Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 410279, 13 pages
http://dx.doi.org/10.1155/2013/410279

Research Article

Reference Function Based Spatiotemporal Fuzzy Logic Control Design Using Support Vector Regression Learning

Xian-Xia Zhang, Ye Jiang, Shiwei Ma, and Bing Wang

Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronics and Automation, Shanghai University, Shanghai 200072, China

Correspondence should be addressed to Xian-Xia Zhang; xianxia_zh@shu.edu.cn

Received 25 August 2013; Accepted 1 November 2013

Academic Editor: Baocang Ding

This paper presents a reference function based 3D FLC design methodology using support vector regression (SVR) learning. The concept of reference function is introduced to 3D FLC for the generation of 3D membership functions (MF), which enhance the capability of the 3D FLC to cope with more kinds of MFs. The nonlinear mathematical expression of the reference function based 3D FLC is derived, and spatial fuzzy basis functions are defined. Via relating spatial fuzzy basis functions of a 3D FLC to kernel functions of an SVR, an equivalence relationship between a 3D FLC and an SVR is established. Therefore, a 3D FLC can be constructed using the learned results of an SVR. Furthermore, the universal approximation capability of the proposed 3D fuzzy system is proven in terms of the finite covering theorem. Finally, the proposed method is applied to a catalytic packed-bed reactor and simulation results have verified its effectiveness.

1. Introduction

Many industrial processes and systems are "distributed" in space [1] and are usually called spatially distributed systems. Recently, a novel three-dimensional fuzzy logic controller (3D FLC) [2] has been developed for the control of such systems. The 3D FLC employs a three-dimensional (3D) fuzzy set [2], which is composed of the traditional fuzzy set plus a third dimension for the spatial information, and carries out a 3D rule inference engine; thus, it has the inherent capability to process spatiotemporal dynamic systems. The control strategy of the 3D FLC is similar to how human operators or experts control the temperature in a space domain. Actually, it is a kind of spatiotemporal fuzzy control system with the traditional model-free advantage.

Currently, most 3D FLC designs are based on expert knowledge [2–5], which requires that the human knowledge to the control solution must exist and be structured [6]. However, in many real-world applications, experts may have problems structuring the knowledge. 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 [6]. On the contrast, data sets hidden with effective control rules are usually available. The motivation of this study is to design a 3D FLC using spatiotemporal data information.

To date, few literatures are found to be focused on data-based 3D FLC design methods. In [7], a table look-up scheme was employed to design 3D FLC in terms of input-output pairs. In [8], a fuzzy c-means algorithm (FCM) and gradient-descent approach were used to design a data-based 3D FLC, where FCM was used to learn the initial 3D fuzzy rule base and then the gradient-descent approach was used to optimize the parameters of MFs. In [9, 10], a clustering and linear support vector regression based 3D FLC design method was proposed, where the nearest neighborhood clustering was used to construct the antecedent part of 3D fuzzy rules and a linear support vector regression (SVR) was used to learn the consequent parameters. These methods either yield lots of fuzzy control rules (e.g. in [7]) or require additional algorithm to reduce redundant 3D fuzzy sets or 3D fuzzy rules [8–10]. As a complementary, Zhang et al. proposed a data-based 3D FLC design method using SVR learning [11], where the learned support vectors and associated learning parameters are directly used to design antecedent part and consequent part of 3D fuzzy rules. The best advantage of this method is that reasonable 3D fuzzy control rules are directly extracted and constructed by SVR learning. The limitation of the design is that Gaussian shape membership function (MF) is the only choice for MF design.

In this study, we focus on a reference function based 3D FLC design using SVR learning, which integrates the merits of SVR learning and flexible MF choice. Utilizing the concept of reference function, the 3D FLC can cope with more kinds of MFs, for example, Symmetric triangle, Gaussian, Cauchy, Laplace, Hyperbolic Secant, and Squared Sinc. A nonlinear mathematical description of a reference function based 3D FLC can be derived, and spatial fuzzy basis functions are defined. Via relating spatial fuzzy basis functions of a 3D FLC to kernel functions (KFs) of an SVR, an equivalence relationship between a 3D FLC and an SVR is established. Therefore, a 3D FLC can be constructed using the learned results of an SVR. In addition, the universal approximation capability of the proposed 3D fuzzy system is proven in terms of the finite covering theorem.

The paper is organized as follows. Preliminaries about the reference function, 3D MF generated by reference function, and the nonlinear mapping of reference function based 3D FLC are addressed in Section 2. In Section 3, the methodology and design scheme of the reference function based 3D FLC design using SVR learning are presented. Then, the finite covering theorem is used to prove that the 3D FLC is a universal approximator in Section 4. In Section 5, a catalytic packed-bed reactor is presented as an example to illustrate the proposed 3D FLC and validate its effectiveness. In Section 6, conclusions are given.

2. Preliminaries

2.1. Reference Function

Definition of Reference Function (see [12, 13]). A function $v : R \to [0,1]$ is a reference function if and only if the following two conditions hold:

(1) $v(x) = v(-x)$,

(2) $v(0) = 1$.

Many functions may be reference functions. For instance, Symmetric triangle, Gaussian, Cauchy, Laplace, Hyperbolic Secant, and Squared Sinc as listed in Table 1 are reference functions. The reference functions can be used to generate 3D MFs, which provide a way for 3D FLC to access more kinds of 3D MFs.

Table 1: Reference functions.

Classification	Mathematical expression
Symmetric triangle	$v(x) = \max(1 - d\lvert x\rvert, 0), \quad d > 0$
Gaussian	$v(x) = e^{-dx^2}, \quad d > 0$
Cauchy	$v(x) = \dfrac{1}{1 + dx^2}, \quad d > 0$
Laplace	$v(x) = e^{-d\lvert x\rvert}, \quad d > 0$
Hyperbolic Secant	$v(x) = \dfrac{2}{e^{dx} + e^{-dx}}, \quad d > 0$
Squared Sinc	$v(x) = \dfrac{\sin^2(dx)}{d^2x^2}, \quad d > 0$

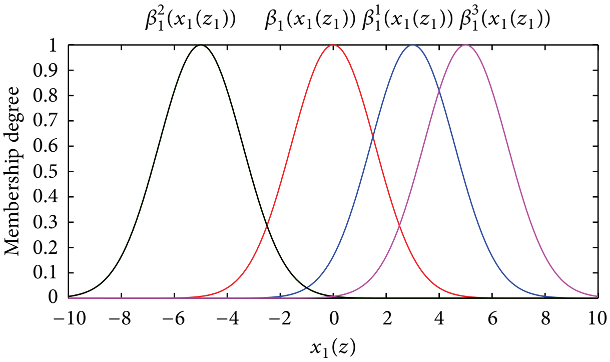


Figure 1: Gaussian MF distribution of spatial input variable $x_1(z)$ at sensing location $z = z_1$. $\beta_1^1(x_1(z_1))$, $\beta_1^2(x_1(z_1))$, and $\beta_1^3(x_1(z_1))$ are generated by the Gaussian type reference function $\beta_1(x_1(z_1))$.

2.2. Reference Function Based 3D FLC

2.2.1. 3D MF Generated by Reference Function. A 3D MF is an extension of a traditional MF by adding a third coordinate for the spatial information. In detail, the 3D MF has three coordinates: one is for the universe of discourse of the variable, another one is for the spatial information, and the third one is for the membership degree. If finite sensors are used, the 3D MF can be considered as the assembly of the traditional 2D MFs at each sensing location. In this way, we can generate a 3D MF by location transformation of a reference function at each sensing location.

For example, we have a spatial input variable $x_1(z)$ defined in a discrete spatial domain $Z = \{z_1, z_2, \ldots, z_p\}$. A 3D MF of $x_1(z)$ can be an assembly of the traditional 2D MFs at each sensing location $z = z_j$ $(j = 1, 2, \ldots, p)$. The MF distribution of $x_1(z)$ at sensing location $z = z_1$ can be shown in Figure 1, where $\beta_1(x_1(z_1))$ is a Gaussian type reference function; $\beta_1^1(x_1(z_1))$, $\beta_1^2(x_1(z_1))$, and $\beta_1^3(x_1(z_1))$ are generated by location transformation of $\beta_1(x_1(z_1))$.

