

Automatic T-S fuzzy model with application to designing predictive controller¹

Zhi-gang Su, Pei-hong Wang, and Yu-fei Zhang

School of energy & environment, Southeast University, Nanjing, Jiangsu 210096,
China
Zhigangsu@seu.edu.cn

Abstract. A novel methodology is proposed to automatically extract T-S fuzzy model with enhanced performance using VABC-FCM algorithm, a novel Variable string length Artificial Bee Colony algorithm (VABC) based Fuzzy C-Mean clustering technique. Such automatic methodology not requires a priori specification of the rule number and has low approximation error and high prediction accuracy with appreciate rule number. Afterward, a new predictive controller is then proposed by using the automatic T-S fuzzy model as the dynamic predictive model and VABC as the rolling optimizer. Some experiments were conducted on the superheated steam temperature in power plant to validate the performance of the proposed predictive controller. It suggests that the proposed controller has powerful performance and outperforms some other popular controllers.

Keywords: T-S fuzzy model, fuzzy c-means; automatic; Artificial Bee Colony; predictive control, superheated steam temperature

1. Introduction

System modeling and (rolling) optimization are the two main issues to design a predictive controller for an uncertain nonlinear system [27]. In this paper, we aim to propose a novel methodology used to automatically extract model from data and then to design a predictive controller using such novel methodology.

Fuzzy system modeling approach is an effective tool for the approximation of uncertain nonlinear systems on the basis of observed data [13]. Among the different fuzzy modeling techniques, the Takagi-Sugeno (T-S) model [34] has attracted most attention and is popularly applied [12, 14, 28, 30, 32]. The construction of a T-S fuzzy model is generally done in two steps. In the first step, the fuzzy sets in the rule antecedents are determined. This can be done manually, using expert knowledge of the process or by some data-driven techniques. In the second step, the parameters of the consequent functions

¹ This article is supported by the National natural science foundation of China (51106025). This is an extended version of the paper [38] originally presented at the international conference on ICNC-FSKD, 25-28, July, 2011.

are estimated. As these functions are usually chosen to be linear in their parameters, standard linear least-squares methods can be used. The bottleneck of the construction procedure is the identification of the fuzzy sets in the antecedents. In the traditional T-S fuzzy model, the expert knowledge is used to define linguistically the fuzzy sets, which leads to the drawbacks of subjectivity and suffers lack of generalization. To overcome these issues, reduce expert knowledge intervention and rather build self learning systems, more objective T-S fuzzy models are developed, such as the first family of population stochastic optimization based methods [4, 9, 10, 11, 16, 19, 25, 39] and the second family of clustering based methods [1, 6, 20, 21, 33, 37, 40]. Because of the powerful global search ability of the population stochastic optimization algorithms, this first family of method has high prediction accuracy [15]. Attractive features of the second family of method are the simultaneous identification of the antecedent membership functions along with the consequent local linear models and the implicit regularization [15].

Although the aforementioned approaches improve the efficiency of the T-S fuzzy model, they have various challenges that should not be neglected [35]. The two main challenges among these are how to automatically determine the rule number and to obtain a global optimal model. For the above first family of method, they usually can get a global structure of T-S fuzzy model due to their powerful global search ability. However, they cannot automatically determine the rule number. In some literature [39], the rule number is labeled as a cell in the fixed length of individual so as to automatically determine the rule number. It is crisp in nature. This drawback also exists for the second family of method due to the reason that the cluster number is assumed to be fixed *a priori*. In addition, the second family of method usually cannot get global optimization solutions, because the traditional fuzzy clustering techniques often get stuck at suboptimal solutions based on the initial configuration. A methodology, used to extract T-S fuzzy model with enhanced performance and without knowing the rule number as a priori, is desirable. In [31], we viewed the fuzzy clustering as an optimization problem where the length of the optimal solution(s) is not known as a priori, and we thus proposed a novel version of Fuzzy C-Means clustering technique, called VABC-FCM algorithm, based on the so-called Variable string length Artificial Bee Colony (VABC) algorithm, another contribution in [31]. By using such novel VABC-FCM algorithm, in this paper, we further propose a novel methodology to automatically extract T-S fuzzy models from data. Use of VABC-FCM allows such methodology not require a priori specification of the rule number and have low approximation error and high prediction accuracy.

For the later issue in traditional predictive controller, the optimal control sequences are computed using the derivation of a given quadratic objective function that is usually represented in the form of vector or matrix. However, it is hard to separate the variables to be optimized from the given quadratic objective function. Thus, it is difficult to obtain analytic solution by using such derivation methods. In addition, the derivation methods exist defects on dealing with control problem with constraint. Therefore, some population stochastic optimization algorithms are used instead of the traditional

derivation methods [14, 22, 27, 32]. Here, we apply the (V)ABC (i.e., VABC or ABC) algorithm as the optimizer, due to the following two reasons [18]: (1) ABC algorithm is recently a popular swarm intelligent algorithm, which outperforms the evolutionary algorithms and other swarm intelligent algorithm in the most of cases; and (2) The number of control parameters involved in ABC is less than that of other population stochastic optimization algorithms. Note that, the (rolling) optimization in predictive control is a constrained optimization problem where the length of optimal solutions is fixed and known as a priori. So, the VABC algorithm and ABC algorithm are equivalent in this case, which will be discussed later in the paper. By using the automatic T-S fuzzy model as the dynamic predictive model and the VABC algorithm as the rolling optimizer, we design a new predictive controller in this paper.

The rest of this paper is organized as follows. Section 2 introduces VABC algorithm and VABC-FCM algorithm. Section 3 presents the automatic T-S fuzzy model. In Section 4, a new fuzzy model predictive controller is designed. The last section concludes this paper.

2. VABC algorithm and VABC based Fuzzy clustering

2.1. The VABC algorithm

In VABC algorithm, the foraging artificial bees are categorized into three groups: employed bees, onlookers and scouts. The task of each category of bees is also the same as that in the ABC algorithm [17, 18] except for some differences. The main difference is the string or food source representation. In ABC, all the lengths of strings are fixed to be equal, whereas those in VABC are variable. This difference is shown in a d -dimensional space in Fig. 1.

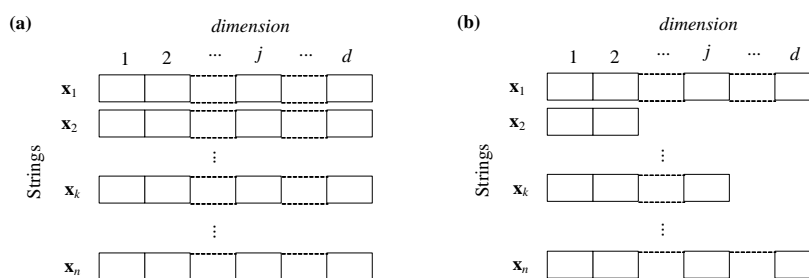


Fig. 1. String representation for (a) ABC and (b) VABC

Firstly, a randomly distributed initial population of n solutions is generated according to mechanism (1).

$$x_{i,j} = x_{\min,j} + \text{rand}[0,1] \times (x_{\max,j} - x_{\min,j}), \quad j \in \{1, 2, \dots, J_i\} \quad (1)$$

where J_i is the length of the i^{th} string and suppose its value is no less than 2. Thus, given an upper bound d_{\max} for J_i ($i = 1, 2, \dots, n$), it can be randomly derived according to:

$$J_i = \text{round}(\text{rand}[0,1] \times (d_{\max} - 2)) + 2 \quad (2)$$

where each solution \mathbf{x}_i ($i = 1, 2, \dots, n$) is a d -dimensional vector.

After initialization, the population is subjected to repeated cycles of four major steps: updating feasible solutions, selecting feasible solutions, avoiding suboptimal solutions and solutions mutation.

All employed bees select a new candidate food source position. The choice is based on the neighborhood of the previously selected food source. The position of such new candidate food source is calculated from equation below:

$$x_{\text{new},j} = x_{i,j} + \text{rand}[-1,1] \times (x_{i,j} - x_{k,q}) \quad (3)$$

where $k \in \{1, 2, \dots, n\}$ and $k \neq i$, and $\text{rand}[-1, 1]$ is a uniformly distributed real number in interval $[-1, 1]$. Subscript j and q denotes respectively the j^{th} and q^{th} dimension of \mathbf{x}_i and \mathbf{x}_k , determined as

$$\begin{cases} q \leftarrow j = \text{round}(\text{rand}[0,1] \times (J_i - 1)) + 1, & \text{if } J_k \geq J_i \\ j \leftarrow q = \text{round}(\text{rand}[0,1] \times (J_k - 1)) + 1, & \text{otherwise} \end{cases} \quad (4)$$

with $\text{round}(\cdot)$ function rounding a number to the nearest integer and the symbol " $q \leftarrow j$ " assigning value of variable j to variable q .

The old food source position in the employed bees' memory will be replaced by the new candidate food source position, if the new position has a better fitness value. Employed bees will return to their hive and share the fitness value of their new food sources with the onlooker bees.

