

Received August 27, 2020, accepted August 29, 2020, date of publication September 1, 2020, date of current version September 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020980

# Multiple Linear Regression Prediction and Wavelet Neural Network Based Intelligent Online Tuning Control Method

JINGWEI LIU<sup>1,2</sup>, MENGXUAN ZHAO<sup>1</sup>, GUANCHEN LI<sup>1</sup>, AND JIAMING CHEN<sup>1,2</sup>

<sup>1</sup>Information College, Capital University of Economics and Business, Beijing 100070, China

<sup>2</sup>Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

Corresponding author: Jingwei Liu (liujingwei@cueb.edu.cn)

This work was supported by the Reform and Innovation Project of Undergraduate Teaching of Beijing in 2019 (No. 202), Philosophy and Social Science Foundation of Beijing (16JDYJB028), and Reserve Subject Leader Project of Capital University of Economics.

**ABSTRACT** Manual-based parameters tuning methods of control system are widely used in many industrial fields. However, better control performances such as faster control speed, more stable, less workload and safer working conditions are expected. Therefore, artificial intelligent based improved classical control (AI-CC) is highly valued in control fields. **Main works are as follows:** Firstly, wavelet neural network based PID (WNN-PID) method is proposed, BPNN of BPNN based PID method (BPNN-PID) is replaced by WNN. Secondly, multiple linear regression-based wavelet neural network PID (MR-WNN-PID) is proposed, the control parameters are adjusted according to the predicted output of control system. Thirdly, simulations are implemented to compare the performance of the proposed methods and the existing methods. Finally, the stability of the proposed methods is analyzed by theory. **Effects are as follows:** Firstly, the proposed AI-CC methods have better performances such as faster control speed, smaller overshoot, better ability of anti-interference. Secondly, theoretical analysis of stability of proposed methods is proven.

**INDEX TERMS** Control parameters online tuning, artificial intelligent control, auto-tuning control, wavelet neural network PID, multiple linear regression-based prediction.

## I. INTRODUCTION

Manual-based parameters tuning control method is a common and widely used method [1], [2]. In the field of industrial control, classical control (CC) methods and systems [3], such as PID control, have been applied for a long time. PID control method, as the earliest controller in practical industry, possesses a history of nearly one hundred years, because the theories of PID method are complete and mature. Recently, improved artificial intelligence based classical control methods (AI-CC) are highly valued in automatic control fields [4], [5]. AI-CC methods have many advantages such as self-adaptation, self-learning, and sufficient theoretical support. Therefore, AI-CC methods are more and more widely applied in practical projects.

Problems of existing control parameters online tuning methods are as follows: (1) The control performances are unsatisfied, because more and more practical applications need faster control speed, lower steady-state error and better stability and anti-interference ability. (2) The control system

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang .

cannot maintain the best performance continuously, because the parameters of the control system cannot be tuned online by the control system itself. (3) Long time working in harmful industrial environments is bad for engineers' health, such as chemically pollution, radiant [6], [7], etc.

There are 3 existing advanced control methods [8]:

(1) Model predictive control (MPC) is built by multi-step testing, rolling optimization, etc., which is more suitable for multivariable systems, it focuses on the optimization of control systems [9]. MPC is developed from the dynamic matrix control (DMC) in 1980s. MPC can present a dynamic response quickly, and it allows constraints to be incorporated into the control law. Moreover, MPC has good control performance and strong robustness, it can overcome the uncertainty of control process. However, MPC has some disadvantages such as large computational burden, monotonic feedback correction method and less rolling optimizing strategy. Furthermore, for practical application of non-linear system, MPC still remains some questions such as reliability and efficiency of the on-line computation scheme [10].

(2) Adaptive control (ADC) can continuously identify control parameters of models and tune them online to reach a

better performance of systems [11]. Adaptive control method is widely applied to uncertain systems, especially for controlling strict-feedback nonlinear systems [12]. Recently, some new adaptive control strategies are proposed for non-strict-feedback nonlinear systems [13]. However, the general solution of nonlinear random systems is difficult to obtain in theoretical level, the estimations of the uncertain parameters and convergence cannot be guaranteed, the parameters in adaptive control systems are much more difficult to be adjusted than in classical control systems [14], [15].

(3) Intelligent control methods (not based on PID) are a class of control techniques that use artificial intelligence computing approaches such as expert control methods, fuzzy control methods, and neural network control methods [16]. However, intelligent control algorithms are nonlinear and complex, which make the intelligent control system difficult to be theoretically analyzed.

(4) Artificial intelligence based improved classical control method (AI-CC methods) is based on the classical control theories, thus the classical control theories can be used for analyzing of AI-CC system such as non-liner, robustness and stability. AI-CC methods have both advantages of intelligent methods and classical control methods. Classical control method is widely used in many applications, so that research of AI-CC methods is more practical and expected.

Compare with the above three methods, AI-CC have its own advantages: (1) It is valuable to improve the existing classical control systems, because there are various practical applications are based on classical control. (2) The AI-CC methods can be theoretically analyzed easily on the basis of a large number of mature and existing classical control theories. (3) Variety of AI-CC methods can be implemented more easily because the output of AI is the input of CC, the dependence and coupling between AI and CC are very low [17]. Therefore, this study focuses on AI-CC methods and its improvements.

#### A. RELATED WORK

The existing AI-CC methods are Expert-PID (E-PID), Fuzzy-PID (F-PID), Neural Network-PID (NN-PID), etc [18].

(1) E-PID and F-PID are rule-based reasoning method. Control parameters of E-PID and F-PID are online tuned by expert rules tables (ERT) and fuzzy rules calculation (FRC) [19], [20]. ERT and FRC need to be offline set, the expert experiences are directly transformed into ERT and FRC. The advantage of E-PID and F-PID is that the control speed is faster because both E-PID and F-PID have fixed ERT and FRC.

(2) NN-PID is neural network optimization-based method. In each adjustment cycle, the control parameters are tuned online based on the training and calculation results of neural network (NN). The inputs of NN include the system error and the differential of error, and the output is the new tuning control parameters. The control effects of NN-PID are determined by the structure of NN, training methods and sample data. The advantage of NN-PID is that the control system is

smarter, which means that less manual configuration work is required, and better performances might be obtained.

The problems of E-PID and F-PID are as follows: (1) E-PID uses simple rules to get faster control speed but causes overshoot and instability. (2) The configuration work of F-PID is more complex for engineers, so the improvements of control performances are hard to achieve. (3) The reasoning process is based on fixed rules and strict formulas, ERT and FRC must be offline configured accurately before the system starting up. And the ERT and FRC cannot self-optimized during the control process [21], [22].

The NN-PID method can solve the above problems of E-PID and F-PID: (1) NN-PID has better online learning and optimizing ability, the rules of online tuning can be adjusted by NN-PID itself. (2) the configuration work of NN-PID is simpler, because the weights of NN do not need to be configured precisely before the system is running [23].

The existing NN-PID methods include Back Propagation Neural Network based PID (BPNN-PID) and Radical Basis Function Neural Network based PID (RBFNN-PID) [24], [25], etc. The problems of the existing BPNN-PID and RBFNN-PID are as follows: (1) Some training processes of RBFNN-PID are unstable, which means that the system output cannot converge in the end of the control process. (2) Some processes of BPNN-PID and RBFNN-PID systems appear overshoot which cannot be acceptable in some applications. (3) Performances such as control speed, steady-state error and ability of anti-interference are expected to be improved. Therefore, this study focuses on the improvements of NN-PID [26], [27].

#### B. CONTRIBUTION

Main contributions to this study are as follows: (1) WNN-PID method is proposed to improve the performances. The training methods of existing BPNN-PID and RBFNN-PID are BPNN and RBFNN, while WNN is the training method of WNN-PID. Compared with BPNN and RBFNN, WNN has better learning ability which is expected to improve the performances of NN-PID systems. (2) MR-WNN-PID is further proposed to improve the performances. Multiple Linear Regression (MR) predictor is adopted to predict the system output, while WNN-PID has no predictor. MR predictor improves the accuracy of control processes. (3) Comparative simulations of all the above AI-CC methods are implemented to verify the effects of improvements. (4) The stability of all the above AI-CC methods is proven theoretically.

The rest of this article is organized as follows: In section II, the existing CC and AI-CC models are introduced. In section III, WNN-PID method is proposed. In section IV, MR-WNN-PID is proposed, which combines MR predictor with WNN-PID method. In section V, comparative simulations of are implemented with the results analysis of all AI-CC methods. In Section VI, theoretical stability analysis is discussed. In Section VII, conclusions are summarized including all the comparative methods.

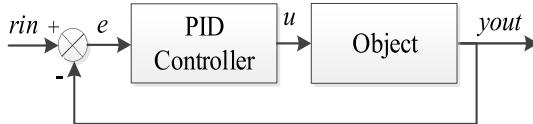
## II. CLASSICAL PID AND NN-PID (EXISTING MODEL)

Section II is organized as follows: In part A, the model of existing PID is described. In part B, the models of BPNN-PID and RBFNN-PID methods are described. Structure, formula and pseudocode of the above existing methods are described in detail.

