Research Article

Model Predictive Control of Nonlinear System Based on GA-RBP Neural Network and Improved Gradient Descent Method

Youming Wang ^{1,2} and **Didi Qing** ¹

¹*School of Automation, Xi'an Key Laboratory of Advanced Control and Intelligent Process, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*

²*State Key Laboratory for Strength and Vibration of Mechanical Structure, Xi'an 710049, China*

Correspondence should be addressed to Youming Wang; xautroland@126.com

Received 1 December 2020; Revised 28 February 2021; Accepted 23 March 2021; Published 31 March 2021

Academic Editor: Thach Ngoc Dinh

Copyright © 2021 Youming Wang and Didi Qing. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A model predictive control (MPC) method based on recursive backpropagation (RBP) neural network and genetic algorithm (GA) is proposed for a class of nonlinear systems with time delays and uncertainties. In the offline modeling stage, a multistep-ahead predictor with GA-RBP neural network is designed, where GA-BP neural network is used as a one-step prediction model and GA is employed to train the initial weights and bias of the BP neural network. The incorporation of GA into RBP can reduce the possibility of the BP neural network falling into a local optimum instead of reaching global optimization. In the online optimizing stage, a multistep-ahead GA-RBP neural network predictor and an improved gradient descent method (IGDM) are proposed to efficiently solve the online optimization problem of nonlinear MPC by minimizing a modified quadratic criterion. The designed MPC strategy can avoid information loss while linearizing the controlled system and computing the Hessian matrix and its inverse matrix. Experimental results show that the proposed approach can reduce the computational burden and improve the performance of MPC (i.e., the maximum overshoots, calculation time, rise time, and RMSE tracking error value) for the solution of nonlinear controlled systems.

1. Introduction

As an advanced technique, model predictive control (MPC) including dynamic matrix control (DMC), generalized predictive control (GPC), and receding horizon control (RHC) has been commonly used in the industrial processes because of its ability to deal with constraints and time delays and adapt to various dynamic models and control objectives [1–4]. In this control strategy, the process behavior is predicted based on an approximate model, and the control sequence is obtained by solving a performance evaluation function. Compared to conventional control methods, such as the PID controllers, MPC is a desirable optimal control strategy based on linear dynamic models to solve a finite horizon open-loop optimal control problem when the operating points of the process are limited in a small area [5]. However, most of the industrial processes operate in a wide

range of operating points and exhibit severe nonlinearities, which may affect the MPC's ability to deliver better control and optimization.

The development of nonlinear MPC has been motivated by inherent nonlinearity and production requirements in various operating areas. The nonlinear MPC has fascinated enormous attention for the last two decades, which can predict the future behavior of the system via constructing a nonlinear prediction model. The effectiveness of nonlinear MPC depends on the construction of an appropriate first-principle or empirical model from industrial or laboratory data and the derivation of an accurate online linear approximation of the nonlinear model for optimization and prediction [6–8]. However, it is always challenging to generate system models that are both accurate and computationally efficient for complex and nonlinear control problems [9, 10].

Generally, there are two approaches including first-principle and data-driven models for the generation of system models. The first-principle models are derived from theory to describe the real physical behavior of a system. However, the solution of the first-principle models is often a computationally demanding task due to the need for solving the model for any given candidate value of the unknown parameters. The other models are constructed by algorithmically inferring the relationships or parameters from a dataset using machine learning without prior knowledge of the system. As increasingly promising models, many data-driven models have recently been applied to nonlinear MPC to improve the model prediction accuracy, such as Takagi-Sugeno fuzzy [11], Nonlinear Autoregressive with the eXogenous variable (NARX) [12], Support Vector Machine (SVM) [13], Volterra series [14], and Artificial Neural Network (ANN) [15–18]. Out of all these modeling methods, the ANN method employs a nonlinear empirical model that represents input-output data and can mimic process models by learning patterns from a sufficiently large dataset and thus model nonlinear systems compared to other methods. ANN is considered an excellent tool for modeling the nonlinear MPC, which can approximate any continuous and differentiable functions to an arbitrarily high degree of accuracy [19, 20]. Hornik et al. [21] proposed a fuzzy wavelet NN for MPC to model the liquid level system. Han et al. [22] presented a self-organizing NN as the prediction model of nonlinear system to perform the real-time MPC. Ławryńczuk [23] developed nonlinear MPC by using a radial basis function neural network (RBFNN). The MPC based on Elman NN for delayed dynamic systems was introduced in [24].

As one of the simple and effective feed-forward neural networks, the backpropagation neural network (BPNN) was also extensively used in nonlinear plant identification and modeling. However, the BP networks have many drawbacks, such as difficulty in choosing appropriate network architecture and training the weights of a fixed architecture and the resulting local minimum and slow convergence. To overcome these problems and improve the reliability of BPNN, many heuristic algorithms such as genetic algorithm (GA) [25], Particle Swarm Optimization (PSO) [26], and Simulated Annealing (SA) [27] are combined to avoid local minima and to achieve global convergence quickly and correctly. As the key to the problem is based on natural selection and genetics, the GA has been introduced as an evolutionary computation method that does not require problem-specific knowledge and thus becomes superior in terms of robustness and solution of nonlinear, parallel, and complex problems [28]. The optimization of the neural network architecture and the training of the weights of a fixed architecture are two important ways to be used as a robust global optimization method for neural networks [29]. The related experimental results demonstrated that the combination of GA and BPNN methods is superior to the single method in the generalization ability and convergence speed [30, 31]. Despite the promising benefits, the prediction

results of the GA-BP neural network reported become inaccurate when used iteratively over the prediction horizon. For the determination of the optimal input moves by solving an optimization problem in the nonlinear MPC framework, the capability to perform satisfactory multi-step-ahead predictions is considered to be important to provide accurate multistep-ahead predictions rather than one-step-ahead predictions [8, 32]. Hence, it is necessary to evaluate and study the multistep-ahead prediction analysis of the MPC model based on the neural network.