2.2.2. Reference Function Based 3D FLC as a Nonlinear Mapping. The basic structure of a 3D FLC is composed of 3D fuzzifier, 3D rule inference, and defuzzifier. Due to its unique 3D nature, some detailed operations of a 3D FLC are different from a traditional one for spatial information expression, processing, and compression. For their detailed operations, one can refer to [2]. Once each component of a reference function based 3D FLC is set, the nonlinear mathematical description of the 3D FLC can be derived (see Appendix A for a brief derivation). Assuming that we employ 3D singleton fuzzifier, 3D fuzzy rules as shown in (A.4) of Appendix A, "product" t-norm and "weighted aggregation" dimension reduction [3] in the 3D rule inference, singleton fuzzy sets for the output variable, and "linear" defuzzifier [14], the reference function based 3D FLC can be mathematically expressed as

$$\begin{aligned} u\left(x_z\right) &= b^0 + \sum_{l=1}^{N} b^l \sum_{j=1}^{p} a_j \prod_{i=1}^{s} \beta_i^l\left(x_i\left(z_j\right)\right) \\ &= b^0 + \sum_{l=1}^{N} b^l \sum_{j=1}^{p} a_j \prod_{i=1}^{s} \beta_i\left(x_i\left(z_j\right) - \tau_{ij}^l\right), \end{aligned} \tag{1}$$

where $x_i(z_j)$ denotes the input of the ith spatial input variable $x_i(z)$ from the sensing location $z = z_j$; $\beta_i^l(x_i(z_j)) = \beta_i(x_i(z_j)-\tau_{ij}^l)$ denotes MF generated by the reference function $\beta_i(x_i(z_j))$; τ_{ij}^l denotes a location parameter, that is, the location transformation of the MF of $x_i(z_j)$ with respect to $\beta_i(x_i(z_j))$; a_j denotes the spatial weight from the jth sensing location; b^0 and b^l are constants; p denotes sensor number.

In (1), let

$$\Psi^l\left(x_z\right) = \sum_{j=1}^{p} a_j \prod_{i=1}^{s} \beta_i^l\left(x_i\left(z_j\right)\right); \tag{2}$$

then (1) can be rewritten as

$$u\left(x_z\right) = b^0 + \sum_{l=1}^{N} b^l \Psi^l\left(x_z\right). \tag{3}$$

We define $\Psi^l(x_z)$ as a Spatial Fuzzy Basis Function (SFBF) [11]. Each SFBF corresponds to a 3D fuzzy rule, and all the SFBFs correspond to a 3D rule base. Mathematically, a 3D FLC is a linear combination of all the SFBFs. Furthermore, we rewrite (2) into (4)

$$\Psi^l\left(x_z\right) = \sum_{j=1}^{p} a_j \varphi^l\left(x\left(z_j\right)\right), \tag{4}$$

where $\varphi^l(x(z_j)) = \prod_{i=1}^{s} \beta_i^l(x_i(z_j))$.

From (4), we can find that, at each sensing location, there exists a traditional FBF [15] $\varphi^l(x(z_j))$; in the whole space domain, multiple traditional FBFs are assembled by the spatial weights $a_1, \ldots, a_p$ into a SFBF $\Psi^l(x_z)$. All the spatial information expression and processing as well as the fuzzy linguistic expression and rule inference are integrated into SFBFs.

Equation (1) (or (3)) shows that the reference function based 3D FLC is a nonlinear mapping from the input space $x_z \in \Omega \subset R^{p\times s}$ to the output space $u(x_z) \in U \subset R$. In particular, using (3) a reference function based 3D FLC can be represented by a three-layer network structure as show in Figure 2.

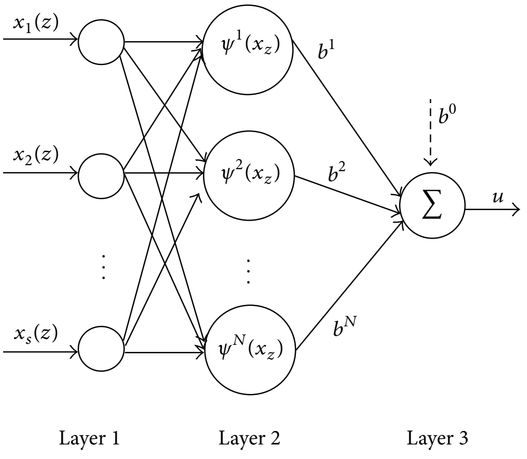


Figure 2: Three-layer network structure of a 3D FLC.

3. Reference Function Based 3D FLC Design Using SVR Learning

3.1. Design Methodology. The design methodology can be depicted by Figure 3. The SFBFs from a 3D FLC are input to an SVR as the KFs, and the learned spatial support vectors as leading spatiotemporal data points from the SVR are imported for the design of a 3D fuzzy rule base. The design theory will involve two crucial issues. The first one is whether SFBFs from a 3D FLC can be used as KFs in an SVR. The second one is what the relationship between a 3D FLC and an SVR would be like on the basis of the first issue.

3.1.1. Spatial Fuzzy Basis Function as Mercer KF. When relating the SFBFs with the KFs in an SVR, for instance, SFBFs are regarded as KFs, the SVR and the 3D FLC will have the same network structures and then have the same mathematical expressions, which will be discussed in Section 3.1.2. Generally speaking, a function satisfying Mercer theorem can be used as a KF for an SVR [16]. In this study, we will prove that an SFBF is a Mercer KF.

In (3), we rewrite $\Psi^l(x_z)$ into $K(x_z, \tau^l)$, which can be further expressed as follows:

$$\begin{gathered} K\left(x_z, \tau^l\right) = K\left(x_z - \tau^l\right) = \sum_{j=1}^{p} a_j K^j\left(x\left(z_j\right), \tau_j^l\right), \\ K^j\left(x\left(z_j\right), \tau_j^l\right) = K^j\left(x\left(z_j\right) - \tau_j^l\right) = \prod_{i=1}^{s} \beta_i\left(x_i\left(z_j\right) - \tau_{ij}^l\right), \end{gathered} \tag{5}$$

where $x_z \in R^{p\times s}$ is a spatial input, $\tau^l \in R^{p\times s}$ is the location transformation parameter of 3D MF in the lth rule, and $K^j(x(z_j), \tau_j^l)$ and $K(x_z, \tau^l)$ are translation invariant KFs [13].

In terms of [13], if the reference functions are positive definite functions, then we do get a Mercer kernel. The reference functions as listed in Table 1 are positive definite functions. Using these reference functions to generate MF, from [13], we can conclude that $K^j(x(z_j), \tau_j^l)$ is a Mercer kernel. Since the linear combination of KFs is still a KF [16], we can derive that $K(x_z, \tau^l)$ is still a Mercer KF. Therefore, SFBFs are Mercer KFs, which can be used as KFs for SVR learning.

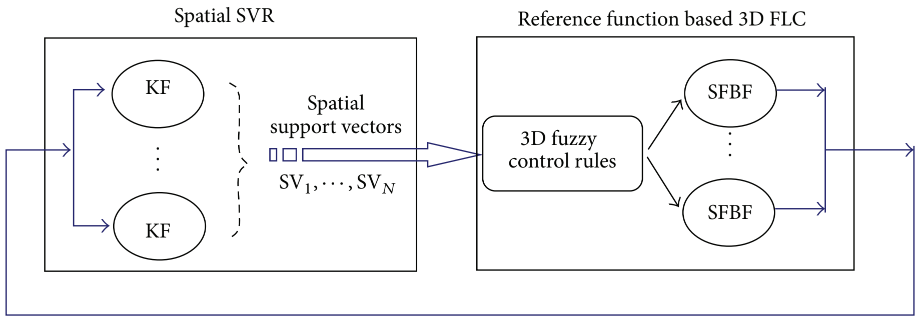


FIGURE 3: Design methodology of reference function based 3D FLC design using SVR learning.

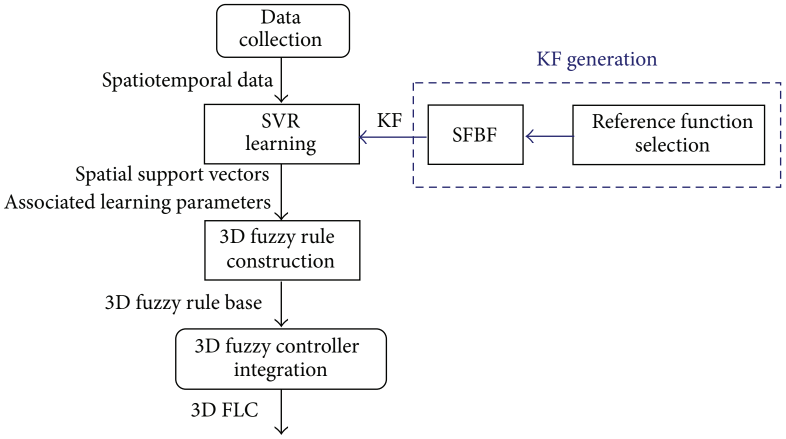

Data collection
Spatiotemporal data
SVR learning
KF
KF generation
SFBF
Reference function selection
Spatial support vectors
Associated learning parameters
3D fuzzy rule construction
3D fuzzy rule base
3D fuzzy controller integration
3D FLC

FIGURE 4: Design scheme of reference function based 3D FLC using SVR learning.

