plant well; As can be seen from the error graph of particle swarm identification, particle swarm algorithm has little error in identifying nonlinear U model objects, and the identification accuracy is relatively high.

7.2. Simulation of neural network control system

Laboratory level system and continuous stirred tank reactor are used as nonlinear controlled plants to verify the effectiveness of extended Kalman filtering neural network control system based on particle swarm identification.

The structure of RBF neural network adopts 1-7-1, 90% of the dataset as a training set, 10% as a test set and the initial values of neural network weights and threshold parameters are all 0. $Q = 0.1$, $\gamma = 1$, $p = 3$[26]. Particle swarm optimization algorithm parameters: $m = 20$, $n = 1$, $\omega = 0.5$, $c_1 = c_2 = 1$, The initialization values of position vector and velocity vector are random Numbers, both of which follow normal distribution, and the maximum number of cycles is 30.

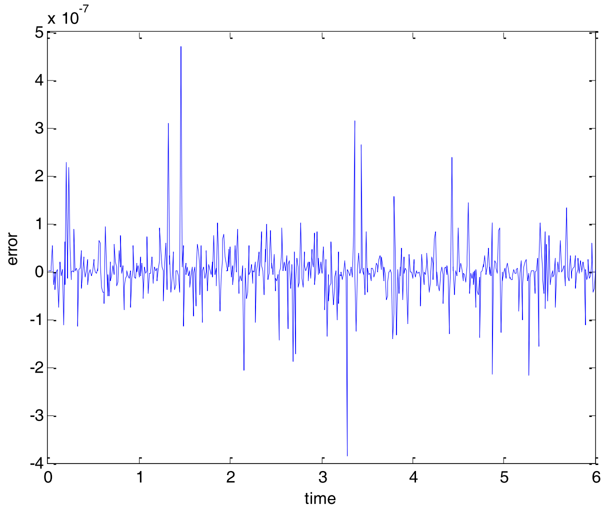

Figure 13. Error graph of particle swarm identification.

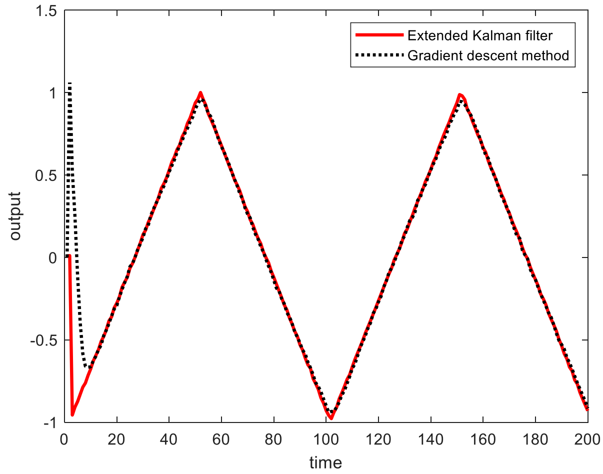

Figure 15. System output response under triangular wave input.

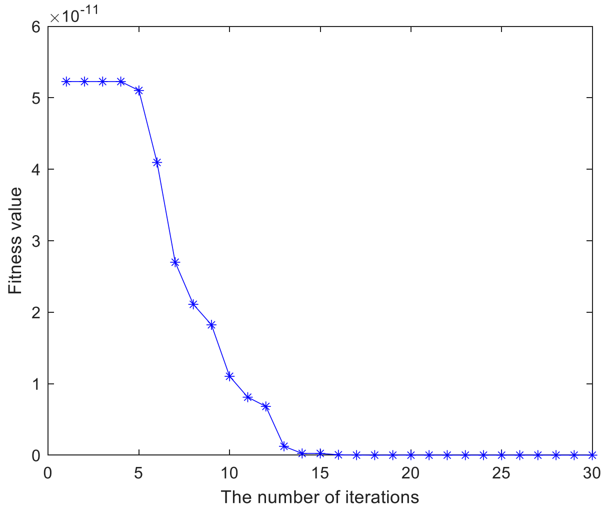

Figure 14. Particle swarm fitness.