Despite the algorithmic developments to improve the convergence rate, it is still required to solve numerical optimization problems online at each iteration, which is a fundamental limiting factor for practical implementation of MPC due to low computational accuracy and efficiency in the process of receding horizon optimization of MPC. In the optimization process of MPC, the optimal control sequence is obtained and the first element is applied to the nonlinear system at each sampling instant. A large number of algorithms were studied to solve the control problem of nonlinear MPC. Lu [33] proposed a systematic design methodology to develop the stable predictive control (SPC) based on recurrent wavelet neural networks for the nonlinear plants. Stefan et al. [34] presented Nesterov's fast gradient method combined with multilayer neural network to accomplish the control of the nonlinear system with input constraints. Kittisupakorn et al. developed a Newton-Raphson algorithm to optimize the control problem of nonlinear MPC for a steel pickling process. Yang et al. [36] designed a nonlinear MPC optimization problem for reference tracking of dynamic positioning ships by combining the disturbance estimation into the rolling optimization.

This paper proposes a novel strategy for nonlinear MPC, in which a recursive neural network is combined with the improved gradient descent method (IGDM) to model the nonlinear dynamic system with time delays and uncertainties and solve the online optimization problem. The receding horizon optimization problem of the nonlinear system is solved by using the proposed IGDM, where the control input sequence can be calculated by solving a modified cost function. The designed control strategy can avoid information loss while linearizing the nonlinear system and solve the online optimization problem on the implementation of MPC. Two different nonlinear systems are selected to verify the abilities of modeling and tracking the proposed algorithm.

The main contributions of this paper can be summarized as follows:

- (1) To obtain an accurate prediction model, we present a recursive BP neural network optimized by GA (GA-RBP) to construct the multistep-ahead model. It can be adapted to predict the future dynamic behavior of nonlinear systems with delay times and uncertainties.
- (2) An IGDM is developed to solve the online optimization problem, which can improve the control performance and reduce the computational burden.

- (3) Combining the GA-RBP neural network predictor with online optimization based on IGDM, we propose a new online MPC algorithm for a class of nonlinear systems with time delays and uncertainties. The simulation results of the nonlinear system have proven that the proposed strategy has better robustness and tracking ability than the traditional GPC.

The rest of this paper is organized as follows. Section 2 introduces the principle of the nonlinear MPC and BPNN model. In Section 3, GA-BP neural network and multistep-ahead GA-RBP neural network predictor are described. An IGDM is proposed to address the online optimization problem of MPC in Section 4. The experimental results are presented in Section 5 and the conclusions are summarized in Section 6.

2. Preliminaries

2.1. Nonlinear Model Predictive Control (NMPC). The essence of NMPC is to predict the future output of the nonlinear dynamic process based on a nonlinear prediction model. The NMPC controller is designed to obtain the control input sequence and consists of applying a control

sequence that minimizes a multistage cost function of the form

$$J = \sum_{i=N_0}^{N_p} [y_r(t+i) - y_c(t+i)]^2 + \sum_{j=1}^{N_u} \lambda_j [\Delta u(t+j-1)]^2, \quad (1)$$

subject to

$$\begin{aligned} \Delta u_{\min} &\leq \Delta u(t) \leq \Delta u_{\max}, \\ u_{\min} &\leq u(t) \leq u_{\max}, \\ y_{c-\min} &\leq y_c(t) \leq y_{c-\max}, \end{aligned} \quad (2)$$

where y_r and y_c represent the desired reference signal and estimated output values, respectively, N_0 denotes the minimum prediction horizon, N_p is the maximum prediction horizon, N_u is the control horizon, and λ_j is a weighting factor of control input and generally chosen as zero or a small value. The only constraint for the values of N_0 and N_u is that these bounds must be less than or equal to N_p .

In the process of modeling a dynamic stochastic system, the system can be represented as a nonlinear autoregressive moving average model with exogenous input (NARMAX) [37, 38] in the form of

$$y(t) = f(y(t-1), \dots, y(t-n_y), u(t-1-t_d), \dots, u(t-n_u-t_d)) + d(t), \quad (3)$$

where f is an unknown nonlinear function, u and y denote the input and output of the process, d represents zero-mean white noise, n_u and n_y represent the maximum lags of the inputs and outputs, respectively, and t_d is the delay time.

2.2. BP Neural Network Model. The BPNN is a kind of classical feed-forward neural networks, which employs an error backpropagation algorithm in networks of neurons and utilizes the gradient descend method to adjust weights and bias. A typical BPNN architecture consists of input layers, hidden layers, and output layers. As shown in Figure 1, the input layer and the hidden layer of BPNN contain m and n neurons, respectively. The weights and bias of the BPNN are adjusted by

$$F(k) = \frac{1}{2} \sum (y(k) - y_m(k))^2, \quad (4)$$

where $y(k)$ and $y_m(k)$ are the real system output and the output from the BPNN, respectively.

In Figure 1, the basic BPNN structure contains one input layer, one hidden layer, and one output layer. I_j , $j = 1, 2, \dots, m$, H_i , $i = 1, 2, \dots, n$, and O represent each layer node of BPNN, respectively. ω_{ij} is the connecting weights between j th input node and i th hidden node. ω_i denotes the connecting weight between the i th hidden node and the output node. b_i and b represent the bias vector of the hidden layer and the output layer, respectively. H_I and H_O are the input and output of the hidden layer, respectively,

and O_O is the output of the network. The output of the BPNN can be expressed as

$$\begin{cases} H_I = b_i + \sum_{j=1}^m \omega_{ij} x_j, \\ H_O = \varphi(H_I), \\ O_O = b + \sum_{i=1}^n \omega_i H_O, \end{cases} \quad (5)$$

where φ is the sigmoid function, x_j is the input of BPNN, and the output of the one-step-ahead prediction model can be expressed as

$$y_m(k) = \sum_{i=1}^n \omega_i \times \left(1 + \exp \left(\sum_{j=1}^m \omega_{ij} x_j + b_i \right) \right)^{-1} + b. \quad (6)$$

The testing dataset is chosen randomly to determine the performance of the BPNN model and the validation dataset is used to ensure that the model does not overfit or underfit in the training stage.