3.1.2. Mathematical Equivalence of a Spatial SVR and a 3D FLC. Once the SFBFs from the 3D FLC are employed as the KFs for an SVR, an inherent equivalence relationship will be built between the SVR and the 3D FLC (comparing Figure 2 and Figure 10). By combining (1) and (B.7), we have the following mathematical expressions:

$$\begin{aligned} u(x_z) &= b^0 + \sum_{l=1}^{N} b^l \Psi^l(x_z) \\ &= b^0 + \sum_{l=1}^{N} b^l \sum_{j=1}^{p} a_j \varphi^l\left(x\left(z_j\right)\right) \\ &= b + \sum_{l=1}^{N} (\alpha_l^* - \alpha_l) K\left(x_z, x_z^l\right), \end{aligned} \tag{6}$$

where $b^0 = b$, $b^l = (\alpha_l^* - \alpha_l)$, and $\Psi^l(x_z) = K(x_z, \tau^l) = K(x_z, x_z^l)$.

From (6), we can find that each spatial support vector x_z^l and its associated learning parameter $(\alpha_l^* - \alpha_l)$ correspond to one 3D fuzzy rule, where x_z^l is applied to set the center of the 3D MF of the 3D fuzzy set $\overline{C}_i^l$ $(i = 1, \ldots, s)$ in the lth rule, that is, the location transformation of the 3D fuzzy set with respect to reference function $\beta(x_z, \tau^l)$, and $(\alpha_l^* - \alpha_l)$ is used to set b^l (the constant for the consequent set of the lth rule in 3D FLC).

3.2. Design Scheme. The design of a reference function based 3D FLC consists of five parts: data collection, KF generation, SVR learning, 3D fuzzy rule construction, and 3D fuzzy controller integration, as shown in Figure 4.

(1) Data Collection. A set of spatiotemporal data will be collected. The data should contain effective control laws. Essentially, the reference function based 3D FLC design is a fuzzy modeling [17] that extracts fuzzy control rules from the spatiotemporal data.

(2) KF Generation. Before SVR learning, KFs should be properly designed. In this step, via properly selecting reference function, SFBFs from a 3D FLC will be formulated (as in (4)) to set KFs for SVR learning.

(3) SVR Learning. With proper KFs, the SVM learning algorithm directly executes the spatiotemporal data set and yields spatial support vectors $x_z^1, \ldots, x_z^N$ and their associated learning parameters $\alpha_1^* - \alpha_1, \ldots, \alpha_N^* - \alpha_N$.

(4) 3D Fuzzy Rule Construction. The spatial support vectors and their associated learning parameters, as leading control laws, are used to construct 3D fuzzy control rules. In detail,

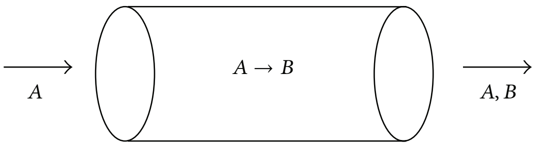


Figure 5: Sketch of a catalytic packed-bed reactor.

the spatial support vector x_z^l is employed to construct the antecedent part of the lth rule; $\alpha_l^* - \alpha_l$ is employed to construct the consequent part of the lth rule. The form of each 3D fuzzy rule is shown as below

$$\overline{R}^l : \text{if } x_z \text{ is close to } x_z^l \text{ then } u \text{ is close to } (\alpha_l^* - \alpha_l). \quad (7)$$

It is shown that the result of the SVM learning can be easily interpreted using structured linguistic knowledge. Finally, a 3D rule base with N rules is established.

(5) 3D Fuzzy Controller Integration. Once the 3D rule base is established, a 3D FLC can be achieved by integrating other components including 3D fuzzifier, 3D rule inference, and defuzzifier. The detailed settings are given in Section 2.2.2. Finally, we obtain a complete 3D FLC, which can be used as a controller for a spatially distributed dynamic system.

4. Universal Approximation of Reference Function Based 3D FLC

The reference function based 3D FLC design method is used to construct a 3D FLC from spatiotemporal data hidden with effective control laws. In other words, the 3D FLC aims at approximating an unknown nonlinear control function. In this study, we use the finite covering theorem to prove that the 3D FLC is a universal approximator; that is, it can approximate continuous control functions to arbitrary accuracy.

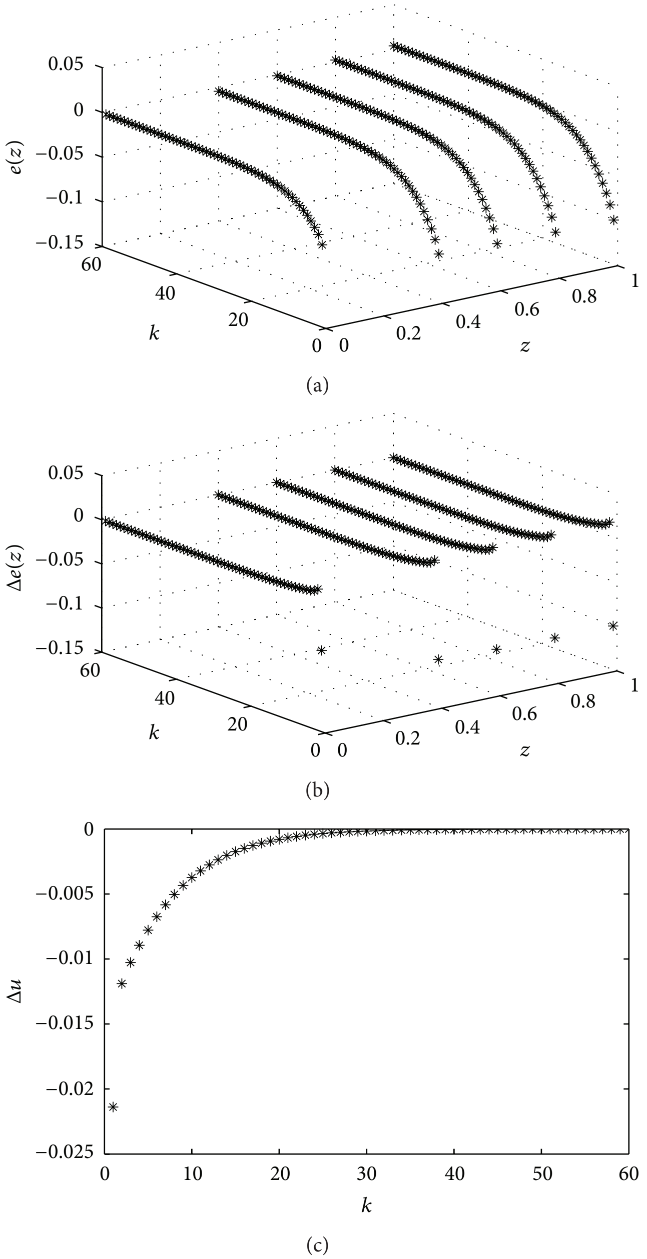
The universal approximation capability of the reference function based 3D FLC can be described by Theorem 1.

Theorem 1. *Let $\tilde{g}(x_z) : R^{p\times s} \rightarrow R$ be a continuous function defined on a compact Ω. For each $\varepsilon > 0$, there exists a reference function based 3D FLC $u(x_z)$ such that*

$$\sup_{x_z\in\Omega} \left(\left|u(x_z) - \tilde{g}(x_z)\right|\right) < \varepsilon. \quad (8)$$

From (A.4), it is shown that $\overline{R}^0$ is an universal rule, namely, for any spatial input x_z, $\overline{R}^0$ will be fired. In (A.4), the fired rule $\overline{R}^0$ will produce the constant b^0. Let

$$\begin{aligned} F(x_z) &= u(x_z) - b^0 \\ &= \sum_{l=1}^{N} b^l \sum_{j=1}^{p} a_j \prod_{i=1}^{s} \beta_i \left(x_i(z_j) - \tau_{ij}^l\right) \\ &= \sum_{l=1}^{N} b^l K(x_z, \tau^l). \end{aligned} \quad (9)$$



(a)

(b)

(c)

Figure 6: Spatial-temporal data set: (a) spatial error $e(z)$; (b) spatial error in change $\Delta e(z)$; (c) incremental output Δu (z: spatial dimension; k: serial number of input-output data).

$F(x_z)$ can be regarded as a 3D FLC generated by rule base $\{\overline{R}^1, \overline{R}^2, \ldots, \overline{R}^N\}$. Then, Theorem 1 can be restated as Theorem 2 as follows.

