Taylor & Francis
Taylor & Francis Group

Automatika
Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Research on parallel nonlinear control system of PD and RBF neural network based on U model

Fengxia Xu, Deqiang Tang & Shanshan Wang

To cite this article: Fengxia Xu, Deqiang Tang & Shanshan Wang (2020) Research on parallel nonlinear control system of PD and RBF neural network based on U model, Automatika, 61:2, 284-294, DOI: 10.1080/00051144.2020.1731227

To link to this article: https://doi.org/10.1080/00051144.2020.1731227

AUTOMATIKA
2020, VOL. 61, NO. 2, 284–294
https://doi.org/10.1080/00051144.2020.1731227

Taylor & Francis
Taylor & Francis Group

REGULAR PAPER

Research on parallel nonlinear control system of PD and RBF neural network based on U model

Fengxia Xu[a], Deqiang Tang[b] and Shanshan Wang[b]

[a]College of Mechanical and Electrical Engineering, Qiqihar University, Qiqihar, People's Republic of China; [b]College of Computer and Control Engineering, Qiqihar University, Qiqihar, People's Republic of China

ABSTRACT
The modelling problem of nonlinear control system is studied, and a higher generality nonlinear U model is established. Based on the nonlinear U model, RBF neural network and PD parallel control algorithm are proposed. The difference between the control input value and the output value of the neural network is taken as the learning target by using the online learning ability of the neural network. The gradient descent method is used to adjust the PD output value, and ultimately track the ideal output. The Newton iterative algorithm is used to complete the transformation of the nonlinear model, and the nonlinear characteristic of the plant is reduced without loss of modelling precision, consequently, the control performance of the system is improved. The simulation results show that RBF neural network and PD parallel control system can control the nonlinear system. Moreover, the control system with Newton iteration can improve the control effect and anti-interference performance of the system.

ARTICLE HISTORY
Received 23 June 2019
Accepted 13 February 2020

KEYWORDS
Nonlinear U-model system; RBF neural networks; Newton iteration; PD

1. Introduction

In daily production activities, nonlinear characteristics are ubiquitous, especially with the rapid development of large-scale mechanical informationization, intelligence, and integration, and the study of nonlinear system characteristics has become particularly important. The nonlinear system first studies the plant modelling problem, and due to the highly complex nonlinearity of controlled plant, it is difficult to describe it with accurate mathematical model, which brings great challenges to the modelling of nonlinear system. Nonlinear modelling is the basis of studying nonlinear characteristics. The quality of the model directly determines the control effect of the system.

After decades of research on nonlinear modelling, NARMAX model (non-linear autoregressive moving average model), polynomial NARMAX model, Hammerstein model, Wiener model, nonlinear FIR model, Lure model, finite Volterra model, bilinear input–output model, NARX model and output radiation model were proposed [1,2]. Among them, Wiener model and Hammerstein model are a kind of nonlinear model with relatively simple structure, which has a certain position in the theoretical research and practical application of nonlinear systems [3]. Compared with linearization of nonlinear systems, these nonlinear models can more accurately describe the nonlinear characteristics of nonlinear systems and meet the requirements of nonlinear controller design. However, these nonlinear models also have some shortcomings: the complex model structure expression and a great many parameters which causes certain difficulties in controller design; for the same nonlinear system, the description capabilities of each model are different, that is, each model has poor universality and can only be used for a special class of nonlinear controlled objects, which is not easy to be widely applied. For this reason, proposing a simple and universal nonlinear model is a problem worth studying. In this context, the U model came into being. It is a polynomial with time-varying parameters and can represent a large class of smooth nonlinear systems. Moreover, during the transformation of the nonlinear plant into the expression of the U model, no linearization processing is performed, and the accuracy of the model can be guaranteed, which is the key to the wide application of the nonlinear U model. At the same time, considering the polynomial expression of the model, Newton iterative algorithm is adopted to transform the model. The transformed nonlinear objects can easily complete the system design with the linear system design method, which provides a good way to solve the nonlinear modelling problem [4].

In the past decade, great progress has been made in the design of nonlinear control systems based on the U model. A pole placement controller is proposed for a

CONTACT Fengxia Xu xufengxia_hit@163.com College of Mechanical and Electrical Engineering, Qiqihar University, Qiqihar 161006, People's Republic of China

This article has been republished with minor changes. These changes do not impact the academic content of the article.

class of known dynamic nonlinear objects based on the U model [5]. A forward adaptive tracking control based on U model is proposed [6]. Under the framework of U model, adaptive tracking of unknown multiple-input and multi-output systems is adopted [2]. An adaptive inverse control structure based on the standard leakage minimum mean square error (NLLMS) algorithm is proposed [7]. An internal model control based on the U model is proposed to solve the control of a class of known dynamic nonlinear objects [8]. The minimum mean variance adaptive inverse method based on the U model is used to control complex nonlinear industrial process systems [9]. The IMC control design based on the minimum mean square error of U model is proposed [10]. A generalized predictive control algorithm based on U model is proposed [11]. The least-square method is used to identify the U model coefficients of stochastic nonlinear objects, and the radial basis neural network is used to construct the controller [12]. For a class of unknown nonlinear delay objects, an adaptive control algorithm is proposed, using neural network to identify U model of time-varying parameters [13]. In [14], nonlinear U-model polynomial plant controller, which was designed by the method of linear polynomial control system, was selected as the research plant, and the U-block model framework was founded, which allowed the method of linear state space to design the nonlinear control system and offered a new direction for the design of nonlinear control system.

RBF neural network has a good generalization ability and simple network structure, which avoids complex mathematical calculation. The researches on the function approximation ability of RBF neural network show that RBF neural network can approximate any nonlinear function with any precision. Therefore, it has attracted much attention in the research of nonlinear control, and some research results have been obtained [15]. A variable learning operator RBF neural network algorithm based on nonlinear U model is proposed [16]. Based on a pole placement PID controller, a compound control method for tracking control of nonlinear dynamic system based on RBF and PD is proposed [17]. An adaptive control algorithm based on online identification of MIMO bilinear object model by RBF neural network is proposed [18]. For a class of uncertain nonlinear systems, an adaptive controller design scheme based on RBF network is proposed, so that the output of the nonlinear system is the expected output in case of uncertainty or unknown interference [19]. For the tracking control problem of a class of uncertain strictly feedback nonlinear systems, a new robust adaptive control design method is proposed by using RBF neural network to approximate all unknown parts of the system [20]. Aiming at the fact that the industrial robot control system is inevitably affected by random noise in practical work, and a design of RBF neural network and PD compound controller based on Kalman filter is proposed [21]. In order to eliminate the adverse effect of external interference and model uncertainty on the control system, a new intelligent control algorithm based on radial basis function neural network is designed for the control of MEMS gyroscopes [22]. According to the characteristics of nonlinear U model, this paper proposes the parallel control system based on RBF neural network and PID, and analyses the importance of Newton iteration algorithm in the control system. After the addition of Newton iteration algorithm, the output of the nonlinear control system can track the input signal well after a small delay, and the system error is small, which obviously improves the control effect of the nonlinear control system and can track the ideal output better.