In the next step, each onlooker bee selects one of the proposed food sources depending on the fitness value obtained from the employed bees. The probability that a food source will be selected can be obtained from following equation:

$$p_i = \text{fit}_i / \sum_{i=1}^n \text{fit}_i \quad (5)$$

where fit_i is the fitness value of the i^{th} food source \mathbf{x}_i . Obviously, the higher the fit_i is, the more probability the i^{th} food source is selected.

The probability of a food source being selected by the onlooker bees increases as the fitness value of a food source increases. After the food source is selected, onlooker bees will go to the selected food source and select a new candidate food source position in the neighborhood of the selected food source. The new candidate food source can be also calculated by Eq. (3).

In the third step, any food source position that does not improve the fitness value will be abandoned and replaced by a new position that is randomly determined by a scout bee. This helps avoid suboptimal solutions. The new randomly chosen position will be calculated according to Eq. (1).

The final step is a mutation operation on the strings/food sources that are selected for mutation. The mutation operator is defined of three types:

1. If the length of the selected string is longer than that of the string holding the best fitness, one randomly generated cell (i.e., a real-number in a random dimension) is removed from the string.
2. If the length of the selected string is shorter than that of the string holding the best fitness, one randomly generated cell is added to the string.
3. Otherwise, the selected string is hold on.

The mutation probability is selected adaptively for each string as in [29]. Let fit_{max} be the maximum fitness, \overline{fit} be the average fitness of the population and fit be the fitness value of the solution to be mutated. The expression for mutation probability, p_m , is given below:

$$p_m = \begin{cases} k_1 (fit_{max} - fit) / (it_{max} - \overline{fit}), & \text{if } fit > \overline{fit} \\ k_2, & \text{otherwise} \end{cases} \quad (6)$$

where k_1 and k_2 are constant. Here, both k_1 and k_2 are kept equal to 0.5.

Given a threshold p_0 for the mutation, the string is selected when its mutation probability is bigger than p_0 , i.e., $p_m > p_0$. It is evident that p_0 cannot be higher than the value of k_2 , otherwise the mutation operation is useless: once p_0 is higher than the value of k_2 , the length of each string in the population will keep the same as its initial length when the VABC is terminated. In such viewpoint, the VABC algorithm degenerates to ABC algorithm once the lengths of string x_i are equally initialized and/or p_0 is set to be bigger than k_2 .

The maximum number of cycles (C_{max}) is used to control the number of iterations and is a termination criterion. The process will be repeated until the number of iteration equals the C_{max} .

2.2. VABC- FCM: VABC based fuzzy c-means clustering approach

Let $Z = \{z_1, z_2, \dots, z_N\}$ be a collection of vectors in R^d describing N objects. Let c ($2 \leq c \leq N$) be the desired cluster number. Each cluster is represented by a prototype or a center $v_k \in R^d$. Let V denotes a matrix of size $(c \times d)$ composed of the coordinates of the cluster centers such that V_{kq} is the q^{th} component of the cluster center v_k . FCM looks for a partition matrix $U = (u_{ik})$ of size $(N \times c)$ and for the matrix V by minimizing the following objective function:

$$J_{FCM}(U, V, Z) = \sum_{i=1}^N \sum_{k=1}^c u_{ik}^m D_{ik}^2 = \sum_{i=1}^N \sum_{k=1}^c u_{ik}^m \|z_i - v_k\|^2 \quad (7)$$

subject to the following constraints (8) and (9):

$$\sum_{k=1}^c u_{ik} = 1, \quad \forall i \in \{1, 2, \dots, N\} \tag{8}$$

$$\sum_{i=1}^N u_{ik} > 0, \quad \forall k \in \{1, 2, \dots, c\} \tag{9}$$

For a given cluster number c , the algorithm starts from an initial guess for either the partition matrix or the cluster centers and iterates until convergence. Convergence is guaranteed but it may lead to local minimum [2]. To choose optimal cluster number c , clustering validity index is necessary. A sufficient review for fuzzy clustering validity indexes can refer to Refs. [23, 24]. Among them, the most popular two are XB-index [36] and PBM-index [23], defined as:

$$XB(U, V, Z) = \sum_{k=1}^c \sum_{i=1}^N u_{ik}^2 D_{ik}^2 / N \times \min_{i \neq j} \{ \|v_i - v_j\|^2 \} \tag{10}$$

$$PBM(U, V, Z) = E_1 \times \max_{i,j} \{ \|v_i - v_j\| \} / c \sum_{k=1}^c \sum_{i=1}^N u_{ik} D_{ik} \tag{11}$$

where E_1 is the constant avoiding the PBM to be too smaller. Here, $E_1 = 100$.

As can be seen from validity indexes (10) and (11), the cluster number minimizing XB-index leads to the optimal partition of data set Z , and on the contrary, proper fuzzy partitioning can be obtained for a data set by maximizing the PBM-index. Thus, the fuzzy c-means clustering can be viewed as an optimization problem where the best partition is considered to be the one that corresponds to the minimum value of the XB-index or to the maximum value of the PBM-index. Therefore, the framework of the VABC-FCM is the same as that of the VABC algorithm except for some differences, which are interpreted as followings.

The first difference may be the string representation and population of food sources initialization. In VABC-FCM, center based encoding is used where centers are considered to be indivisible. Real numbers are encoded in the strings which represent the coordinates of the centers of the partitions. The length of a particular string x_i is now J'_i , calculated by

$$J'_i = d \times \lceil \text{round}(\text{rand}[0,1] \times (d_{\max} - 2)) + 2 \rceil \tag{12}$$

Note that d_{\max} here is the upper bound of the cluster number, not the maximal dimension of data space as interpreted in VABC algorithm. The initial cluster number will therefore range from two to d_{\max} . The centers encoded in a string are randomly selected distinct points from the data set. The selected points are distributed randomly in the string.

The membership value u_{ik} in VABC-FCM algorithm is updated by using the same criterion as the FCM algorithm. After obtaining the membership value u_{ik} , the centers encoded in a string are further updated using the same

criterion as the FCM algorithm. Centers are further updated so as to incorporate a limited amount of local search for speeding up the convergence of VABC-FCM.

The second difference between VABC-FCM algorithm and VABC algorithm is the fitness computation. The fitness in the VABC-FCM algorithm can be calculated as:

- XB-index case: $fit = 1 / (1 + XB(U, V, Z))$
- PBM-index case: $fit = PBM(U, V, Z)$

The final difference exists in mutation operation on the selected string(s). In VABC algorithm, once the length of the selected string(s) is not equal to that of the string holding the best fitness, one random dimension is removed or increased. However, in VABC-FCM algorithm, the indivisible cell is not one dimension but a cluster center. Thus, one random cluster center is removed or increased when the selected string(s) do not have the best fitness.

With the above interpretations, the VABC-FCM algorithm can be directly derived from VABC algorithm.

3. Automatic T-S fuzzy model based on VABC-FCM

The typical T-S fuzzy model rule can be described as follow:

$$R_i: \text{ If } \mathbf{x} \text{ is } A^i, \text{ Then } y^i = f^i(\mathbf{x}) = a_0^i + a_1^i x_1 + a_2^i x_2 + \dots + a_d^i x_d \quad (13)$$

where $i = 1, 2, \dots, N_R$, N_R is the number of fuzzy rules; $\mathbf{x} = [x_1, x_2, \dots, x_d]$ is the input variable vector for the i^{th} rule, d is the dimension of input variable; $\theta_i = [a_0^i, a_1^i, a_2^i, \dots, a_d^i]$ are the consequent parameters; y^i is an output of the i^{th} fuzzy rule; A^i is a fuzzy variable on product variable space.

Given the input variables \mathbf{x} , the fuzzy model output is computed by a weighted mean defuzzification as follows:

$$\hat{y} = \frac{\sum_{i=1}^{N_R} w_i y^i}{\sum_{i=1}^{N_R} w_i} \quad (14)$$

where the weight strength w_i of the i^{th} rule is calculated as

$$w_i = \mu_{A^i}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{v}_i^x\|^2}{\sigma_i^2}\right) \quad (15)$$

where μ_{A^i} is the membership function of A^i , \mathbf{v}_i^x and σ_i denote the center (or mean) and width (or standard deviation) of the fuzzy set A^i , respectively.

The VABC-FCM based T-S fuzzy model, called *automatic T-S fuzzy model*, is proposed in the following ways. Firstly, the VABC-FCM algorithm is used to automatically partition the input-output data space. In VABC-FCM, only the centers of clusters are encoded. Secondly, the membership matrix is used to indicate the fuzzy partition. Thus, the *automatic T-S fuzzy model* can be represented as below:

$$R_i: \text{IF } \mathbf{x} \text{ is in cluster } C_i(v_i^x, u_i) \text{ Then} \quad (16)$$

$$y^i = a_0^i + a_1^i x_1 + a_2^i x_2 + \dots + a_d^i x_d$$

where vector \mathbf{u}_i , the i^{th} row of membership degree matrix U , is calculated as:

$$\mathbf{u}_i = \left\{ u_{ij} \mid u_{ij} = D_{ji}^{-2/(m-1)} / \sum_{i=1}^{N_R} D_{ji}^{-2/(m-1)}, j = 1, 2, \dots, N \right\} \quad (17)$$

The following Algorithm 1 depicted the procedure used to automatically extract T-S fuzzy model.