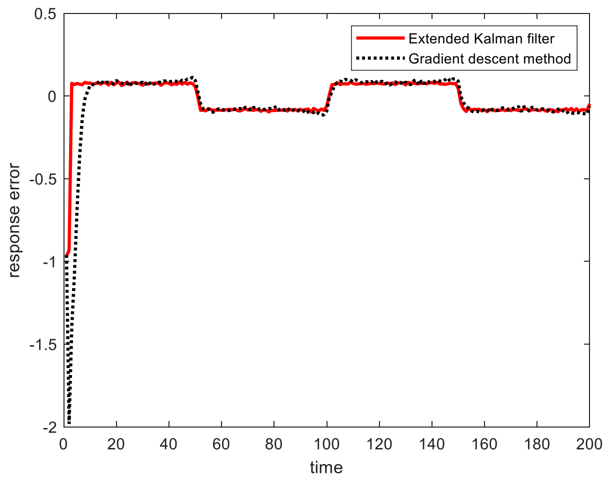

Figure 16. System response error under triangular wave input.

The control interference $\Delta(t)$ is a random interference signal with an amplitude of 0.01.

Simulation 1: The U-model expression of the laboratory level system is:

$$y(t) = \alpha_0(t) + \alpha_1(t)u(t-1)$$

where,

$$\begin{aligned}\alpha_0(t) = {} & 0.9722y(t-1) - 0.04288y^2(t-2) \\ & + 0.1663y(t-2)u(t-2) + 0.2573y(t-2) \\ & * e(t-1) - 0.03259y^2(t-1)y(t-2) \\ & - 0.3513y^2(t-1)u(t-2) + 0.3084y \\ & * (t-1)y(t-2)u(t-2) \\ & + 0.2939y^2(t-2)e(t-1) - 0.1295u(t-2) \\ & + 0.6389u^2(t-2)e(t-1) \\ \alpha_1(t) = {} & 0.3578 - 0.3103y(t-1) + 0.1087y \\ & * (t-2)u(t-2) + 0.4770y(t-2)e(t-1)\end{aligned}$$

Under the influence of disturbance, the input signal selects the triangle wave and the sine wave respectively for simulation. The simulation results of the system output and system response error are shown in Figures 15–18. Figures 15 and 17 are system output responses, and Figures 16 and 18 are system response errors.

Simulation 2: the U-model of the continuous stirred tank reactor can be expressed as:

$$\begin{aligned}y(t) = {} & \alpha_0(t) + \alpha_1(t)u(t-1) \\ & + \alpha_2(t)u^2(t-1) + \alpha_3(t)u^3(t-1)\end{aligned}$$

where,

$$\begin{aligned}\alpha_0(t) = {} & 0.8606y(t-1) - 0.0401y^2(t-1) \\ & + 0.0017y^3(t-1) - 0.000125y^4(t-1) \\ \alpha_1(t) = {} & 0.0464 - 0.045y(t-1) \\ & + 0.0034y^2(t-1) - 0.00025y^3(t-1) \\ \alpha_2(t) = {} & -0.0012 + 0.0013y(t-1) \\ & - 0.0001458y^2(t-1)\end{aligned}$$

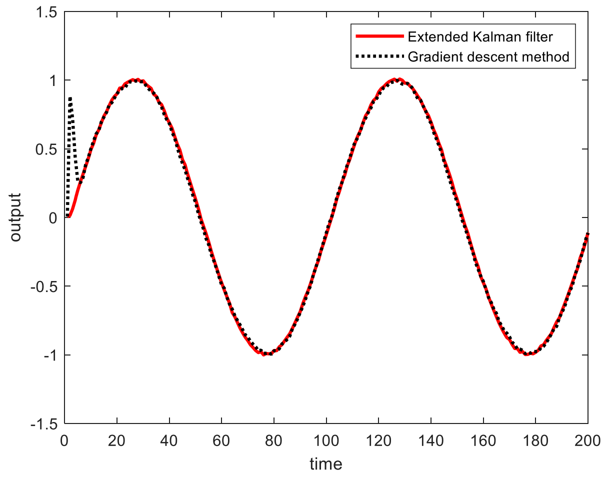

Figure 17. System output response under sinusoidal input.