The rest organization of this paper is organized as follows: Section 1 provides a basic discussion of the nonlinear U model. Section 2 analyses the working principle of RBF neural network. Section 3 proposes a parallel control system of RBF neural network and PD based on U model. Section 4 presents a parallel control system of RBF neural network and PD based on model transformation. In Section 5, two nonlinear plants are selected and the corresponding simulation results are analysed. Section 6 is the conclusion.

2. Nonlinear U-model

Nonlinear modelling is crucial in researching nonlinear control systems. All control algorithm design is based on the model of the controlled plant. The quality of the nonlinear model is directly related to the success or failure of the control algorithm design. In 2002 [23], a nonlinear U model was proposed to express the controlled plant. The U model used polynomial functions with time-varying parameters to represent a large class of smooth nonlinear systems, and its expression was derived on the basis of NARMAX model expression.

Assuming a single-input single-output (SISO) nonlinear controlled plant, the NARMAX model can be expressed as

$$y(t) = f(y(t-1), \ldots, y(t-n), u(t-1), \ldots, u(t-n), e(t), \ldots, e(t-n)) \quad (1)$$

In (1),$y(t)$ and $u(t)$ represent the output signal and input signal of the nonlinear controlled plant, respectively, $e(t)$ represents unknown and unpredictable quantities, such as modelling error, external interference, etc., $f(\cdot)$ represents a function of the nonlinear controlled plant model.

Extend (1), and get

$$y(t)=\sum_{j=0}^{M}\left[\prod_{l=1}^{n}\left(a_l+\sum_{j=1}^{m}y^l(t-1)\right)\right]\cdot \times\left[\prod_{l=1}^{n}\left(b_l+\sum_{j=2}^{m}u^l(t-1)\right)\right] u^j(t-1)+e(t) \quad (2)$$

In (2), both a_l and b_l are constants, $u(t-j)$ and $y(t-j)$ are the input and output signals of the past time, respectively. Simplified (2) can be obtained

In Equation (2), both A and B are constants, U and Y are input

$$y(t)=\sum_{j=0}^{m}\alpha_j(t)u^j(t-1)+e(t) \quad (3)$$

In (3),

$$\begin{aligned}\alpha_j(t)&=\left[\prod_{l=1}^{n}\left(a_l+\sum_{j=1}^{m}y^l(t-1)\right)\right]\cdot \times\left[\prod_{l=1}^{n}\left(b_l+\sum_{j=2}^{m}u^l(t-1)\right)\right]\\&=[a_1+y(t-1)+y(t-2)+\cdots+y(t-m)]\\&[a_1+y^2(t-1)+y^2(t-2)+\cdots+y^2(t-m)]\cdots\\&[a_n+y^n(t-1)+y^n(t-2)+\cdots+y^n(t-m)]\\&[b_1+u(t-2)+u(t-3)+\cdots+u(t-m)]\\&[b_2+u^2(t-2)+u^2(t-3)+\cdots+u^2(t-m)]\cdots\\&[b_n+u^n(t-2)+u^n(t-3)+\cdots+u^n(t-m)]\\&j=0,1,2,\ldots,n\end{aligned}$$

where the parameter $\alpha_j(t)$ represents the function of input $u(t-2), \ldots, u(t-n)$ and output $y(t-1), \ldots, y(t-n)$ at the past moment, and m represents the order of the system model, further converting Equation (3) to:

$$y(t)=U(t)=\sum_{j=0}^{m}\alpha_j(t)u^j(t-1)+e(t) \quad (4)$$

Equation (4) is the nonlinear U model expression.

An example is given to illustrate the universality and simplicity of the nonlinear U model. It is assumed that the mathematical expression of the nonlinear controlled object is

$$y(t)=0.1y(t-1)y(t-2)+0.8u(t-2)u(t-1) -0.3y(t-1)u^2(t-1)$$

According to (4), it can be obtained that:

$$y(t)=\alpha_0(t)+\alpha_1(t)u(t-1)+\alpha_2(t)u^2(t-1)$$

where

$$\alpha_0(t)=0.1y(t-1)y(t-2),$$
$$\alpha_1(t)=0.8u(t-2),\alpha_2(t)=0.3y(t-1).$$

In summary, the U model can use a mapping to transform the smooth nonlinear discrete time input–output dynamic system model into a model that can be designed by using linear control theory. At the same time, there is no linearization in the transformation process, which ensures the high accuracy of the model. This is the key to the widespread application of the nonlinear U model.

Compared with other nonlinear model expressions, U model expressions have the following advantages:

Compared with the NARMAX model and the Hammerstein model, the expression of the U model is more practical, simpler and concise.

The U model establishes a simple and universal mapping between almost all smooth nonlinear discrete time input–output dynamic systems, and the mapping is reversible.

By using the U-model representation to describe the mapping relationship between the sampled data of most input–output nonlinear dynamic systems, the discrete-time model of the nonlinear system can be easily obtained.

The U model expression adopts a polynomial structure, through which the design method of linear control can be applied to the nonlinear control system, thus greatly reducing the difficulty of controller design.

3. RBF neural network

The RBF neural network is a three-layer feedforward neural network with a hidden layer. It demonstrates functions such as local response local response, mutual partial coverage, mutual indirect relationship and so on. Therefore, the RBF neural network has good function approximation, optimal mapping ability and fast learning convergence. In this paper, a single-input-single-output RBF neural network is used. The neural network structure is shown in Figure 1.

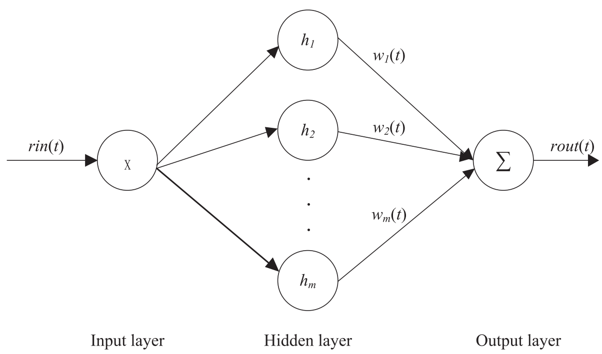

Figure 1. SISO radial basis neural network structure.