### A. CLASSICAL PID MODEL (EXISTING METHOD)

#### 1) STRUCTURE OF MODEL

The structure of classical PID control system can be drawn as Fig.II-A.2. In PID control,  $K_p$ ,  $K_i$ ,  $K_d$  are the control parameters which have been set as a fixed value before the system starts up. After the system starts up, in each adjustment cycle, the system error ( $e$  and  $de/dt$ ) can be calculated based on system input and system output at time  $t$ . Then, the PID controller calculates the output of controller  $u$  as the input of controlled object according to system error. Finally, Under the influence of  $u$ , the output of controlled object is updated to be the system output at time  $t + 1$ .



**FIGURE 1.** Structure of classical PID control systems.

Each simulation process has many adjustment cycles, which can be defined as follows:

*Definition 1:* adjustment cycle (AC). 1 AC represents 1 control process at 1 fixed simulation time (e.g.  $t = 100$ ). In each AC, the system error  $error(t)$ ,  $d_{error}(t)$ , the adjusted control parameters  $k_p(t)$ ,  $k_i(t)$ ,  $k_d(t)$ , the control variable  $u(t)$  (the output of controller) and the output of control system  $yout(t)$  are updated once.

*Definition 2:* simulation process (SP). 1 SP represents 1 independent control process which is from start time  $t = 1$  to the end time  $max\_t$  (e.g. max simulation time  $max\_t = 10000$ ). Before the end time, the control system should be steady and stable.

The relationship between SP and AC is that 1 SP is composed of  $n$  AC.

#### 2) CALCULATION OF MODEL

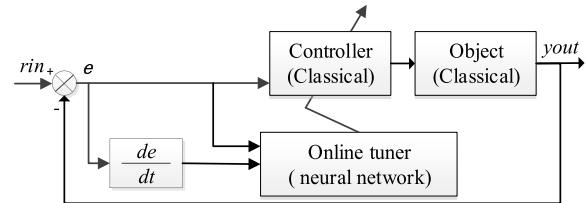
The model of classical PID Controller can be defined as Eq. (1):

$$u(t) = K_p \cdot (e(t)) + \frac{1}{K_i} \int e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

In PID control, the control parameters  $K_p$ ,  $K_i$  and  $K_d$  are fixed values. While in AI-CC, the control parameters  $K_p(t)$ ,  $K_i(t)$  and  $K_d(t)$  are online tuned in each AC. Therefore, the model of AI-CC controller can be defined as Eq. (2):

$$u(t) = K_p(t) \cdot (e(t)) + \frac{1}{K_i(t)} \int e(t) dt + K_d(t) \frac{de(t)}{dt} \quad (2)$$

The output of controller is  $u(t) = u(t - 1) + \Delta u(t)$ . The updated  $u(t)$  is calculated according to the incremental PID



**FIGURE 2.** Structure of NN based intelligent control system.

method, which can be expressed as Eq. (3):

$$\begin{aligned} \Delta u(t) = & k_p(t) (e(t) - e(t - 1)) + k_i(t) e(t) \\ & + k_d(t) (e(t) - 2e(t - 1) + e(t - 2)) \end{aligned} \quad (3)$$

The output of the control system is  $yout(t)$ , which can be calculated by the digital PID method of discrete system. The transfer function of the controlled object is discretized as Eq. (4).

$$\begin{aligned} yout(t) = & -a(2)y(t-1) - a(3)y(t-2) - a(4)y(t-3) \\ & + b(2)u(t-1) + b(3)u(t-2) + b(4)u(t-3) \end{aligned} \quad (4)$$

#### 3) IMPLEMENT OF MODEL

The pseudocode of PID control system is presented as Method 1:

##### Method 1 PID

1. Begin
2. Initialize PID control system;
3. While ( $t \leq max\_t$ ) do:
  4. Simulation time updated:  $t = t + 1$ ;
  5. System input  $rin(t)$  is updated;
  6. Calculate  $\Delta u(t)$  and  $u(t)$  according to Eq. (3);
  7. Calculate  $yout(t)$  according to Eq. (4);
  8. Calculate  $e(t) = yout(t) - rin(t)$ ;
  9. End while
10. End

## B. BPNN-PID MODEL (EXISTING METHOD)

#### 1) STRUCTURE OF MODEL

The structure of neural network (NN) based AI-CC system is drawn as Fig.2. Existing NN based AI-CC methods are BPNN-PID, RBFNN-PID etc. The control parameters of NN-PID are adjusted in each AC by a neural network.

#### 2) INPUT AND OUTPUT OF MODEL

The inputs of NN are  $rin(t)$ ,  $yout(t - 1)$ ,  $error(t)$ , which are expressed as Eq.(5).

$$\begin{cases} net_1^{(1)}(t) = rin(t) \\ net_2^{(1)}(t) = yout(t - 1) \\ net_3^{(1)}(t) = error(t) \end{cases} \quad (5)$$

The outputs of NN are  $K_p(t)$ ,  $K_i(t)$ ,  $K_d(t)$ , which are expressed as Eq.(6).

$$\begin{cases} K_p(t) = O_1^{(3)}(t) \\ K_i(t) = O_2^{(3)}(t) \\ K_d(t) = O_3^{(3)}(t) \end{cases} \quad (6)$$

### 3) FORWARD CALCULATION

The outputs of NN are calculated according to Eq. (7) to Eq.(12).

$$net_j^{(2)}(t) = \sum_{j=1}^Q w_{ji}^{(2)}(t) \cdot O_i^{(1)}(t) \quad (7)$$

$$O_j^{(2)}(t) = f(net_j^{(2)}(t)), \quad (j = 1, 2, \dots, Q) \quad (8)$$

$$net_k^{(3)}(t) = \sum_{j=1}^Q w_{kj}^{(3)}(t) O_j^{(2)}(t) \quad (9)$$

$$O_k^{(3)}(t) = g(net_k^{(3)}(t)), \quad (k = 1, 2, 3) \quad (10)$$

$$g(x) = \frac{e^x}{e^x + e^{-x}} \quad (11)$$

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (12)$$

$f(x)$  represents the scaling transformation function of the hidden layers of BPNN.  $O_j^{(2)}(t)$  represents the output of the hidden layer, and  $O_k^{(3)}(t)$  represents the output of the output layer.  $w_{jk}^{(3)}$  and  $w_{ij}^{(2)}$  are weights of the neural network.

### 4) OBJECT FUNCTION

The target of adjusting process is to minimize the mean square error of control system. The object function of online tuner of NN-PID can be expressed as Eq. (13).

$$\min E(t) = \frac{1}{2}(rin(t) - yout(t))^2 \quad (13)$$

The object function can also be expressed as Eq. (14).

$$\min E(t) = |rin(t) - yout(t)| \quad (14)$$

Although the object function is different from Eq. (13) to Eq. (14), the effects are similar. In this study, all of the NNs will apply the same object function for comparison.

### 5) TRAINING CALCULATION

According to the above object function, gradient descent method is adopted as the training algorithm in this study, because the gradient descent method is simpler and faster than other training methods.  $\alpha$  and  $\mu$  are inertia coefficient and learning rate respectively. The adjustment formulas of  $w_{jk}^{(3)}$  and  $w_{ij}^{(2)}$  are provided as Eq. (15) to Eq. (19):

$$\Delta w_{kj}^{(3)}(t) = \alpha \cdot \Delta w_{kj}^{(3)}(t-1) - \eta \cdot \delta_k^{(3)}(t) \cdot O_j^{(2)}(t) \quad (15)$$

$$\Delta w_{ji}^{(2)}(t) = \alpha \cdot \Delta w_{ji}^{(2)}(t-1) - \eta \cdot \delta_j^{(2)}(t) \cdot O_i^{(2)}(t) \quad (16)$$

$$\delta_k^{(3)} = error(t) \cdot sgn\left(\frac{\partial y(t)}{\partial \Delta u(t)}\right) \cdot \frac{\partial \Delta u(t)}{\partial O_k^{(3)}(t)} \cdot g'(net_k^{(3)}(t)) \quad (17)$$

$$\delta_j^{(2)} = f'(net_j^{(2)}(t)) \sum_{k=1}^3 \delta_k^{(3)} \cdot w_{kj}^{(3)}(t) \quad (18)$$

$$g'(x) = g(x)(1-g(x)), \quad f'(x) = (1-f^2(x))/2 \quad (19)$$

### 6) IMPLEMENT OF MODEL

The pseudocode of NN-PID can be designed as Method 2:

#### Method 2 NN-PID

- 
1. Begin
  2. Initialize NN-PID control system model;
  3. Initialize NN weights  $W_{mn}(t)$  and offsets  $b_l(t)$ ;
  4. While ( $t \leq max\_t$ ) do:
    5. Update the simulation time:  $t = t + 1$ ;
    6. System input  $rin(t)$  is updated;
    7. Calculate  $error(t)$  and  $d_{error}(t)$ ;
    8. Adjust  $W_{mn}(t)$  and  $b_l(t)$  according to (15) to (29);
    9. Adjust  $K_p(t)$ ,  $K_i(t)$ ,  $K_d(t)$  according to (7) to (14);
    10. Calculate  $\Delta u(t)$  and  $u(t)$  according to Eq. (3);
    11. Calculate  $yout(t)$  according to Eq. (4);
    12. End while
  13. End
- 

## III. WNN-PID MODEL (IMPROVED MODEL)

Section III is organized as follows: In part A, the structure of WNN-PID is described. In part B, the input and output of WNN-PID is expressed. In part C, the objective function of WNN-PID is described. In part D, the training algorithm of WNN-PID is provided. In part E, the pseudocode of WNN-PID is shown.