In the process of modeling nonlinearities, overfitting is a typical problem for the neural network method when a model has many parameters and the neural network structure is very complex. In order to avoid overfitting, a general technique named early stopping is used to reduce overfitting with the use of a validation set to stop model training at the peak of the validation set accuracy [39]. The

FIGURE 1: The structure of BPNN.

error on the validation set is monitored during the training session. In the early stopping method, when the validation error starts to increase after a number of iterations, the training is stopped, and the weights at the minimum of the validation error are returned for optimum network complexity.

Furthermore, the problem of underfitting for the neural network method exists because the model is not well trained in the training process. The underfitting of NN means that a model has too few parameters to capture the underlying structure of the observed data and thus provides poor prediction or generalization. To solve this problem, the hyperparameters including the number of hidden neurons and layers, activation function, loss function, and regularization parameters are adjusted to prevent the underfitting of NN. To obtain an optimal neural network, grid search is employed in this paper, which is an exhaustive search approach for hyperparameter tuning [40]. In grid search, each hyperparameter is assigned with several possible values, comprising a multidimensional grid. It was reported that random grid search has a superior efficiency than full grid search theoretically and empirically [41, 42].

3. Model Predictive Control Based on GA-RBP Neural Network Model

3.1. GA-BP Neural Network. To solve the sensitivity to initial weight and easily fall into the local optimum of conventional BP algorithm, this paper combines GA with BP to establish the prediction model. GA is a global searching technique based on the mechanics of natural selection and natural genetic, which can search for the optimal weight and bias vector of the BP algorithm without falling into the local extremum [43]. Based on self-adaptive probabilistic optimization of GA and good nonlinear fitting ability of BPNN, the combined algorithm has the advantages of strong global search capability and high learning accuracy.

In the GA, the individual parameters are encoded as strings of numbers called chromosomes. The process starts by randomly initializing a random population of potential solutions; these are then evaluated using a fitness function to identify the best individuals in the population. Additionally, genetic operators such as mutation and crossover are

implemented, where the crossover is to select two random parent chromosomes and combine them to form two child chromosomes and mutation is to maintain the population diversity and to avoid convergence to a local optimum. The main characteristics that affect the convergence of the GA include population size, number of iterations, number of survivals, mutation, and crossover rates. The combination of these parameters has a major influence over the convergence time and the optimal solution. Thus, it is necessary to train the GA and find the best configuration regarding the convergence time to minimize the average difference between the GA's output and measured data. However, the number of possible parameters and their different values always means that it is very time-consuming for parameter tuning [44]. In this paper, a factorial design approach is used to design the appropriate parameters of GA, which is a well-known methodology based on statistical considerations. The description of the tuning parameters approach can be found in [45, 46].

Figure 2 shows the process of GA-BP neural network modeling, which includes the selection of BPNN structure, the optimization of GA, and the prediction of the BPNN. The procedure of the GA-BP neural network is introduced as follows. Specifically, the BPNN first reads in training data and combines with the initial training weights and bias transferred by GA to complete the first training and obtain the initial prediction model. Then, the BPNN reads in the validation data, uses the initial prediction model to complete the predictions, and evaluates the initial prediction results. If fitness convergence conditions are satisfied, the initial prediction model is the final prediction model; if not, the procedures of GA will iterate and the next group of training iterative parameters will transfer to the BPNN until the fitness convergence conditions of the prediction model are met.

3.2. Multistep-Ahead Prediction Model Based on Recursive Neural Network. Since the controlled output has to be predicted several steps ahead for the optimal control of the nonlinear system, a multistep prediction model is required [26, 32]. The methods of calculating multistep-ahead prediction for the MPC model using the NN model are based on the recursive prediction strategy, which is a successive one-step-ahead prediction whereas the outputs in each consecutive step are considered as inputs for the next step of prediction. A new GA-RBP neural network is presented as a multistep-ahead predictor to obtain the ahead predicted values of nonlinear systems. According to equation (3), the NARMAX one-step ahead model can be derived as

$$y(t+1) = f(y(t), \dots, y(t-n_y+1), u(t), \dots, u(t-n_u-t_d+1)), \quad (7)$$

where $y(t+1)$ is the one-step predicted output value of the nonlinear model. The next step $y(t+2)$ can be obtained of the form

FIGURE 2: Flowchart of the GA-BP neural network.

$$y(t+2) = f(y(t+1), \dots, y(t-n_y+2), u(t+1), \dots, u(t-n_u-t_d+2)). \quad (8)$$

The d -step-ahead predicted value can be expressed as

$$y(t+d) = f(y(t+d-1), \dots, y(t+d-n_y), u(t+d+1), \dots, u(t+d-n_u-t_d)). \quad (9)$$

The structure of the recursive neural network predictor is shown in Figure 3. Specifically, d iterations are required for prediction. Based on the fact that a continuous feed-forward neural network with a single hidden layer and the sigmoid transfer function can approximate any complex continuous mapping with arbitrary precision [19–21], the recursive BPNN is constructed by a single hidden layer neural network to prevent the problem of gradient vanishing. When the prediction horizon is larger, the prediction accuracy will be affected by the accumulative errors generated by the recursive procedure. In order to offset this error, a variable $y_a(t)$ is introduced to define the system output and predicted output of the predictor

$$y_a(t) = y(t) - y_m(t). \quad (10)$$

The estimated output value of the ahead predictor can be derived as

$$y_c(t+j) = y_m(t+j) + y_a(t), \quad j = 1, 2, \dots, d. \quad (11)$$

4. IGDM Online Optimization

To solve numerical optimization problems online at each iteration for the MPC algorithm, the online optimization

technique has involved solving dynamic optimization problems in MPC for superior performance. The essence of online optimization problem is to acquire an optimal control input sequence at each sampling instant. Optimal control input can be obtained from the minimization of the cost function in equation (1) over a finite prediction horizon, which has the form

$$J = \alpha[y_r(t) - y_c(t)]^2 + (1 - \alpha)\Delta u^2(t), \quad (12)$$

let initial prediction error $e(t) = y_r(t) - y_c(t)$, and the minimization of the cost function can be described as

$$\min J = \min(\alpha e^2(t) + (1 - \alpha)\Delta u^2(t)) = \min(\alpha \|E(t)\|_2^2 + (1 - \alpha)\|\Delta U(t)\|_2^2), \quad (13)$$

where α and $1 - \alpha$ represent weighting factor of input and output, respectively. α determines the control performance of the system. $e(t)$ is the prediction error between predicted output and desired reference signal. $y_r(t)$, $y_m(t)$, $e(t)$, and $\Delta u(t)$ are denoted as follows:

$$\begin{aligned} y_r(t) &= [y_r(t+1), y_r(t+2), \dots, y_r(t+N_p)]^T, \\ y_m(t) &= [y_m(t+1), y_m(t+2), \dots, y_m(t+N_p)]^T, \\ e(t) &= [y_r(t+1) - y_m(t+1), y_r(t+2) - y_m(t+2), \dots, y_r(t+N_p) - y_m(t+N_p)]^T, \\ \Delta u(t) &= [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u-1)]^T. \end{aligned} \quad (14)$$

FIGURE 3: The structure of recursive neural network predictor.