The RBF neural network consists of three layers: the input layer, the hidden layer, and the output layer. The first layer is the input layer, which is only responsible for transferring data to the next layer; the second layer is the hidden layer, which is responsible for nonlinear processing of the data of the previous layer, and transmits the processed data to the next layer. Generally, there are more neurons in this layer than in the previous layer; the third layer is the output layer, which is responsible for linear processing of the data of the previous layer, that is, the data of the previous layer is weighted to improve the learning rate of the network.

In Figure 1, m is the number of nodes in the hidden layer, $rin(t)$ is the input value, $rout(t)$ is the output value, $H = [h_1, h_2, \ldots, h_j, \ldots h_m]^T$ is the radial basis vector of the network, h_j is the output of Gaussian function, and $W = [w_1(t), w(t)_2, \ldots, w_j(t), \ldots, w_m(t)]^T$ is the variable of the network weight vector.

The following points should be noted when designing the RBF neural network:

The number of neurons in the hidden layer, that is, the value of m;

The setting of the weight matrix between the hidden layer and the output layer, that is, the matrix W;

The setting of the weight matrix between the input layer and the hidden layer, the values in this matrix are all 1;

The selection of radial basis function of neuron in hidden layer. Gaussian function is usually used.

The selection of radial basis function plays an important role in the design of RBF neural network. Generally, there are several options as follows:

(1) Gaussian function:

$$y(x) = \exp\left(- \frac{x^2}{2\sigma^2} \right)$$

(2) Inversion sigmoid function

$$y(x) = \frac{1}{1 + \exp\left(\frac{x^2}{\sigma^2}\right)}$$

(3) Quasi-multiple quadratic function

$$y(x) = \frac{1}{(x^2 + \sigma^2)^{1/2}}$$

where σ represents the extended constant or width of the basis function. These frequently used radial basis functions are monotonically decreasing as the Euclidean distance between the variable x and the 0 point increases. When the variable x is equal to 0, the value of the function is the maximum. From the radial basis function, it can be seen that the radial basis neural network has good local characteristics. Gaussian functions are widely used in radial basis neural networks because of their simple expression, good symmetry and the existence of arbitrary derivatives. Therefore, Gaussian function is adopted as the radial basis function of neurons in the hidden layer in this paper.

The algorithm of RBF neural network is as follows:

$$h_j = \exp\left(-\frac{||x - C_j||^2}{2b^2} \right) \tag{5}$$

In (5), $j = 1, 2, \ldots, m$, $C_j = [c_{j1}, c_{j2}, \ldots, c_{ji}, \ldots, c_{jm}]^T$ is the vector of the centre point of Gaussian function, and b is the basis width parameter of Gaussian function. The output of the neural network is

$$rout(t) = w_1(t)h_1 + w_2(t)h_2 + \cdots + w_m(t)h_m \tag{6}$$

The performance index function of the network is

$$J = \frac{1}{2}(routm(t) - rout(t))^2 \tag{7}$$

In (7), $routm(t)$ is the target output value. According to gradient descent method, the partial derivative with respect to $w_j(t)$ can be obtained as follows:

$$\Delta w_j(t) = -\eta \frac{\partial J}{\partial w_j(t)} h_j = \eta(routm(t) - rout(t))h_j \tag{8}$$

In (8), η is the learning rate of the network and is a normal number. The iterative algorithm of output weight is as follows:

$$w_j(t) = w_j(t-1) + \Delta w_j(t) + \alpha(w_j(t-1) - w_j(t-2)) \tag{9}$$

In (9), α is the inertia coefficient and is a normal number.

4. Parallel control system of RBF neural network and PD based on U-model

The system adopts design method of RBF neural network PD parallel control, in which RBF is feedforward controller, and its network control system structure diagram is shown in Figure 1. The system uses PD to complete the feedback control to ensure the stability of the system and suppress the disturbance. The RBF neural network controller is used to realize the feedforward control to ensure the control response speed of the system, reduce the overshoot and enhance the control precision. The structure of the control system is shown in Figure 2.

The controller takes the given signal $r(t)$ of the system as the input value of RBF neural network, and adjusts the weight $w_j(t)$ by the difference between the actual output value $u_2(t)$ of the neural network and the control quantity $U(t)$ of the system. When the system starts running, $u_2(t) = 0$, $U(t) = u_1(t)$, PD controller plays a role. PD controller can optimize the learning process of neural network according to the dynamic

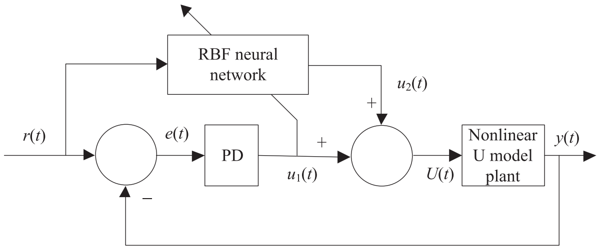

Figure 2. Composite control structure diagram of RBF neural network and PID.

characteristics of the system and improve the stability and anti-interference of the controller. At the end of each control cycle, the output $u_2(t)$ of RBF neural network is compared with the system control quantity $U(t)$, and the weight is modified to enter the learning phase of neural network. The purpose of RBF neural network learning is to minimize the difference between the control quantity of the system and the output quantity of the neural network, that is, the control quantity of the system is mainly generated by the neural network controller, and finally the actual output $y(t)$ of the controlled plant tracks the ideal output $r(t)$. PD algorithm is adopted here to make the learning of RBF only depend on the current measured value and variation of the error.

The output of RBF neural network is

$$u_2(t) = w_1(t)h_1 + w_2(t)h_2 + \cdots + w_m(t)h_m \quad (10)$$

The output of PD is

$$u_1(t) = k_p e(t) + k_d(e(t) - e(t-1)) \quad (11)$$

The total control input is

$$U(t) = u_1(t) + u_2(t) \quad (12)$$

The performance index function of the network is

$$J = \frac{1}{2}(U(t) - u_2(t))^2 = \frac{1}{2}{u_1}^2 \quad (13)$$

According to gradient descent method, the partial derivative with respect to $w_j(t)$ can be obtained as follows:

$$\begin{aligned}\Delta w_j(t) &= -\eta \frac{\partial J}{\partial w_j(t)} h_j \\ &= \eta(U(t) - u_2(t))h_j = \eta u_1(t) h_j \end{aligned} \quad (14)$$

In (14), η is the learning rate of the network and is a normal number. The iterative algorithm of output weight is as follows:

$$\begin{aligned} w_j(t) &= w_j(t-1) + \Delta w_j(t) \\ &\quad + \alpha(w_j(t-1) - w_j(t-2)) \end{aligned} \quad (15)$$

In (15), α is the inertia coefficient and is a normal number.