### A. STRUCTURE OF MODEL

The structure of WNN-PID is similar to NN-PID, which is shown in Fig.2. The difference between WNN-PID and BPNN-PID is that they have different structures of neural network and different training algorithms. The training method of WNN-PID is the gradient descent algorithm which is applied to WNN. Meanwhile, the training method of BPNN-PID is the gradient descent algorithm which is applied to BPNN.

### B. INPUT AND OUTPUT OF MODEL

$K_p(t)$ ,  $K_i(t)$ ,  $K_d(t)$  are the control parameters which are the result of forward propagation of BPNN according to Eq. (20) to Eq. (27).

$$\begin{cases} net_1^{(1)}(t) = rin(t) \\ net_2^{(1)}(t) = yout(t-1) \\ net_3^{(1)}(t) = error(t) \end{cases} \quad (20)$$

$$\begin{cases} K_p(t) = O_1^{(3)}(t) \\ K_i(t) = O_2^{(3)}(t) \\ K_d(t) = O_3^{(3)}(t) \end{cases} \quad (21)$$

$$O_j^{(2)}(t) = \Psi_{ab}\left(\frac{net_j^{(2)}(t) - b_j(t)}{a_j(t)}\right) \quad (22)$$

$$net_k^{(3)}(t) = \sum_{j=1}^Q w_{jk}^{(3)}(t) \cdot O_j^{(2)}(t) \quad (23)$$

$$O_k^{(3)}(t) = g\left(net_k^{(3)}(t)\right) \quad (24)$$

$$g(x) == \frac{e^x}{e^x + e^{-x}} \quad (25)$$

$$\Psi_{ab}(t) = \cos(1.75t) \cdot e^{-\frac{t^2}{2}} \quad (26)$$

$$\Psi'_{a,b}(t) = -1.75 \cdot \sin(1.75t) \cdot e^{-\frac{t^2}{2}} - t \cdot \cos(1.75t) \cdot e^{-\frac{t^2}{2}} \quad (27)$$

$\Psi_{ab}(t)$  represents the scaling transformation function of the hidden layers of WNN. The selection of  $\Psi_{ab}(t)$  must satisfy the framework condition.  $O_j^{(2)}(t)$  represents the output of the hidden layer, and  $O_k^{(3)}(t)$  represents the output of the output layer.  $w_{jk}^{(3)}$  and  $w_{ij}^{(2)}$  are weights of the neural network.  $\Delta a_j(t)$  and  $\Delta b_j(t)$  are scale parameters of the activation function in the hidden layer.

### C. OBJECT FUNCTION

WNN-PID method is proposed based on BPNN-PID. Therefore, the object function of WNN is same as BPNN mentioned above, which is expressed as Eq. (13).

### D. TRAINING ALGORITHM OF MODEL

$w_{jk}^{(3)}, w_{ij}^{(2)}, \Delta a_j(t)$  and  $\Delta b_j(t)$  are adjusted in each AC (adjustment cycle) as Eq. (28) to Eq. (37).

$$\Delta w_{jk}^{(3)}(t) = \alpha \cdot \Delta w_{jk}^{(3)}(t-1) - \eta \cdot \frac{\partial E(t)}{\partial w_{jk}^{(3)}(t)} \quad (28)$$

$$\Delta w_{ij}^{(2)}(t) = \alpha \cdot \Delta w_{ij}^{(2)}(t-1) - \eta \cdot \frac{\partial E(t)}{\partial w_{ij}^{(2)}(t)} \quad (29)$$

$$\Delta a_j(t) = \alpha \cdot \Delta a_j(t-1) - \eta \cdot \frac{\partial E(t)}{\partial a_j(t)} \quad (30)$$

$$\Delta b_j(t) = \alpha \cdot \Delta b_j(t-1) - \eta \cdot \frac{\partial E(t)}{\partial b_j(t)} \quad (31)$$

$$\begin{aligned} \frac{\partial E(t)}{\partial w_{jk}^{(3)}(t)} &= error(t) \cdot sgn\left(\frac{\partial y(t)}{\partial \Delta u(t)}\right) \cdot \frac{\partial \Delta u(t)}{\partial o_k^{(3)}(t)} \\ &\cdot g'(net_k^{(3)}(t)) \cdot o_j^{(2)}(t) \end{aligned} \quad (32)$$

$$\begin{aligned} \frac{\partial E(t)}{\partial w_{ij}^{(2)}(t)} &= \sum_{k=1}^L \delta_k^{(3)}(t) \cdot w_{jk}^{(3)}(t) \cdot \psi'_{a,b}\left(\frac{net_j^{(2)}(t) - b_j(t)}{a_j(t)}\right) \\ &\cdot \frac{1}{a_j(t)} \cdot o_i^{(1)}(t) \end{aligned} \quad (33)$$

$$\begin{aligned} \frac{\partial E(t)}{\partial a_j(t)} &= \sum_{k=1}^L \delta_k^{(3)}(t) \cdot w_{jk}^{(3)}(t) \cdot \psi'_{a,b}\left(\frac{net_j^{(2)}(t) - b_j(t)}{a_j(t)}\right) \\ &\cdot \left(-\frac{net_j^{(2)}(t) - b_j(t)}{a_j^2(t)}\right) \end{aligned} \quad (34)$$

$$\frac{\partial E(t)}{\partial b_j(t)} = \sum_{k=1}^L \delta_k^{(3)}(t) \cdot w_{jk}^{(3)}(t) \cdot \Psi'_{ab}\left(\frac{net_j^{(2)}(t) - b_j(t)}{a_j(t)}\right)$$

$$\times \left(-\frac{1}{a_j(t)}\right) \quad (35)$$

$$\delta_k^{(3)}(t) = error(t) \cdot sgn\left(\frac{\partial y(t)}{\partial u(t)}\right) \cdot \frac{\partial u(t)}{\partial o_k^{(3)}(t)} \cdot g'\left(net_k^{(3)}(t)\right) \quad (36)$$

$$g(x) = \frac{e^x}{e^x + e^{-x}}, \quad g'(x) = \frac{2}{(e^x + e^{-x})^2} \quad (37)$$

### E. IMPLEMENT OF MODEL

The pseudocode of WNN-PID can be designed as Method 3:

#### Method 3 WNN-PID

1. Begin
2. Initialize WNN-PID control system model;
3. Initialize WNN weights  $W_{mn}(t)$  and offsets  $b_l(t)$ ;
4. While ( $t \leq max\_t$ ) do:
  5. Update the simulation time:  $t = t + 1$ ;
  6. System input  $rin(t)$  is updated;
  7. Calculate  $error(t)$  and  $d_{error}(t)$ ;
  8. Adjust  $W_{mn}(t)$  and  $b_l(t)$  according to (28) to (37);
  9. Adjust  $K_p(t), K_i(t), K_d(t)$  according to (20) to (27);
  10. Calculate  $\Delta u(t)$  and  $u(t)$  according to Eq. (3);
  11. Calculate  $yout(t)$  according to Eq. (4);
  12. End while
  13. End

### IV. MR-WNN-PID (PROPOSED MODEL)

Section IV is organized as follows: In part A, the multiple regression-based (MR) predictor is proposed. In part B, MR-WNN-PID is proposed based on MR predictor and WNN-PID. Structure, formula and pseudocode of MR-WNN-PID are described in detail.

#### A. DESIGN OF MRP

MR predictor is proposed in order to improve the performance of control system such as control speed, steady-state error and anti-interference ability, etc. Multiple regression is a classic regression method with multiple inputs and single output. In this study, MR predictor is proposed to solve the problems of vector time series prediction.

The control parameters will be calculated according to the predicted output of MR predictor, which means the current system output  $yout(t)$  at time  $t$  is replaced by the predicted output  $yout(t+p)$  at time  $t+p$ .

#### 1) MODEL OF MR PREDICTOR

The input series of MR predictor at time  $t$  are expanded to series at time  $t, t-3$  and  $t-6$ . Therefore, more time series of control system with more historic information are participated in the prediction.