Theorem 2. *Under the condition of Theorem 1, let $g(x_z) : R^{p\times s} \rightarrow R$ be a continuous function defined on a compact Ω. For any constant b^0, one has $\tilde{g}(x_z) = g(x_z) + b^0$. For each $\varepsilon > 0$, there exists a reference function based 3D FLC $F(x_z)$ such that*

$$\sup_{x_z\in\Omega} \left(\left|F(x_z) - g(x_z)\right|\right) < \varepsilon. \quad (10)$$

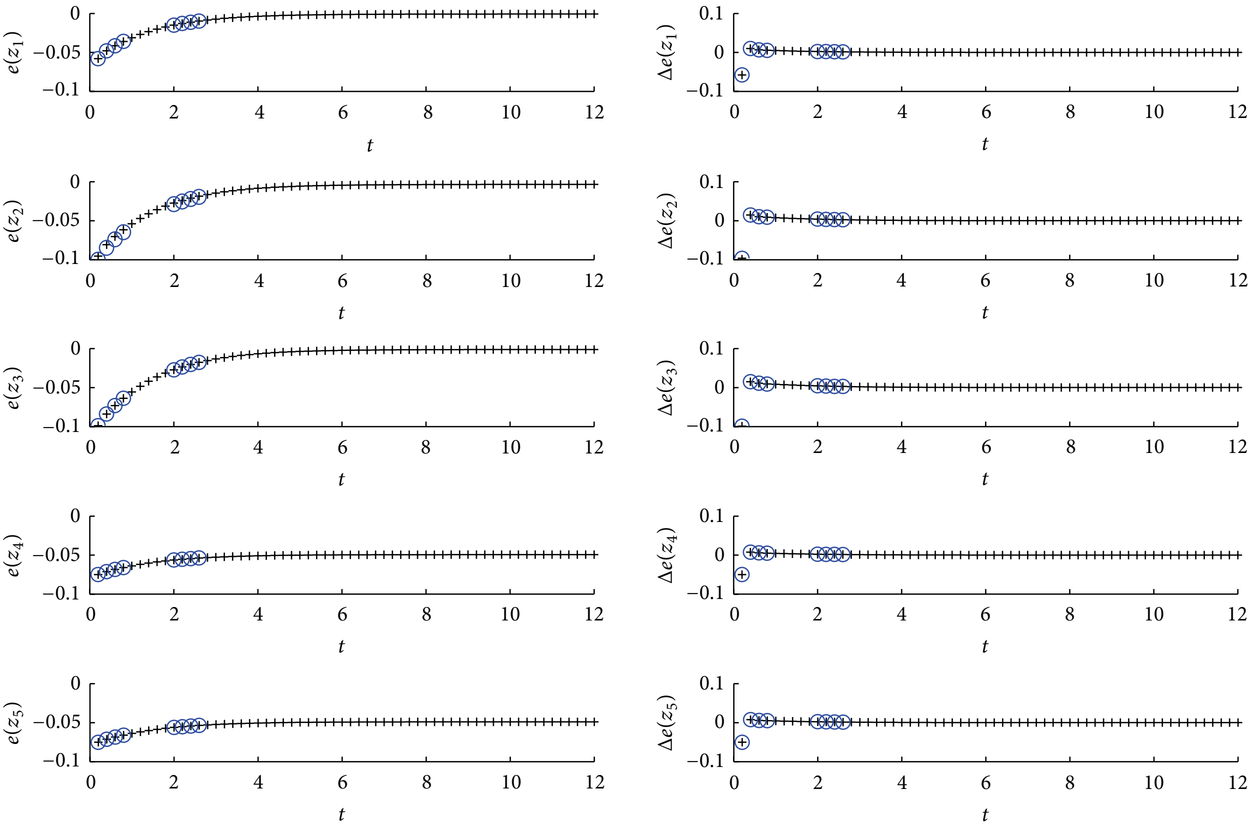


Figure 7: Input data and support vectors from each sensing location. Circle (o): support vector; dot (+): input data.

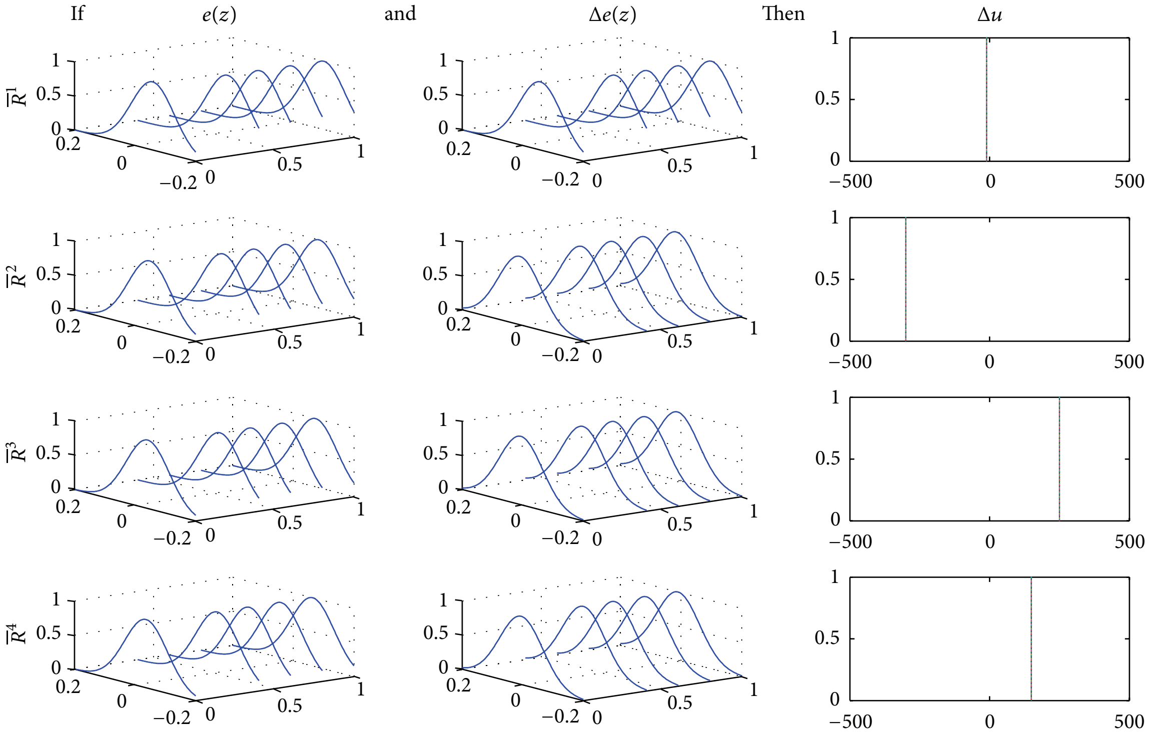
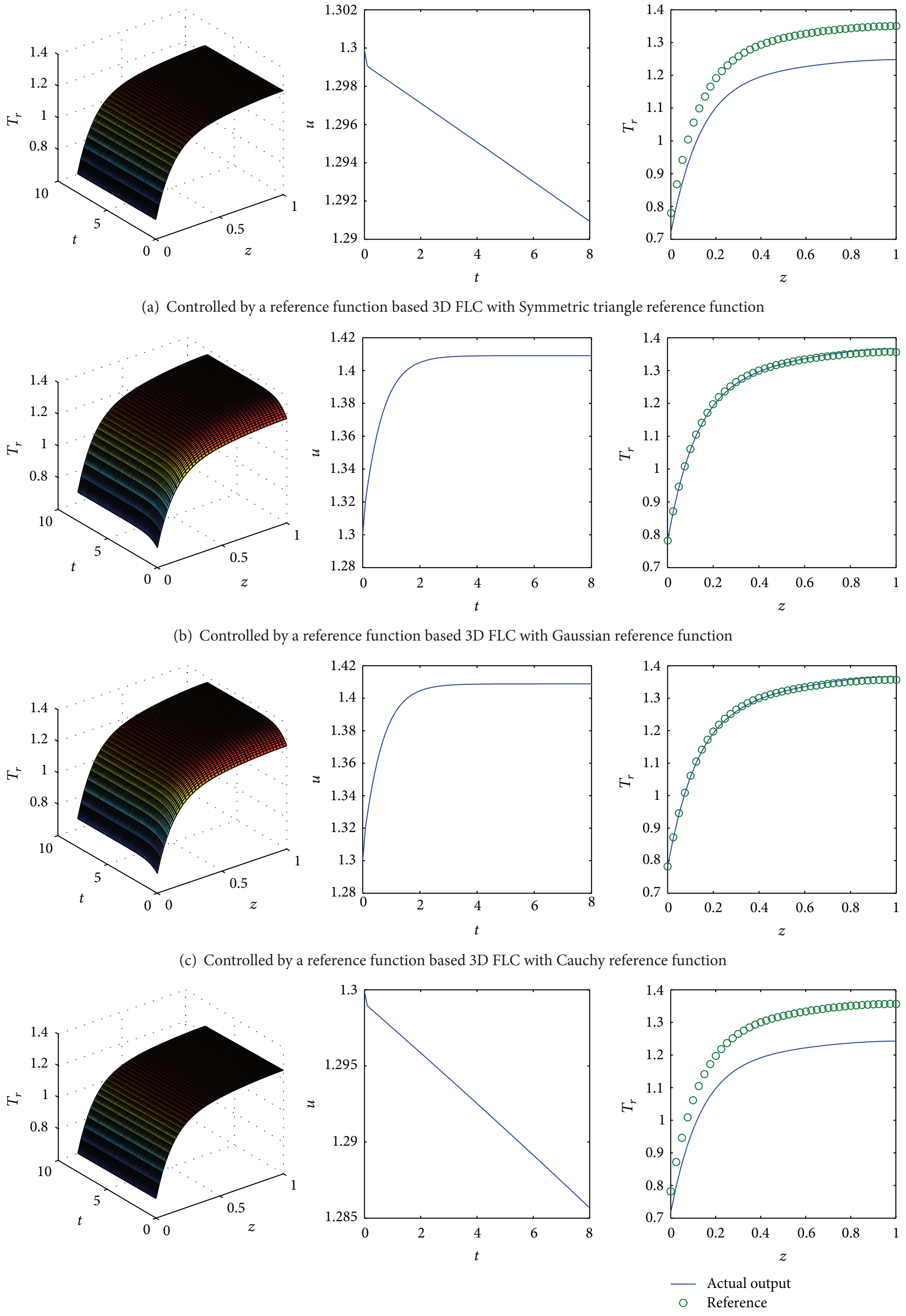