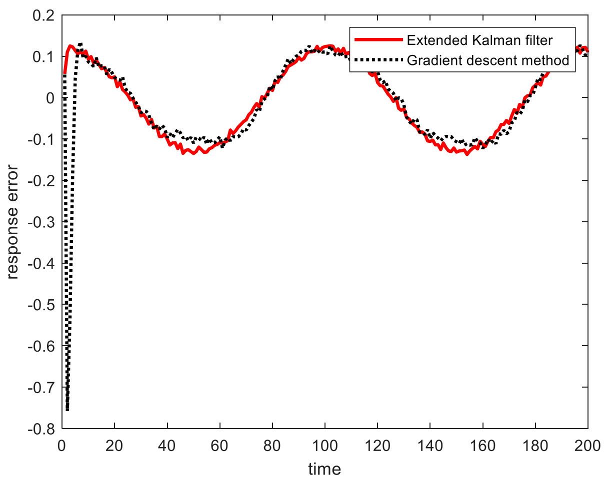

Figure 18. System response error under sinusoidal input.

$$\alpha_3(t) = 0.00002083 - 0.00002083y(t-1)$$

Under the influence of disturbance, the input signal selects the triangle wave and the sine wave respectively for simulation. The simulation results of the system output and system response error are shown in Figures 19–22. Figures 19 and 21 are system output responses, and Figures 20 and 22 are system response errors.

It can be seen from the simulation results that the extended Kalman filtering neural network control system based on particle swarm identification can track the expected output well in the presence of control interference. Compared with the gradient descent method, the extended Kalman filtering algorithm is used as the learning algorithm of the RBF neural network, which makes the system have a better control effect, faster response speed, smaller response error, and smoother output. In addition, when the system starts to respond, the response error has no obvious jump change, which can reduce the unnecessary wear and tear of the controlled device and prolong the device's service life. The simulation results show

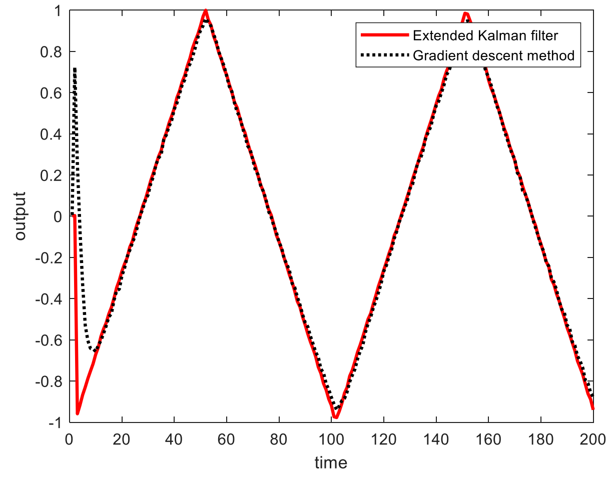

Figure 19. System output response under triangular wave input.

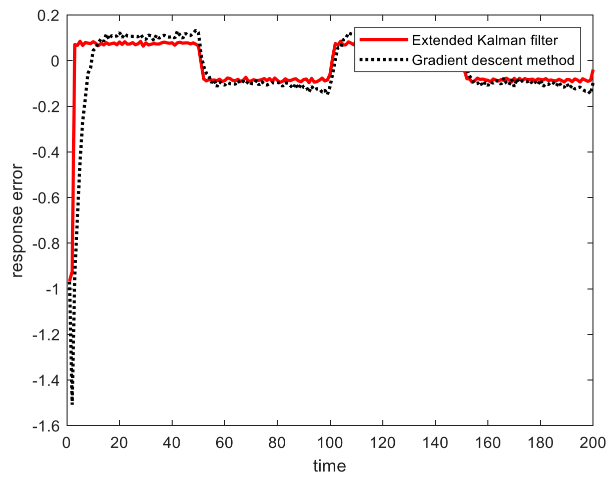

Figure 20. System response error under triangular wave input.

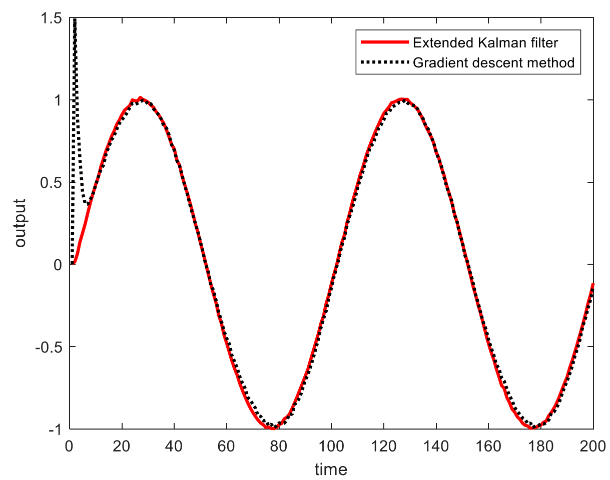

Figure 21. System output response under sinusoidal input.

that the proposed algorithm improves the convergence speed and anti-noise capability compared with the gradient descent method.

