Research Article

A Clustering and SVM Regression Learning-Based Spatiotemporal Fuzzy Logic Controller with Interpretable Structure for Spatially Distributed Systems

Xian-xia Zhang,[1] Jun-da Qi,[1] Bai-li Su,[2] Shi-wei Ma,[1] and Hong-bo Liu[3]

[1] *Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronics and Automation, Shanghai University, Shanghai 200072, China*

[2] *School of Electrical Information and Automation, Qufu Normal University, Shandong 276826, China*

[3] *School of Control Science and Engineering, Shandong University, Jinan 250061, China*

Correspondence should be addressed to Xian-xia Zhang, xianxia_zh@shu.edu.cn

Many industrial processes and physical systems are spatially distributed systems. Recently, a novel 3-D FLC was developed for such systems. The previous study on the 3-D FLC was concentrated on an expert knowledge-based approach. However, in most of situations, we may lack the expert knowledge, while input-output data sets hidden with effective control laws are usually available. Under such circumstance, a data-driven approach could be a very effective way to design the 3-D FLC. In this study, we aim at developing a new 3-D FLC design methodology based on clustering and support vector machine (SVM) regression. The design consists of three parts: initial rule generation, rule-base simplification, and parameter learning. Firstly, the initial rules are extracted by a nearest neighborhood clustering algorithm with Frobenius norm as a distance. Secondly, the initial rule-base is simplified by merging similar 3-D fuzzy sets and similar 3-D fuzzy rules based on similarity measure technique. Thirdly, the consequent parameters are learned by a linear SVM regression algorithm. Additionally, the universal approximation capability of the proposed 3-D fuzzy system is discussed. Finally, the control of a catalytic packed-bed reactor is taken as an application to demonstrate the effectiveness of the proposed 3-D FLC design.

1. Introduction

Many industrial processes and physical systems such as industrial chemical reactor [1, 2], semiconductor manufacturing [3], and thermal processing [4] are "distributed" in space. They are usually called spatially distributed systems, or distributed parameter systems [1].
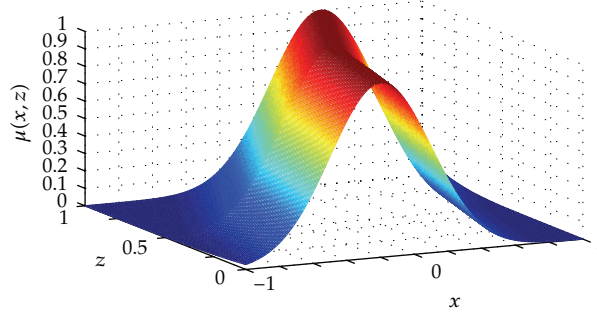
Figure 1: A three-dimensional fuzzy set.

The states, controls, and outputs of such systems depend on the space position as well as on the time [2]. Traditionally, model-based methods are used to control such systems, where a good mathematical model is definitely required. However, the process model may not be easily obtained in many complex situations, and then, a model-free control method has to be used. This leads to the recent development of the novel three-dimensional fuzzy-logic control (3-D FLC) [5–8], which has the inherent capability to process spatiotemporal dynamic systems. The 3-D FLC uses one kind of three-dimensional (3-D) fuzzy set (shown in Figure 1), which is composed of the traditional fuzzy set and a third dimension for the spatial information, and executes a 3-D rule inference engine. It is actually a kind of spatiotemporal fuzzy-control system with the traditional model-free advantage.

To date, the 3-D FLC design has been focused on an expert-knowledge-based approach [5], that is, the fuzzy-rule design is from human experts' knowledge. In this approach, human knowledge to the control solution must exist, and be structured. Practically, experts may have problems structuring the knowledge [9]. Sometimes, although experts have the structured knowledge, they may sway between extreme cases: offering too much knowledge in the field of expertise, or tending to hide their knowledge [9]. Thus, we often lack expert knowledge for control that is usually hidden in an input-output data set. Under this circumstance, a data-driven design becomes a good choice for the 3-D FLC, that is, extraction of fuzzy rules from a spatiotemporal input-output data set. Since the research on the 3-D FLC is just at the beginning stage, extracting 3-D fuzzy control rules from a spatiotemporal data set is still a challenging and open problem for spatially distributed systems.

Traditional data-driven FLC design methods have been developed in the past three decades. They are usually composed of three parts: rule generation, structure optimization, and parameter optimization [10]. For instance, grid partitioning of multidimensional space [11] and clustering technique [12] can be used to generate rules automatically; reducing redundancy variable [12], fusing similar clusters [13], and fusing similar fuzzy set [14] can be applied to reduce the rule number and realize the structure optimization; genetic algorithm [15] and gradient decent approach [16] can be adopted for fine tuning of membership function and realize the parameter optimization. For a complete review of data-driven fuzzy system design, one can further refer to [10]. These methods provide useful solutions to a traditional FLC design.

In this study, we aim at developing a new data-driven 3-D FLC design method based on clustering and SVM-regression learning. The initial 3-D rule base is first generated by a nearest-neighborhood-clustering method from a spatiotemporal data set via defining Frobenius norm as a distance. Then, the initial 3-D rule base is simplified based on similarity
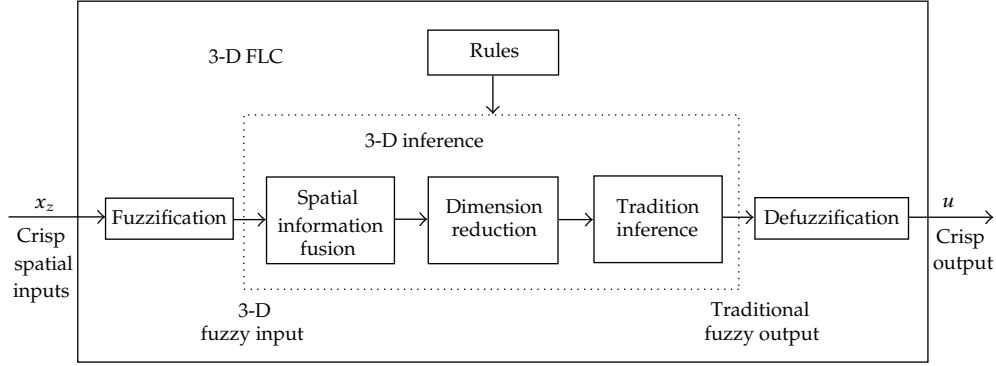
Figure 2: Basic structure of a 3-D FLC.

measure technique defined for 3-D fuzzy sets and 3-D fuzzy rules. Subsequently, an SVM-regression learning algorithm is used to learn the parameters of the rule consequent parts. In addition, the universal approximation capability of the proposed 3-D fuzzy system is discussed.

The paper is organized as follows. Preliminaries about 3-D FLC and SVM regression are addressed in Section 2. In Section 3, a clustering and SVM-regression learning-based 3-D fuzzy control design methodology is presented in detail. In Section 4, the universal approximation capability of the proposed 3-D fuzzy system is presented. In Section 5, a catalytic packed-bed reactor is presented as an example to illustrate the proposed design scheme of a 3-D FLC and validate its effectiveness. Finally, conclusions are given in Section 6.

2. Preliminaries

2.1. 3-D FLC

The 3-D FLC is designed to have the inherent capability to deal with spatial information and its basic structure is shown in Figure 2. It has a similar functional structure similar to the traditional FLC, which consists of three basic blocks: fuzzification, rule inference, and defuzzification. However, it will differ in the detailed operations because of the spatial processing requirement. Generally, the 3-D FLC will be involved with the following basic designs: 3-D membership function (MF), 3-D fuzzification, 3-D rule base, 3-D rule inference, and defuzzification. One can refer to [5] for detailed description. Once each component of a 3-D FLC is set, a precise mathematical formula of the 3-D FLC can be derived.

Assumed that we have 3-D fuzzy rules represented by the following expression:

$$\overline{R}^l : \text{IF } x_1(z) \text{ is } \overline{C}_1^l \text{ and} \cdots \text{and } x_s(z) \text{ is } \overline{C}_s^l, \quad \text{Then } u \text{ is } B^l, \tag{2.1}$$

where $x_i(z) = (x_i(z_1)x_i(z_2)\cdots x_i(z_p))^T$ denotes the ith spatial input variable $(1 \leq i \leq s)$, $x_i(z_j)$ is the input of $x_i(z)$ from the sensing location $z = z_j$ $(1 \leq j \leq p)$; z denotes one-dimensional space in a discrete space domain $Z = \{z_1, z_2, \ldots, z_p\}$; $\overline{C}_i^l$ denotes a 3-D fuzzy set, $l = 1, \ldots, N$; u denotes output variable (the control action); B^l denotes a traditional fuzzy set.

If Gaussian type 3-D membership functions (MF) are used to describe 3-D fuzzy sets in (2.1), then we have

$$\mu_{Gi}^l(x_i(z)) = \exp\left(-\left(\frac{x_i(z) - c_i^l(z)}{\sigma_i^l(z)}\right)^2\right), \tag{2.2}$$

where μ_{Gi}^l denotes the Gaussian type 3-D MF of the ith spatial input $x_i(z)$ in the lth rule; $c_i^l(z) = (c_i^l(z_1), \ldots, c_i^l(z_p))^T$ and $\sigma_i^l(z) = (\sigma_i^l(z_1), \ldots, \sigma_i^l(z_p))^T$ are the center and width of μ_{Gi}^l, respectively; $c_i^l(z_j)$ and $\sigma_i^l(z_j)$ denote center and width of the Gaussian type 2D MF of the ith spatial input $x_i(z)$ at the sensing location $z = z_j$. The Gaussian type 3-D MF μ_{Gi}^l can be regarded as an assembly of multiple Gaussian type 2-D MFs over the space domain Z. Then, the Gaussian type 2-D MF of the ith spatial input $x_i(z)$ at the sensing location $z = z_j$ is given as

$$\mu_{Gij}(x_i(z_j)) = \exp\left(-\left(\frac{x_i(z_j) - c_{ij}^l}{\sigma_{ij}^l}\right)^2\right), \tag{2.3}$$

where $c_{ij}^l = c_i^l(z_j)$ and $\sigma_{ij}^l = \sigma_i^l(z_j)$.