The objective of the receding horizon optimization is to minimize the cost function in equation (12). For each iteration of the cost function, an intermediate control vector is generated; then, a series of control increment $\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u-1)$ can be derived. For the multivariable function $E(\theta)$ with the input $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$, the optimal θ^* can be obtained by minimizing $E(\theta)$, and the parameters θ_{k+1} can be calculated by

$$\theta_{k+1} = \theta_k + \eta p_k, \quad k = 1, 2, \dots, \quad (15)$$

where $p_k = \nabla E(\theta_k)^T \nabla E(\theta_k)$ represents the search direction, η denotes step size, and k is the current iteration. Based on the Taylor expansion formula, we can obtain

$$E(\theta^{k+1}) \approx 0 = E(\theta_k) + \eta \nabla E(\theta_k)^T \nabla E(\theta_k),$$

$$\eta = \frac{\nabla E(\theta_k)^T \nabla E(\theta_k) \nabla E(\theta_k)^T E(\theta_k)}{\nabla E(\theta_k)^T \nabla E(\theta_k) \nabla E(\theta_k)^T \nabla E(\theta_k) \nabla E(\theta_k)^T E(\theta_k)}. \quad (16)$$

According to the IGDM update rule, the optimization problem is solved by

$$\Delta u(t) = u(t) - u(t-1) = \eta \left(-\frac{\partial J}{\partial u(t)} \right), \quad (17)$$

$$\frac{\partial J}{\partial u(t)} = -\alpha \left(\frac{\partial e(t)}{\partial u(t)} \right)^T e(t) + (1-\alpha) \Delta u(t),$$

where the control increment $\Delta u(t)$ is denoted by

$$\Delta u(t) = \frac{\eta \alpha}{1 + \eta(1-\alpha)} \left(\left(\frac{\partial e(t)}{\partial u(t)} \right)^T e(t) \right), \quad (18)$$

and the Jacobian matrix is expressed as

$$\frac{\partial e(t)}{\partial u(t)} = \frac{\partial y_c(t)}{\partial u(t)} = \begin{bmatrix} \frac{\partial y_c(t+N_0)}{\partial u(t)} & \dots & \frac{\partial y_c(t+N_u)}{\partial u(t+N_u-1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_c(t+N_p)}{\partial u(t)} & \dots & \frac{\partial y_c(t+N_p)}{\partial u(t+N_u-1)} \end{bmatrix}. \quad (19)$$

It is noted that the optimization process will be recursive until the optimum control input $u^*(k)$ can be obtained.

Furthermore, the predicted output of the system $y_m(t+i)$ and its derivatives are required to calculate the last variable $u(k+1)$. Based on the chain rule, the element of the Jacobian matrix can be expressed as

$$\frac{\partial y_c(t+s)}{\partial u(t+N_u+1)} = \sum_{i=1}^m \frac{\partial y_c(t+s)}{\partial H_O} \frac{\partial H_O}{\partial H_I} \frac{\partial H_I}{\partial u(t+N_u+1)}. \quad (20)$$

The proposed GA-RBP neural network predictive control algorithm mainly includes identification and optimization. The multistep-ahead predictor can be obtained by using the recursive technique. The process of the proposed predictive control algorithm by the following procedure is as follows:

- (1) Select the training dataset, validation dataset, and testing dataset.
- (2) Determine the number of neurons in each layer of the BPNN, and select the best individuals by GA.
- (3) Train the BPNN based on the initial weights and bias until the prediction accuracy meets the requirements.
- (4) Establish GA-RBP neural network predictor via recursive technique, and the predictor will be used to predict the system output.
- (5) Specify the initial sequence of control input $u_0(k)$ and initialize the minimum prediction horizon N_0 and the input weighting factor α .
- (6) At the current sampling time k , the past input and output samples and the current sample are utilized to predict the current output and future output of the system.
- (7) Correct the multistep-ahead predicted values y_c according to the error between y_c and the current reference signal y_r in equation (11).
- (8) Compute Jacobian matrix equation (19) and obtain control moves; the control input sequence at the current sampling time k is calculated such that a set of y_c approximates the reference signal y_r in an optimal way.
- (9) Obtain the sequence of control input in step (8) and its first control element $u(k)$ will be applied to control the system.
- (10) Repeat steps (6)–(9) until the simulation time is reached.

Since a model of the process is directly used online for prediction, MPC techniques can be applied to multivariable processes and also when the number of control inputs and controlled variables is different. The IGDM technique aims at identifying advanced starting points for the optimization to reduce the number of iterations required to reach the optimum. It is considered that the iteration process of GA stops when the number of iterations reaches the maximum. So the computational complexity of GA can be approximately described as $O(i_{\max} * p)$, where $i_{\max}$ is the maximum number of iterations and p is the population size. The computational complexity of the BP algorithm depends on the number of neurons. Supposing that the number of neurons in the input layer, hidden layer, and output layer is m , n , and l , the computational complexity of the feed-forward calculation of the sample k is $O[k(m * n + n * l)]$, and the complexity depends on the number of hidden nodes n , the complexity of feed-forward calculation is $O(k * n)$. The computational complexity of the GA-RBP algorithm is $O[(i_{\max} * p) + d(2 * k * n)]$, where d is the number of recursive networks. Considering the influence of the control input in the algorithm, the complexity per iteration of the IGDM in this paper is $O[(n_u + n_y)^3 (H_p^2 + H_u^2)]$, where H_p is the prediction horizon, H_u is the control horizon, and n_u and n_y are the input and output dimensions, respectively. It is noted that the control horizon H_u has a significantly bigger impact on computational complexity than the prediction horizon H_p [18, 34].