When PD is used for individual control, the value of PD gain largely determines the control index of the system. When the RBF control algorithm is added, the degree to which the control effect of the system depends on PD gain is significantly reduced.

5. Stability analysis

Lyapunov"s second law is a general method for the stability of deterministic systems, nonlinear systems and time-varying systems. The stability can be discriminated without having to solve the differential equations (difference equations) of the system. The scalar function of any definite symbol (positive or negative) is

$$V = V[x(t)] \quad (16)$$

If $V(0) = 0, x(k)$ is the solution of the following state equation

$$x(t+1) = f[x(t)] = x'(t)Px(t) = Ax(t) \quad (17)$$

Then, V is the Lyapunov function, and the difference of V is defined as

$$\Delta V[x(t)] = V[x(t+1)] - V[x(t)] \quad (18)$$

Make

$$V[x(t)] = \frac{1}{2}\sum_{i=1}^{t} e^2(i) \quad (19)$$

As can be seen from the second law of Lyapunov, it can be found that the system is stable as long as it is proved that $\Delta V[x(t)] < 0$.

Combined with the given Equation (19), according to the learning process, the $V[x(t)]$ changes are

$$\begin{aligned} \Delta V[x(t)] &= V[x(t+1)] - V[x(t)] \\ &= \frac{1}{2}\left(\sum_{i=1}^{t+1} e^2(i) - \sum_{i=1}^{t} e^2(i)\right) \\ &= \frac{1}{2}\sum_{i=0}^{t} [e^2(i+1) - e^2(i)] \\ &= \frac{1}{2}\sum_{i=0}^{t} \{[e(i) + \Delta e(i)]^2 - e^2(i)\} \\ &= \frac{1}{2}\sum_{i=0}^{t} [2e(i) \cdot \Delta e(i) + \Delta e^2(i)] \end{aligned} \quad (20)$$

Due to the learning process, $V[x(t)]$ changes to

$$\begin{aligned} e(t+1) &= e(t) + \left(\frac{\partial e(t)}{\partial w(t)}\right)^T \times \Delta w(t) \\ &= e(t) + \Delta e(t) \end{aligned} \quad (21)$$

By Equation (7):

$$J = \frac{1}{2}(routm(t) - rout(t))^2 = \frac{1}{2}e^2(t) \tag{22}$$

In order to ensure that the weight coefficient is modified along the negative gradient direction of J corresponding to $w(t)$, according to Equation (8), it can be obtained

$$\Delta w_j(t) = -\eta \frac{\partial J}{\partial w_j(t)} h_j = -\eta \frac{\partial J}{\partial e(t)} \frac{\partial e(t)}{\partial w(t)} h_j \tag{23}$$

From Equations (21) and (22) and let $A = \frac{\partial e(t)}{\partial w(t)}$, it can be obtained

$$\Delta e(t) = \left(\frac{\partial e(t)}{\partial w(t)}\right)^T \times \Delta w(t) = \left(\frac{\partial e(t)}{\partial w(t)}\right)^T \times \left(-\eta \frac{\partial J}{\partial e(t)} \frac{\partial e(t)}{\partial w(t)} h_j\right) = -\eta h_j A A^T \cdot e(t) \tag{24}$$

Let $K = \eta h_j$, it can be obtained

$$\Delta e(t) = -KAA^T \cdot e(t) \tag{25}$$

Substituting (25) into (20), it can be obtained

$$\Delta V[x(t)] = \frac{1}{2}\sum_{i=0}^{t}[2e(i) \cdot \Delta e(i) + \Delta e^2(i)] = -\frac{1}{2}\sum_{i=0}^{t}(A^T \cdot e(t))^T \times (2K - K^2 AA^T)(A^T \cdot e(t)) \tag{26}$$

According to Lyapunov stability theory: When $\Delta V[x(t)] < 0$, the whole system is stable, and there should be

$$2K - K^2 AA^T > 0 \tag{27}$$

Therefore, the value field of K is $0 < K < 2(AA^T)^{-1}$, and then $\Delta V[x(t)] < 0$, the following are available

$$\frac{1}{2}e^2(t+1) < \frac{1}{2}e^2(t), \lim e(t) = 0, t \to \infty \tag{28}$$

That is, as k increases, $e(k)$ tends to zero, and the learning algorithm converges.

Conclusion: taking the learning weight according to Equation (27), the whole system is stable and the learning algorithm converges.

The learning step determines the amount of modification of each cycle weight. If the value is too large, the system will oscillate and will not converge. If the value is too small, the convergence time will be extended accordingly. Therefore, we need to select the appropriate learning step according to the actual situation of the network. Lyapunov stability principle, according to the value range, the learning step of the neural network PID should satisfy the relationship

$$0 < \eta < \frac{1}{\varepsilon^2} \tag{29}$$

where

$$\varepsilon = \frac{1}{2\sqrt{e(t)}} \frac{\partial e(t)}{\partial w(t)} \tag{30}$$

The selection of the learning rate in the simulation of Section 7 is selected based on the formula (29) and the formula (30).

6. Parallel control system of RBFNN and PD based on model transformation

According to the expression of the nonlinear U model in (4), the coefficients of the nonlinear U-model are time-varying, and the time-varying velocity of the coefficients of different nonlinear systems cannot be estimated. The RBF neural network algorithm needs to continuously learn according to the nonlinear object model to complete the system control. Therefore, the tracking speed of the nonlinear system based on the U model has certain limitations, which in turn causes the system to have certain tracking error. In order to improve the response speed and control precision of the nonlinear system, the Newton iterative algorithm is used to transform the nonlinear U model, which reduces the time-varying speed of the nonlinear model and improves the control precision of the system.

The Newton-Raphson iterative algorithm is used to solve the polynomial, which provides a transformation method for the expression of nonlinear objects in the U model [24]. In order to obtain the output of the nonlinear model controller by applying the linear control design method, the (4) is further converted into the following form:

$$y(t) = U(t) \tag{31}$$

where $U(t) = \sum_{j=0}^{M} \alpha_j(t)u^j(t-1) + e(t)$, $U(t)$ is the output of the controller. In the model transformation part, As long as one root of the nonlinear equation is obtained, the output of the controller can be obtained. It should be noted that the transformation of the U model does not lose any characteristics of the original nonlinear model, and improves the design accuracy and efficiency of the nonlinear control system. The output of the Newton iteration formula is $u(t-1)$, and the

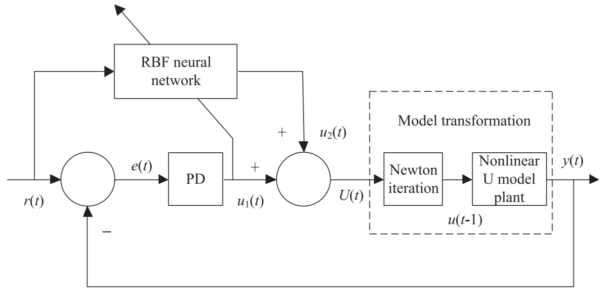

Figure 3. Structure diagram of the improved control system.