Furthermore, if we employ singleton fuzzification, "product" t-norm and "weighted aggregation" dimension reduction [6] in the 3-D rule inference, singleton fuzzy sets for the output variable, and "center of sets" defuzzification [17], the 3-D FLC can be mathematically expressed as

$$
\begin{aligned}
u(x_z) &= \frac{\sum_{l=1}^N \zeta^l \sum_{j=1}^p a_j \prod_{i=1}^s \mu_{Gij}(x_i(z_j))}{\sum_{l=1}^N \sum_{j=1}^p a_j \prod_{i=1}^s \mu_{Gij}(x_i(z_j))} \\
&= \frac{\sum_{l=1}^N \zeta^l \sum_{j=1}^p a_j \prod_{i=1}^s \exp\left(-\left((x_i(z_j) - c_{ij}^l)/\sigma_{ij}^l\right)^2\right)}{\sum_{l=1}^N \sum_{j=1}^p a_j \prod_{i=1}^s \exp\left(-\left((x_i(z_j) - c_{ij}^l)/\sigma_{ij}^l\right)^2\right)},
\end{aligned}
\tag{2.4}
$$

where $x_z = (x_1(z)x_2(z)\cdots x_s(z)) \in \Omega \subset R^{p \times s}$ is a spatial input vector with Ω as the input domain, p as the number of sensors, and s as the number of spatial inputs; N is the number of rules; a_j is the spatial weight from the jth spatial point [6]; $\zeta^l \in U$ is the nonzero value in the singleton fuzzy set of the output variable for the lth rule.

In (2.4), let

$$\phi^l(x_z) = \frac{\sum_{j=1}^p a_j \prod_{i=1}^s \mu_{Gij}(x_i(z_j))}{\sum_{l=1}^N \sum_{j=1}^p a_j \prod_{i=1}^s \mu_{Gij}(x_i(z_j))}, \tag{2.5}$$

then (2.4) can be rewritten as

$$u(x_z) = \sum_{l=1}^{N} \zeta^l \phi^l(x_z).$$ (2.6)

Similar to a traditional FLC [16], we define $\phi^l(x_z)$ as a spatial fuzzy basis function (SFBF). Each SFBF corresponds to a 3-D fuzzy rule, and all the SFBFs correspond to a 3-D rule base. Mathematically, a 3-D FLC is a linear combination of all the SFBFs.

Equation (2.6) shows that the 3-D FLC is a nonlinear mapping from the input space $x_z \in \Omega \subset R^{p \times s}$ to the output space $u(x_z) \in U \subset R$. It provides us a way to understand and analyze the 3-D FLC from the point of view of function approximation. In Section 4, we will prove that the 3-D FLC has a universal approximation property based on the nonlinear mapping in (2.6).

2.2. Linear SVM Regression

An SVM is a learning algorithm that originated from theoretical foundations of the statistical learning theory [18] and has been widely used in many practical applications, such as bioinformatics, machine vision, text categorization, handwritten character recognition, time series analysis, and so on. The distinct advantage of the SVM over other machine learning algorithms is that it has a good generalization ability and can simultaneously minimize the empirical risk and the expected risk [19]. The SVM algorithms can be categorized into two categories: SVM classification and SVM regression. In this study, we are concerned with the SVM regression with ε-insensitive loss function [20].

Suppose we have a training set $D = \{[x_i, y_i] \in R^s \times R, \ i = 1, \dots, q\}$ consisting of q pairs $(x_1, y_1), (x_2, y_2), \dots, (x_q, y_q)$, where the inputs are s-dimensional vectors, and the labels are continuous values. In ε-SVM regression, the goal is to find a function $f(x, w)$ so that for all training patterns x has a maximum deviation ε from the target values y_i and has a maximum margin. The ε-insensitive loss function is defined as follows:

$$\left|y - f(x, w)\right|_\varepsilon = \begin{cases} 0, & \text{if } \left|y - f(x, w)\right| \leq \varepsilon, \\ \left|y - f(x, w)\right| - \varepsilon, & \text{otherwise.} \end{cases}$$ (2.7)

The ε-insensitive loss function defines an ε tube [9].

The regression problem can be formulated as a convex optimization problem as follows:

$$\min_{w, b, \xi_i, \xi_i^*} = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{l} \xi_i + \sum_{i=1}^{l} \xi_i^*\right)$$ (2.8)

subject to

$$
\begin{aligned}
y_i - \langle w \cdot x_i \rangle - b &\leq \varepsilon + \xi_i \\
\langle w \cdot x_i \rangle + b - y_i &\leq \varepsilon + \xi_i^* \\
\xi_i \geq 0, \quad \xi_i^* \geq 0, &\quad i = 1, \ldots, q,
\end{aligned}
\tag{2.9}
$$

where ξ_i and ξ_i^* are slack variables, and the constant C is a design parameter chosen by the user, which determines the trade off between the complexity of $f(x, w)$ and the approximate error.

The above optimization problem can be solved in a dual space. By introducing the Lagrange multipliers, the primal optimization problem can be formulated in its dual form as follows:

$$
\max_{\alpha_i, \alpha_i^*} \left\{ -\frac{1}{2} \sum_{i=1}^{q} \sum_{j=1}^{q} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \langle x_i \cdot x_j \rangle - \varepsilon \sum_{i=1}^{q} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{q} (\alpha_i^* - \alpha_i) y_i \right\}
\tag{2.10}
$$

subject to

$$
\sum_{j=1}^{q} \alpha_i^* = \sum_{i=1}^{q} \alpha_i,
\tag{2.11}
$$

$$
0 \leq \alpha_i^* \leq C, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, q.
$$

Solving the dual quadratic programming problem, we can find an optimal weight vector w and an optimal bias b of the regression hypersurface given as follows:

$$
w = \sum_{i=1}^{q} (\alpha_i^* - \alpha_i) x_i,
$$

$$
b = \frac{1}{q} \left(\sum_{i=1}^{q} (y_i - \langle w \cdot x_i \rangle) \right).
\tag{2.12}
$$

Then, the best regression hypersurface is given by

$$
f(x, w) = \sum_{i=1}^{q} (\alpha_i^* - \alpha_i) \langle x \cdot x_i \rangle + b = \sum_{i \in \text{SV}} (\alpha_i^* - \alpha_i) \langle x \cdot x_i \rangle + b
\tag{2.13}
$$

The training pattern x_i with nonzero $(\alpha_i^* - \alpha_i)$ is called support vector (SV).

3. Clustering and SVM-Regression Learning-Based 3-D FLC Design

Clustering and SVM-regression learning-based 3-D FLC design is a novel design of a 3-D FLC by integrating a nearest-neighborhood-clustering and an SVM-regression. The design
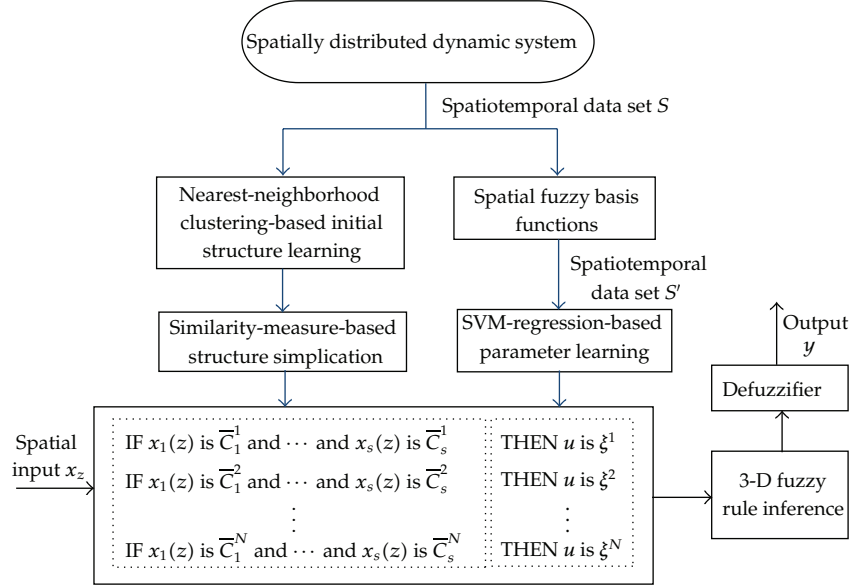
Figure 3: Conceptual configuration of a clustering and SVM-regression learning-based 3-D FLC.

methodology can be depicted by Figure 3. Firstly, a nearest-neighborhood-clustering method with Frobenius norm defined as a distance is employed to mine the underlying knowledge of the spatiotemporal data set S and yield the initial structure, that is, antecedent part of 3-D fuzzy rules. Because the obtained input space partition may have redundancy in terms of highly overlapping MFs, it is necessary to optimize the obtained initial fuzzy partition. Then, a similarity measure technique is utilized to merge similar 3-D fuzzy sets and to merge similar 3-D fuzzy rules, and then to simplify the initial rule structure. Finally, a linear SVM-regression algorithm is used to learn the parameters of the consequent parts based on an equivalence relationship between a linear SVM regression and a 3-D FLC.