Algorithm 1 VABC-FCM based automatic T-S fuzzy model

Identification stage

Input: Input-output $Z = [X, \mathbf{y}]$, where X is input matrix and \mathbf{y} is output vector.

Output: Structure of automatic T-S fuzzy model: $\{V, U, \theta\}$.

Step 1: Initialization.

Step 2: Apply the VABC-FCM to automatically determine the rule number N_R , the center coordinates $V = \{\mathbf{v}_k \mid k = 1 \text{ to } N_R\}$ of the N_R rules, and the membership matrix $U = \{u_{ik} \mid i = 1 \text{ to } N\}$.

Step 3: Determine the local model in each rule by minimizing:

$$\min_{\theta_k} \frac{1}{N} (\mathbf{y} - X_e \theta_k)^T \Phi_k (\mathbf{y} - X_e \theta_k)$$

where $X_e = [1, X]$ is the matrix extended by a unitary column and the diagonal matrix Φ_k is defined as:

$$\Phi_k = \text{diag}[u_{k1}, u_{k2}, \dots, u_{kN}]$$

Thus, the consequent parameters of the k^{th} local model is given by:

$$\theta_k = (X_e^T \Phi_k X_e)^{-1} X_e^T \Phi_k \mathbf{y}$$

Evaluation stage

Input: An arbitrary input $\mathbf{z} = [\mathbf{x}, y]$ and T-S fuzzy model: $\{V, U, \theta\}$.

Output: Prediction \hat{y} of \mathbf{x} .

Step 1: Calculate the distances between \mathbf{z} and center vectors in V ;

Step 2: Calculate the membership degrees of sample \mathbf{z} belonging to each local model; take the membership degrees as the weights of \mathbf{z} belong to the N_R local models

Step 3: Calculate the response y^k of \mathbf{x} for local model θ_k , and obtain the prediction \hat{y} of \mathbf{x} .

To validate the performance of the automatic T-S fuzzy model, the nonlinear static system [33] is used to validate the prediction accuracy of the proposed model, whereas the Box-Jenkins gas furnace system [3] is used to validate the approximation ability of the proposed model. In the two examples, the parameters of automatic T-S fuzzy model are initialized as follows: food number $n = 100$, $C_{\max} = 500$, $P_0 = 0.3$ and $d_{\max} = 15$. The Mean Square Error (MSE) is used as a performance measure, defined as:

$$MSE = \frac{1}{N} \sum_{k=1}^N (\hat{y}_k - y_k)^2$$

To show the powerful performance, our proposed method is compared with other advanced T-S fuzzy models as mentioned in the Introduction.

Example 1: Nonlinear static system [33]

The nonlinear static system with two inputs, x_1 and x_2 , and a single output, y :

$$y = (1 + x_1^2 + x_2^{1.5})^2, \quad 1 \leq x_1, x_2 \leq 5$$

As in [33], 50x2 input-output data, consisting of 50 training and testing samples respectively, are used. With the 50 input-output training samples, the automatic T-S fuzzy model can be constructed. The experiments are run for 50 times. The automatic T-S fuzzy model for random one of the 50 running times is presented as follows.

R_1 : IF \mathbf{x} is in C_1 ((1.2721, 1.4796), \mathbf{u}_1)

Then $y^1 = 2.6648 + 0.7508x_1 + 0.7408x_2$

R_2 : IF \mathbf{x} is in C_2 ((2.6415, 2.0677), \mathbf{u}_2) Then $y^2 = 4.1955 - 0.2718x_1 - 0.6389x_2$

R_3 : IF \mathbf{x} is in C_3 ((4.302, 4.3758), \mathbf{u}_3)

Then $y^3 = 0.9269 + 0.0274x_1 + 0.0542x_2$

where \mathbf{u}_1 , \mathbf{u}_2 and \mathbf{u}_3 among the rules R_1 , R_2 and R_3 are indicated in Fig. 2.

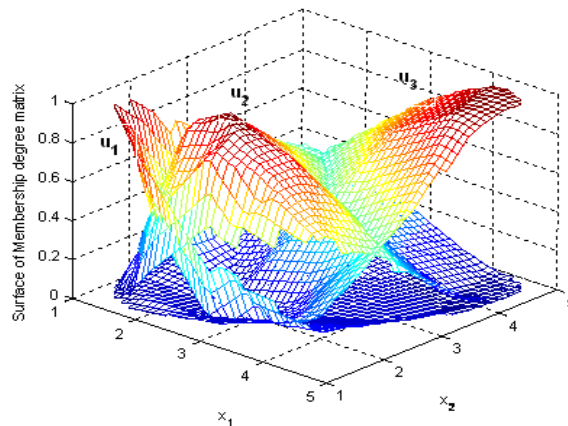


Fig. 2. Membership matrix for the nonlinear static system by using VABC-FCM algorithm

Fig. 3 shows the prediction of the T-S fuzzy model and the prediction errors for the 50 testing samples. In this case, the training MSE is 0.024911, and the prediction/testing MSE is 0.055616.

The average MSE of 50 running times is used as the total performance criterion. The average training MSE is $0.0249 \pm 3.851e-5$, and average prediction MSE is $0.0556 \pm 1.8136e-5$, as shown in Table 1. For comparison,

the other three advanced methods [20, 33, 40] are also run here. The results are presented in Table 1. It suggests from Table 1 that our method has high prediction accuracy with appreciated rules.

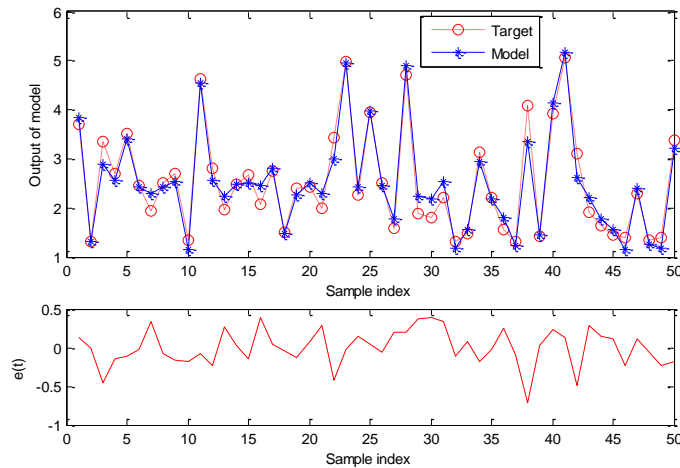


Fig. 3. Predictions of the 50 testing samples using automatic T-S fuzzy model

Table 1. Comparison results for nonlinear static system*1

Method	Rules (mean)	Prediction MSE (mean \pm std.)
Literature [33]	6	0.079
Literature [20]	4/5	0.114 / 0.102
VGA-FCM [40]	4	0.069 \pm 0.012
Our method	3	0.0556 \pm 1.8136e-5

*1 In literatures [20 33], the Fuzzy T-S models are identified using clustering technique, thus the Std. is absence.

Example 2: Box-Jenkins (BJ) gas furnace [3]

This example uses BJ gas furnace data to show the approximation ability of the proposed model. The model holding lower approximation error with fewer rules is the best one. The BJ dataset, recorded from a combustion process of a methane-air mixture, has been used widely to validate the performance of the new proposed modeling methods. There are originally 296 data points, $t = 1$ to 296. $y(t)$ is the output CO₂ concentration and $u(t)$ is the input gas flowing rate. It has been found in the most literatures that three popular sets of input variables for predicting $y(t)$: (1) $y(t-1)$ and $u(t-4)$, (2) $y(t-1)$, $y(t-2)$, $u(t-3)$ and $u(t-4)$, and (3) $y(t-1)$, $y(t-2)$, $y(t-3)$, $u(t-1)$, $u(t-2)$ and $u(t-3)$. All these three kinds of input variables are considered in our study.

As that done in Example 1, the experiment is implemented for 50 times, and the average MSE is taken as the performance criterion. The average training/Approximation MSE for different type of input variables are presented

in Table 2, as well as the Std. values of the MSE. In Table 2, we also present the results obtained by using other advanced methods. It shows that our proposed method outperforms other methods.