5. Numerical Experiments

To illustrate the effectiveness and efficiency of the GA-RBP neural network algorithm, some numerical experiments are presented compared with other three models including BPNN, RBFNN, and Support Vector Regression (SVR). The GA controlling parameters are selected based on the factorial design approach as follows: the population size is chosen from 30 to 100, generation is set within 25 to 75, and crossover probability and mutation probability are selected from 0.30 to 0.85 and 0.05 to 0.55. Based on the prediction model using the GA-RBP neural network strategy, the IGDM will be applied to calculate the control law for the system. We select three criteria to evaluate the performance of the proposed algorithm as the mean absolute error (MAE), root mean square error (RMSE), and calculation time. Generally, the smaller MAE and RMSE are, the better prediction performance is. The mathematical formulas can be expressed as

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y(i) - y_m(i)|, \quad (21)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y(i) - y_m(i))^2}, \quad (22)$$

where N is the number of samples and $y_m(i)$ and $y(i)$ are the predicted output and the system output, respectively. All the experiments are implemented in MATLAB (version 9.6

(R2019a)) on the personal computer of Windows 7 operating system, an Intel Core TM i7 CPU with 2.5 GHz, and 12 GB RAM.

5.1. System I. A typical highly nonlinear discrete system is described as

$$y(k+1) = u(k)^3 + \frac{y(k)}{y(k)^2 + 1} - 1.4 \sin(1.5y(k)). \quad (23)$$

The time-varying reference signal $y_r(t)$ is

$$y_r(k) = \sin(k), \quad 0 < k < 600, \quad (24)$$

where the delay time is $t_d = 0$ and the maximum lags of the system are $n_u = 1$ and $n_y = 1$, respectively. The prediction and control horizons of the proposed control method are selected as $N_p = 2$ and $N_u = 1$, respectively. The initial values are $u(1) = 1$ and $y(1) = 0$, respectively. A random function is used to generate 2000 data as the input of the nonlinear discrete system and the output data of the controlled system is obtained according to equation (23). The input vector of the GA-BP prediction model is $[u(k-1), y(k-1)]^T$, whereas $y(k)$ is output.

In BP and GA-BP algorithms, each hidden layer consists of 20 neurons. The population size of GA is chosen 30, the generation is selected as 50, and the crossover and mutation probabilities are selected as 0.4 and 0.2, respectively. The penalty coefficient is $C = 150$ and the hyperparameter of the Gaussian kernel is $\sigma = 0.6$ in SVR. Each modeling algorithm is trained offline with the input-output data pairs obtained from the training data and the initial RMSE is set as 0.001. The prediction errors of GA-BP, BP, RBF, and SVR are shown in Figure 4.

The MAE and RMSE values of the training dataset, validation dataset, and testing dataset are shown in Table 1. It can be observed that GA-BP has good predictive performance and MAE and RMSE test error is small, which proves that the model has good generalization abilities. To further verify the nonlinear accuracy of the GA-BP neural network model, the performance of each model is compared in Table 2. It can be seen that the GA-BP algorithm achieves a smaller MAE and RMSE than the other models, which demonstrate that the GA-BP is suitable for nonlinear system modeling. Although GA-BP requires more training time for the system modeling, GA-BP neural network can avoid falling into the local optimum via adjusting the network parameters, and thus its prediction accuracy is greatly improved.

Combined with the proposed GA-RBP neural network predictor, the IGDM is used to calculate the control moves of the system. The system output is required to track the reference signal as equation (22). Figure 5 shows the comparison results between the system output and the reference signal. The satisfactory performance obtained is due to the accurate representation of the nonlinear dynamics of the process by these neural network models. The tracking time for GA-RBP neural network predictive control (GA-RBPNNPC) is less than those of the other methods, which

FIGURE 4: Prediction error of different models on System I.

TABLE 1: Design specifications of GA-BP neural network model.

GA-BP	Training dataset	Validation dataset	Testing dataset
Sample size	1600	300	100
MAE	0.1897	0.0607	0.0242
RMSE	0.2318	0.0758	0.0294

TABLE 2: The performance of different models.

Prediction model	MAE	RMSE	Training time (s)
GA-BP	0.0242	0.0294	6.83
BP	0.6642	0.7823	1.01
RBF	0.8571	0.9494	1.27
SVR	0.9047	0.7733	3.20

FIGURE 5: Tracking performance of the proposed algorithm on System I.

reflects the proposed controller which can track reference signal quickly. The absolute tracking error between the system output and the reference signal of the proposed algorithm can be shown in Figure 6, which demonstrates that the proposed GA-RBP neural network model can highly approximate the dynamic process, and the tracking performance of the proposed IGDM is satisfactory.

The simulation results for the nonlinear system are compared with an RBF neural network predictive control (RBFNNPC) and a traditional GPC. The control parameters of RBFNNPC are the same as those of GA-RBP neural network predictive control. The GPC parameters are set as $N_p = 16$, $N_u = 1$, and $\lambda = 5$. The control performance of the proposed algorithm and the other algorithms is listed in Table 3. It can be observed that the proposed control method has a superior performance for this system, which indicates that the GA-RBP neural network predictive controller has a better calculate ability than the other methods. It can be inferred that the proposed algorithm improves the control performance of MPC including the maximum overshoot value, calculation time, and RMSE error value.

FIGURE 6: Tracking error.