The spatiotemporal data set S from a spatially distributed system is composed of n spatiotemporal input-output data pairs given as follows:

$$S = \left\{ \left(x_z^k, u^k \right) \mid x_z^k \in R^{p \times s}, \ u^k \in R, \ k = 1, \ldots, n \right\}, \tag{3.1}$$

where $x_z^k = (x_1^k(z), \ldots, x_s^k(z))$ denotes the value of s spatial input variables at the kth sampling time, $x_i^k(z) = (x_i^k(z_1), \ldots, x_i^k(z_p))^T$ denotes the value of ith spatial input variable at the kth sampling time $(i = 1, \ldots, s)$, u^k denotes the output value at the kth sampling time, n denotes the number of sampling time, and p denotes the number of sensors. Since infinite sensors are used, x_z^k is a matrix with p rows and s columns.

3.1. Initial Structure Learning

3.1.1. Nearest Neighborhood Clustering Method

Clustering method is one of the data-driven learning tools for unlabeled data. It can mine underlying knowledge (or data structure) from a dataset that is difficult for humans to manually identify. One of the simplest clustering algorithms is the nearest-neighborhood clustering algorithm [16]. However, the existing nearest-neighborhood clustering algorithm has not the capability to deal with spatiotemporal data. In this study, we expand its capability to deal with spatiotemporal data set S, which is of matrix form. The key point is that the Frobenius norm given in (3.2) is used for defining a distance in a nearest neighborhood clustering algorithm.

$$\|X\|_F = \sqrt{\text{tr}(X^T X)} \quad (X \in R^{p \times s}). \tag{3.2}$$

The nearest neighborhood clustering algorithm is summarized as follows.

(i) Step 1: Begin from the first spatiotemporal data x_z^1. Let the first cluster center c_z^1 be x_z^1, the number of data pairs m_1 be 1, and the threshold be ρ_0 for generating new fuzzy rules.

(ii) Step 2: Suppose that the kth spatiotemporal data x_z^k $(k = 2, \ldots, n)$ is considered, when N clusters have been generated and their centers are $c_z^1, c_z^2, \ldots,$ and c_z^N respectively. Firstly, compute the distance between x_z^k and each center of N clusters using $\|x_z^k - c_z^l\|_F$ $(l = 1, \ldots, N)$. Then, compute the threshold ρ using

$$\rho = \max_{l=1,\ldots,N} \left(\frac{1}{1 + \left\|x_z^k - c_z^l\right\|_F} \right) \tag{3.3}$$

Hence, the corresponding cluster center $c_z^{l_k}$ is taken as the nearest neighborhood cluster of x_z^k.

(iii) Step 3: (a) If $\rho < \rho_0$, then x_z^k is taken as a new cluster center, and let $N = N+1$, $m_N = 1$, and $c_z^N = x_z^k$. (b) If $\rho \geq \rho_0$, x_z^k belongs to the cluster with the center $c_z^{l_k}$. The center of l_kth cluster is tuned by introducing a learning rate $\eta = \eta_0/(m_{l_k} + 1)$ $(\eta_0 \in [0, 1])$ as follows:

$$c_z^{l_k} = c_z^{l_k} + \eta \left(x_z^k - c_z^{l_k} \right), \tag{3.4}$$

and let $m_{l_k} = m_{l_k} + 1$.

(iv) Step 4: Let $k = k + 1$. If $k \geq n + 1$, then quit. Otherwise, back to Step 2.

3.1.2. Rule Extraction and 3-D MF Construction

After clustering learning, we obtain an input space partition with N cluster centers $c_z^1, c_z^2, \ldots, c_z^N$. Then, we will produce antecedent part of rule base and construct 3-D MFs in

terms of the partition. Each cluster corresponds to a 3-D fuzzy rule. Assumed that we employ Gaussian type 3-D MF. Then, the cluster center corresponds to the center of Gaussian type 3-D MFs in the antecedent part. Thus, the number of fuzzy rules is equal to the number of clusters N. In addition, we determine the width of the Gaussian MFs in terms of the domain of variables. For instance, the width of the Gaussian type 3-D MFs from the same sensing location are defined as

$$\sigma(z_j) = \max_{1 \le i \le s} \left(\frac{x_i^{\max}(z_j) - x_i^{\min}(z_j)}{10} \right), \tag{3.5}$$

where $x_i^{\max}(z_j)$ and $x_i^{\min}(z_j)$ are the maximum and the minimum bound values of the ith spatial input variable, respectively.

3.2. Structure Simplication

After the initial structure learning, the obtained fuzzy partition of the input space and fuzzy rules may have redundancy in terms of highly overlapping MFs. In this step, we will simplify the fuzzy partition and fuzzy rules. The crucial technique for simpification is similarity measure. The previous similarity measure techniques [14, 21, 22] developed for traditional fuzzy sets and traditional fuzzy rules are not suitable to 3-D fuzzy sets and 3-D fuzzy rules. In this study, we will define a new similarity measure technique.

3.2.1. Similarity Measure

Firstly, we define the similarity of two 3-D fuzzy sets $\overline{A}$ and $\overline{B}$ as below.

$$\overline{S}\left(\overline{A}, \overline{B}\right) = \frac{1}{1 + d\left(\overline{A}, \overline{B}\right)}, \quad \overline{S}(\cdot) \in (0, 1], \tag{3.6}$$

where $d(\overline{A}, \overline{B})$ is a distance between $\overline{A}$ and $\overline{B}$. Since Gaussian type 3-D MFs are chosen, the following simple expression can be used to approximate the distance:

$$d\left(\overline{A}, \overline{B}\right) = \left\| \begin{bmatrix} c_{\overline{A}(z_1)} & \sigma_{\overline{A}(z_1)} \\ \vdots & \vdots \\ c_{\overline{A}(z_p)} & \sigma_{\overline{A}(z_p)} \end{bmatrix} - \begin{bmatrix} c_{\overline{B}(z_1)} & \sigma_{\overline{B}(z_1)} \\ \vdots & \vdots \\ c_{\overline{B}(z_p)} & \sigma_{\overline{B}(z_p)} \end{bmatrix} \right\|_F, \tag{3.7}$$

where $c_{\overline{A}(z_j)}(c_{\overline{B}(z_j)})$ and $\sigma_{\overline{A}(z_j)}(\sigma_{\overline{B}(z_j)})$ are center and width of the Gaussian type 3-D MF $\overline{A}(\overline{B})$ at sensing location $z = z_j$ $(j = 1, \ldots, p)$, respectively.

Based on the similarity measure, we can merge similar 3-D fuzzy sets, or merge similar 3-D fuzzy rules.

(i) Merge of Two Similar 3-D Fuzzy Sets $\overline{A}$ and $\overline{B}$

Firstly, the similarity between $\overline{A}$ and $\overline{B}$ is computed according to (3.6). If $\overline{S}(\overline{A}, \overline{B})$ is higher than a threshold, we can conclude that $\overline{A}$ and $\overline{B}$ are similar, and then merge them into a new 3-D fuzzy set $\overline{C}$. The center and width of $\overline{C}$ are viewed as the average values of $\overline{A}$ and $\overline{B}$, and are given as the following:

$$
\begin{aligned}
c_{\overline{C}(z_j)} &= \frac{c_{\overline{A}(z_j)} + c_{\overline{B}(z_j)}}{2}, \\
\sigma_{\overline{C}(z_j)} &= \frac{\sigma_{\overline{A}(z_j)} + \sigma_{\overline{B}(z_j)}}{2}.
\end{aligned}
\tag{3.8}
$$

(ii) Merge of Two Similar 3-D Fuzzy Rules $\overline{R}^{l_1}$ and $\overline{R}^{l_2}$

The similarity $\overline{R}^{l_1}$ and $\overline{R}^{l_2}$ is inferred by measuring their similarity in the antecedent part. For instance, the similarity computation between $\overline{R}^{l_1}$ and $\overline{R}^{l_2}$ is given by

$$
\overline{S}_{\text{rule}}\left(\overline{R}^{l_1}, \overline{R}^{l_2}\right) = \min_{1 \le i \le s}\left\{\overline{S}\left(\overline{C}_i^{l_1}, \overline{C}_i^{l_2}\right)\right\},
\tag{3.9}
$$

where $\overline{R}^{l_1}$ and $\overline{R}^{l_2}$ have the same rule form as in (2.1), $\overline{C}_i^{l_1}$ ($\overline{C}_i^{l_2}$) denotes the 3-D fuzzy set for the ith spatial input variable $x_i(z)$ in the l_1th (l_2th) rule. If $\overline{S}_{\text{rule}}(\overline{R}^{l_1}, \overline{R}^{l_2})$ is higher than a threshold, we can conclude that $\overline{R}^{l_1}$ and $\overline{R}^{l_2}$ are similar, and then merge them into a new 3-D fuzzy rule $\overline{R}^{l_1 l_2}$. The merging of two 3-D fuzzy rules is realized by merging the two fuzzy sets of each spatial input variable in the two 3-D fuzzy rules, respectively.

3.2.2. Similarity Measure-Based Structure Simplification

Based on the similarity measure, the simplification task includes removing 3-D fuzzy sets similar to the universal set, merging similar 3-D fuzzy sets, and merging similar rules. The detailed procedure of structure simplification is summarized as follows.

(i) Step 1: Given a 3-D fuzzy rule base $\overline{\mathfrak{R}} = \{\overline{R}^l\}_{l=1}^{K}$. Firstly, set proper thresholds: $\lambda_u \in (0, 1]$ for removing 3-D fuzzy sets that are similar to the universal set, $\lambda_{\text{set}} \in (0, 1]$ for merging similar 3-D fuzzy sets, and $\lambda_{\text{rule}} \in (0, 1]$ for merging 3-D fuzzy rules with similar antecedents.