Table 2. Comparison results for BJ gas furnace data set

Methods	Input variables	rules (mean)	MSE (mean \pm std)
PSO [4, 19, 39]	$y(t-1), u(t-4)$	3.64	0.1550 ± 0.0422
CRPSO [39]	$y(t-1), u(t-4)$	3.96	0.1428 ± 0.0138
GA [9, 10, 16, 39]	$y(t-1), u(t-4)$	4.32	0.1478 ± 0.0109
DE [11, 39]	$y(t-1), u(t-4)$	4.96	0.1768 ± 0.0602
VGA-FCM [40]	$y(t-1), u(t-4)$	3.74	0.1515 ± 0.0312
Literature [37]	$y(t-1), u(t-4)$	25	0.328
Literature [6]	$y(t-1), u(t-4)$	3	0.2678
Literature [20]	$y(t-1), u(t-4)$	4	0.426
Literature [21]	$y(t-1), y(t-2),$ $u(t-3), u(t-4)$	5	0.248
Literature [33]	$y(t-1), y(t-2),$ $y(t-3), u(t-1),$ $u(t-2), u(t-3)$	2	0.068
Our method	$y(t-1), u(t-4)$	4	$0.1256 \pm 2.7755e-5$
	$y(t-1), y(t-2),$ $u(t-3), u(t-4)$	3	$0.0581 \pm 1.5160e-4$
	$y(t-1), y(t-2),$ $y(t-3), u(t-1),$ $u(t-2), u(t-3)$	3	$0.0548 \pm 1.5164e-4$

4. Application to design fuzzy predictive controller

Section 4.1 shows the procedures used to design predictive controller. Consequently, some experiments are conducted to validate the performance of the designed predictive controller in section 4.2.

4.1. Automatic T-S model and (V)ABC based predictive controller

There are various families of strategies used to design predictive controller since the basic concept of predictive control is proposed [26]. Although there exist differences in assumptions and designing ideas, the key idea of these strategies are similar. More precisely, for a given Single Input (control input or controller output u) Single Output (process output or plant output y) system (SISO) illustrated in Fig. 4, such key idea can be interpreted as follows:

At time t , the current plant state is sampled. The (corrected) plant outputs $\{\hat{y}(t), \hat{y}(t+1), \dots, \hat{y}(t+N_p-1)\}$ are computed for a time horizon $[t, t+N_p-1]$ in the further. These predicted outputs depend on the information that has been

collected, the dynamic predictive model and the assumed control input trajectory $\{v(t), v(t+1), \dots, v(t+N_u-1)\}$ for a appreciated time horizon $[t, t+N_u-1]$ (i.e., N_u is generally smaller than the prediction horizon N_p);

Select optimal control sequence $\{v^*(t), v^*(t+1), \dots, v^*(t+N_u-1)\}$ in the assumed input trajectory v to make the plant outputs \hat{y} approximate the reference trajectory y_r as possible. Such approximation can be realized by minimizing a objective function, generally defined as:

$$\min J = \sum_{k=1}^{N_p} [\hat{y}(t+k) - y_r(t+k)]^2 + \lambda \sum_{k=0}^{N_u-1} (\Delta v(t+k))^2$$

where, $\Delta v(t+k) = v(t+k) - v(t+k-1)$, y_r is reference trajectory and λ is constant weight vector.

Apply the first element of the optimal control sequence v^* as the actual control input signal to the plant, i.e., $u(t) = v^*(t)$;

Repeat the above operations, once the later sampling time comes.

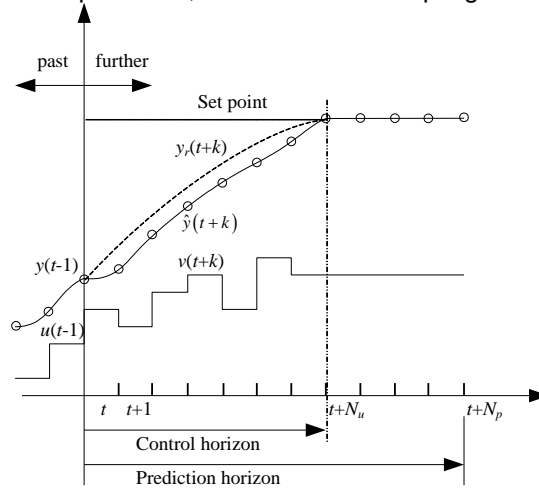


Fig. 4. Key idea of predictive control strategy by using SISO system

In practice, the plant (i.e., object) to be controlled is much more complex and Multiple Inputs Multiple Outputs (MIMO) system is usually faced. In this case, the MIMO can be viewed as the composition of some subsystems with multiple inputs and single output (MISO). By comparing with SISO, note that the MISO only has some more inputs. To avoid confusion with scalar variables such as u , v , y , y_r and \hat{y} in the SISO, the corresponding vector variables are denoted by \underline{u} , \underline{v} , \underline{y} , \underline{y}_r and $\underline{\hat{y}}$ in MIMO system. In what follows, the predictive controller is designed for MIMO system in general.

According to the above key idea, we can see that there are three significant characteristics for the model predictive control algorithm: dynamic predictive model, rolling optimization and feedback correction. In our study, the above key idea is also applied. The predictive controller is designed according to the following four steps.

- (1) Structure design for the predictive controller

For a given MIMO system, the predictive controller is designed as shown in Fig. 5. It can be seen that the structure maintains the three significant characteristics, holding by the traditional predictive controller. The two main differences are the applications of automatic T-S fuzzy model and (V)ABC algorithm respectively as the dynamic prediction model and rolling optimizer.

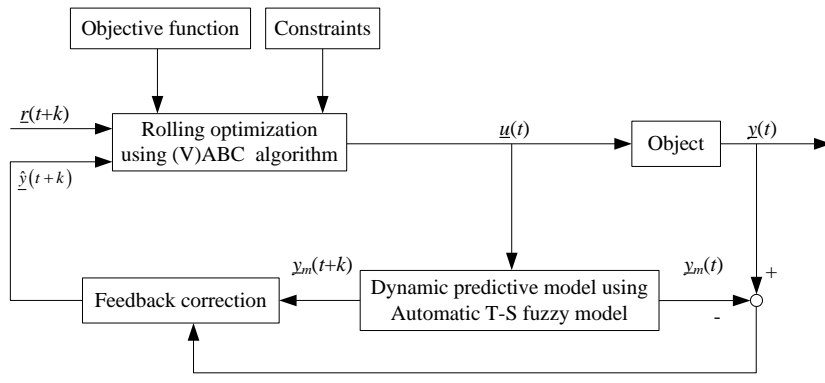


Fig. 5. Structure diagram of the designed predictive controller

(2) Dynamic predictive model by using automatic T-S fuzzy model

For MIMO system with n_i inputs and n_o outputs, the dynamic predictive model holds the following form in general:

$$\underline{y}_m(t) = f(\underline{y}(t-1), \underline{y}(t-2), \dots, \underline{y}(t-n_y), \underline{u}(t-1), \underline{u}(t-2), \dots, \underline{u}(t-n_u)) \quad (18)$$

where $f(\cdot)$ is the (nonlinear) function describing the plant system, n_u and n_y are the orders of inputs and outputs respectively, $\underline{u}(t-k)$ and $\underline{y}(t-k)$ are respectively the control input vector and plant/object output vector, defined respectively as

$$\underline{u}(t-k) = [u_1(t-k), \dots, u_j(t-k), \dots, u_{n_u}(t-k)]^T$$

$$\underline{y}(t-k) = [y_1(t-k), \dots, y_i(t-k), \dots, y_{n_o}(t-k)]^T$$

Such general form (18) can be decomposed into number of n_o subsystems with n_i inputs and single output. Thus, we have

$$\underline{y}_m(t) = \begin{bmatrix} f_1(y_1(t-1), \dots, y_1(t-n_y), \underline{u}(t-1), \dots, \underline{u}(t-n_u)) \\ f_2(y_1(t-1), \dots, y_2(t-n_y), \underline{u}(t-1), \dots, \underline{u}(t-n_u)) \\ \vdots \\ f_{n_o}(y_{n_o}(t-1), \dots, y_{n_o}(t-n_y), \underline{u}(t-1), \dots, \underline{u}(t-n_u)) \end{bmatrix} \quad (19)$$

where $f_i(\cdot)$ is the dynamic predictive model for the i^{th} MISO subsystem.

The dynamic predictive model $f_i(\cdot)$ ($1 \leq i \leq n_o$) can be in various forms such as Neural network, fuzzy model and statistic regression model. In our study,

we apply the automatic T-S fuzzy model as the dynamic predictive model $f(\cdot)$ due to its powerful approximation ability and prediction ability, as already discussed in Section 3. Once the input variables of the automatic T-S fuzzy model are determined, the automatic T-S model can be identified to predict the outputs of the object in the further by using the information of these input variables collected at the past and current states. The way how to establish and identify the automatic T-S fuzzy model can refer to Section 3.

(3) Feedback correction strategy

As we known, the performance of automatic T-S fuzzy model may be degenerated along with the running times, although the automatic T-S fuzzy model holds powerful approximation and prediction abilities. This is because the automatic T-S fuzzy model may be established under limited information, i.e., the information is collected at the past and current states. Along with the running times, some new useful information maybe occurs. As well, the characteristics of the object maybe change along with the running times. For instance, characteristics of the object will degenerate when the service life of object is closed. In addition, it is unfeasible to frequently identify the dynamic predictive model in practice. Therefore, feedback correction is necessary.