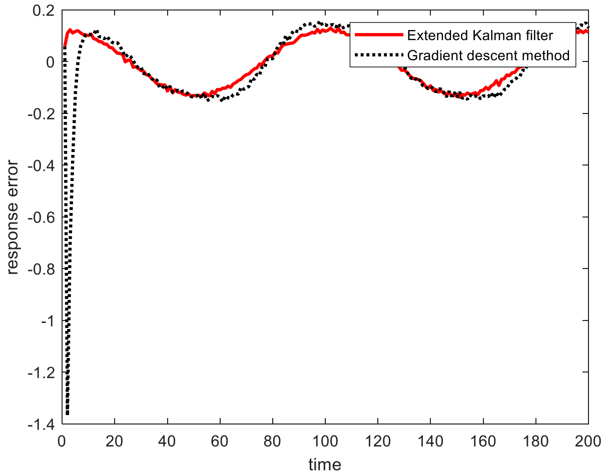

Figure 22. System response error under sinusoidal input.

8. Conclusions

In this paper, the U model framework is directly used to establish the nonlinear model through online identification under slow time variation. For the nonlinear U-model plant with known structures but unknown parameters, a particle swarm optimization algorithm is used to identify the time-varying parameters of the nonlinear U-model, and a better identification result is obtained. The identification error is small, and the accuracy is high. The newton iterative algorithm is introduced to transform the nonlinear plant model. The extended Kalman filtering neural network control based on particle swarm identification is proposed as the control scheme of the nonlinear U-model system. Considering that the interference in the nonlinear system will lead to the decline of network learning ability, the extended Kalman filtering algorithm is introduced in the learning of RBF neural network to reduce the influence of the disturbance on the stability and accuracy of the system, improve the control effect and the anti-interference of the system, and realize the accurate control of the complex nonlinear system.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the [Heilongjiang Provincial Natural Science Foundation of China] under Grant [number LH2021F057]; [State Grid Heilongjiang Electric Power Co., Ltd.] under Grant [number 5224162000JK]; and Heilongjiang Provincial institutions of higher learning basic scientific research funds scientific research project [number 135509309, 135409102].

References

[1] Xu F, Zhu Q, Zhao D, et al. U-model based design methods for nonlinear control systems a survey of the development in the 1st decade. Control Decis. 2013;28(7):961–971.

[2] Zhu QM, Guo LZ. A pole placement controller for nonlinear dynamic plant. Proc of IMechE, Part I: J Systemsand Control Engineering. 2002;216(16):467–476.

[3] Li R, Zhu QM, Kiely J, et al. (2020). Algorithms for U-model-based dynamic inversion (UM-dynamic inversion) for continuous time control systems. Paper 3640210. Complexity.

[4] Chang W, Wang W, Jia H. "Radial basis functions neural network of vary learning rate based stochastic U-model," *Int Conf on Electrical and Control Engineering(ICECE 2011 — Proceedings). Yichang: IEEE*, pp. 278-281, 2011.

[5] Wu X, Liu L, Zhu Q, et al. U-model-based adaptive control for a class of stochastic non-linear dynamic plants with unknown parameter. Int J of Modelling, Identification and Control. 2011;13(3):135–143.

[6] Azhar ASS, Al-Sunni FM, Shafiq M. "U-model based adaptive tracking scheme for unknown MIMO bilinear systems," *1st IEEE Conf on Industrial Electronics and Applications. Wuhan: IEEE*, pp. 473-486, 2006.

[7] Afande N, Saad S, Ridao P, et al. Implementation of nonlinear adaptive U-model control synthesis using a robot operating system for an unmanned underwater vehicle. IEEE Access. 2020;8(2020):205685–205695.