Newton iteration formula can be described as

$$u_{k+1}(t-1) = u_k(t-1) - \left.\frac{\sum_{j=0}^{M} \alpha_j(t) u_k^j(t-1) - U(t)}{d\left[\sum_{j=0}^{M} \alpha_j(t) u_k^j(t-1)\right] / du(t-1)}\right|_{u^j(t-1)=u_k^j(t-1)} \quad (32)$$

In (32), k is the number of iterations, that is, $k+1$ iteration is obtained from k iteration, and $k \geq 0$.

According to the nonlinear plant expression, the Newton iteration formula and the U model are inverse functions of each other. When the iterative formula is completely inverse with the U model, the system model transformation is matched, which is equivalent to offsetting the nonlinear part. The system output can completely track the input, and the effect of the control system is better. When the iterative algorithm is not completely inverse with the U model or the calculation order of the iterative algorithm is limited, the system model transformation has deviation, and the nonlinear system needs to be designed to complete the control requirements. However, in the real nonlinear system, the U model cannot completely describe the nonlinear object, and it is difficult to achieve a state in which the U model and the iterative algorithm are perfectly matched. Therefore, the Newton iterative algorithm is used to improve the system. The improved nonlinear control system is shown in Figure 3.

7. Simulation

The effectiveness and control effect of the designed controller were verified by continuous stirred tank reactor and laboratory liquid level control system.

Plant 1: This example is to verify the proposed control system by using the continuous stirred tank reactor as the nonlinear plant. The general model of the system is $\dot{y} = -(1+2a)y + au - uy - ay^2$. In the above equation, y represents the output of the controlled plant, as a dimensionless of component concentration. u represents the input of the controlled plant, as a dimensionless of flow rate. The above formula is discretized and converted into the expression of the U model:

$$U(t) = a_0(t) + a_1(t)u(t-1) + a_2(t)u^2(t-1) + a_3(t)u^3(t-1)$$

where

$$a_0(t) = 0.8606y(t-1) - 0.0401y^2(t-1) + 0.0017y^3(t-1) - 0.000125y^4(t-1)$$

$$a_1(t) = 0.0464 - 0.045y(t-1) + 0.0034y^2(t-1) - 0.00025y^3(t-1)$$

$$a_2(t) = -0.0012 + 0.0013y(t-1) - 0.0001458y^2(t-1)$$

$$a_3(t) = 0.00002083 - 0.00002083y(t-1)$$

The structure of RBF neural network adopts 1-7-1, where in the input layer is a 1-layer, the hidden layer is a 7-layer, the output layer is a 1-layer. Control system without Newton iteration: learning rate $\eta = 0.2$, inertia coefficient $\alpha = 0.01$, initial value of C is $[-3\ -2\ -1\ 0\ 1\ 2\ 3]$, initial value of b is 7, initial value of weight is 0, initial values of k_p and k_d are 0.4 and 0.01 respectively. Control system with Newton iteration: learning rate $\eta = 0.3$, inertia coefficient $\alpha = 0.01$, initial value of C is $[-3\ -2\ -1\ 0\ 1\ 2\ 3]$, initial value of b is 7, initial value of weight is 0, initial values of k_p and k_d are 0.1 and 0.02 respectively. Triangular wave, sine wave and square wave were selected for the input signal in simulation experiment, and the simulation diagram of system output and error signal was shown in Figures 4–9. Figures 4, 6 and 8 are system output response graphs, while Figures 5, 7 and 9 are system error graphs.

As it can be seen from the output response graph of the system, the control system without Newton iterative algorithm cannot track the input signal well, and

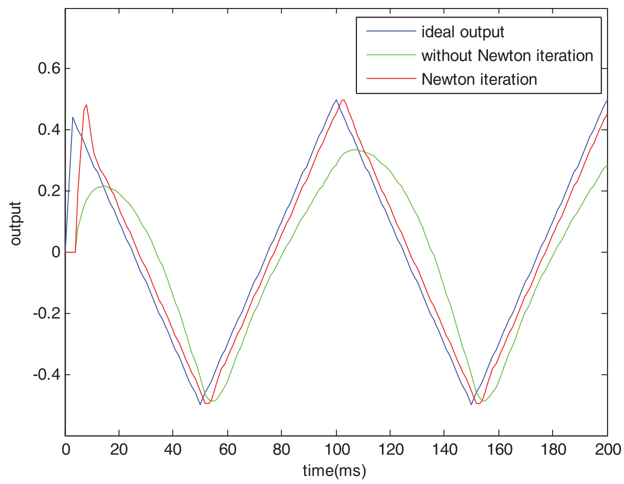

Figure 4. System output response graph when triangular wave input.

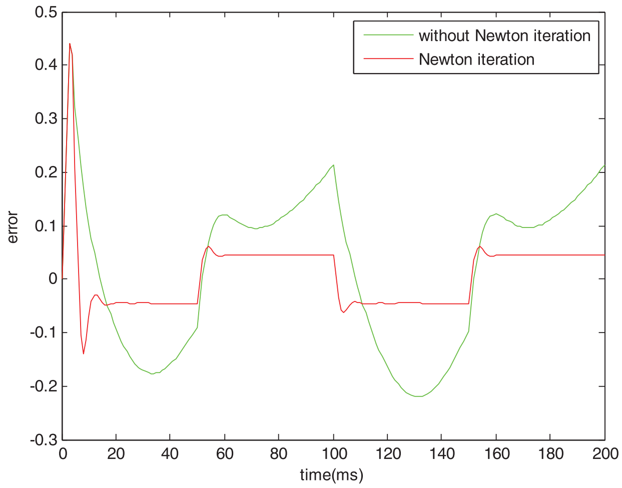

Figure 5. System error graph when triangular wave input.