In this study, we feed back the prediction error and apply it to correct the output of the dynamic predictive model. More precisely, we have

$$\begin{aligned} \hat{y}(t+k) &= \underline{e}(t) + \underline{y}_m(t+k) \\ &= \underline{e}(t) + f\left(\begin{array}{c} \hat{y}(t+k-1), \dots, \hat{y}(t+1), \underline{y}(t), \dots, \underline{y}(t+k-n_y), \\ \underline{v}(t+k-1), \dots, \underline{v}(t), \underline{u}(t-1), \dots, \underline{u}(t+k-n_u) \end{array}\right), k=1, \dots \end{aligned} \quad (20)$$

where $\underline{u}(t+k-n_u), \dots, \underline{u}(t-1)$ are the actual input trajectory at the past cases, $\underline{v}(t), \dots, \underline{v}(t+k-1)$ are the assumed control input trajectory at the current and further cases, and $\underline{e}(t)$ is the feedback prediction error defined as:

$$\underline{e}(t) = \underline{y}(t) - \hat{y}(t) = \underline{y}_m(t) - f(\underline{y}(t-1), \dots, \underline{y}(t-n_y), \underline{u}(t-1), \dots, \underline{u}(t-1)) \quad (21)$$

(4) Rolling optimization by using (V)ABC algorithm

The rolling optimization aims to find the optimal control sequence $\{\hat{v}^*(t+k) \mid k = 0, 1, \dots, N_U-1\}$ under the following two constraints:

$$\text{Input constraint:} \quad \underline{u}_{\min} \leq \underline{v}(t+k) \leq \underline{u}_{\max} \quad (22)$$

$$\text{Input move constraint:} \quad -\Delta \underline{u}_{\max} \leq \Delta \underline{v}(t+k) \leq \Delta \underline{u}_{\max} \quad (23)$$

where \underline{u}_{\min} and \underline{u}_{\max} are the lower and upper constraints of the control input \underline{u} (Note that \underline{v} is the assumed control inputs. Therefore, we use \underline{u}_{\min} and \underline{u}_{\max} instead of \underline{v}_{\min} and \underline{v}_{\max}), $\Delta \underline{v}(t+k) = \underline{v}(t+k) - \underline{v}(t+k-1)$ and $\Delta \underline{u}_{\max}$ is the maximal variation in the control input.

To apply the (V)ABC as the optimizer, we should encode the assumed control inputs \underline{v} as population of food sources in advance. The i^{th} food source can be encoded in the following form:

$$\mathbf{x}_i = \left[\underline{v}'(t)^T, \underline{v}'(t+1)^T, \dots, \underline{v}'(t+k-1)^T, \underline{v}'(t+k)^T, \dots, \underline{v}'(t+N_u-1)^T \right] \quad (24)$$

where $\underline{v}'(t+k) = \left[v'_1(t+k), v'_2(t+k), \dots, v'_{n_i}(t+k) \right]^T, k = 0, 1, \dots, N_u - 1$

To satisfy the two constraints in Eqs. (22) and (23) simultaneously, it is unfeasible to produce the food sources by applying operation in Eq. (1). Here, the initial population of control inputs at time t is determined according to

$$v'_j(t) = \min \left\{ u_{j,\max}, \max \left\{ u_{j,\min}, u_j(t-1) + \text{rand}[-1,1] \times \Delta u_{j,\max} \right\} \right\}, j = 1, 2, \dots \quad (25)$$

and the control inputs at other times are defined according to

$$v'_j(t+k) = \min \left\{ u_{j,\max}, \max \left\{ u_{j,\min}, v'_j(t+k-1) + \text{rand}[-1,1] \times \Delta u_{j,\max} \right\} \right\} \quad (26)$$

where, $k = 1$ to N_u , $u_{j,\min}$, $u_{j,\max}$ and $\Delta u_{j,\max}$ are the j^{th} element of the \underline{u}_{\min} , \underline{u}_{\max} and $\underline{\Delta u}_{\max}$, respectively.

Note that, the new produced candidate food source \mathbf{x}_{new} by applying operation (3) might not satisfy the input move constraint. This situation can be avoided by using the following correction mechanism for each input variable j .

If $\Delta = v_j^{new}(t+k) - v_j^{new}(t+k-1) > \Delta u_{j,\max}$, we have

$$v_j^{new}(t+k) = v_j^{new}(t+k) - (\Delta - \Delta u_{j,\max});$$

If $\Delta = v_j^{new}(t+k) - v_j^{new}(t+k-1) < -\Delta u_{j,\max}$, we have

$$v_j^{new}(t+k) = v_j^{new}(t+k) - (\Delta + \Delta u_{j,\max}).$$

Once the assumed control inputs are encoded as population of food sources, the rest task is to design an objective function for the rolling optimization. For convenience, we apply the quadratic function, defined as:

$$J_{MIMO} = \sum_{i=1}^{n_o} \gamma_i J_i \quad (27)$$

where γ_i are constant weights, and J_i is the quadratic objective function of the i^{th} MISO system and is defined as

$$J_i = \sum_{k=N_0}^{N_p} \left\| y_{r,i}(t+k) - \hat{y}_i(t+k) \right\|^2 + \lambda_i \sum_{k=0}^{N_u-1} \left\| \Delta v_i(t+k) \right\|^2 \quad (28)$$

where N_0 is time delay, reference trajectory $y_{r,i}(t+k) = \alpha_{i,k} y(t) + (1 - \alpha_{i,k}) y_{sp,i}$, $y_{sp,i}$ is set point and $\alpha_{i,k}$ is constant weight, $\alpha_{i,k} = 0.1$ is used in general.

The following Algorithm 2 shows the steps to realize the proposed predictive controller.

Algorithm 2: Automatic T-S model & (V)ABC based predictive controller

1. Initialization
Predetermine the parameters such as prediction horizon N_p , control horizon N_u , weight coefficients γ_i and λ_i , the limits of control input \underline{u}_{\min} , \underline{u}_{\max} and $\Delta\underline{u}_{\max}$, and the control parameters n , d_{\max} , P_0 and C_{\max} in VABC.
 2. At time t , read the information such as the plant outputs $\underline{y}(t-1), \dots, \underline{y}(t-n_y)$ and actual control inputs $\underline{u}(t-1), \dots, \underline{u}(t-n_u)$.
 3. Generate the population of food sources for the current time t .
 4. By taking Eq. (27) as the objective function, the optimal assumed control input trajectory $\{\underline{v}^*(t+k) \mid k = 0, 1, \dots, N_u-1\}$ are obtained by using the (V)ABC algorithm.
 5. Apply the first element of the optimal control sequence \underline{v}^* as the actual control input to the plant, i.e., $\underline{u}(t) = \underline{v}^*(t)$.
 6. Go to the second step and repeat the operations interpreted in steps 2 ~ 5, when the new state comes, i.e., $t = t + 1$.
-

To implement the Algorithm 2, some parameters should be predetermined such as N_p , N_u , γ_i , λ_i , \underline{u}_{\min} , \underline{u}_{\max} , $\Delta\underline{u}_{\max}$, n , d_{\max} , P_0 and C_{\max} . Among them, the latter four parameters can be initialized as that done in [31]. The weight coefficients λ_i are used to reduce the influence of large changing in control input. If the control system is stable and the changing in control input is not large, the coefficients λ_i can take a relatively smaller value. On the contrary, large value can be taken by λ_i . In our method, the coefficients λ_i can take a small value because the control input has been restricted in a certain interval in advance, as shown in Eqs. (22) and (23). Here, coefficients $\lambda_i = 0.5$ for $i = 1$ to n_o . The weight coefficients γ_i are used to indicate the importance of each sub objective function J_i . Here, $\gamma_i = 1$ for $i = 1$ to n_o , because we consider all the objective functions J_i play the same contribution.

The prediction horizon N_p and control horizon N_u play important roles for the predictive controller. The prediction horizon controls the stability and speediness of the dynamic response. The smaller N_p is, the speediness is faster but the stability is worse; on the contrary, the stability is better but the speediness is slower. In addition, N_p taking large value will lead to the increasing computational complexity of the controller. The control horizon N_u is critical to the sensitivity of the controller: the higher N_u is, the controller is more sensitive. However, large N_u will lead to the worse stability. In the most of cases, the control horizon plays opposite role to the controller as that of the prediction horizon. Therefore, we can firstly determine the control horizon according to the dynamic characteristics of the object and then determine the prediction horizon according to the simulation results. Usually, $N_u = 1 \sim 2$ is used for the object with simple dynamic characteristics, and $N_u = 4 \sim 8$ is used for the complex object.

4.2. Superheated steam temperature control system

Continuous process in power plant and power station are complex systems characterized by nonlinearity, uncertainty and load disturbance. The superheater is an important part of the steam generation process in the boiler-turbine system, where steam is superheated before entering the turbine that drives the generator.