(ii) Step 2: Calculate $s_{jki} = \overline{S}(\overline{C}_i^j, \overline{C}_i^k)$ with $j \ne k, j = 1, \ldots, K, k = 1, \ldots, K$, and $i = 1, \ldots, s$. Let $s_{rmq} = \max_{j \ne k}\{s_{jki}\}$ and select $\overline{C}_q^r$ and $\overline{C}_q^m$.

(iii) Step 3: If $s_{rmq} \ge \lambda_{\text{set}}$, merge $\overline{C}_q^r$ and $\overline{C}_q^m$ into a new 3-D fuzzy set $\overline{C}_q^{rm}$, set $\overline{C}_q^r = \overline{C}_q^{rm}$ and $\overline{C}_q^m = \overline{C}_q^{rm}$, and back to step 2. If no more two 3-D fuzzy sets have the similarity with $s_{rmq} \ge \lambda_{\text{set}}$ ($j \ne k$), then go to step 4.

(iv) Step 4: Remove the 3-D fuzzy set similar to the universal set and the rule with membership function that is always near zero over the space domain.

(v) Step 5: Calculate the similarity of two rules $s_{l_1 l_2} = \overline{S}_{\text{rule}}(\overline{R}^{l_1}, \overline{R}^{l_2})$ with $l_1 \neq l_2$, $l_1 = 1, \ldots, N, l_2 = 1, \ldots, N$. Let $s_{rm} = \max_{l_1 \neq l_2} \{s_{l_1 l_2}\}$.

(vi) Step 6: If $s_{rm} \geq \lambda_{\text{rule}}$, merge the rth and the mth rules into a new rule R_{new} and substitute them. Let $N = N - 1$, and back to step 5. If no more rules have similarity with $s_{rm} \geq \lambda_{\text{rule}}$ $(r \neq m)$, then quit.

Generally speaking, the threshold λ_u is higher than the threshold λ_{set}, while the choice of a suitable threshold λ_{rule} depends on the application. The lower λ_{set} is set, the less fuzzy sets and less fuzzy rules are yielded in the resulting rule base. In this study, we set $\lambda_u = 0.95$, $\lambda_{\text{set}} = 0.75$, and $\lambda_{\text{rule}} = 1$.

3.3. Parameter Learning

After the structure simplification, we obtain a rule base with optimized antecedent parts. For a complete rule base, the rest task is to determine the consequent part parameters. In this study, we employ an SVM regression algorithm to learn the consequent part parameter ξ^l $(l = 1, \ldots, N)$ in the 3-D FLC.

Firstly, the original input samples are transformed into new samples. Utilizing the spatial fuzzy basis functions $\phi^l(x_z^k)$ $(l = 1, \ldots, N)$ in (2.5), we can transform each spatial input sample x_z^k $(k = 1, \ldots, n)$ in S into a new input sample $\phi(x_z^k) = (\phi^1(x_z^k), \phi^2(x_z^k), \ldots, \phi^N(x_z^k))$. Then, the original data set S in (3.1) can be transformed into a new data set S' as follows:

$$S' = \left\{ \left(\phi\left(x_z^k\right), u^k\right) \mid \phi\left(x_z^k\right) \in R^N, \ u^k \in R, \ k = 1, \ldots, n\right\}. \tag{3.10}$$

Secondly, an equivalence relationship of an SVM regression and a 3-D FLC can be derived based on the new data set S'. From (2.13), the final decision function $f(\phi(x_z^k))$ of an SVM can be described with the following form:

$$f\left(\phi\left(x_z^k\right)\right) = \sum_{k=1}^{n} (\alpha_k^* - \alpha_k) \left\langle \phi\left(x_z^k\right), \phi(x_z) \right\rangle + b, \tag{3.11}$$

where α_k^* and α_k are associated learning parameters in a SVM, The training pattern $\phi(x_z^k)$ with nonzero $(\alpha_k^* - \alpha_k)$ is called support vector (SV). Furthermore, (3.11) can further be expressed by

$$
\begin{aligned}
f\left(\phi\left(x_z^k\right)\right) &= \sum_{k=1}^{n} (\alpha_k^* - \alpha_k) \sum_{l=1}^{N} \phi^l\left(x_z^k\right) \phi^l(x_z) + b \\
&= \sum_{l=1}^{N} \left(\sum_{k=1}^{n} (\alpha_k^* - \alpha_k) \phi^l\left(x_z^k\right) \right) \phi^l(x_z) + b \\
&= \sum_{l=1}^{N} \xi^l \phi^l(x_z) + b \\
&= u(x_z)
\end{aligned}
\tag{3.12}
$$

In (3.12), the bias term b in a 3-D FLC can be realized by adding a fuzzy rule as follows:

$$\overline{R}^0 : \text{IF } x_1(z) \text{ is } \overline{C}_1^0 \text{ and} \cdots \text{and } x_s(z) \text{ is } \overline{C}_s^0, \quad \text{THEN } u \text{ is } b, \tag{3.13}$$

where $\overline{C}_i^0$ is a universal 3-D fuzzy set, whose fuzzy degree is 1 over the space domain for any spatial input $x_i(z)$, $i = 1, \ldots, s$. From (3.12), we can see that an SVM will be equivalent to a 3-D FLC if (3.14) holds.

$$\xi^l = \sum_{k=1}^{n} (\alpha_k^* - \alpha_k) \phi^l \left(x_z^k \right). \tag{3.14}$$

Finally, a linear SVM regression is employed to learn the consequent part parameters. Using (3.14), the parameters ξ^l $(l = 1, \ldots, N)$ in consequent parts are obtained in terms of the SVM learning, that is,

$$\xi^l = \sum_{k \in \text{SV}} (\alpha_k^* - \alpha_k) \phi^l \left(x_z^k \right). \tag{3.15}$$

4. Universal Approximation of Clustering and SVM-Regression Learning-Based 3-D FLC

In essence, the clustering and SVM-regression learning-based 3-D FLC design is a fuzzy modeling that extracts fuzzy control rules and constructs a 3-D FLC from spatiotemporal data hidden with effective control laws. In other words, the proposed 3-D FLC aims at approximating an unknown nonlinear control function. Thus, in this subsection, we are concerned with its universal approximation capability. The universal approximation capability of the SVM learning-based 3-D FLC can be described by the following theorem.

Theorem 4.1. *Suppose that the input universe of discourse Ω is a compact set in $R^{p \times s}$. Then, for any given real continuous function $g(x_z)$ on Ω and arbitrary $\varepsilon > 0$, there exists a 3-D FLC $u(x_z)$ as described in (2.4) satisfying the following inequality:*

$$\sup_{x_z \in \Omega} \left(|u(x_z) - g(x_z)| \right) < \varepsilon. \tag{4.1}$$

The proof of the theorem is given in the appendix by using Stone-Weierstrass theorem [23]. Theorem 4.1 indicates that the clustering and SVM-regression learning-based 3-D FLC is a universal approximator, that is, it can approximate continuous control functions to arbitrary accuracy.

5. Application

5.1. A Catalytic Packed-Bed Reactor

We take a catalytic packed-bed reactor [1, 5] as an example. The reactor is long and thin as shown in Figure 4. It is fed with gaseous reactant C from the right side, and the zero-order
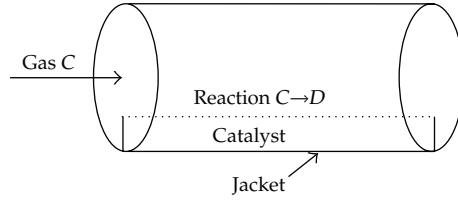
Figure 4: Sketch of a catalytic packed-bed reactor.

gas phase reaction $C \rightarrow D$ is carried out on the catalyst. The reaction is endothermic, and a jacket is used to heat the reactor. A dimensionless model that describes this nonlinear tubular chemical reactor is provided as follows:

$$\varepsilon_p \frac{\partial T_g}{\partial t} = -\frac{\partial T_g}{\partial z} + \alpha_c (T_s - T_g) - \alpha_g (T_g - u),$$

$$\frac{\partial T_s}{\partial t} = \frac{\partial^2 T_s}{\partial z^2} + B_0 \exp\left(\frac{\gamma T_s}{1 + T_s}\right) - \beta_c (T_s - T_g) - \beta_p (T_s - b(z)u) \tag{5.1}$$

subject to the boundary conditions

$$z = 0, \quad T_g = 0, \quad \frac{\partial T_s}{\partial z} = 0; \quad z = 1, \quad \frac{\partial T_s}{\partial z} = 0, \tag{5.2}$$

where T_g, T_s, and u denote the dimensionless temperature of the gas, the catalyst, and jacket, respectively. The values of the process parameters are given as follows:

$$\varepsilon_p = 0.01, \quad \gamma = 21.14, \quad \beta_c = 1.0, \quad \beta_p = 15.62, \tag{5.3}$$

$$B_0 = -0.003, \quad \alpha_c = 0.5, \quad \alpha_g = 0.5. \tag{5.4}$$

The concerned control problem is to control the catalyst temperature $T_s(z,t)$ throughout the reactor to track a spatial reference profile $(T_{sd}(z) = 0.42 - 0.2\cos(\pi z))$ in order to maintain a desired degree of reaction rate using the measurements of catalyst temperature from five sensing locations $z' = [0 \ 0.25 \ 0.5 \ 0.75 \ 1]$ and manipulating one spatially distributed heating source $(b(z) = 1 - \cos(\pi z))$. The mathematical model (5.1)-(5.2) is only for the process simulation for evaluation of the control scheme. The method of lines [24] is used to simulate the model.