Figure 8: The first four 3D fuzzy rules and their associated MFs ($b = -0.0752$).



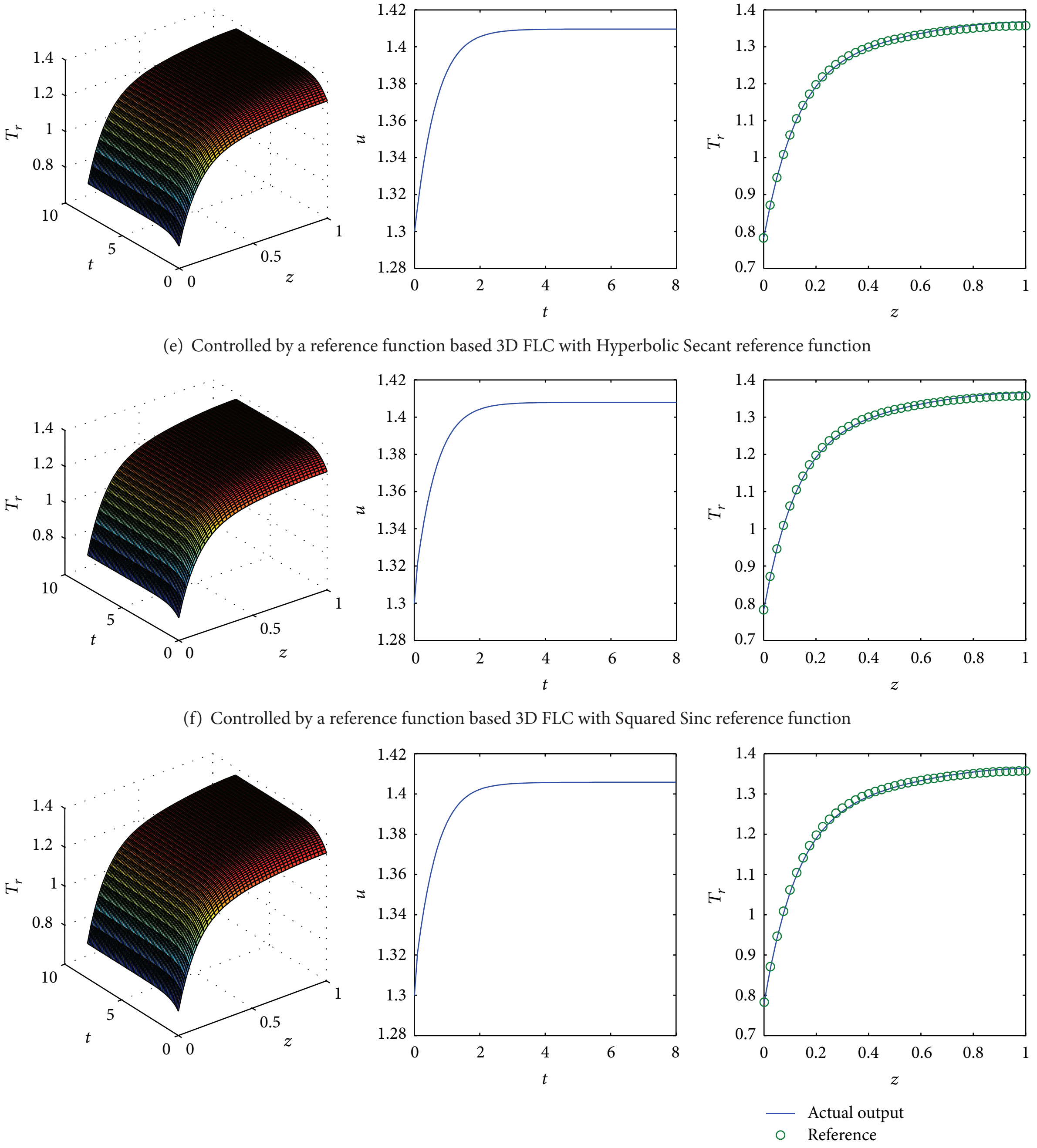
(a) Controlled by a reference function based 3D FLC with Symmetric triangle reference function

(b) Controlled by a reference function based 3D FLC with Gaussian reference function

(c) Controlled by a reference function based 3D FLC with Cauchy reference function

(d) Controlled by a reference function based 3D FLC with Laplace reference function

FIGURE 9: Continued.



(e) Controlled by a reference function based 3D FLC with Hyperbolic Secant reference function

(f) Controlled by a reference function based 3D FLC with Squared Sinc reference function

(g) Controlled by an expert based 3D FLC

Figure 9: Control performance comparisons. From left to right in (a)–(g): catalyst temperature varying with time and space, manipulated input, and catalyst temperature at steady state.

Before the proof of Theorem 2, we first present some preparation work.

When $x_z = \overline{x}_z \in \Omega$, the firing level of the fired rule $\overline{R}^l$ ($l = 1, \ldots, N$) is

$$\mu_{\varphi^l}(\overline{x}_z) = \sum_{j=1}^{p} a_j \prod_{i=1}^{s} \beta_i\left(\overline{x}_i(z_j) - \tau_{ij}^l\right). \qquad (11)$$

The inference result of $\overline{R}^l$ is given by

$$D^l(u) = \begin{cases} 0 & \text{if } \mu_{\varphi^l}(\overline{x}_z) = 0 \text{ or } u \neq b^l, \\ b^l \mu_{\varphi^l}(\overline{x}_z) & \text{in other case,} \end{cases} \qquad (12)$$

where u is the output variable of the 3D FLC, which is corresponding to the "u" of the consequent part of the fired rule. The composition result of all the fired rules is given as follows:

$$D(u) = \bigcup_{l=1}^{N} D^{l}(u). \tag{13}$$

Based on the above preparation, Lemma 3 is presented as follows.