Superheater is made of material that have characteristic of high temperature resistant during normal operation, and the average temperature of superheater is close to highest permitted temperature of material. It will result in malfunction and damage when superheater is overheated. On the other hand, when superheated steam temperature is too low, it will reduce the thermal efficiency of power plant and influence the safe operation of turbine. Therefore, superheated steam temperature is one of the most important factor determining the safety and economic of thermal power plant.

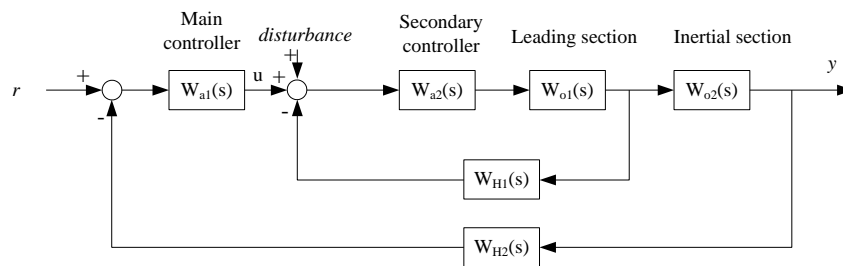


Fig. 6. Structure of superheated steam temperature control system

Sources of disturbances, affecting the superheated steam temperature, are various, such as, the steam flow, combustion conditions of boiler, enthalpy of steam entering the superheater, the change of temperature and flow speed of fume that flows through the superheater, etc. It is difficult to reach small dynamic deviation. Moreover, the temperature adjustment of export steam is required to prevent the superheater been overheated. At present the Spray Water control strategy is adopted as steam temperature adjustment approach for most thermal power plants. In such case, the input, denoted as $u(t)$, is the amount of Spray Water (kg/s), and the output, denoted as $y(t)$, is the superheated steam temperature ($^{\circ}\text{C}$). Thus, the structure of superheated steam temperature control system can be interpreted in Fig. 6.

Some parameters of the superheated steam system in Fig. 6 are presented as follows [5]:

$$W_{o1}(s) = \frac{8}{(1+15s)^2} \quad ^{\circ}\text{C} / \text{mA}, \quad W_{o2}(s) = \frac{1.125}{(1+25s)^3} \quad ^{\circ}\text{C} / \text{mA}$$

$$W_{H1}(s) = 0.1 \quad \text{mA} / ^{\circ}\text{C}, \quad W_{H2}(s) = 0.1 \quad \text{mA} / ^{\circ}\text{C}, \quad W_{a2}(s) = 25$$

According to these parameters above, the generalized transfer function of the superheated temperature can be calculated as:

$$G(s) = \frac{W_{a1}(s)W_{o1}(s)W_{o2}(s)W_{H2}(s)}{1 + W_{a1}(s)W_{o1}(s)W_{H2}(s)}$$

$$= \frac{22.5}{3516000s^5 + 890625s^4 + 401250s^3 + 41850s^2 + 1605s + 21}$$

To simulate for the superheated steam temperature system (see in Fig. 6), suppose the input $u(t) \in [-10, 10]$, i.e., $u_{\min} = -10$, $u_{\max} = 10$, is generated from a pseudo random signal generator. The sampling period is set as 5 second (s) and we continuously record a set of samples for 2000 seconds. In other word, four hundreds of points of signals $u(t)$ are collected. Corresponding to the input $u(t)$, the output $y(t)$ can be obtained by using some ways. An intuitionistic way is to deduce $y(t)$ in the SIMULINK environment in MATLAB® software according to the calculus $G(s)$. Fig. 7 shows the input $u(t)$ and output $y(t)$ generated by using this way.

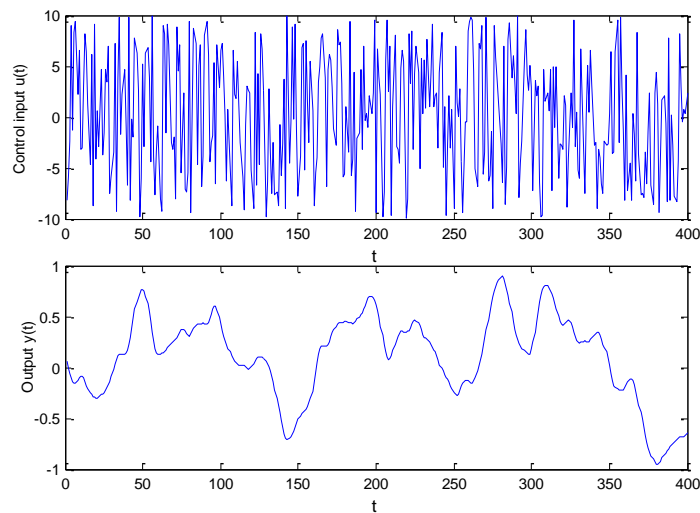


Fig. 7 The output and input of training samples

By selecting $\mathbf{x} = [y(t-1), y(t-2), y(t-3), u(t-1), u(t-2), u(t-3)]$ as the input variables (i.e., $n_y = n_u = 3$) and $y(t)$ as the output, thus the amount of 400 training samples can be constructed. By using these samples, the automatic T-S fuzzy model can be constructed for the superheated steam temperature, as follows.

R_1 : If \mathbf{x} is in C_1 ((-0.9556, -0.9556, -0.9556, -2.7559, -3.0635, -3.6739), \mathbf{u}_1)
Then

$$y_m(t) = -0.0015 + 2.718y(t-1) - 2.5562y(t-2) + 0.8403y(t-3)$$

$$+ 0.0003u(t-1) + 0.0001u(t-2) - 0.0001u(t-3)$$

R_2 : If \mathbf{x} is in C_2 ((0.2858, 0.2865, 0.2864, 0.6995, 0.7949, 0.8041), \mathbf{u}_2) Then

$$y_m(t) = -0.0015 + 2.718y(t-1) - 2.5562y(t-2) + 0.8403y(t-3) \\ + 0.0003u(t-1) + 0.0001u(t-2) - 0.0001u(t-3)$$

where the membership degrees u_1 and u_2 are indicated in Fig. 8.

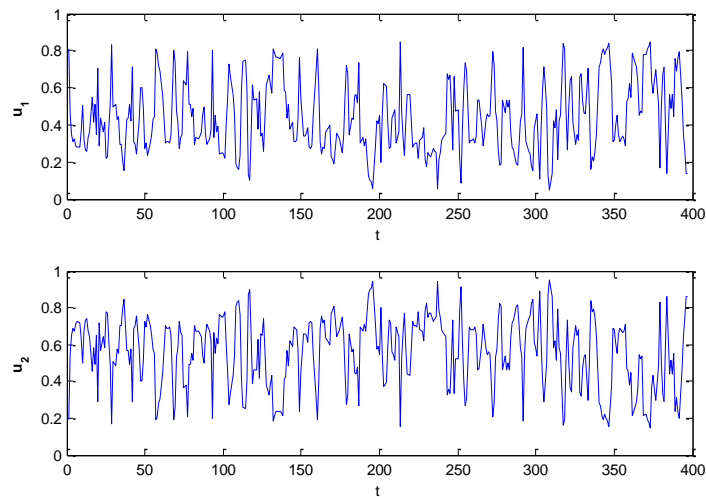


Fig. 8 Membership degrees distribution of 2 clusters for the superheated steam temperature

With the above established dynamic predictive model, we still should predetermine the prediction horizon and control horizon. According to the discussions about these two parameters in Section 4.1, here we assume $N_u = 1$ and $N_p = 10$ by considering the fact that the superheated steam temperature system is not too complex and does not have system time delay, i.e., $N_0 = 0$.

To validate the performance of our method, the generalized predictive controller (GPC) [7, 8] and the traditional PI controller are also studied. The prediction horizon N_p and control horizon N_u taken by GPC are the same as that in our method. The main controller for the traditional PI is designed as

$$W_{ai}(s) = \frac{1}{0.5} \left(1 + \frac{1}{74s} \right).$$

Four typical study cases are conducted to show whether our method is robust or not when the disturbance, prediction horizon, inertial coefficient and gain increase respectively.

Case 1: Study on the influence of step disturbance

In this case, we analyze the laws how the step changes of reference value (r) and disturbance (d) influence on the performance of our method. Fig. 9 shows the results when $r = +1$ mA, and Fig. 10 shows the results when $d = +4$ mA. It can be seen from Figs. 9 and 10 that our controller can rapidly approach to the new set point. As well, we can see that the performance of our method outperforms the traditional PI and GPC.