In this application, we aim at extracting 3-D fuzzy rules from a spatiotemporal data set using clustering and SVM regression learning algorithm and constructing a complete 3-D FLC without any prior knowledge.
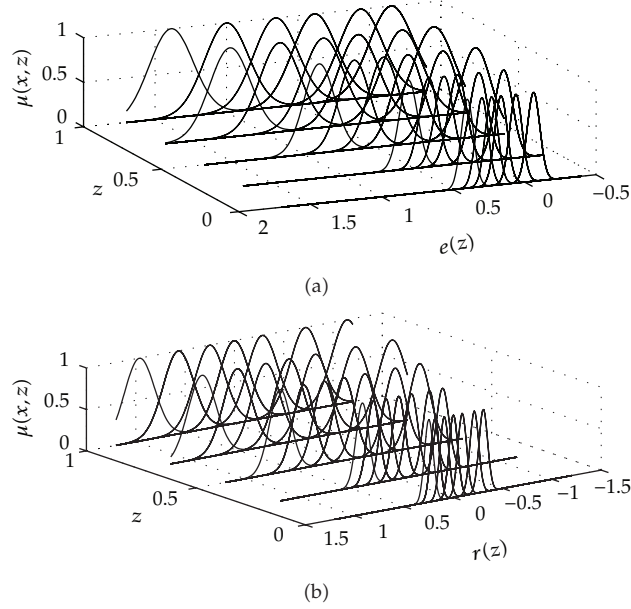
(a)

(b)

Figure 5: Distributions of Gaussian type 3-D fuzzy sets in the simplified fuzzy-rule base.

5.2. Design of a Clustering and SVM-Regression Learning-Based 3-D FLC

5.2.1. Spatiotemporal Data Collection

The spatiotemporal input-output data set is collected from the catalytic packed-bed reactor controlled by expert-knowledge-based 3-D FLC [5], where pseudorandom quinary signal (PRQS) [25] with maximum length of 124 as perturbed signal is added to the control input. Each spatiotemporal input-output data pair consists of a spatial error input $e^*(z) = [e_1^*, \ldots, e_5^*]^T$, a spatial error in change input $r^*(z) = [r_1^*, \ldots, r_5^*]^T$, and an incremental output Δu^*, where $e_i^* = T_s(z_i, q) - T_{sd}(z_i)$ $r_i^* = e_i^*(q) - e_i^*(q - 1)$; q and $q - 1$ denotes the qth and q-1th sampling time, respectively. The detailed design of the expert-knowledge-based 3-D FLC, including fuzzification, 3-D rule inference, and defuzzification, can refer to [5]. The scaling factors for the spatial error, the spatial error in change, and the incremental output are set as 2.0, 0.001, and 0.8716, respectively. The parameters of PRQS are chosen with the following settings: the number of the levels is 5, the length of the period is 124, the sampling time is 0.2 s, and the minimum switching time (i.e., clock period) is 0.2 s.

Two groups of data sets are obtained by adding PRQS signal with different scaling factor (i.e., 0.447 and 0.1) to the control input. The first group with 150 data pairs is generated for training by adding PRQS perturbation signal with a scaling factor 0.447, and the other group with 150 data pairs is generated for test by adding PRQS perturbation signal with a scaling factor 0.1. To evaluate the performance, we employ the following root-mean-squared error (RMSE) as the criteria:

$$\text{RMSE} = \sqrt{\sum_{k=1}^{n} \frac{\left(\Delta u_k^* - \Delta u_k\right)^2}{n}}, \tag{5.5}$$

Table 1: Learning results of an SVM regression with different values of C and ε.

C	ε	Number of SV	RMSE training ($\times 10^{-2}$)	RMSE testing ($\times 10^{-2}$)	SSE ($\times 10^{-2}$)	IAE ($\times 10^{-1}$)	ITAE ($\times 10^{-1}$)
	0.00001	149	4.89	3.48	1.70	2.566	8.656
	0.0001	140	4.89	3.48	1.70	2.566	8.655
	0.001	129	4.86	3.45	1.71	2.573	8.699
1	0.01	105	4.75	3.13	1.84	2.699	9.389
	0.1	13	6.35	3.24	1.99	2.901	10.163
	0.2	12	12.80	6.44	3.74	4.653	19.063
	0.3	10	19.29	7.76	4.77	5.715	24.248
	0.00001	149	4.89	3.48	1.70	2.566	8.656
	0.0001	140	4.89	3.48	1.70	2.566	8.655
	0.001	129	4.86	3.45	1.71	2.573	8.699
10	0.01	105	4.75	3.13	1.84	2.699	9.389
	0.1	13	6.35	3.24	1.99	2.901	10.163
	0.2	12	12.80	6.44	3.74	4.653	19.063
	0.3	10	19.29	7.76	4.77	5.715	24.248
	0.00001	149	4.89	3.48	170	2.566	8.656
	0.0001	141	4.89	3.48	1.70	2.566	8.655
	0.001	129	4.86	3.45	1.71	2.573	8.699
100	0.01	105	4.75	3.13	1.84	2.699	9.389
	0.1	13	6.35	3.24	1.99	2.901	10.163
	0.2	12	12.80	6.44	3.74	4.653	19.063
	0.3	10	19.29	7.76	4.77	5.715	24.248
	0.00001	149	4.89	3.48	170	2.566	8.656
	0.0001	141	4.89	3.48	1.70	2.566	8.655
	0.001	129	4.86	3.45	1.71	2.574	8.702
1000	0.01	106	4.75	3.13	1.84	2.700	9.395
	0.1	13	6.35	3.24	1.99	2.901	10.163
	0.2	12	12.80	6.44	3.74	4.653	19.063
	0.3	10	19.29	7.76	4.77	5.715	24.248

where n denotes the number of samples, Δu_k^* denotes actual output, and Δu_k denotes expected output.

5.2.2. Design of a Clustering and SVM-Regression Learning-Based 3-D FLC

The design procedure of the proposed 3-D FLC is given as follows:

(i) Employ the nearest neighborhood clustering algorithm to deal with the spatiotemporal data set for the input space partition with $\rho_0 = 0.7$ and $\eta_0 = 0$, and then generate 16 3-D fuzzy rules with 32 3-D fuzzy sets, where the width of Gaussian type 3-D fuzzy sets is $\sigma_z = [0.0620, 0.0902, 0.1518, 0.2008, 0.2175]^T$ from (3.5).

(ii) Simplify the 3-D fuzzy sets and 3-D fuzzy rules based on similarity measure (as described in Section 3.2) with $\lambda_u = 0.95$, $\lambda_{set} = 0.75$, and $\lambda_{rule} = 1$, and then obtain

Table 2: Performance comparisons.

Performance index	Clustering and SVM-regression learning-based 3-D FLC	Expert-knowledge-based 3-D FLC
Number of rules	15	49
No disturbance		
ISS ($\times 10^{-2}$)	1.70	1.69
IAE ($\times 10^{-1}$)	2.566	2.557
ITAE ($\times 10^{-1}$)	8.655	8.646
With 50% increase disturbance in velocity of gas		
ISS ($\times 10^{-2}$)	1.77	1.78
IAE ($\times 10^{-1}$)	2.675	2.680
ITAE ($\times 10^{-1}$)	9.058	9.062

15 3-D fuzzy rules with 15 3-D fuzzy sets. The distributions of Gaussian type 3-D fuzzy sets are shown in Figure 5.

(iii) SVM algorithm described in Section 3.2 is used to learn the consequent part parameters with $C = \{1, 10, 100, 1000\}$ and $\varepsilon = \{0.00001, 0.0001, 0.001, 0.01, 0.1, 0.2\}$. The RMSE in (5.5) for training and test are listed in Table 1. From Table 1, we can find that: (1) smaller ε yielded more support vectors and led to reasonable training and test performance; while larger ε yielded less support vectors and led to worse training and test performance. (2) C almost had no influence on the training and test performance, once ε was fixed. In this study, we choose $C = 100$ and $\varepsilon = 0.0001$. Finally, a complete 3-D FLC is constructed with 15 3-D fuzzy rules and 15 3-D fuzzy sets as shown in Figure 6. Using the linguistic hedges approach [14, 21], we can interpret these 3-D fuzzy rules using linguistic words. For instance, the first 3 fuzzy rules are interpreted as follows.

   (a) $\overline{R}^1$: IF $e^*(z)$ is less than POSITIVE SMALL and $r^*(z)$ is more than POSITIVE SMALL, THEN Δu^* is sort of POSITIVE MEDIUM.

   (b) $\overline{R}^2$: IF $e^*(z)$ is very ZERO and $r^*(z)$ is very NEGATIVE SMALL, THEN Δu^* is very ZERO.

   (c) $\overline{R}^3$: IF $e^*(z)$ is sort of POSITIVE SMALL and $r^*(z)$ is more than POSITIVE MEDIUM, THEN Δu^* is more than POSITIVE MEDIUM.