The input of the MR predictor at time  $t, t-3, t-6$  are as follows:

- (1) Control target:  $rin(t), rin(t-3), rin(t-6)$ ;
- (2) Control parameter:  $K_p(t), K_i(t), K_d(t), K_p(t-3), K_i(t-3), K_d(t-3), K_p(t-6), K_i(t-6), K_d(t-6)$ ;

- (3) Control variable, which is the output of controller:  $u(t)$ ,  $u(t-3)$ ,  $u(t-6)$ , and the output of the control system:  $yout(t)$ ,  $yout(t-3)$ ,  $yout(t-6)$

The vector time series can be expressed as Eq. (38):

$$\begin{aligned} X_n = \{ &rin(t), K_p(t), K_i(t), K_d(t), u(t), yout(t), \\ &rin(t-3), K_p(t-3), K_i(t-3), K_d(t-3), \\ &u(t-3), yout(t-3), rin(t-6), K_p(t-6), \\ &K_i(t-6), K_d(t-6), u(t-6), yout(t-6) \} \quad (38) \end{aligned}$$

The predicted output  $yout(t+p)$  of  $p$  steps predictive control system can be predicted according to multiple regression prediction. For example, the output  $yout(t+10)$  of MRP at future time  $t+10$  can be expressed as Eq. (39):

$$\begin{aligned} yout(t+10) = &\beta_0 + \beta_1 rin(t) + \beta_2 K_p(t) + \beta_3 K_i(t) + \beta_4 K_d(t) \\ &+ \beta_5 u(t) + \beta_6 yout(t) + \beta_7 rin(t-3) \\ &+ \beta_8 K_p(t-3) + \beta_9 K_i(t-3) + \beta_{10} K_d(t-3) \\ &+ \beta_{11} u(t-3) + \beta_{12} yout(t-3) \\ &+ \beta_{13} rin(t-6) + \beta_{14} K_p(t-6) \\ &+ \beta_{15} K_i(t-6) + \beta_{16} K_d(t-6) + \beta_{17} u(t-6) \\ &+ \beta_{18} yout(t-6) \quad (39) \end{aligned}$$

$\beta_0$  is a constant, and  $\beta_1, \beta_2, \dots, \beta_{18}$  are the regression coefficients.

## 2) MODEL FITTING OF MR PREDICTOR

The above model of MRP is solved by Ordinary least squares (OLS) method. The target of solving process is to find the minimal residual sum of square (RSS), which can be expressed as Eq. (40):

$$\sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (yout(t+10) - \beta_0 - \beta_1 rin(t) - \beta_2 K_p(t) - \beta_3 K_i(t) - \beta_4 K_d(t) - \beta_5 u(t) - \beta_6 yout(t) - \beta_7 rin(t-3) - \beta_8 K_p(t-3) - \beta_9 K_i(t-3) - \beta_{10} K_d(t-3) - \beta_{11} u(t-3) - \beta_{12} yout(t-3) - \beta_{13} rin(t-6) - \beta_{14} K_p(t-6) - \beta_{15} K_i(t-6) - \beta_{16} K_d(t-6) - \beta_{17} u(t-6) - \beta_{18} yout(t-6))^2 \quad (40)$$

In order to find out the minimal RSS, one-order partial derivative conditions are calculated, and the normal equations of MR model are as follows:

$$\left\{ \begin{array}{l} \sum_{N=1}^N (yout(t+10) - \beta_0 - \beta_1 rin(t) - \dots - \beta_{18} yout(t-6)) = 0 \\ \sum_{N=1}^N (yout(t+10) - \beta_0 - \beta_1 rin(t) - \dots - \beta_{18} yout(t-6)) rin(t) = 0 \\ \vdots \\ \sum_{N=1}^N (yout(t+10) - \beta_0 - \beta_1 rin(t) - \dots - \beta_{18} yout(t-6)) yout(t-6) = 0 \end{array} \right. \quad (41)$$

Therefore, the best parameters matrix  $\beta$  of MR model are also the OLS estimators which satisfy the normal equations. After matrix calculation,  $\beta$  can be expressed as Eq.(42):

$$\begin{aligned} \beta &= (\beta_0, \beta_1, \beta_2, \dots, \beta_{18})' \\ &= (X_n' X_n)^{-1} X_n' yout(t+10) \end{aligned} \quad (42)$$

## B. DESIGN OF MR-WNN-PID

### 1) STRUCTURE OF MODEL

The structure of MR-WNN-PID control system can be drawn as Fig.3. MR-WNN-PID is designed by combining MR predictor and WNN-PID algorithm. The difference between MR-WNN-PID and WNN-PID is that: the system output of WNN-PID at time  $t$  is replaced by the predicted system output of MR predictor at time  $t+p$  in MR-WNN-PID.

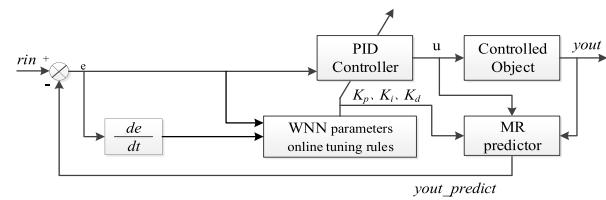


FIGURE 3. Structure of MR-WNN-PID control system.

### 2) IMPLEMENT OF MODEL

The pseudocode of the MR-WNN-PID is presented as Method 4:

#### Method 4 MR-WNN-PID

1. Begin
2. Initialize the control system of MR-WNN-PID;
3. Initialize weights  $W_{mn}(t)$  and offsets  $b_l(t)$  of WNN;
4. While ( $t \leq max\_t$ ) do:
  5. Update the simulation time:  $t = t + 1$ ;
  6. System input  $rin(t)$  is updated;
  7. If predict begin==1 do:
    8. Solve parameters of MRP according to (39);
    9. Calculate  $yout\_predict(t)$  according to MRP;
    10. Calculate  $error(t)$  according to  $yout\_predict(t)$
  11. Else
    12. Calculate  $error(t)$  according to  $yout(t)$ ;
  13. End If
  14. Adjust  $W_{mn}(t)$  and  $b_l(t)$  according to (28) to (37);
  15. Adjust  $K_p(t)$ ,  $K_i(t)$ ,  $K_d(t)$  according to (20) to (27);
  16. Calculate  $\Delta u(t)$  and  $u(t)$  according to Eq. (3) ;
  17. Calculate  $yout(t)$  according to Eq. (4);
  18. End while
  19. End

## V. SIMULATION AND COMPARATIVE ANALYSIS

Section V is organized as follows: In part A, general design of simulation is provided. In part B, standard (unsaturated)

simulation is provided. In part C, anti-interference (saturated) simulation is provided.

#### A. GENERAL DESIGN OF SIMULATION

In this study, 2 types of simulations are designed:

- (1) standard (unsaturated) simulation: this simulation aims to verify the feasibility (stability and convergence), performance (dynamic speed and steady-state error) and anti-interference ability of each algorithm. Standard simulation is an unsaturated simulation, which is discussed in the previous section.
- (2) anti-interference (saturated) simulation: this simulation aims to verify the stability (whether the algorithm can converge after long-term interference). Meanwhile, Saturated simulation proves that the system output is unsaturated in standard simulation.

Same configurations of both above 2 types of simulations are designed as follows:

- (1) In each AC, parameters are adjusted once;
- (2) Incremental digital PID algorithm is applied to implement the PID system;
- (3) The model of controlled object is  $\text{sys}(s) = \frac{180}{1.5s^2+25s+1} + e^{-5s}$ ;
- (4) Sampling period is set as 1 ( $ts = 1$  second);
- (5) The input of control system is set as  $rin(t) = 1$ , which can make the system output unsaturated;
- (6) Simulations process is repeated 10 times (10 SPs) to find the statistical characteristics.

#### B. STANDARD (UNSATURATED) SIMULATION

##### 1) SIMULATION DESIGN

Special configuration of standard (unsaturated) simulation is as follows: 1) Maximal simulation time is set as 8000 ( $\text{max\_t} = 8000$ ); 2) The start time of interference is at  $d\_t = 4000$ ; 3) The interference duration is set as  $I\_t = 10$ ; 4) Interference intensity is set as  $dst\_u(t) = 0.5$ , which is added to the output of the PID controller. In summary: from simulation time 4500 to 4510, the control variable (output of PID controller) is changed to  $u(t) + dst_u(t)$ .

##### 2) SIMULATION AND RESULT

Simulation results of 10 SPs of BPNN-PID, WNN-PID, and MR-WNN-PID are drawn in Fig.4.