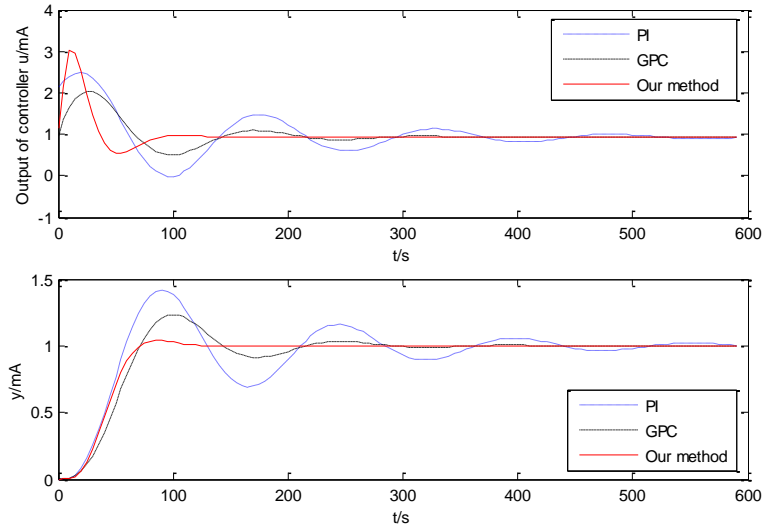


Fig. 9 The response curve in the case $r = 1 \text{ mA}$

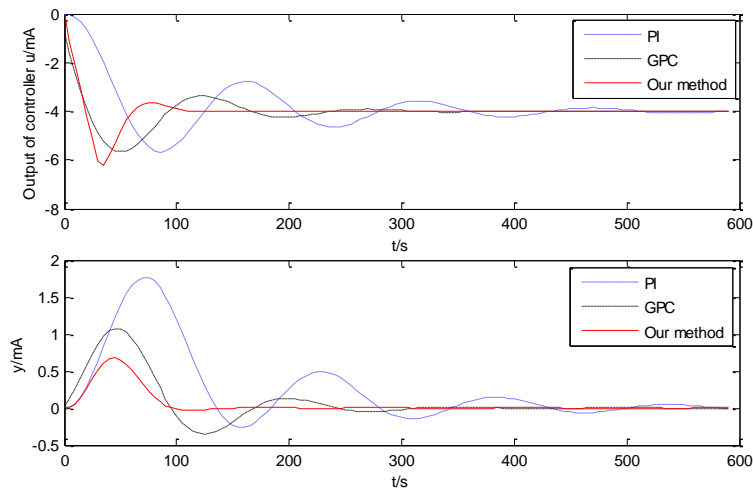


Fig. 10 The simulations under the condition disturbance = +4 mA

Case 2: Study on the influence of prediction horizon

In this case, we analyze the laws how the prediction horizon N_p influence on the performance of our method. Here, we only consider the following three cases: $N_p = 6, 10,$ and 16 under the condition that disturbance $d = +4 \text{ mA}$. The results are shown in Fig. 11. From Fig. 11, we can see that our method has powerful ability on tracking and anti-jamming. The higher the N_p is, the controller is more robust in stability.

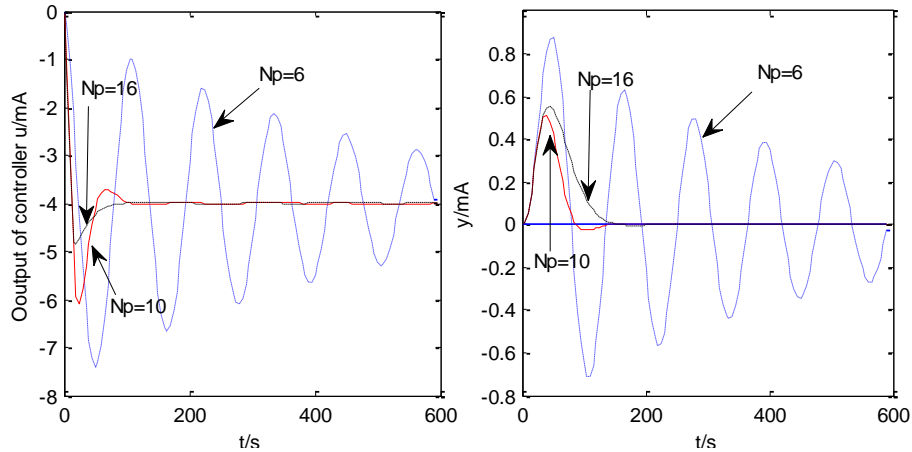


Fig. 11 Influence of N_p on results under the condition disturbance = 4 mA and

Case 3: Study on the influence of inertial coefficient

The dynamic characteristics of superheated steam temperature always vary with the changing working conditions. By considering this fact, we want to see whether our method is valid or not when the parameters (i.e., the inertial coefficient in W_{o2}) of the superheated steam temperature system change in a large range. In other words, we analyze whether our method is sensitive to the inertia of the object or not. To study on the influence of inertial coefficient, we suppose $W_{o1}(s)$ is unchanged and the inertial coefficient in $W_{o2}(s)$ increases from 25 to 35, i.e., the $W_{o2}(s)$ is redefined as follows (Note that the established automatic T-S fuzzy model is not repeatedly identified according to training samples derived from the following redefined transfer function):

$$W'_{o2}(s) = \frac{1.125}{(1 + 35s)^3} \text{ } ^\circ\text{C} / \text{mA}$$

The simulation results are shown in Fig. 12. We can see from Fig. 12 that our method has small oscillation, and the performance of our method outperforms the traditional PI and GPC.

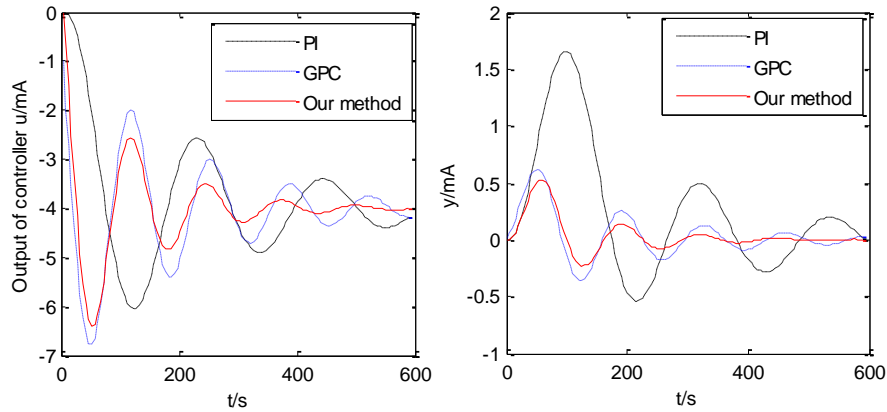


Fig. 12 Results in the case when the inertia of the object increases

Case 4: Study on the influence of gain

In this case, we want to see whether the proposed method is sensitive to the change of gain or not. We suppose $W_{o1}(s)$ is unchanged and gain in $W_{o2}(s)$ increases from 1.125 to 1.875, i.e., the $W_{o2}(s)$ is redefined as:

$$W'_{o2}(s) = \frac{1.875}{(1 + 25s)^3} \text{ } ^\circ\text{C} / \text{mA}$$

The simulation results are shown in Fig. 13. It can be seen from Fig. 13 that our method is robust in the changes of gain and its performance outperforms the traditional PI and GPC.

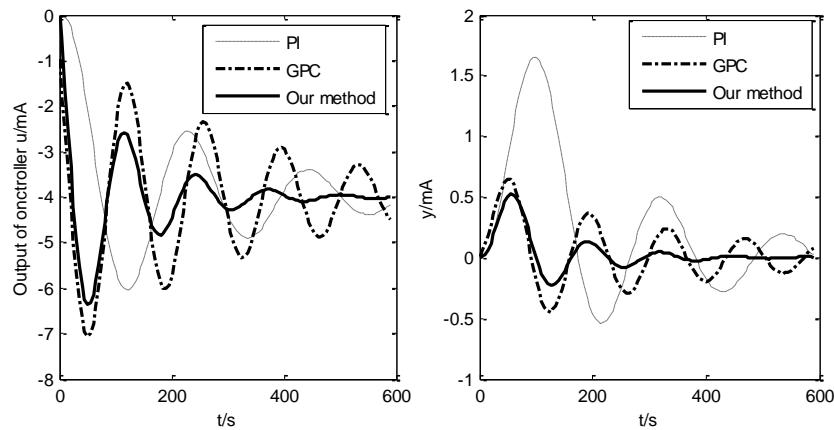


Fig. 13 Results in the case when proportional gain increases from 1.125 to 1.875

In the study cases 3 and 4, the inertial coefficient and gain can also decrease and we have the similar results as above two study cases.

Form the above four study cases, we can see that our method has powerful performance and outperforms the traditional PI strategy and GPC strategy.

5. Conclusions

In this study, a novel methodology for automatically extracting T-S fuzzy model with enhanced performance is proposed by using a novel swarm intelligent fuzzy clustering technique, i.e., the Variable string length Artificial Bee Colony (ABC) algorithm based Fuzzy C-Mean clustering (VABC-FCM). Use of the VABC-FCM algorithm makes the proposed methodology can automatically evolve the rule number from data without knowing the rule number as a priori. Moreover, the output fuzzy partition matrix of VABC-FCM is sufficiently applied in the identification process of T-S fuzzy model, which brings the convenience for model construction and reduces the approximation error. Some numerical examples are used to validate the proposed T-S fuzzy model. The results suggest that it has high prediction accuracy with appreciated rule number.