5.2.3. Control-Performance Validation

The designed clustering and SVM regression learning-based 3-D FLC is applied to the control of the catalytic packed-bed reactor, where simulation time is 10 s. We select the same quantitative performance criteria as in [5]: steady-state error (SSE), integral of the absolute error (IAE), and integral of time multiplied by absolute error (ITAE). The control performance is given in Table 2, and the control profile is given in Figures 7 and 8, where (a), (b), and (c) represent catalyst temperature evolution profile, manipulated input, and catalyst temperature profiles in steady state, respectively. We can find that the proposed 3-D FLC has
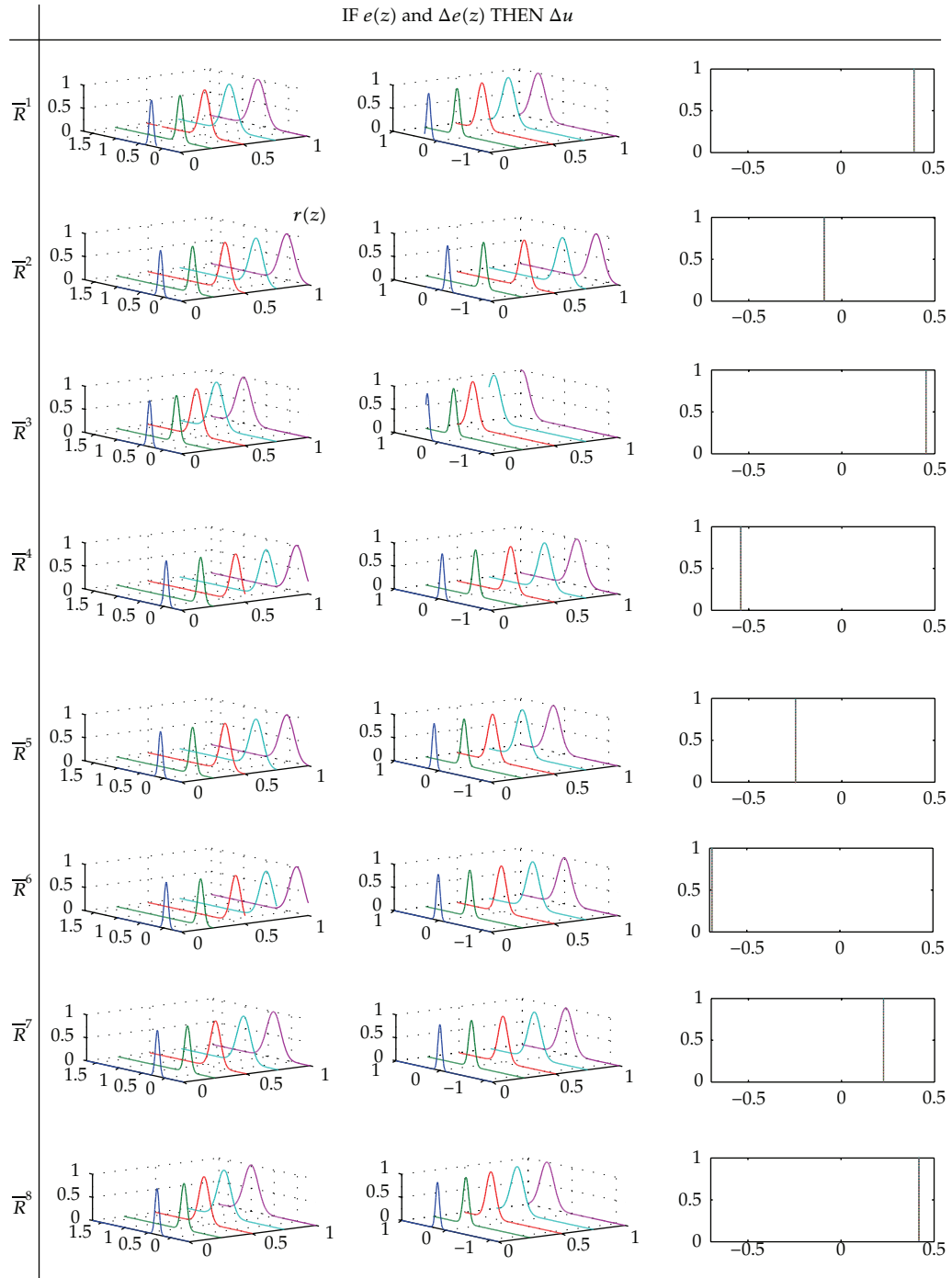
IF $e(z)$ and $\Delta e(z)$ THEN Δu
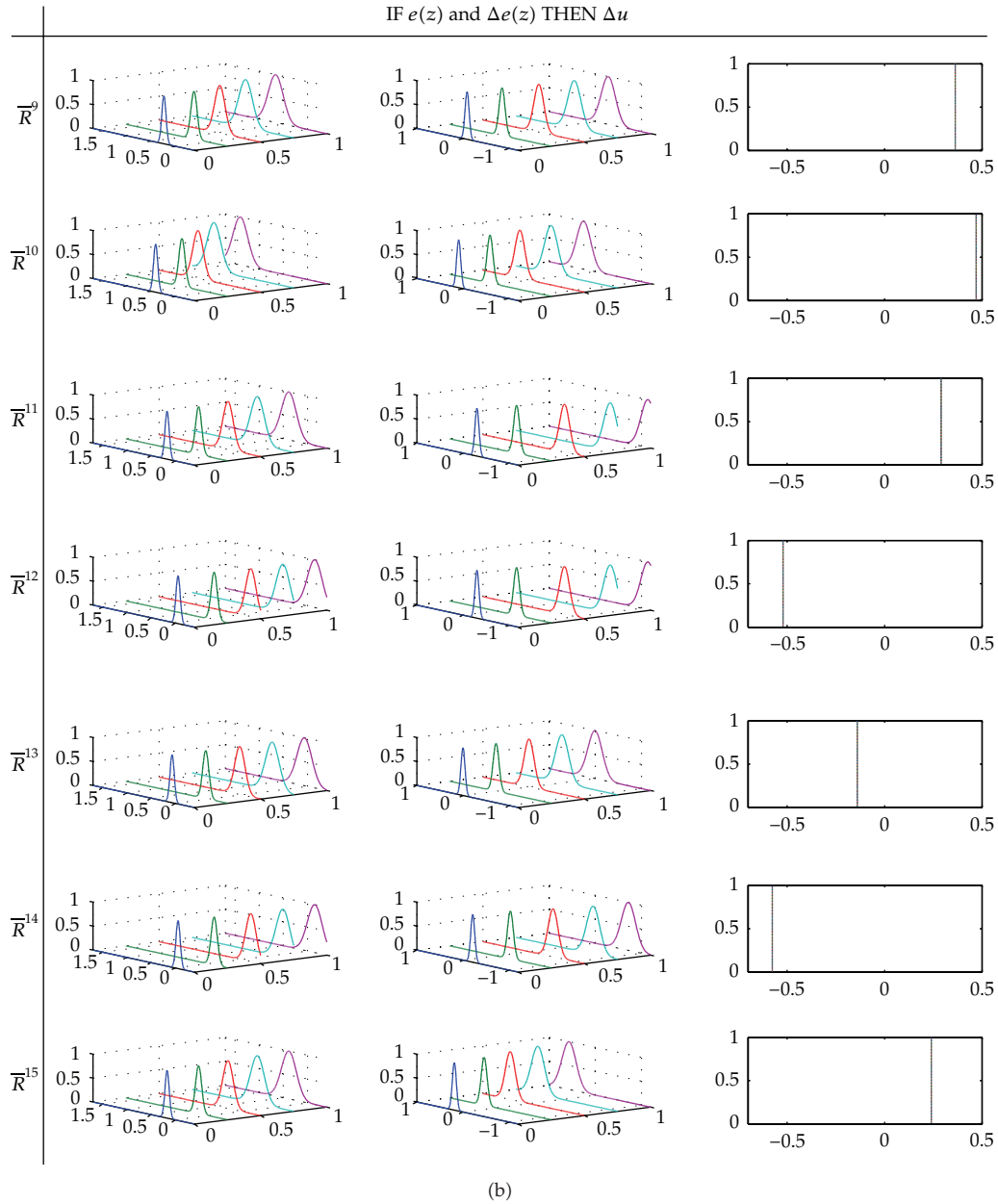

(a)

Figure 6: Continued.

(b)

Figure 6: 3-D fuzzy rules and their associated 3-D fuzzy sets of a clustering and SVM-regression learning-based 3-D FLC.

comparable control performance to the expert-knowledge-based 3-D FLC in [5] both in ideal condition and in disturbed condition.

In addition, we do more control experiments when the SVM-learning algorithm adopts different C and ε. According to the experimental results (see the last three columns in Table 1), we can find that the proposed 3-D FLC shows good control performance when a smaller ε is chosen.

(a)   (b)   (c)
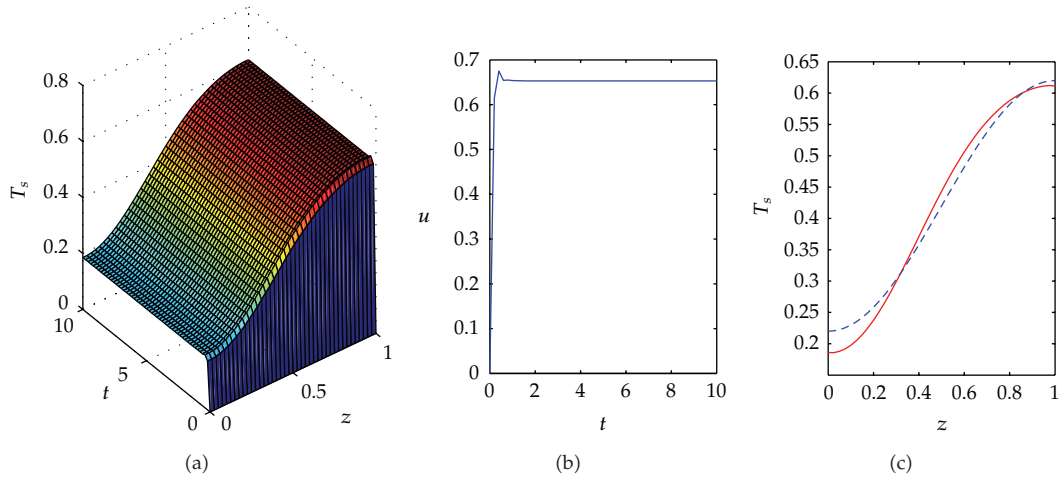
Figure 7: Controlled by a clustering and SVM-regression learning-based 3-D FLC under ideal situation (dotted line: reference profile).