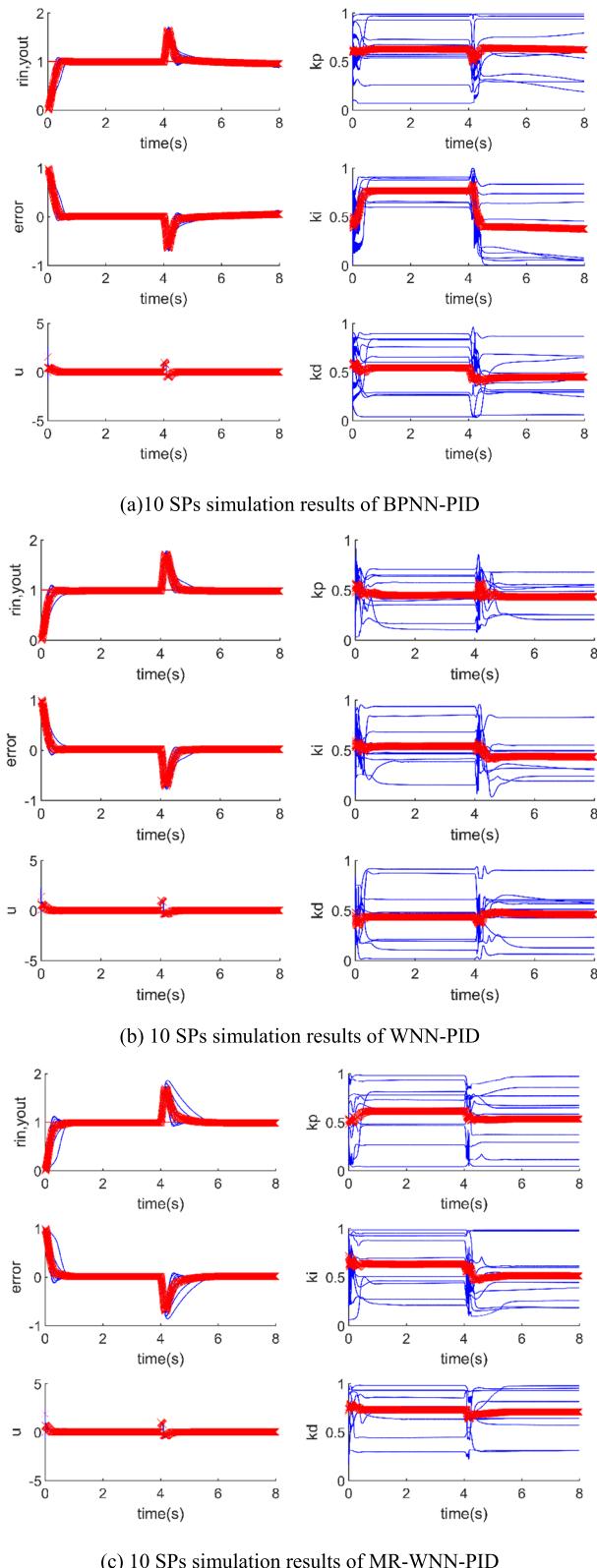
Curves of system input, output, error, output of controller, parameters of  $K_p$ ,  $K_i$ ,  $K_d$ , and simulation time are plotted.

The details of 10 SPs (each SP has 8000ACs,  $\text{max\_t} = 8000$ ) of BPNN-PID, WNN-PID and MR-WNN-PID are shown in Table 1.

Table 1 includes 6 indices (Rising Time, Settling Time, Steady-state error, overshoot, max deviation, Recovery time of deviation ) of 3 AI-CC control method.

#### 3) COMPARATIVE ANALYSIS

According to Fig.4 and Table 1, results of anti-interference (unsaturated) simulations of BPNN-PID, WNN-PID and



**FIGURE 4. 10 SPs simulations of 3 algorithms**

MR-WNN-PID are provided separately for the following comparative analyses:

### a: COMPARATIVE ANALYSIS ABOUT CONTROL SPEED

MR-WNN-PID method has the fastest speed (smallest rising time and settling time, mean of  $Tr = 263$  and mean of  $Ts = 331$ ) than BPNN-PID (mean of  $Tr = 290$ , mean of  $Ts = 357$ ), while WNN-PID has minimal settling time (mean of  $Ts = 356$ ). In addition, minimal rising time and settling time of BPNN-PID (mean of  $Tr = 290$ , mean of  $Ts = 357$ ) are reduced by both WNN-PID (minimal of  $Tr = 162$ , minimal of  $Ts = 206$ ), and MR-WNN-PID (minimal  $Tr = 143$ , minimal  $Ts = 185$ ) algorithm.

### b: COMPARATIVE ANALYSIS ABOUT OVERSHOOT

Several results of WNN-PID and MR-WNN-PID have no overshoot in 10SPs, but all the results of BPNN-PID have overshoots in each SP. Besides, the overshoot of WNN-PID (mean of peak values of overshoot=0.0321) is smaller than BPNN-PID (mean of peak values of overshoot =0.035)

## C. ANTI-INTERFERENCE (SATURATED) SIMULATION

### 1) SIMULATION DESIGN

Special configuration of anti-interference (saturated) simulation is as follows: 1) Maximal simulation time is set to 10000 ( $\max\_t = 10000$ ), which support enough time to show the effects of interference; 2) The start time of interference is to 3000 ( $d\_t = 3000$ ); 3) The interference duration is set to 2000 ( $I\_t= 2000$ ) which is a long-term interference; 4) Interference intensity (amplitude or strength of interference) is set to 0.5 ( $dst\_u(t) = 0.5$ ), which is added to the output of the PID controller, and the amplitude of interference is half of the system input. From simulation time 3000 to 5000, the control variable (output of PID controller) is changed into  $u(t) + dst\_u(t)$ .

### 2) SIMULATION AND RESULT

The results of both standard simulation and unsaturated simulation of BPNN-PID, WNN-PID and MR-WNN-PID are drawn in Fig. 5.

### 3) COMPARATIVE ANALYSIS

Results of the above anti-interference simulations show that:

- 1) **Comparative analysis about overshoot:** WNN-PID and MR-WNN-PID have smaller overshoot than BPNN in both standard and unsaturated simulations. This result is shown in Fig. 5(a) and Fig. 5(b).
- 2) **Comparative analysis about anti-interference:** The deviation of BPNN-PID after interference is greater than WNN-PID and MR-WNN-PID, and it cannot converge after interference. This result shows the BPNN-PID is most affected by interference. In contrast, WNN-PID and MR-WNN-PID have lower deviation after interference, and they can converge in the end. This result is shown in Fig.5(a).
- 3) **Comparative analysis about steady-state error:** With long-term interference, WNN-PID and MR-WNN-PID have smaller steady-state error after the interference.

BPNN-PID has the greatest steady-state error. This result is shown in Fig.5(a).

## VI. THEORETICAL ANALYSIS FOR STABILITY

Section VI is organized as follows: In part A, the basis of theoretical analysis is described. In part B, theoretical analysis of stability is provided. In part C, the example of stability analysis is provided. In part D, the unstable AI-CC system is improved, which includes the detail of RBFNN-PID method and the improved RBFNN-S-PID method.

### A. BASIS OF THEORETICAL ANALYSIS

Based on the above discussions, BPNN-PID, RBFNN-PID and WNN-PID are AI-CC method, which combine NN method with classical PID control method. Therefore, all the system analysis methods of classical PID theories are applied to NN based method.

The research scale of controlled objects is limited to the common practical applications of classical PID control. In each AC, the control parameters ( $K_p(t), K_i(t), K_d(t)$ ) are fixed value. Based on maturely recognized PID theories, AI-CC method can be converted to the linear time-invariant and approximate linear time-invariant system [28]. Moreover, stability analysis for high-order systems, delay systems, and inertial systems are discussed, and a specific stability analysis case for second-order delay system is provided as well.

According to the classical PID theories, the theoretical analysis and simulation of AI-CC can also be applied for multi-order systems with delay linear time-invariant and approximate linear time-invariant systems.

### B. THEORETICAL ANALYSIS OF STABILITY

The transfer functions of AI-CC system include open-loop transfer function and closed-loop transfer function, which are expressed as Eq. (43) to (46).

$$G_{Ctr}(s) = K_p + K_i \cdot s + K_d \cdot \frac{1}{s} \quad (43)$$

$$G_{Obj}(s) = \frac{K}{\tau \sigma s^2 + (\tau + \sigma) s + 1} \cdot e^{-\tau s} \quad (44)$$

$$G_{Open}(s) = G_{Ctr}(s) \cdot G_{Obj}(s) \quad (45)$$

$$G_{Close}(s) = \frac{G_{Open}(s)}{1 + G_{Open}(s)} \quad (46)$$

It is assumed that  $G_{Obj}(s)$  can be divided into 2 parts:  $G_1(s) = e^{-\tau s}$ , the delay section, and  $G_2(s) = \frac{M(s)}{N}$ , the system model.  $G_{Close}(s)$  represents closed-loop system, which can be expressed as  $G_{Close}(s) = \frac{M(s) \cdot e^{-\tau s}}{N(s) + M(s) \cdot e^{-\tau s}}$ , and the frequency characteristic of open loop system is  $G_k(j\omega) = \frac{M(j\omega)}{N(j\omega)} \cdot e^{-j\omega t}$ . If  $G_2(s)$  is calculated as  $G_2(j\omega) = A(\omega) \cdot e^{-j\theta(\omega)}$ ,  $G_{Obj}(s)$  can be expressed as  $G_{Obj}(s) = A(\omega) \cdot e^{-j[\theta(\omega) - \omega t]}$ .

The equivalent criterions of Nyquist stability criterion are as followsč

- 1) When  $L(\omega) = 0dB$ , if  $\varphi(\omega) > -\pi$ , the system is stable.