5.2. System II. A nonlinear dynamic system with time delay has the following form:

$$\begin{aligned}
 y(t) = & 0.9722y(t-1) + 0.3578u(t-1) - 0.1295u(t-2) - 0.3103y(t-1)u(t-1) \\
 & - 0.04228y^2(t-2) + 0.1663y(t-2)u(t-2) - 0.03259y^2(t-1)y(t-2) \\
 & - 0.3513y^2(t-1)u(t-2) + 0.3084y(t-1)y(t-2)u(t-2) \\
 & + 0.1087y(t-2)u(t-1)u(t-2) + d(t).
 \end{aligned} \tag{25}$$

The reference signal $y_r(t)$ and the external disturbance signal $d(t)$ are

$$y_r(t) = \begin{cases} 0.9, & 0 < t \leq 200, \\ 0, & 200 < t \leq 400, \\ 0.7, & 400 < t \leq 600, \\ -0.3, & 600 < t \leq 800, \\ 0.4, & 800 < t \leq 1000, \end{cases} \tag{26}$$

$$d(t) = \begin{cases} 0, & 0 < t \leq 100, \\ 0.05, & 100 < t \leq 600, \\ -0.1, & 600 < t \leq 800, \\ 0.1, & 800 < t < 1000. \end{cases} \tag{27}$$

The sampling time is 1 second. The delay time is $t_d = 1$, and the initial values of the system input and output are $u(1) = u(2) = 1$ and $y(1) = y(2) = 0$, respectively. It can be obtained that the maximum lags in the outputs and inputs are 1 and 2, respectively. A random function is used to generate 3000 data as the input of the nonlinear discrete system and the output data of the controlled system are obtained based on equation (25). The crossover and mutation probabilities of GA are chosen as 0.3 and 0.1, respectively. The SVR parameters are selected as $C = 80$ and $\sigma = 0.2$. To compare the

predictive performance of the BP and GA-BP, the same number of neurons and learning rate are used in the modeling process. The input and output variables of the proposed GA-BP neural network model are specified by $y_m(k) = f(u(k-1), u(k-2), y(k-1), y(k-2))$.

The design specifications for the GA-BP neural network model obtained are given in Table 4. It can be inferred from the cross-validation results that the GA-BP is well trained and can be used as a prediction model. It can be seen from Figure 7 and Table 5 that the GA-BP model is more accurate compared with the other methods for the time-delay nonlinear system. After the offline training stage, the weights and bias were used as connection parameters of the neural network, and the two-step-ahead prediction results can be obtained by a recursive algorithm. The constructed GA-RBP predictor is combined with the IGDM to track set-point changes signal as equation (26).

As reference trajectory equation (26), a series of set-point changes is considered. As shown in Figures 8 and 9, the proposed algorithm can always reach a stable value in a relatively small time when the operating point changes because the controller adopts a receding horizon optimization method. It also can be seen that there is no steady-state error in the tracking of the desired reference. The MPC based on the GA-RBP algorithm is stable to track the reference trajectory fast enough. In order to test the robustness

TABLE 3: Comparison of the performance of different controllers.

Controller	Hidden nodes	Maximum overshoots	Calculation time (s)	Rise time (s)	RMSE
GA-RBPNNPC	20	0.2945	0.16	21	0.0576
RBFNNPC	11	0.4357	0.25	28	0.2869
GPC	0	0.4829	0.20	24	0.3907

TABLE 4: Design specifications of GA-BP neural network model.

GA-BP	Training dataset	Validation dataset	Testing dataset
Sample size	2500	400	100
MAE	0.0132	0.0035	0.0011
RMSE	0.0293	0.0097	0.0026

FIGURE 7: Prediction error of different models on System II.

TABLE 5: The performance of different models.

Prediction model	MAE	RMSE	Training time (s)
GA-BP	0.0011	0.0014	15.17
BP	0.0026	0.0033	2.03
RBF	0.0175	0.0235	2.09
SVM	0.0115	0.0038	5.96

and disturbance rejection of the proposed algorithm, the external disturbance is introduced in the system. The simulation results of RBFNNPC and traditional GPC algorithms are compared under set-point changes and constant external disturbances. The control parameters of RBFNNPC are the same as those of GA-RBP neural network predictive control. The GPC parameters are selected the same as [2], which can be listed as $N_p = 20$, $N_u = 1$, and $\lambda = 10$. Figures 10 and 11 show the output response and corresponding absolute errors of the system.

As shown in Figures 10 and 11, the system output starts fluctuating when an external disturbance occurs. However, the effect of the disturbance and the overall performance become

FIGURE 8: Tracking performance of the proposed algorithm on System II.

FIGURE 9: Tracking error.

reasonable after around 50 seconds, and the proposed MPC algorithm damps. It can be seen that the proposed method ensures that the process response can track the set points without unrestrained overshoot and oscillations. Compared with the set-point tracking results of the other controllers, our controller has smaller overshoots and rise time. The system output can be faster convergent at the desired value due to the accurate representation of the nonlinear dynamics of the

FIGURE 10: Tracking performance of the proposed algorithm on System II under external disturbance.

FIGURE 11: Tracking error.

process by GA-RBP neural network predictive controller and the use of IGDM at the current sampling instant to obtain the value of control in equation (18). Meanwhile, the evaluation criteria of the proposed algorithm include the hidden nodes, the maximum overshoots, the calculation time, the rise time, and the RMSE. It can be seen from Table 6 that the proposed control method has superior performance on the calculation time and RMSE error values, which indicates that the proposed

approach has a better adjustment ability than other methods and thus can reduce computational burden and optimize the control performance of MPC. The proposed control method has satisfied prediction and tracking abilities and has no difficulty in rejecting the external disturbance and the influence of model mismatch can be compensated in time. It can also be concluded that the presentation of IGDM realizes the real-time control of complex processes of nonlinear system.

TABLE 6: Comparison of the performance of different controllers.

Controller	Hidden nodes	Maximum overshoots	Calculation time (s)	Rise time (s)	RMSE
GA-RBPNNPC	30	0.2105	0.19	13	0.1247
RBFNNPC	13	0.2545	0.32	16	0.2128
GPC	0	0.3092	0.29	15	0.4406

6. Conclusions

In this paper, an MPC method based on the GA-RBP neural network predictor is proposed and the reference signal tracking and disturbance rejection behavior of the nonlinear system was studied. Due to the global searching characteristic of GA, the optimum interconnection values can be obtained, which improves the convergence rate and stability of the BP algorithm. The GA-RBP neural network predictor is applied to model two highly nonlinear systems and the prediction accuracy proves that it has satisfied performance. Simultaneously, combine the multistep-ahead GA-RBP neural network predictor with the IGDM to address the online optimization problem of nonlinear MPC. Finally, the proposed method has been utilized in these systems to evaluate our strategy performance of set-point changes signal tracking and external disturbance rejection, and the effectiveness of the results from computer simulation has been proven.