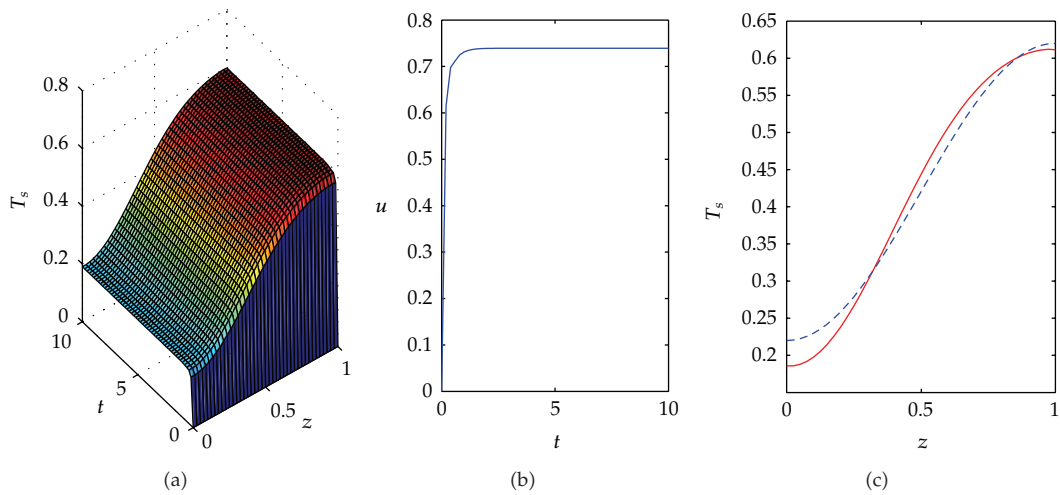(a)   (b)   (c)

Figure 8: Controlled by a clustering and SVM-regression learning-based 3-D FLC under disturbed situation (dotted line: reference profile).

The above simulation results demonstrate that the proposed design method of a clustering and SVM-regression learning-based 3-D FLC is effective. It provides a beneficial complementary design method to 3-D FLCs.

6. Conclusions

In this paper, we have proposed a new 3-D FLC design methodology based on clustering and SVM regression learning from a spatiotemporal data set. The 3-D FLC design is divided into three steps. Firstly, an initial rule structure is extracted by a nearest neighborhood clustering method, which is modified to be suitable for spatio-temporal data. Secondly, the initial structure is simplified via using similarity measure technique, which is defined for 3-D fuzzy

sets and 3-D fuzzy rules. Thirdly, the parameters of the rule consequent parts are learned by a spatial fuzzy basis function-based SVM regression learning algorithm. Besides, the universal approximation capability of the proposed 3-D fuzzy system is discussed. Finally, effectiveness of the proposed 3-D FLC design methodology is validated on a catalytic packed-bed reactor.

Appendix

A. Proof of the Clustering and SVM-Regression Learning-Based 3-D FLC as a Universal Approximator

Let Θ be a set of 3-D FLCs defined in Ω, which is a compact set in $R^{p \times s}$. Then, Preliminary 1 is given as follows.

Preliminary 1

Let $d_\infty(u, g)$ be a semimetric [26] with the following definition

$$d_\infty(u, g) = \sup_{x_z \in \Omega} \left(\left| u(x_z) - g(x_z) \right| \right) \tag{A.1}$$

Therefore, (Θ, d_∞) is a metric space. Since there is at least one fuzzy rule in the rule base of a 3-D FLC, Θ is non-empty. Thus, (Θ, d_∞) is strictly defined.

Subsequently, we will prove that (Θ, d_∞) is dense in $(C[\Omega], d_\infty)$ using Stone-Weierstrass theorem, where $C[\Omega]$ is a set of real continuous functions defined in a compact set Ω. The Stone-Weierstrass theorem is first stated here as follows.

Stone-Weierstrass Theorem (see [16, 23])

Let Z be a set of real continuous functions on a compact set U. If $(1) Z$ is an algebra, that is, the set Z is closed under addition, multiplication, and scalar multiplication; (2) Z separates points on U, that is, for every $x, y \in U, x \neq y$, there exists $f \in Z$ such that $f(x) \neq f(y)$; and (3) Z vanishes at no point of U, that is, for each $x \in U$ there exists $f \in Z$ such that $f(x) \neq 0$; then, the uniform closure of Z consists of all real continuous functions on U, that is, (Z, d_∞) is dense in $(C[U], d_\infty)$.

Proof. (1) Firstly, we prove (Θ, d_∞) is an algebra. Let $u_1, u_2 \in \Theta$, then we can write them as

$$
\begin{aligned}
u_1(x_z) &= \frac{\sum_{l_1=1}^{N_1} \overline{u}_1^{l_1} \sum_{j_1=1}^{p_1} \overleftrightarrow{a}_{j_1} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_1}) - \overleftrightarrow{c}_{ij_1}^{l_1}\right)/\overleftrightarrow{\sigma}_{ij_1}^{l_1}\right)^2\right)}{\sum_{l_1=1}^{N_1} \sum_{j_1=1}^{p_1} \overleftrightarrow{a}_{j_1} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_1}) - \overleftrightarrow{c}_{ij_1}^{l_1}\right)/\overleftrightarrow{\sigma}_{ij_1}^{l_1}\right)^2\right)} + b_1, \\[2em]
u_2(x_z) &= \frac{\sum_{l_2=1}^{N_2} \overline{u}_2^{l_2} \sum_{j_2=1}^{p_2} \underset{\leftrightarrow}{a}_{j_2} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_2}) - \underset{\leftrightarrow}{c}_{ij_2}^{l_2}\right)/\underset{\leftrightarrow}{\sigma}_{ij_2}^{l_2}\right)^2\right)}{\sum_{l_2=1}^{N_2} \sum_{j_2=1}^{p_2} \underset{\leftrightarrow}{a}_{j_2} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_2}) - \underset{\leftrightarrow}{c}_{ij_2}^{l_2}\right)/\underset{\leftrightarrow}{\sigma}_{ij_2}^{l_2}\right)^2\right)} + b_2.
\end{aligned}
\tag{A.2}
$$

Subsequently, we have three derivation procedures. $\square$

(i) Addition

$$u_1(x_z) + u_2(x_z)$$

$$= \frac{\sum_{l_1=1}^{N_1} \sum_{l_2=1}^{N_2} \left(\overline{u}_1^{l_1} + \overline{u}_2^{l_2}\right) \sum_{j_1=1}^{p_1} \sum_{j_2=1}^{p_2} \overleftrightarrow{a}_{j_1} \underset{\leftrightarrow j_2}{a} \prod_{i=1}^{s} \exp \mathscr{Z}}{\sum_{l_1=1}^{N_1} \sum_{l_2=1}^{N_2} \sum_{j_1=1}^{p_1} \sum_{j_2=1}^{p_2} \overleftrightarrow{a}_{j_1} \underset{\leftrightarrow j_2}{a} \prod_{i=1}^{s} \exp \mathscr{Z}} \tag{A.3}$$

$$+ (b_1 + b_2),$$

where $\mathscr{Z}$ denotes $\left(-((x_i(z_{j_1}) - \overleftrightarrow{c}_{ij_1}^{l_1})/\overrightarrow{\sigma}_{ij_1}^{l_1})^2 - ((x_i(z_{j_2}) - \underset{\leftrightarrow ij_2}{c}^{l_2})/\underset{\leftrightarrow ij_2}{\sigma}^{l_2})^2\right)$. Equation (A.3) has the same form as (3.12), then $u_1(x_z) + u_2(x_z) \in \Theta$.

(ii) Multiplication

$$u_1(x_z) u_2(x_z)$$

$$= \frac{\sum_{l_1=1}^{N_1} \sum_{l_2=1}^{N_2} \overline{u}_1^{l_1} \overline{u}_2^{l_2} \sum_{j_1=1}^{p_1} \sum_{j_2=1}^{p_2} \overrightarrow{a}_{j_1} \underset{\leftrightarrow j_2}{a} \prod_{i=1}^{s} \exp \mathscr{Z}}{\sum_{l_1=1}^{N_1} \sum_{l_2=1}^{N_2} \sum_{j_1=1}^{p_1} \sum_{j_2=1}^{p_2} \overrightarrow{a}_{j_1} \underset{\leftrightarrow j_2}{a} \prod_{i=1}^{s} \exp \mathscr{Z}}$$

$$+ b_2 \frac{\sum_{l_1=1}^{N_1} \overline{u}_1^{l_1} \sum_{j_1=1}^{p_1} \overrightarrow{a}_{j_1} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_1}) - \overleftrightarrow{c}_{ij_1}^{l_1}\right)/\overrightarrow{\sigma}_{ij_1}^{l_1}\right)^2\right)}{\sum_{l_1=1}^{N_1} \sum_{j_1=1}^{p_1} \overrightarrow{a}_{j_1} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_1}) - \overleftrightarrow{c}_{ij_1}^{l_1}\right)/\overrightarrow{\sigma}_{ij_1}^{l_1}\right)^2\right)} \tag{A.4}$$

$$+ b_1 \frac{\sum_{l_2=1}^{N_2} \overline{u}_2^{l_2} \sum_{j_2=1}^{p_2} \underset{\leftrightarrow j_2}{a} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_2}) - \underset{\leftrightarrow ij_2}{c}^{l_2}\right)/\underset{\leftrightarrow ij_2}{\sigma}^{l_2}\right)^2\right)}{\sum_{l_2=1}^{N_2} \sum_{j_2=1}^{p_2} \underset{\leftrightarrow j_2}{a} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_2}) - \underset{\leftrightarrow ij_2}{c}^{l_2}\right)/\underset{\leftrightarrow ij_2}{\sigma}^{l_2}\right)^2\right)} + b_1 b_2,$$

where $\mathscr{Z}$ denotes $\left(-((x_i(z_{j_1}) - \overleftrightarrow{c}_{ij_1}^{l_1})/\overrightarrow{\sigma}_{ij_1}^{l_1})^2 - ((x_i(z_{j_2}) - \underset{\leftrightarrow ij_2}{c}^{l_2})/\underset{\leftrightarrow ij_2}{\sigma}^{l_2})^2\right)$. In terms of algebraic operation, the product of functions in Gaussian form is also a function in Gaussian form. Thus, (A.4) has the same form as (3.12), and $u_1(x_z) u_2(x_z) \in \Theta$.