**TABLE 1.** Detail of 10 SPs simulations of 3 algorithms.

A. RISING TIME, SETTLING TIME, STEADY-STATE ERROR STATISTICS OF 10 SIMULATIONS

Comparison→SP number↓	BPNN-PID Tr	BPNN-PID Ts	BPNN-PID St-Err	WNN-PID Tr	WNN-PID Ts	WNN-PID St-Err	MR-WNN-PID Tr	MR-WNN-PID Ts	MR-WNN-PID St-Err
1	324	379	0.0088	199	249	0.0139	489	701	0.0149
2	250	300	0.0076	684	767	0.0415	150	193	0.0071
3	259	319	0.0102	313	374	0.0097	245	299	0.0130
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	471	569	0.0074	173	228	0.0071	333	417	0.0238
minimum	179	228	0.0073	<u>162(improved)</u>	<u>206(improved)</u>	<u>0.0071(improved)</u>	<u>143(improved)</u>	<u>185(improved)</u>	<u>0.0067(improved)</u>
mean	290	357	0.0088	296	<u>356(improved)</u>	0.0152	<u>263(improved)</u>	<u>331(improved)</u>	0.0134
maximum	471	569	0.0111	684	767	0.0415	570	701	0.0307

B. OVERSHOT, MAX DEVIATION, RECOVERY TIME STATISTICS OF 10 SIMULATIONS

Comparison→SP number↓	BPNN-PID Overshoot	BPNN-PID Max deviation	BPNN-PID Recovery time	WNN-PID Overshoot	WNN-PID Max deviation	WNN-PID Recovery time	MR-WNN-PID Overshoot	MR-WNN-PID Max deviation	MR-WNN-PID Recovery time
1	0.0204	1.6318	266	0.0487	1.7378	354	No	1.7394	598
2	0.0434	1.6115	188	No	1.7817	830	0.1095	1.6102	202
3	0.0320	1.6846	1074	0.0112	1.6927	262	0.0190	1.7045	363
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10	0.0293	1.6052	186	0.0965	1.6168	207	No	1.8617	1211
minimum	0.0160	1.6052	186	<u>No(improved)</u>	1.6168	207	<u>No(improved)</u>	1.6052	187
mean	0.0357	1.6487	460	<u>0.0321(improved)</u>	1.7143	384	0.0583	1.7106	477
maximum	0.0820	1.7072	1074	0.0965	1.7817	830	0.1222	1.8617	1211

2) When  $\varphi(\omega) = -\pi$ , if  $L(\omega) < 0dB$ , the system is stable.

Therefore, stability of control system can be judged according to the following steps, which is defined as judging method for stability (abbreviated as AI-S-CC method) in this study:

- 1) In order to ensure the real-time stability of the system, only the stable parameters (which are tuned by AI-CC methods and judged to make the system stable) are accepted;
- 2) All the unstable parameters are rejected. In this case, the previous stable parameters are retained and used continuously.

### C. EXAMPLE OF STABILITY ANALYSIS

For example, if the controlled object model is  $G_{Obj}(s) = \frac{280}{s^2 + 25s + 1} + e^{-2s}$ , the controller model is  $G_{Ctr}(s) = k_p + \frac{k_i}{s} + k_d \cdot s = \frac{k_d \cdot s^2 + k_p \cdot s + k_i}{s}$ , and the current time is  $t$ . The control parameters at time  $t$  are as follows:  $K_p(t) = 0.2$ ,  $K_i(t) = 0.2$  and  $K_d(t) = 0.2$ , which are pre-verified to stabilize the control system. Then the open-loop transfer function  $G_{Open}(s)$  can be calculated.

According to the open-loop transfer function  $G_{Open}(s)$ , the amplitude margin can be calculated as:  $Gm = 20 \cdot$

$\log_{10}(gm) = -2.98dB < 0$ , and phase margin can be calculated as  $Pm = pm = -93.7 \text{ deg} > -\pi$ .

In each AC, amplitude and phase are calculated, the pseudocode is provided as Method 5, which can be implemented by different programming languages:

#### Method 5 Calculation of Amplitude and Phase

```

1.sys=tf(180,[1.5,25,1],'inputdelay',5);
2.sys_pid=tf([kd2bc,kp2bc,ki2bc],[1,0]);
3.sys_open=sys*sys_pid;
4.[bode_mag,bode_phase,bode_w]=bode(sys_open);
5.[bode_gm,bode_pm,bode_wcg,bode_wcp]=margin(bode
_mag,bode_phase,bode_w);

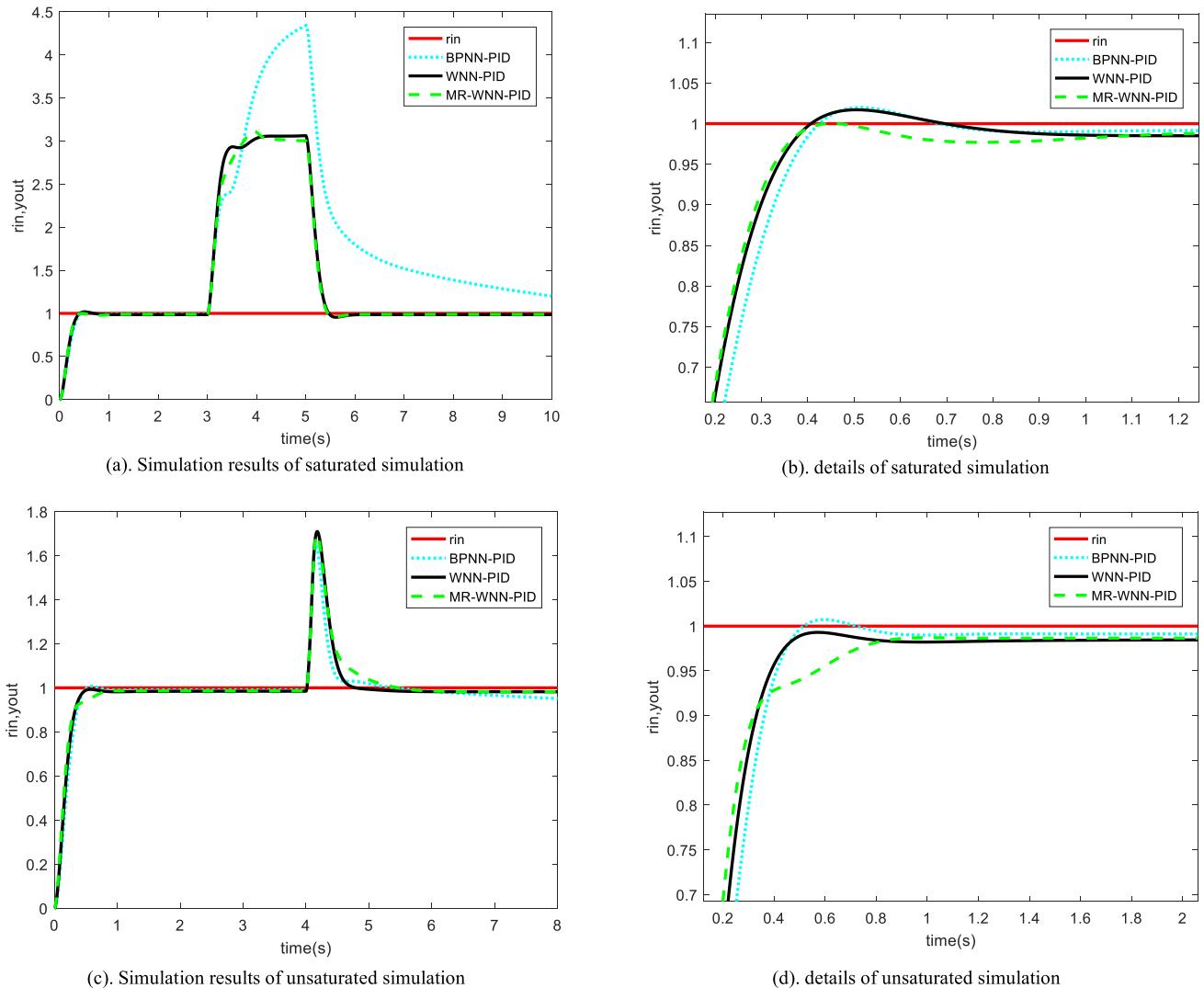
```

The bode plots of the control system with the latest  $K_p(t)$ ,  $K_i(t)$ ,  $K_d(t)$  can be plotted in Fig.6. According to the above theoretical analysis, the current online tuned parameters  $K_p(t)$ ,  $K_i(t)$ ,  $K_d(t)$  can make the system work steadily, so the current online tuned parameters  $K_p(t)$ ,  $K_i(t)$ ,  $K_d(t)$  can be adopted (not abandoned) for PID controller.