Lemma 3. *Under the condition of Theorem 1 there exists a reference function based 3D FLC $F(x_z)$ such that*

$$D(u)\left|u - g(x_z)\right| \le D(u) * \varepsilon \quad \textit{for each } u \in R. \tag{14}$$

Proof. Let $a_z \in \Omega$. As $g(\cdot)$ is continuous at a_z, for each $i = 1, \ldots, s$ there exists a $\delta_{a_z}^{i} > 0$ such that

$$\left|x_z^{i} - a_z^{i}\right| \le \delta_{a_z}^{i} \quad i = (1, \ldots, s) \Longleftrightarrow \left|g(x_z) - g(a_z)\right| \le \varepsilon. \tag{15}$$

For each $a_z \in \Omega$, set

$$O_{a_z} = \left\{x_z \mid \left|x_z^{i} - a_z^{i}\right| \le \delta_{a_z}^{i}\ (i = 1, \ldots, s)\right\}. \tag{16}$$

Then, O_{a_z} is open on $R^{p\times s}$ and $\Omega \subseteq \bigcup_{a_z\in\Omega} O_{a_z}$. As Ω is compact, there exists a finite subfamily $O_{a_z^1}, O_{a_z^2}, \ldots, O_{a_z^k}$ such that

$$\Omega \subseteq O_{a_z^1} \bigcup O_{a_z^2} \bigcup \cdots \bigcup O_{a_z^k}. \tag{17}$$

We can construct a 3D FLC $F(x_z)$, defined by

$$\mu_{\varphi^l}(x_z) = \begin{cases} \sum_{j=1}^{p} a_j^{l} \prod_{i=1}^{s} \beta_i\left(x_i(z_j) - \tau_{ij}^{l}\right) & x_z \in O_{a_z^l} \\ 0 & x_z \notin O_{a_z^l} \end{cases} \quad (l = 1, \ldots, N), \tag{18}$$

$$b^{l} = \begin{cases} g(x_z) & x_z \in O_{a_z^l} \\ 0 & x_z \notin O_{a_z^l} \end{cases} \quad (l = 1, \ldots, N). \tag{19}$$

When $x_z = \overline{x}_z \in \Omega$, we have the following.

(1) If $D(u) = 0$, the lemma is trivial.

(2) If $D(u) > 0$, then $D(u) = \bigcup_{l=1}^{N} D^{l}(u) > 0$; hence, there exists a $k \in [1, N]$ such that $D^{k}(u) > 0$. Therefore, we further have that the following.

   (a) From $D^{k}(u) \neq 0$, it follows that $\mu_{\varphi^k}(\overline{x}_z) \neq 0$; that is, $\overline{x}_z \in O_{a_z^k}$. In terms of the continuity of $g(\cdot)$, we have $|g(\overline{x}_z) - g(a_z^{k})| \le \varepsilon$.

   (b) From $D^{k}(u) \neq 0$, it follows that $D^{k}(u) = b^{k}\mu_{\varphi^k}(\overline{x}_z) \neq 0$, and then $b^{k} \neq 0$. We have $u = b^{k} = g(a_z^{k})$.

Hence, we have

$$\left|u - g(\overline{x}_z)\right| \le \left|u - g\left(a_z^{k}\right)\right| + \left|g\left(a_z^{k}\right) - g(\overline{x}_z)\right| < 0 + \varepsilon = \varepsilon. \tag{20}$$

□

In terms of Lemma 3, the proof of Theorem 2 can be given as follows.

Proof of Theorem 2. Consider

$$\begin{aligned} \left|F(\overline{x}_z) - g(\overline{x}_z)\right| &\le \frac{\sum D(u)\left|u - g(\overline{x}_z)\right|}{\sum D(u)} \\ &\le \frac{\varepsilon * \sum D(u)}{\sum D(u)} \le \varepsilon. \end{aligned} \tag{21}$$

□

5. Application

5.1. A Catalytic Packed-Bed Reactor. This designed 3D FLC is applied to a catalytic packed-bed reactor [1, 4, 18] shown in Figure 5, where a reaction of the form $A \rightarrow B$ takes place 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:

$$\begin{aligned} \frac{\partial T_r}{\partial t} &= -\frac{\partial T_r}{\partial z} + \frac{1}{P_{eT}}\frac{\partial^2 T_r}{\partial z^2} \\ &\quad - B_T B_C C_A \exp\left(\frac{\gamma_r T_r}{1 + T_r}\right) + \beta_T\left(u - T_r\right), \\ \frac{\partial C_A}{\partial t} &= -\frac{\partial C_A}{\partial z} + \frac{1}{P_{eC}}\frac{\partial^2 C_A}{\partial z^2} - B_C C_A \exp\left(\frac{\gamma_r T_r}{1 + T_r}\right) \end{aligned} \tag{22}$$

subject to the boundary conditions

$$\begin{aligned} z = 0, &\quad P_{eT} T_r = \frac{\partial T_r}{\partial z}, \quad P_{eC}\left(C_A - 1\right) = \frac{\partial C_A}{\partial z}, \\ z = 1, &\quad \frac{\partial T_r}{\partial z} = 0, \quad \frac{\partial C_A}{\partial z} = 0, \end{aligned} \tag{23}$$

where T_r, C_A, and u denote the dimensionless temperature, the concentration of reactant A, and jacket temperature, respectively; t and z denote the dimensionless time and space; P_{eT} and P_{eC} are the heat and mass Peclet numbers, B_T is a dimensionless heat of reaction, B_C is a dimensionless preexponential factor, γ_r is a dimensionless activation energy, and β_T is a dimensionless heat transfer coefficient. The values of the process parameters are given as follows:

$$\begin{aligned} P_{eT} = 5.0, &\quad P_{eC} = 5.0, \quad B_C = 0.00001, \\ B_T = 1.0, &\quad \beta_T = 15.62, \quad \gamma_r = 22.14. \end{aligned} \tag{24}$$

The control problem is to maintain a desired reaction rate via tuning the jacket temperature to control catalyst temperature. In this application, the reactor began to work at one steady state; because of the requirement of operation conditions, the reference value of temperature is increased by 8%. Thus, the control objective is to make the temperature of reactor well track the new reference value along the space domain.

5.2. Design of Reference Function Based 3D FLC

(1) Data Collection. The spatiotemporal data is collected from the catalytic packed-bed reactor, which is controlled by

an expert based 3D FLC [4]. Five point sensors are located along the length of the reactor with $Z = [00.40.60.81]$ for collecting the spatial distribution of the temperature T_r. Two spatial inputs are error and error in change; that is, $e^*(Z, k) = T_{sd}(Z) - T_s(Z, k)$ and $\Delta e^*(Z) = e^*(Z, k) - e^*(Z, k-1)$. The detailed design of the expert based 3D FLC, including 3D fuzzifier, 3D rule inference, and defuzzifier, can refer to [4]. The scaling factors for the spatial error, the spatial error in change, and the incremental output are set as 0.5, 0.1, and 0.3, respectively. The sample period is 0.1 s, and the sampling duration is 6 s. Thus, we have 60 input-output data pairs (shown in Figure 6), each of which is represented by $([e(z)_k, \Delta e(z)_k], \Delta u_k)$, $k = 1, \ldots, 60$.

(2) KF Generation. The reference functions (Symmetric triangle, Gaussian, Cauchy, Laplace, Hyperbolic Secant, and Squared Sinc) as listed in Table 1 are employed, respectively, to generate 3D MFs and then formulate SFBFs. SFBFs are used for KFs in an SVR learning.

(3) SVR Learning. With the spatiotemporal data set as above, the SVM learning algorithm is used for the support vector learning. It should be noted that the SFBFs in (3) are taken as the KFs. As a result, spatial support vectors are extracted and their associated learning parameters are obtained. For instance, when the Gaussian type reference function is used for KF generation, eight support vectors were learned from 60 spatiotemporal data pairs when $C = 1000$, $\varepsilon = 0.00005$, and $d = 0.1$, as shown in Figure 7, where the spatiotemporal input data is decomposed into multiple two-dimensional graphical representations over the space domain.

(4) 3D Fuzzy Rule Construction. In terms of the learned results of the SVR in the previous step, we establish 3D fuzzy rules. For instance, with the Gaussian type reference function, eight 3D fuzzy rules are constructed. The first four 3D fuzzy rules are presented as follows.

$\overline{R}_1$: if $e(z)$ is close to $[-0.0580 \ -0.0963 \ -0.0988 \ -0.1000 \ -0.1005]'$ and $\Delta e\ (z)$ is close to $[-0.0580 \ -0.0963 \ -0.0988 \ -0.1000 \ -0.1005]'$, then Δu is close to -9.5172.

$\overline{R}_2$: if $e(z)$ is close to $[-0.0479 \ -0.0817 \ -0.0839 \ -0.0848 \ -0.0851]'$ and $\Delta e(z)$ is close to $[0.0101 \ 0.0146 \ 0.0149 \ 0.0152 \ 0.0154]'$, then Δu is close to -298.9862.

$\overline{R}_3$: if $e(z)$ is close to $[-0.0414 \ -0.0714 \ -0.0731 \ -0.0735 \ -0.0737]'$ and $\Delta e(z)$ is close to $[0.0065 \ 0.0104 \ 0.0109 \ 0.0113 \ 0.0114]'$, then Δu is close to 249.2288.

$\overline{R}_4$: if $e(z)$ is close to $[-0.0359 \ -0.0626 \ -0.0638 \ -0.0639 \ -0.0639]'$ and $\Delta e(z)$ is close to $[0.0055 \ 0.0087 \ 0.0092 \ 0.0096 \ 0.0098]'$, then Δu is close to 148.1335.