(iii) Scalar Multiplication

For arbitrary $c \in R$, we have

$$cu_1(x_z) = c \frac{\sum_{l_1=1}^{N_1} \overline{u}_1^{l_1} \sum_{j_1=1}^{p_1} \overleftrightarrow{a}_{j_1} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_1}) - \overleftrightarrow{c}_{ij_1}^{l_1}\right)/\overrightarrow{\sigma}_{ij_1}^{l_1}\right)^2\right)}{\sum_{l_1=1}^{N_1} \sum_{j_1=1}^{p_1} \overleftrightarrow{a}_{j_1} \prod_{i=1}^{s} \exp\left(-\left(\left(x_i(z_{j_1}) - \overleftrightarrow{c}_{ij_1}^{l_1}\right)/\overrightarrow{\sigma}_{ij_1}^{l_1}\right)^2\right)} + cb_1. \tag{A.5}$$

Equation (A.5) has the same form as (3.12), and $cu_1(x_z) \in \Theta$.

Finally, by combining (A.3)~(A.5) together, we can conclude that (Θ, d_∞) is an algebra.

(2) Secondly, we will prove that (Θ, d_∞) separates point on Ω by constructing a simple 3-D FLC $u(x_z)$ as in (3.12), namely, $u(x_z^0) \neq u(y_z^0)$ holds for arbitrarily given $x_z^0, y_z^0 \in \Omega$ with $x_z^0 \neq y_z^0$.

We choose two fuzzy rules, that is, $N = 2$.

Let

$$
\begin{aligned}
x_z^0 &= \left(\left(x_1^0(z_1), \ldots, x_1^0(z_p) \right)^T, \ldots, \left(x_s^0(z_1), \ldots, x_s^0(z_p) \right)^T \right) \\
y_z^0 &= \left(\left(y_1^0(z_1), \ldots, y_1^0(z_p) \right)^T, \ldots, \left(y_s^0(z_1), \ldots, y_s^0(z_p) \right)^T \right), \\
a_j &= \frac{1}{p}, \qquad \sigma_{ij}^1 = \sigma_{ij}^2 = 1, \qquad c_{ij}^1 = x_i^0(z_j), \qquad c_{ij}^2 = y_i^0(z_j), \\
& \qquad x_z^1 = x_z^0, \qquad x_z^2 = y_z^0 \quad (j = 1, \ldots, p).
\end{aligned}
\tag{A.6}
$$

We have

$$
\begin{aligned}
u\left(x_z^0\right) &= \frac{\overline{u}^1 + \overline{u}^2 (1/p) \sum_{j=1}^p \prod_{i=1}^s \exp\left(-(x_i^0(z_j) - y_i^0(z_j))^2\right)}{1 + (1/p) \sum_{j=1}^p \prod_{i=1}^s \exp\left(-(x_i^0(z_j) - y_i^0(z_j))^2\right)} + b \\
&= \zeta \overline{u}^1 + (1 - \zeta)\overline{u}^2 + b, \\
u\left(y_z^0\right) &= \frac{\overline{u}^2 + \overline{u}^1 (1/p) \sum_{j=1}^p \prod_{i=1}^s \exp\left(-(x_i^0(z_j) - y_i^0(z_j))^2\right)}{1 + (1/p) \sum_{j=1}^p \prod_{i=1}^s \exp\left(-(x_i^0(z_j) - y_i^0(z_j))^2\right)} + b \\
&= \zeta \overline{u}^2 + (1 - \zeta)\overline{u}^1 + b, \\
\zeta &= \frac{1}{1 + (1/p) \sum_{j=1}^p \prod_{i=1}^s \exp\left(-(x_i^0(z_j) - y_i^0(z_j))^2\right)}.
\end{aligned}
\tag{A.7}
$$

Since $x_z^0 \neq y_z^0$, there must be some i and j such that $x_i^0(z_j) \neq y_i^0(z_j)$. Thus, we have $\prod_{i=1}^s \exp(-(x_i^0(z_j) - y_i^0(z_j))^2) \neq 1$. For arbitrary j, $\prod_{i=1}^s \exp(-(x_i^0(z_j) - y_i^0(z_j))^2) \leq 1$ holds, therefore, we have $\sum_{j=1}^p \prod_{i=1}^s \exp(-(x_i^0(z_j) - y_i^0(z_j))^2) \neq p$. If we choose $\overline{u}^1 = 0$ and $\overline{u}^2 = 1$, then

$$
u\left(x_z^0\right) = 1 - \xi + b \neq \xi + b = u\left(y_z^0\right).
\tag{A.8}
$$

Therefore, (Θ, d_∞) separates point on Ω.

(3) Finally, we prove (Θ, d_∞) vanishes at no point of Ω.

For any 3-D FLC $u(x_z)$ expressed as in (3.12), if we choose $\zeta^l \geq 0$ $(l = 1, \ldots, N)$ and $b > 0$, then for any $x_z \in \Omega$, we have $u(x_z) > 0$.

Therefore, (Θ, d_∞) vanishes at no point of Ω.

By combining the results from (1) to (3) together, Theorem 4.1 is proven.

Acknowledgments

References

[1] W. H. Ray, *Advanced Process Control*, McGraw-Hill, New York, NY, USA, 1981.

[2] P. D. Christofides, *Nonlinear and Robust Control of Partial Differential Equation Systems: Methods and Applications to Transport-Reaction Processes*, Birkhäuser, Boston, Mass, USA, 2001.

[3] C. K. Qi, H.-X. Li, X. C. Zhao, S. Y. Li, and F. Gao, "Hammerstein modeling with structure identification for multi-input multi-output nonlinear industrial processes," *Industrial & Engineering Chemistry Research*, vol. 50, no. 19, pp. 11153–11169, 2011.

[4] C. C. Doumanidis and N. Fourligkas, "Temperature distribution control in scanned thermal processing of thin circular parts," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 5, pp. 708–717, 2001.

[5] H.-X. Li, X.-X. Zhang, and S. Y. Li, "A three-dimensional fuzzy control methodology for a class of distributed parameter systems," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 3, pp. 470–481, 2007.

[6] X.-X. Zhang, H.-X. Li, and S. Y. Li, "Analytical study and stability design of a 3-D fuzzy logic controller for spatially distributed dynamic systems," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1613–1625, 2008.

[7] X.-X. Zhang, S. Li, and H.-X. Li, "Structure and BIBO stability of a three-dimensional fuzzy two-term control system," *Mathematics and Computers in Simulation*, vol. 80, no. 10, pp. 1985–2004, 2010.

[8] X.-X. Zhang, H.-X. Li, and C. K. Qi, "Spatially constrained fuzzy-clustering-based sensor placement for spatiotemporal fuzzy-control system," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 946–957, 2010.

[9] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, MIT Press, London, UK, 2001.

[10] S. Guillaume, "Designing fuzzy inference systems from data: an interpretability-oriented review," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 3, pp. 426–443, 2001.

[11] I. Rojas, H. Pomares, J. Ortega, and A. Prieto, "Self-organized fuzzy system generation from training examples," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 1, pp. 23–36, 2000.

[12] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 7–31, 1993.

[13] R. Krishnapuram and C. P. Freg, "Fitting an unknown number of lines and planes to image data through compatible cluster merging," *Pattern Recognition*, vol. 25, no. 4, pp. 385–400, 1992.

[14] M. Setnes, R. Babuška, U. Kaymak, and H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 28, no. 3, pp. 376–386, 1998.

[15] O. Cordón, F. Herrera, and P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 667–674, 2001.

[16] L. X. Wang, *A Course in Fuzzy Systems and Control*, Prentice-Hall, Upper Saddle River, NJ, USA, 1997.

[17] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, Upper Saddle River, NJ, USA, 2001.

[18] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[19] J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[20] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[21] M.-Y. Chen and D. A. Linkens, "Rule-base self-generation and simplification for data-driven fuzzy models," *Fuzzy Sets and Systems*, vol. 142, no. 2, pp. 243–265, 2004.

[22] Y. Jin, "Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 212–221, 2000.

[23] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, New York, NY, USA, 3rd edition, 1976.

[24] W. E. Schiesser, *The Numerical Methods of Lines Integration of Partial Differential Equations*, Academic Press, San Diego, Calif, USA, 1991.

[25] R. Haber and L. Keviczky, *Nonlinear System Identification—Input–Output Modeling Approach*, vol. 1 of *Nonlinear System Parameter Identification*, Kluwer Academic, Dodrecht, The Netherlands, 1999.

[26] J.-J. Chiang and P.-Y. Hao, "Support vector learning mechanism for fuzzy rule-based modeling: a new approach," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 1, pp. 1–12, 2004.