It can be observed from simulation results that GA-RBP neural network predictive control algorithm presents a satisfactory performance when the process with time delays where traditional GPC algorithm fails due to the nonlinearities. The proposed method can achieve more accurate predictive performance and address online optimization of nonlinear MPC, and the MPC strategy based on the GA-RBP and IGDM can track the desired reference signal with satisfactory performances. However, when the GA-RBP modeling method addresses small sample sizes or uneven sample distribution, the prediction ability for GA-RBP neural network is limited and needs to be improved. In general, there are two ways to solve this problem. One is the data augmentation to generate as many training samples as possible. The other is sampling and data synthesis, where the training dataset is processed to obtain more training samples from an unbalanced dataset to a balanced dataset.

In industrial applications, the delay time of the nonlinear system may be unknown or uncertain. The analysis and design of a controller for nonlinear systems with time delay are more complex if the delays are unknown or time varying. This problem can be solved by optimal control methods, including linear matrix inequality (LMI) approach [47, 48], Lyapunov-Krasovskii functional [49, 50], and fuzzy logic model [51, 52], which are proved to be robust MPC controllers for the constraint process in practice. The incorporation of GA-RBP in the optimal control methods for the MPC model will be a new method for a class of nonlinear systems with time delays and uncertainties. It is promising that the new combined algorithm has good prediction and tracking performance in nonlinear systems with unknown time delays. The study of this type of nonlinear system in industrial applications will be the further direction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 51875457 and 61903298), the Key Research and Development Program of Shaanxi Province of China (2019GY-086), and the Open Projects of the State Key Laboratory for Strength and Vibration of Mechanical Structures (SV2015-KF-04). It was also supported by the Graduate Student Innovation Fund of Xi'an University of Post and Telecommunications (CXJJLY2018063).

References

- [1] Y. Ma, F. Borrelli, B. Hencsey et al., "Model predictive control for the operation of building cooling systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, 2012.
- [2] C.-H. Lu and C.-C. Tsai, "Adaptive predictive control with recurrent neural network for industrial processes: an application to temperature control of a variable-frequency oil-cooling machine," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 3, pp. 1366–1375, 2008.
- [3] P. Cortés, A. Wilson, S. Kouro, J. Rodriguez, and H. Abu-Rub, "Model predictive control of multilevel cascaded h-bridge inverters," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 8, pp. 2691–2699, 2010.
- [4] S. Saki and A. Fatehi, "Neural network identification in nonlinear model predictive control for frequent and infrequent operating points using nonlinearity measure," *ISA Transactions*, vol. 97, pp. 216–229, 2020.
- [5] A. W. Hermansson and S. Syafie, "Model predictive control of pH neutralization processes: a review," *Control Engineering Practice*, vol. 45, pp. 98–109, 2015.
- [6] D. He, L. Wang, and L. Yu, "Multi-objective nonlinear predictive control of process systems: a dual-mode tracking control approach," *Journal of Process Control*, vol. 25, pp. 142–151, 2015.
- [7] H. Peng, K. Nakano, and H. Shiota, "Nonlinear predictive control using neural nets-based local linearization ARX model—stability and industrial application," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 1, pp. 130–143, 2006.
- [8] H.-G. Han, J.-F. Qiao, and Q.-L. Chen, "Model predictive control of dissolved oxygen concentration based on a self-