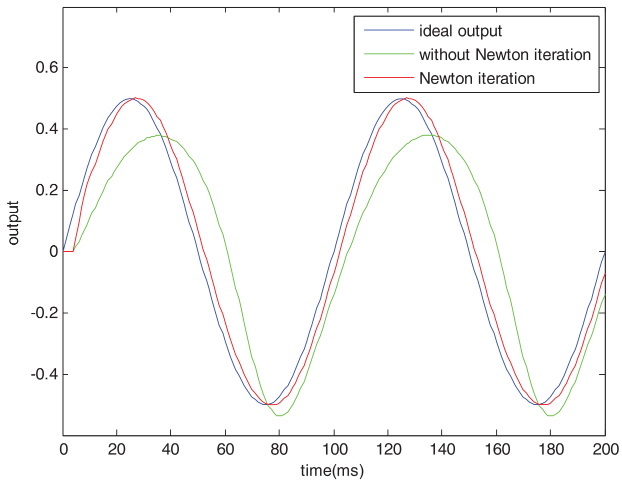

Figure 6. System output response graph when sine wave input.

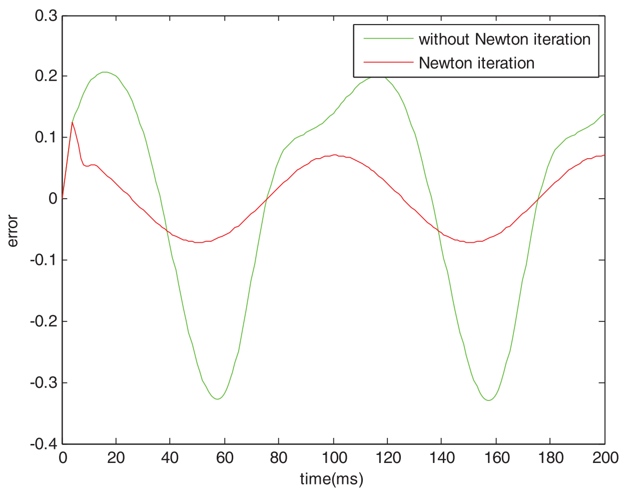

Figure 7. System error graph when sine wave input.

the system error is relatively large. After the addition of Newton iteration algorithm, the output of the non-linear control system can track the input signal well after a small delay, and the system error is small, which obviously improves the control effect of the nonlinear control system and can track the ideal output better. As can be seen from Figure 8, when the input signal is a square wave signal and the jump occurs, the output of the nonlinear control system with Newton iteration will track the input signal after a short period of overshoot adjustment; Although the non-linear system without Newton iteration does not produce overshoot, the adjustment time is long and the tracking effect is poor.

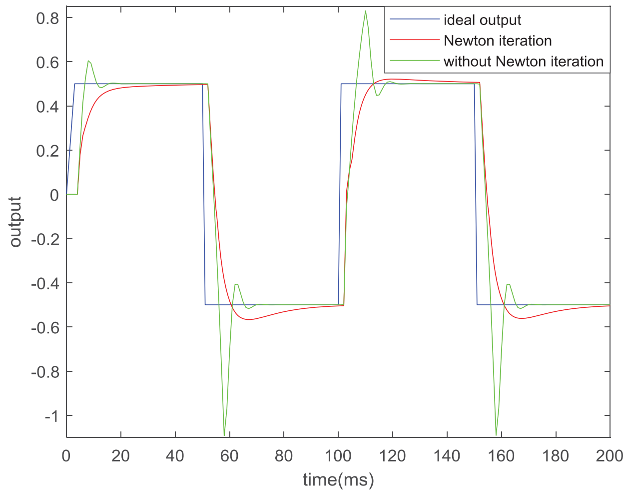

Figure 8. System output response graph when square wave input.

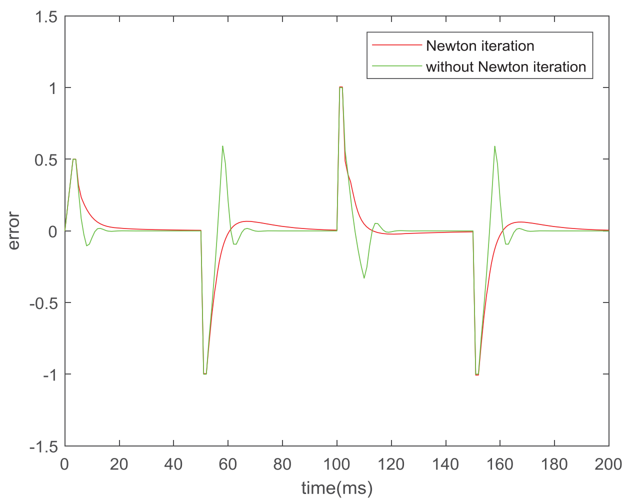

Figure 9. System error graph when square wave input.

Plant 2: laboratory liquid level control system is used to verify the rationality of the designed controller. The U model expression of laboratory liquid level control system is

$$y(t) = a_0 + a_1 u(t-1)$$

where

$$a_0(t) = 0.9722y(t-1) - 0.04288y^2(t-2) + 0.1663y(t-2)u(t-2)$$

$$+0.2573y(t-2)e_0(t-1)$$
$$-0.03259y^2(t-1)y(t-2)$$
$$-0.3513y^2(t-1)u(t-2)$$
$$+0.3084y(t-1)y(t-2)u(t-2)$$
$$+0.2939y^2(t-2)e_0(t-1)-0.1295u(t-2)$$
$$+0.6389u^2(t-2)e_0(t-1)$$

$$a_1(t)=0.3578-0.3103y(t-1)+0.1087y(t-2)u(t-2)+0.4770y(t-2)e_1(t-1)$$

The structure of RBF neural network adopts 1-7-1, where in the input layer is a 1-layer, the hidden layer is a 7-layer, the output layer is a 1-layer. Control system without Newton iteration: learning rate $\eta = 0.2$, inertia coefficient $\alpha = 0.01$, initial value of C is $[-3\ -2\ -1\ 0\ 1\ 2\ 3]$, initial value of b is 7, initial value of weight is 0, initial values of k_p and k_d are 0.4 and 0.01 respectively. Control system with Newton iteration: learning rate $\eta = 0.3$, inertia coefficient $\alpha = 0.01$, initial value of C is $[-3\ -2\ -1\ 0\ 1\ 2\ 3]$, initial value of b is 7, initial value of weight is 0, initial values of k_p and k_d are 0.1 and 0.02 respectively. Triangular wave, sine wave and square wave were selected for the input signal in simulation experiment, and the simulation diagram of system output and error signal was shown in Figures 10–15. Figures 10, 12 and 14 are system output response graphs, while Figures 11, 13 and 15 are system error graphs.