The first four 3D fuzzy rules and their associated 3D MF distributions can be depicted in Figure 8, which show the inherent spatial nature of the 3D fuzzy control system.

(5) 3D Fuzzy Controller Integration. Based on the 3D fuzzy rules established in step (4), we obtain a complete 3D FLC by combining 3D fuzzifier, 3D rule inference, and defuzzifier. The resultant 3D FLC will be used as a controller for the catalytic packed-bed reactor.

5.3. Control Performance Validation. The designed reference function based 3D FLC using SVR learning is validated on the nonlinear catalytic packed-bed reactor. We employed six kinds of reference functions, that is, Symmetric triangle, Gaussian, Cauchy, Laplace, Hyperbolic Secant, and Squared Sinc, and finally produced six 3D FLCs. With the reference function based 3D FLC as the controller, the catalyst temperature varying with time and space, manipulated input, and the catalyst temperature at steady state are presented in Figures 9(a)–9(f). The control performance is given in Table 2, where steady-state error (SSE), integral of the absolute error (IAE), and integral of time multiplied by absolute error (ITAE) [2] are used as the performance criteria. In terms of Figures 9(a)–9(f) and Table 2, we can find that different reference functions will yield different control performance. In this application, Gaussian, Cauchy, Hyperbolic Secant, and Squared Sinc reference functions result in good control performance, while Symmetric triangle and Laplace reference functions lead to poor control performance. The results illustrate that KF will influence the control performance; thus, in the actual application, we should choose proper KF to design a 3D FLC.

As a comparison, we do another control experiment; that is, the expert knowledge-based 3D FLC from [4] is taken as the controller. As for its detailed design including 3D MF, 3D rule base, 3D inference, fuzzification, and defuzzification, one can refer to [4]. The scaling factors for the spatial error, the spatial error in change, and the incremental output are set as 0.5, 0.1, and 0.3, respectively. The controlled profiles and control performance are given in Figure 9(g) and Table 2, respectively.

From Figure 9 and Table 2, we can find that with a proper reference function, the reference function based 3D FLC has nearly the same control performance as the expert knowledge-based 3D FLC. It means that the proposed spatial SVR learning method can well extract the control laws hidden in a spatiotemporal input–output dataset and formulate them in the form of 3D fuzzy rules.

6. Conclusions

A reference function based 3D FLC design methodology using SVR learning is proposed for spatially distributed dynamic systems. Utilizing the concept of reference function, the 3D FLC can access more kinds of 3D MFs, such as Symmetric triangle, Gaussian, Cauchy, Laplace, Hyperbolic Secant, and Squared Sinc. Based on the mathematical expressions of reference function based 3D FLC, we define spatial fuzzy basis functions and then find an equivalence

Table 2: Performance comparisons.

Reference function based 3D FLC	3D fuzzy rules	ISS ($\times 10^{-3}$)	IAE ($\times 10^{-1}$)	ITAE ($\times 10^{-1}$)
Symmetric triangle reference function	7	103.4	16.163	65.457
Gaussian reference function	8	4.7	1.864	3.526
Cauchy reference function	8	4.7	1.881	3.537
Laplace reference function	7	108	16.525	67.408
Hyperbolic Secant reference function	28	4.8	1.918	3.567
Squared Sinc reference function	5	4.7	1.859	3.515
Expert-based 3D FLC	25	4.8	1.931	3.676

relationship between a 3D FLC and an SVR by connecting spatial fuzzy basis functions in the 3D FLC to KFs in the SVR. On the basis of the equivalence relationship, a 3D FLC can be designed using the SVR learning; that is, the learned spatial support vectors as the optimal leading data points can be directly used for 3D fuzzy control rule generation. The proposed reference function based 3D FLC design can be carried out in five steps: data collection, KF generation, SVR learning, 3D fuzzy rule construction, and 3D fuzzy controller integration. Besides, the universal approximation capability of the proposed 3D fuzzy system is discussed. Finally, effectiveness of the proposed 3D FLC design methodology is validated on a catalytic packed-bed reactor.

Appendices

A. Nonlinear Mapping Derivation of a Reference Function Based 3D FLC

Let $x_z = (x_1(z), \ldots, x_s(z))$ be a spatial input vector. Then, the 3D MF of the ith spatial input $x_i(z)$ is given as

$$\mu_i = \beta_i^l \left(x_i (z)\right) = \beta_i \left(x_i (z) - \tau_i^l\right) \tag{A.1}$$

and the Gaussian type 2D MF of the ith spatial input $x_i(z)$ at the sensing location $z = z_j$ is given as

$$\mu_{ij} = \beta_i^l \left(x_i \left(z_j\right)\right) = \beta_i \left(x_i \left(z_j\right) - \tau_{ij}^l\right). \tag{A.2}$$

Via a 3D fuzzifier, the spatial input vector x_z in the universe of discourse X can be transformed into a spatial fuzzy input $\overline{A}_X$ as below:

$$\begin{aligned} &\overline{A}_X \\ &= \sum_{z\in Z} \sum_{x_1(z)\in X_1} \cdots \sum_{x_s(z)\in X_s} \mu_{\overline{A}_X} \left(x_1 (z), \ldots, x_s (z), z\right) \\ &\qquad / \left(x_1 (z), \ldots, x_s (z), z\right) \\ &= \sum_{z\in Z} \sum_{x_1(z)\in X_1} \cdots \sum_{x_s(z)\in X_s} \mu_{X_1} \left(x_1 (z), z\right) * \cdots * \mu_{X_s} \left(x_s (z), z\right) \\ &\qquad / \left(x_1 (z), \ldots, x_s (z), z\right), \end{aligned} \tag{A.3}$$

where $*$ denotes the t-norm operation.

Assume that 3D fuzzy rules are designed with the following form:

$$\begin{aligned} &\overline{R}^0 : \text{if } x_1 (z) \text{ is } \overline{C}_1^0 \text{ and} \cdots \text{and } x_s (z) \text{ is } \overline{C}_s^0 \text{ then } u \text{ is } b^0, \\ &\overline{R}^l : \text{if } x_1 (z) \text{ is } \overline{C}_1^l \text{ and} \cdots \text{and } x_s (z) \text{ is } \overline{C}_s^l \text{ then } u \text{ is } b^l, \end{aligned} \tag{A.4}$$

where $\overline{C}_i^0$ is a universal 3D fuzzy set, whose MF at sensing location $z = z_j$ is $\beta_i^0(x_i(z_j)) \equiv 1$; $\overline{C}_i^l$ is a 3D fuzzy set, whose MF at sensing location $z = z_j$ is $\beta_i^l(x_i(z_j)) : R \to [0, 1]$, $i = 1, \ldots, s$; b^0 and b^l are constants.

Then, for each fired rule, a fuzzy relation is obtained as below:

$$\overline{R}^l : \overline{C}_1^l \times \cdots \times \overline{C}_s^l \longrightarrow b^l, \quad l = 0, 1, 2, \ldots, N. \tag{A.5}$$

A 3D rule inference integrates the spatial information processing and the traditional inference and contains three main operations: spatial information fusion, dimension reduction, and traditional inference. Firstly, using the spatial information fusion operation, we have a spatially distributed set W^l over the space domain with the grade of the MF derived as

$$\begin{aligned} &\mu_{W^l} (z) \\ &= \mu_{\overline{A}_X \circ (\overline{C}_1^l \times \cdots \times \overline{C}_s^l)} \left(x_z, z\right) \\ &= \sup\nolimits_{x_1(z)\in X_1, \ldots, x_s(z)\in X_s} \left[\mu_{\overline{A}_X} \left(x_z, z\right) * \mu_{\overline{C}_1^l \times \cdots \times \overline{C}_s^l} \left(x_z, z\right)\right] \\ &= \left\{\sup\nolimits_{x_1(z)\in X_1} \left[\mu_{X_1} \left(x_1 (z), z\right) * \mu_{\overline{C}_1^l} \left(x_1 (z), z\right)\right]\right\} \\ &\qquad * \cdots * \left\{\sup\nolimits_{x_s(z)\in X_s} \right. \\ &\qquad\qquad \left. \left[\mu_{X_s} \left(x_s (z), z\right) * \mu_{\overline{C}_s^l} \left(x_s (z), z\right)\right]\right\} \\ &= \prod_{i=1}^{s} \beta_i \left(x_i (z) - \tau_i^l\right), \end{aligned} \tag{A.6}$$

where "product" is used for t-norm ($*$) and singleton fuzzifier is used.