- organizing RBF neural network,” *Control Engineering Practice*, vol. 20, no. 4, pp. 465–476, 2012.
- [9] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, “Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: a state of the art review and case study of a residential HVAC system,” *Energy and Buildings*, vol. 141, pp. 96–113, 2017.
 - [10] L. Li, S. Rong, R. Wang, and S. Yu, “Recent advances in artificial intelligence and machine learning for nonlinear relationship analysis and process control in drinking water treatment: a review,” *Chemical Engineering Journal*, vol. 405, Article ID 126673, 2021.
 - [11] T. A. Johansen, R. Shorten, and R. Murray-Smith, “On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models,” *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 3, pp. 297–313, 2000.
 - [12] G. R. Srinivas and Y. Arkun, “A global solution to the nonlinear model predictive control algorithms using polynomial ARX models,” *Computers & Chemical Engineering*, vol. 21, no. 4, pp. 431–439, 1997.
 - [13] C. J. Ong, D. Sui, and E. G. Gilbert, “Enlarging the terminal region of nonlinear model predictive control using the support vector machine method,” *Automatica*, vol. 42, no. 6, pp. 1011–1016, 2006.
 - [14] J. K. Gruber, D. R. Ramirez, D. Limon, and T. Alamo, “Computationally efficient nonlinear Min-Max model predictive control based on volterra series models-application to a pilot plant,” *Journal of Process Control*, vol. 23, no. 4, pp. 543–560, 2013.
 - [15] M. Spisiak and S. Kozak, “Nonlinear predictive control based on artificial neural networks,” *Neural Network World*, vol. 14, no. 3, pp. 247–260, 2004.
 - [16] T. Wang, H. Gao, and J. Qiu, “A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 416–425, 2016.
 - [17] K. Patan, “Two stage neural network modelling for robust model predictive control,” *Isa Transactions*, vol. 72, pp. 56–65, 2018.
 - [18] M. Ławryńczuk, “Accuracy and computational efficiency of suboptimal nonlinear predictive control based on neural models,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2202–2215, 2011.
 - [19] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 3, pp. 303–314, 1989.
 - [20] K.-I. Funahashi, “On the approximate realization of continuous mappings by neural networks,” *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989.
 - [21] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
 - [22] H. G. Han, X. L. Wu, and J. F. Qiao, “Real-Time model predictive control using a self-organizing neural network,” *IEEE Transactions on Neural Networks & Learning Systems*, vol. 24, no. 9, pp. 1425–1436, 2013.
 - [23] M. Ławryńczuk, “Computationally efficient nonlinear predictive control based on RBF neural multi-models,” in *Proceedings of the International Conference on Adaptive & Natural Computing Algorithms*, Springer-Verlag, Berlin, Heidelberg, April 2009.
 - [24] A. Wysocki and M. Ławryńczuk, “Elman neural network for modeling and predictive control of delayed dynamic systems,” *Archives of Control Sciences*, vol. 26, no. 1, pp. 117–142, 2016.
 - [25] X.-J. Wu, X.-J. Zhu, G.-Y. Cao, and H.-Y. Tu, “Predictive control of SOFC based on a GA-RBF neural network model,” *Journal of Power Sources*, vol. 179, no. 1, pp. 232–239, 2008.
 - [26] X. Li and G. Gong, “Predictive control of slurry pressure balance in shield tunneling using diagonal recurrent neural network and evolved particle swarm optimization,” *Automation in Construction*, vol. 107, pp. 1–15, Article ID 102928, 2019.
 - [27] M. M. N. Sudibyo and N. Aziz, “MIMO Neural Wiener based Model Predictive Control (NWMPC) for MTBE reactive distillation using simulated annealing-particle swarm optimization (SA-PSO),” in *Proceedings of 12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, pp. 1631–1636, Copenhagen, Denmark, May 2015.
 - [28] Z. Wang and A. Sobey, “A comparative review between Genetic Algorithm use in composite optimisation and the state-of-the-art in evolutionary computation,” *Composite Structures*, vol. 233, Article ID 111739, 2020.
 - [29] C.-H. Cheng and S.-L. Shu, “Application of GA-based neural network for attitude control of a satellite,” *Aerospace Science and Technology*, vol. 14, no. 4, pp. 241–249, 2010.
 - [30] L. Wang and X. Bi, “Risk assessment of knowledge fusion in an innovation ecosystem based on a GA-BP neural network,” *Cognitive Systems Research*, vol. 66, pp. 201–210, 2021.
 - [31] Y. Liu, J. Dai, S. Zhao et al., “A bidirectional reflectance distribution function model of space targets in visible spectrum based on GA-BP network,” *Applied Physics B*, vol. 126, no. 114, 2020.
 - [32] R. Haber, U. Schmitz, and R. Bars, “Long-range optimal model and multi-step-ahead prediction identification for predictive control,” *IFAC Proceedings Volumes*, vol. 36, no. 16, pp. 477–482, 2003.
 - [33] C. H. Lu, “Design and application of stable predictive controller using recurrent wavelet neural networks,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3733–3742, 2009.
 - [34] R. Stefan, N. J. Colin, and M. Manfred, “Computational complexity certification for real-time MPC with input constraints based on the fast gradient method,” *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1391–1402, 2012.
 - [35] P. Kittisupakorn, P. Thitiyasook, M. A. Hussain, and W. Daosud, “Neural network based model predictive control for a steel pickling process,” *Journal of Process Control*, vol. 19, no. 4, pp. 579–590, 2009.
 - [36] H. Yang, F. Deng, Y. He et al., “Robust nonlinear model predictive control for reference tracking of dynamic positioning ships based on nonlinear disturbance observer,” *Ocean Engineering*, vol. 215, Article ID 107885, 2020.
 - [37] I. J. Leontaritis and S. A. Billings, “Input-output parametric models for non-linear systems part I: deterministic non-linear systems,” *International Journal of Control*, vol. 41, no. 2, pp. 303–328, 1985.
 - [38] L. Bai and D. Coca, “Nonlinear predictive control based on NARMAX models,” in *Proceedings of the International Conference on Optimization of Electrical & Electronic Equipment*, IEEE, Brasov, Romania, May 2008.
 - [39] L. Prechelt, “Early stopping-but when?” *Neural Networks: Tricks of the Trade*, pp. 55–69, Springer, Berlin, Germany, 1998.

- [40] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012.
- [41] V. V. Degtyarev, "Neural networks for predicting shear strength of CFS channels with slotted webs," *Journal of Constructional Steel Research*, vol. 177, p. 106443, 2021.
- [42] Kohji, Omata, Masahiko et al., "Artificial neural network and grid search aided optimization of temperature profile of temperature gradient reactor for dimethyl ether synthesis from syngas," *Industrial & Engineering Chemistry Research*, vol. 48, no. 2, pp. 844–849, 2009.
- [43] S. Wang, N. Zhang, L. Wu, and Y. Wang, "Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method," *Renewable Energy*, vol. 94, pp. 629–636, 2016.
- [44] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.
- [45] C. B. B. Costa, M. R. W. Maciel, and R. M. Filho, "Factorial design technique applied to genetic algorithm parameters in a batch cooling crystallization optimisation," *Computers & Chemical Engineering*, vol. 29, no. 10, pp. 2229–2241, 2005.
- [46] C. B. B. Costa, E. A. C. Rivera, M. C. A. F. Rezende, M. R. W. Maciel, and R. M. Filho, "Prior detection of genetic algorithm significant parameters: coupling factorial design technique to genetic algorithm," *Chemical Engineering Science*, vol. 62, no. 17, pp. 4780–4801, 2007.
- [47] G. S. Huang and S. W. Wang, "Use of uncertainty polytope to describe constraint processes with uncertain time-delay for robust model predictive control applications," *ISA Transactions*, vol. 48, no. 4, pp. 503–511, 2009.
- [48] M.-J. Park and H.-K. Rhee, "LMI-based robust model predictive control for a continuous MMA polymerization reactor," *Computers & Chemical Engineering*, vol. 25, no. 11–12, pp. 1513–1520, 2001.
- [49] S. K. Nguang, "Robust stabilization of a class of time-delay nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 756–762, 2000.
- [50] S. S. Ge, F. Hong, and T. H. Lee, "Adaptive neural network control of nonlinear systems with unknown time delays," *IEEE Transactions on Automatic Control*, vol. 48, no. 11, pp. 2004–2010, 2003.
- [51] V. B. Tarassov, "Development of fuzzy logics: from universal logic tools to natural pragmatics and non-standard scales," *Procedia Computer Science*, vol. 120, pp. 908–915, 2017.
- [52] C. Ariño, A. Querol, and A. Sala, "Shape-independent model predictive control for Takagi-Sugeno fuzzy systems," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 493–505, 2017.