It can be seen from the system simulation graph that the nonlinear control system without Newton iteration has a large error in tracking the input signal, and the addition of Newton iterative algorithm significantly improves the control effect of the non-linear control system and enables the system to track the ideal output

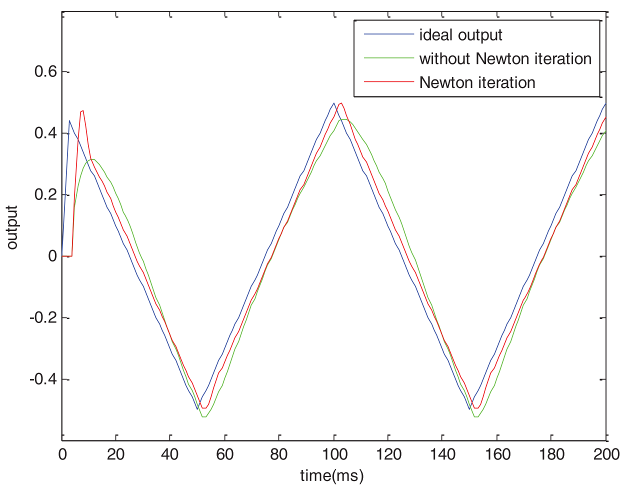

Figure 10. System output response graph when triangular wave input.

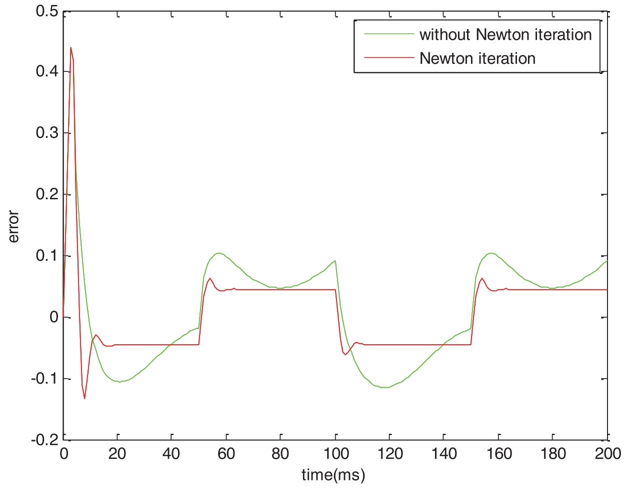

Figure 11. System error graph when triangular wave input.

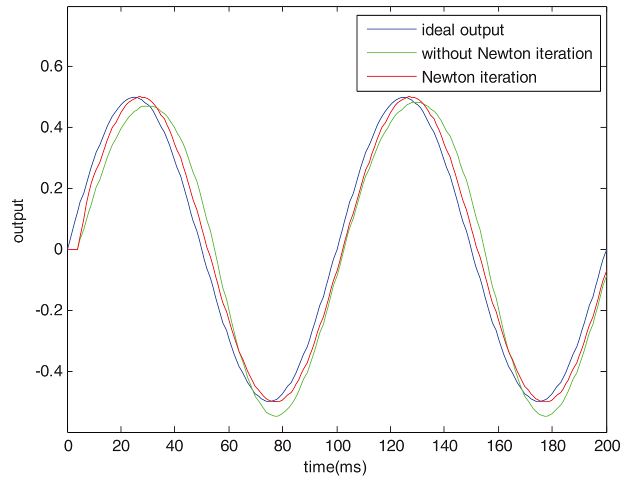

Figure 12. System output response graph when sine wave input.

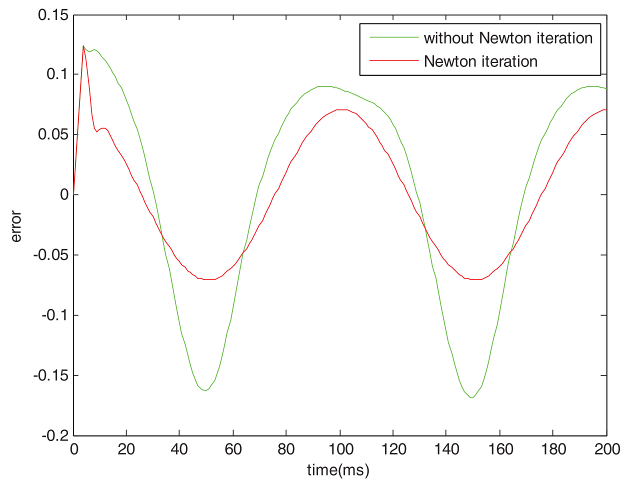

Figure 13. System error graph when sine wave input.

better. It can also be seen from two simulation examples that when the nonlinear plant is the third-order U model, that is, the nonlinear model is more complicated, the control effect of the system without Newton iterative algorithm becomes significantly worse, while

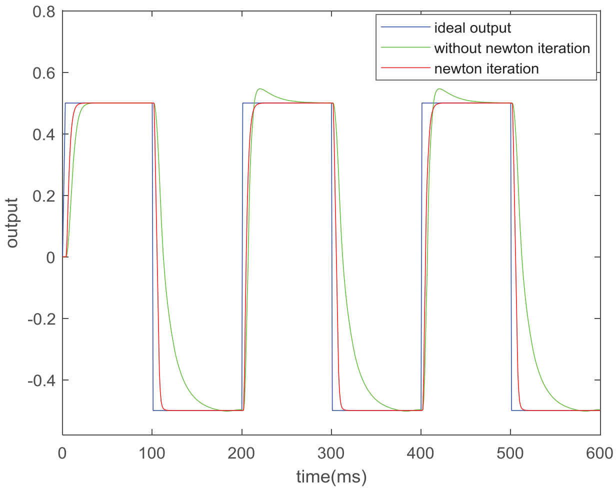

Figure 14. System output response graph when square wave input.

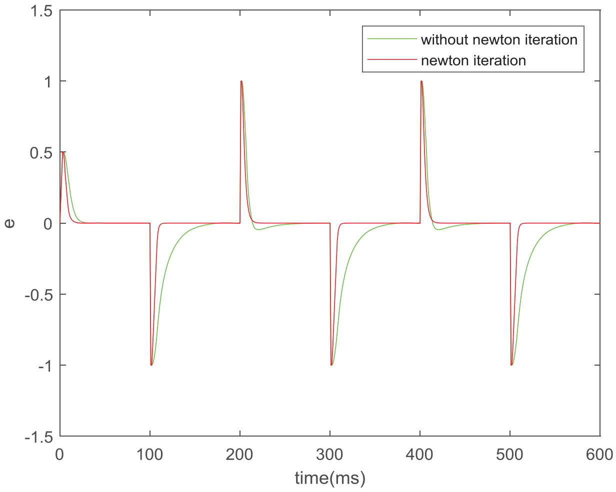

Figure 15. System error graph when square wave input.

the control effect of the system with Newton iterative algorithm remains unchanged. Therefore, after adding the Newton iterative algorithm, the time-varying characteristics of the nonlinear U model are weakened, and the control effect of the nonlinear control system is improved, so that the system can better track the ideal output.