### D. IMPROVEMENT OF UNSTABLE AI-CC SYSTEM

#### 1) RBFNN-PID MODEL (UNSTABLE MODEL)

RBFNN-PID is studied in this section, because RBFNN-PID is an unstable AI-CC system. Assume that  $X$  represents the

**FIGURE 5. Results of saturated and unsaturated simulation.**

input vector of RBFNN and  $y_m(k)$  represents the output of RBFNN.

$$X = [net_1^{(1)}(t), net_2^{(1)}(t), net_3^{(1)}(t)]^T \quad (47)$$

$$h_j = \exp\left(\frac{\|X - C_j\|^2}{2b_j^2}\right), \quad (j = 1, 2, \dots, m) \quad (48)$$

$$y_m(k) = w_1 h_1 + w_2 h_2 + \dots + w_m h_m \quad (49)$$

The detailed parameters of gradient descent method are as follows:  $\Delta k_p$ ,  $\Delta k_i$ ,  $\Delta k_d$  are changed as shown in Eq. (50) to (56).

$$\begin{aligned} \Delta k_p &= -\eta \frac{\partial E}{\partial k_p} = -\eta \frac{\partial E}{\partial k_p} \frac{\partial y}{\partial \Delta u} \frac{\partial \Delta u}{\partial k_p} \\ &= \eta \cdot error(k) \frac{\partial y}{\partial \Delta u} xc(1) \end{aligned} \quad (50)$$

$$\begin{aligned} \Delta k_i &= -\eta \frac{\partial E}{\partial k_i} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial \Delta u} \frac{\partial \Delta u}{\partial k_i} \\ &= \eta \cdot error(k) \frac{\partial y}{\partial \Delta u} xc(2) \end{aligned} \quad (51)$$

$$\begin{aligned} \Delta k_d &= -\eta \frac{\partial E}{\partial k_d} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial \Delta u} \frac{\partial \Delta u}{\partial k_d} \\ &= \eta \cdot error(k) \frac{\partial y}{\partial \Delta u} xc(3) \end{aligned} \quad (52)$$

$$xc(1) = error(k) - error(k-1) \quad (53)$$

$$xc(2) = error(k) \quad (54)$$

$$xc(3) = error(k) - 2error(k-1) + error(k-2) \quad (55)$$

$$\frac{\partial y(k)}{\partial \Delta u(k)} = \sum w_j h_j \frac{C_{ji} - x_1}{b_j^2} \quad (56)$$

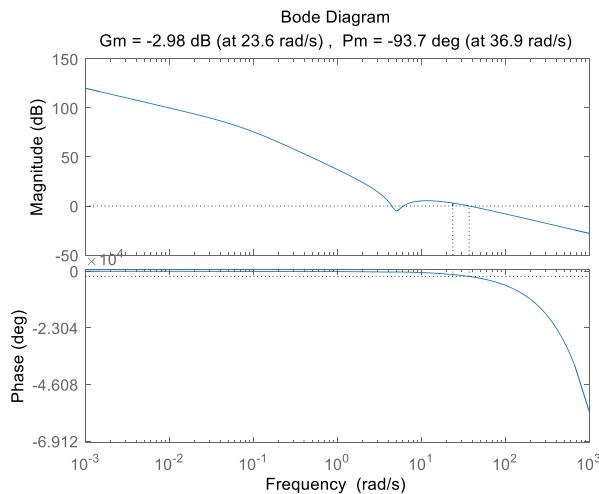
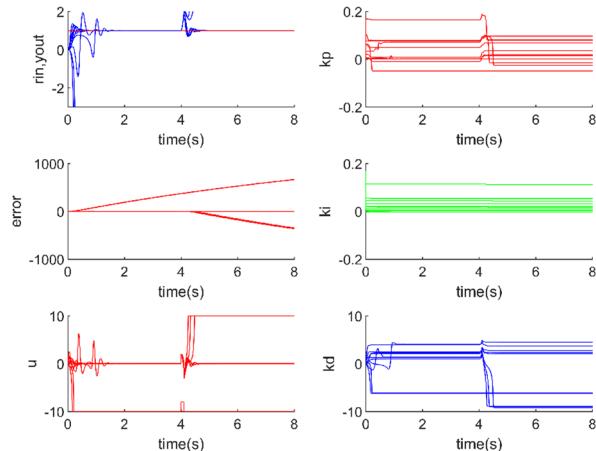
The simulation result of RBFNN-PID is shown in Fig.7. Compare with other methods, RBFNN-PID is unstable. RBFNN-PID has serious deviation and it cannot converge in the end.

## 2) IMPROVED RBFNN-S-PID (STABLE MODEL)

In order to solve the above problem, RBFNN-S-PID method is proposed. In RBFNN-S-PID, the control parameters are judged according to the above AI-S-CC method. In RBFNN-

**TABLE 2.** Comparison of different AI-CC systems.

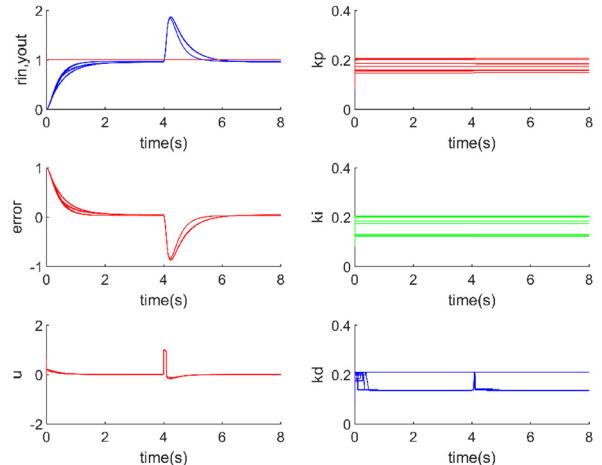
Method→ Comparison Item↓	BPNN- PID[24],[25] (existing method)	WNN-PID (proposed method)	MR-WNN-PID (proposed method)	RBFNN- PID[24],[25] (existing method)	RBFNN-S-PID (proposed method)
Stability analysis (Convergence)	Stable: 100%	Stable: 100%	Stable: 100%	70% stable	100% Stable
Control Speed (means of $T_r$ and $T_s$ ) [minimum of $T_r$ and $T_s$ ]	(290,357) [179,228]	<u>Better</u> <u>(296,356)</u> <u>[162,206]</u>	<u>Best</u> <u>(263,331)</u> <u>[143,185]</u>	-	-
Tracking ability (means of Steady-state error)	(0.0088)	(0.0152)	<u>Better than WNN-PID</u> <u>(0.0134)</u>	-	-
Feature of overshoot (means of overshoot)	(0.0357)	<u>Best</u> <u>(0.0321)</u> <u>Some results</u> <u>have no overshoot</u>	(0.0583) <u>Some results</u> <u>have no overshoot</u>	-	-
Anti-interference speed (means of recovery time)	(460)	<u>Fastest</u> <u>(384)</u>	(477)	-	-

**FIGURE 6.** Bode plot result.**FIGURE 7.** Simulation Result of RBFNN-PID.

S-PID, the adjusted  $K_p(t)$ ,  $K_i(t)$  and  $K_d(t)$  are judged in each AC. The effect of the improved RBFNN-S-PID is shown in Fig.8.

Comparing Fig.7 with Fig.8, the RBFNN-S-PID system is much more stable: 1) All the RBFNN-S-PID results are convergent; 2) RBFNN-S-PID has less overshoot.

Throughout the online tuning process, all adjusted parameters are judged. RBFNN-PID is unstable, but RBFNN-S-PID with AI-CC-S method is stable. From the above theoretical analysis and simulation, stability problem is solved.

**FIGURE 8.** 10SPs Simulation for unstable RBFNN-PID-S systems.

## VII. CONCLUSION

In this study, WNN-PID and MR-WNN-PID are proposed, which improve the performance of existing BPNN-PID and RBFNN-PID methods. The stability analyses of the above methods are implemented, and the unstable problem of RBFNN-PID is solved. Comparative simulation results among BPNN-PID, WNN-PID and MR-WNN-PID methods are summarized in Table 2.

Comparative indicators are designed as follows: control speed characteristic is measured by  $T_r$  and  $T_s$  indicator.

Capability of output tracking input is measured by steady-state error. Contents of parameter configuration work, workload and operating time are estimated, hence the intelligent level of AI-CC system and the ability requirements for engineers can be evaluated.

According to Table 2, conclusions of this study can be summarized as follows:

**1) The improvements of the proposed WNN-PID method are as follows:** WNN-PID method is a stable (100% stable) and available control method in both saturated and unsaturated test. The control speed of WNN-PID method (means of  $Tr = 296$ , means of  $Ts = 356$ ) is better than BPNN-PID method (means of  $Tr = 290$ , means of  $Ts = 357$ ), but it is lower than MR-WNN-PID method (means of  $Tr = 263$ , means of  $Ts = 331$ ). The steady-state error of WNN-PID (steady-state error=0.0152) is bigger than both BPNN-PID method (steady-state error=0.0088) and MR-WNN-PID method (steady-state error=0.0134). WNN-PID has the fastest recovery speed (anti-interference), and it can converge in the end (recovery time=384). The effect of overshoot inhibition of WNN-PID (overshoot=0.0321) is the best, and WNN-PID also have several simulation results without overshoot in 10 SPs.

**2) The improvements of the proposed MR-WNN-PID method are as follows:** MR-WNN-PID method is a stable and feasible method in both saturated and unsaturated test. MR-WNN-PID method has the fastest control speed (means of  $Tr = 263$ , means of  $Ts = 331$ ). The steady-state error of MR-WNN-PID method (steady-state error=0.0134) is smaller than WNN-PID, but bigger than BPNN-PID. the recovery speed of MR-WNN-PID method (recovery time=477) is longer than BPNN-PID (recovery time=460) and WNN-PID (recovery time=384), but it can converge in the end. MR-WNN-PID method has several simulation results without overshoot in 10 SPs.

**3) The improvement of the proposed RBFNN-S-PID:** RBFNN-S-PID is stable after combining RBFNN-S-PID and AI-CC-S method, while RBFNN-S-PID is unstable.