Consequently, a new fuzzy model predictive controller is designed by using the proposed T-S fuzzy model as its dynamic predictive model and by using (V)ABC algorithm as its rolling optimizer. Taking the superheated steam temperature in power plant as the example, some experiments were conducted to validate the performance of the proposed fuzzy model predictive controller. The experimental results show that the proposed fuzzy model predictive controller has powerful performance. In addition, we compare our method with the popular generalized predictive controller and PI. It shows that our method outperforms the generalized predictive controller and traditional PI controller.

References

1. Babuska, R.. Fuzzy modeling for control. Kluwer Academic Publisher, Boston. (1998)
2. Bezdek, J.C.. Pattern recognition with fuzzy objective function algorithms. Plenum, New York. (1981)
3. Box, G.E.P., Jenkins, G.M.. Time series analysis, forecasting and control. Sanfrancisco, CA: Holden Day. (1970)
4. Chen, C.C.. A PSO-based method for extracting fuzzy rules directly from numerical data. *Cybern. Syst.* Vol. 37, No. 7, 707-723. (2006)
5. Chen, L.J.. Principle of automatic control in thermal process (in chinese). Water Resources and Electric Power Press, Beijing. (1982)
6. Chen, J.Q., Xi, Y.G., et al.. A clustering algorithm for fuzzy model identification. *Fuzzy sets and systems*, Vol. 38, 319-329. (1998)
7. Clarke, D.W., Mohtadi, C., Tuffts, P.S.. Generalized predictive control: Part 1, the basic algorithm. *Automatica*, Vol. 23, 137-148. (1987)
8. Clarke, D.W., Mohtadi, C., Tuffts, P.S.. Generalized predictive control: Part 2, extensions and interpretations. *Automatica*, Vol. 23, 149-160. (1987)
9. Cordon, O., Herrera, F.. A two-stage evolutionary process for designing TSK fuzzy rule based systems. *IEEE Transaction on Systems, Man and Cybernetic B*, Vol. 29, No. 6, 703-715. (1999)

10. Cordon, O., Herrera, F., Gomide, F., Hoffmann, F., Magdalena, L.. Ten years of genetic fuzzy systems: current framework and new trends. In: proceedings of IFSA and NAFIPS, Vol. 3, 1241-1246. (2001)
11. Eftekhari, M., Katebi, S.D., Karimi, M., Jahanmiri, A.H.. Eliciting transparent fuzzy model using differential evolution. Applied soft computing, Vol. 8, 466-476. (2008)
12. Fisher, M., Nelles, O., Isermann, R.. Adaptive predictive control of a heat exchanger based on a fuzzy model. Control Eng. Practice, Vol. 6, 259-269. (1998)
13. Hellendoorn, H., Driankov, D.. Fuzzy model identification: selected approaches. Springer, Berlin, Germany. (1997).
14. Jiang, H., Kwong, C.K., Chen, Z., Ysim, Y.C.. Chaos particle swarm optimization and T-S fuzzy modelling approaches to constrained predictive control. Expert systems with applications, Vol. 39, 194-201. (2012)
15. Johansen, T.A., Babuska, R.. On multi-objective identification of Takagi-Sugeno fuzzy model parameters. In preprint 15th IFAC congress, Barcelona, Spain. (2002).
16. Kang, S.J., Woo, C.H., Hwang, H.S., Woo, K.B.. Evolutionary design of fuzzy rule base for nonlinear system modeling and control. IEEE Transaction on Fuzzy Systems, Vol. 8, No. 1, 37-45. (2000)
17. Karaboga, D.. An idea based on honey bee swarm for numerical optimization. Technical report-TR06, Erciyes University, Engineering faculty, Computer Engineering Department. (2005)
18. Karaboga, D., Basturk, B.. On the performance of artificial bee colony (ABC) algorithm. Applied soft computing, Vol. 8, No. 1, 687-697. (2008)
19. Khosla, A., Kumar, S., Aggarwal, K.K.. A framework for identification of fuzzy models through particle swarm optimization. In: IEEE Indicon Conference, Chennai, India, 388-391. (2005)
20. Li, N., Li S.Y., Xi, Y.G.. Multi-model modeling method based on satisfactory clustering. Control theory & application, Vol. 20, No. 5, 783-787. (2003)
21. Lv, J.H., Chen, J.Q., Liu, Z.Y., et al., A study on fuzzy rules based and nonlinear models for thermal processes. Proceedings of the Chinese society for electrical engineering, Vol. 22, No. 11, 132-137. (2002)
22. Martinez, M., Senent, J.S., Blasco, X.. Generalized predictive control using genetic algorithms. Engineering applications of artificial intelligence, Vol. 11, 355-367. (1998)
23. Maulik, U., Bandyopadhyay, S.. Performance evaluation of some clustering algorithms and validity indices. IEEE Transactions on pattern Anal. Machine intelligent, Vol. 24, No. 12, 1650-1654. (2002)
24. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.. Validity index for crisp and fuzzy clusters. Pattern recognition, Vol. 37, No. 3, 487-501. (2004)
25. Pedrycz, W., Reformat, M.. Evolutionary fuzzy modeling. IEEE Transaction on Fuzzy Systems, Vol. 11, No. 5, 652-665. (2003)
26. Richalet, J., Rault, A., Testud, J.L., Papon, J.. Model predictive heuristic control Application to industrial processes. Automatica, Vol. 14, 413-428. (1978)
27. Sarimveis, H., Bafas, G.. Fuzzy model predictive control of non-linear processes using genetic algorithms. Fuzzy sets and systems, Vol. 139, 59-80. (2003)
28. Sousa, J.M., Babuska, R., Verbuggen, H.B.. Fuzzy predictive control applied to an air-conditioning system. Control Eng. Practice, Vol. 5, 1395-1406. (1997)
29. Srinivas, M., Patnaik, L.M.. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, No. 4, 656-667. (1994)

30. Su, B., Chen, Z., Yuan, Z.. Constrained predictive control based on T-S fuzzy model for nonlinear system. *Journal of systems engineering and electronics*, Vol. 18, 95-100. (2007)
31. Su, Z.G., Wang, P.H., Shen, J., Li, Y.G., Zhang, Y.F., Hu, E.J.. Automatic fuzzy partitioning approach using Variable string length Artificial Bee Colony (VABC) algorithm. *Applied soft computing*, Vol. 12, No. 11, 3421-3441. (2012)
32. Sugeno, M., Kang, G.T. Fuzzy modeling and control of multilayer incinerator. *Fuzzy sets and systems*, Vol. 1, No. 8, 329-346. (1986)
33. Sugeno, M., Yasukawa, T.. A fuzzy logic based approach to qualitative modeling. *IEEE Transaction on Fuzzy Systems*, Vol. 1, No. 1, 7–31. (1993)
34. Takagi, T., Sugeno, M.. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, No. 1, 116–132. (1985)
35. Turksen, I.B., Celikyilmaz, A.. Comparison of fuzzy functions with fuzzy rule base approaches. *Int. J. Fuzzy Syst*, Vol. 8, No. 3, 137-149. (2006)
36. Xie, X.L., Beni, G.. A validity measure for fuzzy clustering. *IEEE Transaction on pattern Anal. Machine intelligent*, Vol. 13, No. 8, 841-847. (1991)
37. Xu C.W., Yong, Z.. Fuzzy model identification and self-learning for dynamic systems. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 17, No. 4, 683–689. (1987)
38. Zhang, Y.F., Su, Z.G., Wang, P.H.. A convenient version of T-S fuzzy model with enhanced performance. In: *proceedings of the 8th international conference on Fuzzy Systems and Knowledge Discovery*, 1074-1079. (2011)
39. Zhao, L., Qian, F., Yang, Y.P., et al.. Automatically extracting T-S fuzzy models using cooperative random learning particle swarm optimization. *Applied soft computing*, Vol. 10, 938-944. (2010)
40. Zhu, H.X.. A study on fuzzy modeling and control based on immune optimization algorithms for thermal processes. MS thesis, Southeast University at Nanjing, (2005)

Zhi-gang Su received his M.S. and Ph.D. degrees from Southeast University, Nanjing, China in 2006 and 2010, respectively. He worked with Southeast University from 2010. His research interests are focused on intelligent modeling, machine learning, soft computing and optimization algorithm, as well as their applications.

Pei-hong Wang is a full professor with School of Energy & Environment, Southeast University. His research interests are focused on intelligent modeling, diagnosis and optimization algorithm, as well as the applications of these methods in power plant.

Yu-fei Zhang is an associate professor with School of Energy & Environment, Southeast University. Her research interests are focused on intelligent modeling and control, as well as their applications.

Received: February 22, 2012; Accepted: November 26, 2012.