8. Conclusion

In this paper, the time-varying polynomial U model is adopted to establish the nonlinear plant model. Based on the nonlinear U model, RBF neural network and PID parallel control are proposed as the control scheme of nonlinear system. Considering the time-varying characteristics of nonlinear pants, the Newton iterative algorithm is introduced to complete the transformation of the nonlinear object model and weaken the nonlinear characteristics of the object without approximate processing, which ensures the accuracy of the nonlinear plant model. The simulation shows that the control system with Newton iteration improves the control effect and the anti-interference of the system. At the same time, Newton iterative algorithm is used to transform the model of nonlinear plant, which reduces the requirements of nonlinear control system design and provides a new idea for some mature linear control methods to be applied to nonlinear system design.

Disclosure statement

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

Funding

The Fundamental Research Funds in Heilongjiang Provincial Universities (No :135309505).

References

[1] Ali SSA, Al-Sunni FM, Shafiq M, et al. Learning feedforward control of MIMO nonlinear systems using U model. In Proceedings of the 9th IASTED International Conference on Control and Applications; 2007; Austria: IEEE. p. 278–283.

[2] Ali SSA, Al-Sunni FM, Shafiq M, et al. U-model based learning feedforward control of MIMO nonlinear systems. Electr Eng. 2010;91(8):405–415.

[3] Du Wenxia WX. Research on neural network control and U model control method of nonlinear system [Ph.D. dissertation]. Hebei Province, China: Yanshan university; 2012.

[4] Xu FX, Zhu QM, Zhao DY, et al. U-model based design methods for nonlinear control systems a survey of the development in the 1st decade. Control Decis. Jul. 2013;28(7):961–977. doi:10.13195/j.cd.2013.07.4.xufx.023.

[5] Zhu QM, Guo LZ. A pole placement controller for nonlinear dynamic plant. Proc Inst Mech Eng, Part I: J Syst Control Eng. 2002 Sep;216(16):467–476. doi:10.1177/095965180221600603.

[6] Ali SSA, Al-Sunni FM, Shafiq M, et al. Learning feedforward control of MIMO nonlinear systems using U model. In: Proceedings of the 9th IASTED International Conference on Control and Applications; 2007; USA.

[7] Butt NR, Shafiq M, Khan T. An adaptive root-solving controller for tracking of nonlinear dynamic plants. In: 2005 International Conference on Industrial Electronics and Control Applications; 2005; Ecuador: IEEE. p. 144–149.

[8] Shafiq M, Butt NR. Real-time adaptive tracking of DC motor speed using U-model based IMC. Autom Control Comput Sci. 2007 Feb.;41(1):31–38. doi:10.3103/S0146411607010051.

[9] Khan T, Shafiq M. A novel internal model control scheme for adaptive tracking of nonlinear dynamic plants. In: 2006 the 1st IEEE Confernce on Industrial Electronics and Applications; 2006; Wuhan. p. 123–130.

[10] Shafiq M, Haseebiddon M. U-model-based internal model control for non-linear dynamic plants. Proc Inst Mech Eng, Part I: J Syst Control Eng. 2005 Sep;219(6):449–458. doi:10.1243/095965105X33563.

[11] Du WX, Wu XL, Zhu QM. Direct design of a U model-based generalized predictive controller for a class of non-linear(polynomial) dynamic plants. Proc Inst Mech Eng, Part I: J Syst Control Eng. 2012 Jan;226(1):27–42. doi:10.1177/0959651811409655.

[12] Chang WC, Wang WJ, Jia HR. Radial basis functions neural network of vary learning rate based stochastic U-model. In: 2011 International Conference on Electrical and Control Engineering; 2011; Yichang: IEEE. p. 278–281.

[13] Xu FX, Cheng Y, Ren HL, et al. Research on adaptive neural network control system based on nonlinear U-model with time-varying delay. Math Probl Eng. Aug, 2014;2014(2014):1–7. doi:10.1155/2014/420713.

[14] Shafiq M, Butt NR. Utilizing higher-order neural networks in U-model based controllers for stable nonlinear plants. Int J Control, Autom Syst. 2011 June;9(3):489–496. doi:10.1007/s12555-011-0308-y.

[15] Subudhi B, Jena D. A differential evolution based neural network approach to nonlinear system identification. Appl Soft Comput. 2011;11(1):861–871.

[16] Zhu QM, Zhao DY, Zhang J. A general U-block model-based design procedure for nonlinear polynomial control systems. Int J Syst Sci. 2016;47(14):3465–3475. doi:10.1080/00207721.2015.1086930.

[17] Xu FX, Zhang XJ, Song XH, et al. Composite control of RBF neural network and PD for nonlinear dynamic plants using U-model. J Intell Fuzzy Syst. 2018 Jul;35(1):565–575. doi:10.3233/JIFS-169612.

[18] Azhar ASS, Al-Sunni FM, Shafiq M. U-model based adaptive tracking scheme for unknown MIMO bilinear systems. In: 2006 1ST IEEE Conference on Industrial Electronics and Applications; 2006; Wuhan. p. 473–486.

[19] Ge SS, wang C. Adaptive neural control of uncertain MIMO nonlinear systems. IEEE Trans Neural Netw. 2004 May;15(3):674–692. doi:10.1109/TNN.2004.826130.

[20] Sun G, wang D, Peng ZH, et al. Neural network control for a class of strict feedback nonlinear systems. Control and Decision-Making. 2013 May;28(5):778–781+786. doi:1001-0920(2013)05-0778-04.

[21] Wang XL, Zhou J, Ji CY, et al. Research on compound control of RBF neural network and PD based on kalman filter. Comput Meas Control. 2009 Aug;17(8):1551–1553+1573. doi:10.16526/j.cnki.11-4762/tp.2009.08.052.

[22] Fei JT, Wu D. Adaptive control of MEMS gyroscope using fully tuned RBF neural network. Neural Comput Appl. 2017 Apr;28(4):695–702. doi:10.1007/s00521-015-2098-2.

[23] Zhu QM, Guo LZ. A pole placement controller for non-linear dynamic plant. Proc Inst Mech Eng, Part I: J Syst Control Eng. 2002;216(16):467–476.

[24] Zhu QM, Warwick K, Douce JL. Adaptive general predictive controller for nonlinear systems. IEE Proc D Control Theory Appl. 1991;138(1):33–40. IET Digital Library. doi:10.1049/ip-d.1991.0005.