*Future Work Are as Follows:* More accurate predictors are expected to further improve the performances of AI-CC methods. For example, Pythagorean fuzzy interaction power Bonferroni mean (PBM) operator and weighted Pythagorean fuzzy interaction PBM operator for multiple attribute decision making (MADM) methods [28] can be applied to AI-CC methods. These two new operators considered the relationship between membership and non-membership function of Pythagorean fuzzy numbers (PFNs), which improved the accurate result of the intelligent MADM system. Moreover, better methods of configuration and initialization for the above optimization and neural networks are expected to further improve performance of AI-CC control system.

## REFERENCES

- [1] Y. V. Hote and S. Jain, "PID controller design for load frequency control: Past, present and future challenges," *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 604–609, 2018.
- [2] A. A. El-samahy and M. A. Shamseldin, "Brushless DC motor tracking control using self-tuning fuzzy PID control and model reference adaptive control," *Ain Shams Eng. J.*, vol. 9, no. 3, pp. 341–352, Sep. 2018.
- [3] M. Rennela and S. Staton, "Classical control and quantum circuits in enriched category theory," *Electron. Notes Theor. Comput. Sci.*, vol. 336, pp. 257–279, Apr. 2018.
- [4] P. P. Khargonekar and M. A. Dahleh, "Advancing systems and control research in the era of ML and AI," *Annu. Rev. Control*, vol. 45, pp. 1–4, Apr. 2018.
- [5] N. Ernest, D. Carroll, C. Schumacher, M. Clark, K. Cohen, and G. Lee, "Genetic fuzzy based artificial intelligence for unmanned combat aerial vehicle control in simulated air combat missions," *J. Defense Manage.*, vol. 6, no. 1, pp. 0374–2167, 2016.
- [6] T. Xu, G. Li, Z. Yu, J. Zhang, L. Wang, Y. Li, and D. Li, "Design and application of emergency coordination control system for multi-infeed HVDC receiving-end system coping with frequency stability problem," *Autom. Electr. Power Syst.*, vol. 41, no. 8, pp. 98–104, 2017.
- [7] D. M. Prett, C. E. Garc, and B. L. Ramaker, *The Second Shell Process Control Workshop: Solutions to the Shell Standard Control Problem*. Amsterdam, The Netherlands: Elsevier, 2017, pp. 376–399.
- [8] Y. Shi, C. Shen, H. Fang, and H. Li, "Advanced control in marine mechatronic systems: A survey," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1121–1131, Jun. 2017.
- [9] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 935–947, Feb. 2017.
- [10] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Comput. Chem. Eng.*, vol. 23, nos. 4–5, pp. 667–682, May 1999.
- [11] L. Wang, H. Dan, Y. Zhao, Q. Zhu, T. Peng, Y. Sun, and P. Wheeler, "A finite control set model predictive control method for matrix converter with zero common-mode voltage," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 6, no. 1, pp. 327–338, Mar. 2018.
- [12] H. Wang, W. Sun, and P. X. Liu, "Adaptive intelligent control of nonaffine nonlinear time-delay systems with dynamic uncertainties," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 7, pp. 1474–1485, Jul. 2017.
- [13] Y. Li, S. Tong, L. Liu, and G. Feng, "Adaptive output-feedback control design with prescribed performance for switched nonlinear systems," *Automatica*, vol. 80, pp. 225–231, Jun. 2017.
- [14] S. Tong, Y. Li, and S. Sui, "Adaptive fuzzy tracking control design for SISO uncertain nonstrict feedback nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 6, pp. 1441–1454, Dec. 2016.
- [15] Y.-J. Liu and S. Tong, "Barrier Lyapunov functions for Nussbaum gain adaptive control of full state constrained nonlinear systems," *Automatica*, vol. 76, pp. 143–152, Feb. 2017.
- [16] S. N. Vassilyev, A. Y. KelinaM, Y. I. Kudinov, and F. F. Pashchenko, "Intelligent control systems," *Procedia Comput. Sci.*, vol. 103, pp. 623–628, Feb. 2017.
- [17] A. Kumar and V. Kumar, "Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator," *Expert Syst. Appl.*, vol. 73, pp. 161–177, May 2017.
- [18] B. K. Sahu, T. K. Pati, J. R. Nayak, S. Panda, and S. K. Kar, "A novel hybrid LUS-TLBO optimized fuzzy-PID controller for load frequency control of multi-source power system," *Int. J. Electr. Power Energy Syst.*, vol. 74, pp. 58–69, Jan. 2016.
- [19] R. Hernández-Alvarado, L. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espínosa, and F. Fonseca-Navarro, "Neural network-based self-tuning PID control for underwater vehicles," *Sensors*, vol. 16, no. 9, p. 1429, Sep. 2016.
- [20] F. Khozeimeh, R. Alizadehsani, M. Roshanzamir, A. Khosravi, P. Layegh, and S. Nahavandi, "An expert system for selecting wart treatment method," *Comput. Biol. Med.*, vol. 81, pp. 167–175, Feb. 2017.
- [21] Q. X. Liu, "Design of flow control system based on expert PID," in *Proc. Int. Symp. Comput., Consum. Control (IS3C)*, Jul. 2016, pp. 1031–1034.
- [22] H. J. Live, H. B. Wang, X. M. Zhu, Z. H. Shen, and J. Y. Chen, "Simulation research of fuzzy auto-tuning PID controller based on MATLAB," in *Proc. Int. Conf. Comput. Technol., Electron. Commun. (ICCTEC)*, Dec. 2017, pp. 180–183.
- [23] S. Hou, J. Fei, C. Chen, and Y. Chu, "Finite-time adaptive fuzzy-neural-network control of active power filter," *IEEE Trans. Power Electron.*, vol. 34, no. 10, pp. 10298–10313, Oct. 2019.

- [24] R. Hernandez-Alvarado, L. G. Garcia-Valdovinos, T. Salgado-Jimenez, A. Gomez-Espinosa, and F. F. Navarro, "Self-tuned PID control based on backpropagation neural networks for underwater vehicles," in *Proc. OCEANS MTS/IEEE Monterey*, Sep. 2016, pp. 1–5.
- [25] L. Nie, J. Guan, C. Lu, H. Zheng, and Z. Yin, "Longitudinal speed control of autonomous vehicle based on a self-adaptive PID of radial basis function neural network," *IET Intell. Transp. Syst.*, vol. 12, no. 6, pp. 485–494, Aug. 2018.
- [26] P. Lei, F. Hao, W. Hai-Hua, and Z. Tian-Wei, "Variable pitch control strategy of wind power generation based on BPNN-PID algorithm," in *Proc. IEEE Sustain. Power Energy Conf. (iSPEC)*, Nov. 2019, pp. 105–112.
- [27] J. Fei and T. Wang, "Adaptive based on RBFNN control for active power filter," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 5, pp. 1139–1150, 2019.
- [28] J. W. Liu, T. Y. Li, Z. Y. Zhang, and J. M. Chen, "NARX prediction-based parameters online tuning method of intelligent PID system," *IEEE Access*, vol. 8, pp. 130922–130936, 2020.
- [29] L. Wang and N. Li, "Pythagorean fuzzy interaction power Bonferroni mean aggregation operators in multiple attribute decision making," *Int. J. Intell. Syst.*, vol. 35, no. 1, pp. 150–183, Jan. 2020.



**MENGXUAN ZHAO** was born in Beijing, China, in 1997. She received the bachelor's degree in information system and information management from the Oversea Chinese College, Capital University of Economics and Business, in 2019. She is currently pursuing the master's degree in management science and engineering with the Capital University of Economics and Business.

She received the Scholarship from the Capital University of Economics and Business, from 2016 to 2018, and won the Second Prize at the Contest of Municipal E-Commerce Plus.



**GUANCHEN LI** was born in Beijing, China, in 1999. He is currently pursuing the bachelor's degree with the School of Information, Capital University of Economics and Business.

From 2017 to 2020, he obtained a software copyrights at the Copyright Administration of the People's Republic of China, participated in publishing a book, and won the MCM Award. His research interests include machine learning, deep generation networks, and probability graph model.

Mr. Li received the First-Class Scholarship from the Capital University of Economics and Business.



**JIAMING CHEN** was born in Beijing, China, in 1994. He received the M.S. degree in software engineering from Capital Normal University, Beijing, in 2019. He is currently pursuing the Ph.D. degree in computer science with the Beijing University of Technology.

In 2018, he was a Researcher and an Assistant Tutor with the Big Data College, Huike Education Corporation. From 2016 to 2018, he was the Frontend Architect at Chengdu iDelta Technology Company, Ltd. His research interests include artificial intelligence, brain computer interface, and data mining. He received the Scholarship from Capital Normal University, from 2016 to 2018, and won the Third Prize at the Contest of Lan Qiao Cup, in 2017.



**JINGWEI LIU** was born in Beijing, China, in 1982. He received the Ph.D. degree in control science and engineering (pattern recognition and intelligent system) from the Department of Electronic Information and Control Engineering, Beijing University of Technology, Beijing, in 2014.

Since 2014, he has been an Associate Professor with the Information College, Capital University of Economics and Business. He served as the Director of the Software Research and Development Center, Information College, Capital University of Economics and Business. He is the author of three books, more than 20 articles, and more than 20 inventions. His research interests include artificial intelligence, automatic control, intelligent control, and intelligent systems.