Then, utilizing a weighted aggregation [3] dimension reduction operation, a 2D set χ^l is obtained. Consider the following:

$$\begin{aligned}\mu_{\chi^l} &= a_1\mu_{W^l}(z_1) + a_2\mu_{W^l}(z_2) + \cdots + a_p\mu_{W^l}(z_p) \\ &= \sum_{j=1}^{p} a_j \prod_{i=1}^{s} \beta_i\left(x_i(z_j) - \tau_{ij}^{l}\right).\end{aligned} \tag{A.7}$$

Finally, traditional inference operation (Mamdani implication operation) and linear defuzzifier [14] are carried out successively. We have the nonlinear mathematical expression as follows:

$$\begin{aligned}u(x_z) &= b^0 + \sum_{l=1}^{N} b^l \sum_{j=1}^{p} a_j \prod_{i=1}^{s} \mu_{\overline{C}_i^l}\left(x_i(z_j)\right) \\ &= b^0 + \sum_{l=1}^{N} b^l \sum_{j=1}^{p} a_j \prod_{i=1}^{s} \beta_i\left(x_i(z_j) - \tau_{ij}^{l}\right).\end{aligned} \tag{A.8}$$

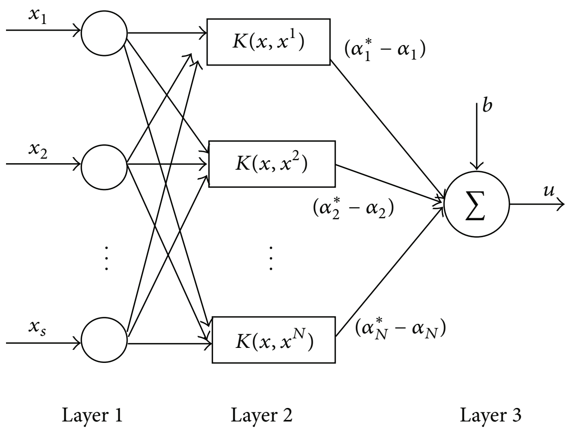
B. Mathematical Preliminaries of ε-Support Vector Regression

In this study, we focus on ε-support vector regression (ε-SVR). Suppose that we have a training set $D = \{[x_i, y_i] \in R^s \times R,\ i = 1, \ldots, q\}$ consisting of q pairs (x_1, y_1), (x_2, y_2), …, (x_q, y_q), where the inputs are s-dimensional vectors and the labels are continuous values. In ε-SVR, 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:

$$\left|y - f(x,w)\right|_{\varepsilon} = \begin{cases} 0 & \text{if } \left|y - f(x,w)\right| \le \varepsilon \\ \left|y - f(x,w)\right| - \varepsilon & \text{otherwise.} \end{cases} \tag{B.1}$$

To make the SVR nonlinear, we may map the input vector $x \in R^s$ into the vector v of a high-dimensional feature space, $v = \Theta(x)$, where Θ represents a mapping $R^s \rightarrow R^f$, and formulate a linear regression problem in this feature space, and then an optimization problem will be solved. The optimization problem can also be solved in a dual space. By introducing the Lagrange multipliers α_i and α_i^*, the primal optimization problem can be formulated in its dual form as follows:

$$\begin{aligned}\max_{\alpha_i, \alpha_i^*} \Bigg\{ &-\frac{1}{2}\sum_{i=1}^{q}\sum_{j=1}^{q}\left(\alpha_i^* - \alpha_i\right)\left(\alpha_j^* - \alpha_j\right)\left\langle \Theta(x_i) \cdot \Theta(x_j)\right\rangle \\ &- \varepsilon \sum_{i=1}^{q}\left(\alpha_i^* + \alpha_i\right) + \sum_{i=1}^{q}\left(\alpha_i^* - \alpha_i\right) y_i \Bigg\}\end{aligned} \tag{B.2}$$



Layer 1: input $x = (x_1, \ldots, x_s)$

Layer 2: support vectors $x^1, \ldots, x^N$ and KFs $K(x, x^1), \ldots, K(x, x^N)$

Layer 3: output $u = \sum_{\mathrm{l}=1}^{N}(\alpha_{\mathrm{i}}^* - \alpha_{\mathrm{i}})K(x, x^i) + b$

Figure 10: Three-layer network structure of an SVR.

subject to

$$\sum_{j=1}^{q}\alpha_i^* = \sum_{i=1}^{q}\alpha_i, \quad 0 \le \alpha_i^* \le C,\ 0 \le \alpha_i \le C,\ i = 1, \ldots, q, \tag{B.3}$$

where the constant C is a design parameter chosen by the user, which determines the tradeoff between the complexity of $f(x, w)$ and the approximate error.

Solving the dual quadratic programming problem, we can find an optimal weight vector w and an optimal bias b of the regression hypersurface given in (B.4):

$$\begin{aligned} w &= \sum_{i=1}^{q}\left(\alpha_i^* - \alpha_i\right)\Theta(x_i), \\ b &= \frac{1}{q}\left(\sum_{i=1}^{q}\left(y_i - \left\langle w \cdot \Theta(x_i)\right\rangle\right)\right).\end{aligned} \tag{B.4}$$

Then, the best regression hypersurface is given by

$$\begin{aligned} f(x,w) &= \sum_{i=1}^{q}\left(\alpha_i^* - \alpha_i\right)\left\langle \Theta(x) \cdot \Theta(x_i)\right\rangle + b \\ &= \sum_{i \in \mathrm{SV}}\left(\alpha_i^* - \alpha_i\right)\left\langle \Theta(x) \cdot \Theta(x_i)\right\rangle + b.\end{aligned} \tag{B.5}$$

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

To avoid a direct mapping $\Upsilon(x)$, the kernel trick is used. AKF $K(x_i, x_j)$, which satisfies the Mercer's theorem, is introduced as below:

$$K\left(x_i, x_j\right) = \left\langle \Theta(x_i) \cdot \Theta(x_j)\right\rangle. \tag{B.6}$$

Using $K(x_i, x_j)$, the SVR can be constructed which operates in an infinite dimensional space. Then, the solution of the SVR has the form

$$f(x, w) = \sum_{i \in \mathrm{SV}} (\alpha_i^* - \alpha_i) K\left(x, x^i\right) + b. \tag{B.7}$$

Let $x^1, x^2, \ldots, x^N$ represent support vectors. The solution of the SVR can be described by a three-layer network structure as shown in Figure 10.

Acknowledgment

This work was supported by the project from the National Science Foundation of China under Grant no. 61273182, no. 51375293, and no. 61171145.

References

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

[2] 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.

[3] 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.

[4] X. X. Zhang, S. Y. 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.

[5] 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.

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

[7] Y. Jiang, X. X. Zhang, T. Zou, and G. T. Cao, "A novel 3-D fuzzy logic controller design using table look-up scheme," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM '10)*, pp. V4-388–V4-393, Taiyuan, China, October 2010.

[8] X. X. Zhang, J. J. Li, Y. Jiang, B. L. Su, C. K. Qi, and T. Zou, "Fuzzy clustering based spatiotemporal fuzzy logic controller design," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pp. 3167–3172, Beijing, China, July 2012.

[9] X. X. Zhang, Y. Jiang, T. Zou, C. K. Qi, and G. T. Cao, "Data-driven based 3-D fuzzy logic controller design using nearest neighborhood clustering and linear support vector regression," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ '11)*, pp. 374–380, Taipei, Taiwan, June 2011.

[10] X. X. Zhang, J. D. Qi, B. L. Su, S. W. Ma, and H. B. Liu, "A clustering and SVM regression learning-based spatiotemporal fuzzy logic controller with interpretable structure for spatially distributed systems," *Journal of Applied Mathematics*, vol. 2012, Article ID 841609, 24 pages, 2012.

[11] X. X. Zhang, Y. Jiang, H. X. Li, and S. Y. Li, "SVR learning-based spatiotemporal fuzzy logic controller for nonlinear spatially distributed dynamic systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1635–1647, 2013.

[12] D. Dubois and H. Prade, "Operations on fuzzy numbers," *International Journal of Systems Science*, vol. 9, no. 6, pp. 613–626, 1978.

[13] Y. Chen and J. Z. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 716–728, 2003.

[14] H. Ying, *Fuzzy Control and Modeling: Analytical Foundations and Applications*, IEEE Press, New York, NY, USA, 2000.

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

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

[17] R. Babuska, *Fuzzy Modeling for Control*, Kluwer Academic, Boston, Mass, USA, 1998.

[18] P. D. Christofides and P. Daoutidis, "Nonlinear control of diffusion-convection-reaction processes," *Computers and Chemical Engineering*, vol. 20, no. 2, pp. S1071–S1076, 1996.