[8] Butt NR, Shafiq M, Khan T. "An adaptive root-solving controller for tracking of nonlinear dynamic plants," *Proc of the 2005 IEEE Int Conf on Industrial Electronics and Control Applications. Ecuador: IEEE*, pp. 144-149, 2005.

[9] Shafiq M, Butt NR. Real-time adaptive tracking of DC motor speed using U-model based IMC. Autom Control Comput Sci. 2007;41(1):31–38.

[10] Khan T, Shafiq M. "A novel internal model control scheme for adaptive tracking of nonlinear dynamic plants," *the 1st IEEE Conf on Industrial Electronics and Applications. Wuhan: IEEE*, pp. 123-130, 2006.

[11] Shafiq M, Haseebuddin M. U-model-based internal model control for non-linear dynamic plants. Proc of the Institution of Mechanical Engineers, Part I: J of Systems and Control Engineering. 2005;219(6):449–458.

[12] Shafiq M, Butt NR. "U-Model based adaptive IMC for nonlinear dynamic plant," *Proc of the 10th IEEE Int Conf on Emerging Technologies & Factory Automation. Catania: IEEE*, vol. 1, pp. 955-959, 2005.

[13] Butt NR, Shafiq M. Adaptive tracking of non-linear dynamic plants, using the U-model. Proc of the Institution of Mechanical Engineers, Part I: J of Systems and Control Engineering. 2006;220(3):473–487.

[14] Butt NR, Shafiq M. Higher-order neural network based root-solving controller for adaptive tracking of stable nonlinear plant, *IEEE Int Conf on Engineering of Intelligent Systems. Islamabad: IEEE*, pp. 1-6, 2006.

[15] Shafiq M, Butt NR. Utilizing higher-order neural networks in U-model based controllers for stable nonlinear plants. Int J of Control, Automation and Systems. 2011;9(3):489–496.

[16] Khan T, Shafiq M. "Innovative inverse control techniques for adaptive tracking of nonlinear dynamic plants," *IEEE Int Conf on Engineering of Intelligent Systems. Beijing: IEEE*, pp. 1-6, 2006.

[17] Zhang J, Li Y, Fei W, et al. U-Model based adaptive neural networks fixed-time backstepping control for uncertain nonlinear system. Math Probl Eng. 2020;2020(1):1–7.

[18] Li R, et al. U-Model-Based Two-degree-of-freedom internal model control of nonlinear dynamic systems. Entropy . 2021;23(2):169.
[19] Chang W, Wang W, Jia H. "Radial basis functions neural network of vary learning rate based stochastic U-model," 2011 *Int Conf on Electrical and Control Engineering(ICECE 2011 —Proceedings). Yichang: IEEE*, pp. 278-281, 2011.
[20] Xu F, Zhang X, Song X, et al. Composite control of RBF neural network and PD for nonlinear dynamic plants using U-model. J Intell Fuzzy Syst. 2018;11(1):565–575.
[21] El Hajjaji A, Khamlichi A. Analysis of a RBF neural network based controller for pitch angle of variable-speed wind turbines. Procedia Eng. 2017;181:552–559.
[22] Soni RK, Nigam MJ, Mitra R. Implementation of hybrid neuro-fuzzy controller based on radial basis function network with gradient descent algorithm for non-linear system. Power, Communication and Information Technology Conference (PCITC), IEEE. 2015: 178–183.
[23] Kilic E, Ozcalik HR, Yilmaz S. Efficient speed control of induction motor using RBF based model reference adaptive control method. automatika. 2016;57(3): 714–723.
[24] Xu F, Wang S, Liu F. Research on RBF neural network model reference adaptive control system based on nonlinear U – model. Automatika. 2020;61(1):46–57.
[25] Du M KL, Swamy N. Neural networks and statistical learning. Springer Science & Business Media. 2014;14:419–450.
[26] Örkcü HH, Bal H. Comparing performances of backpropagation and genetic algorithms in the data classification. Expert Syst Appl. 2011;38(4):3703–3709.
[27] Zhu QM, Zhang WC, Zhang JH, et al. U-neural network-enhanced control of nonlinear dynamic systems. Neurocomputing. 2019;352:12–21.
[28] Khooban MH, Abadi DNM, Alfi A, et al. Optimal type-2 fuzzy controller For HVAC systems. Automatika. 2014;55(1):69–78.