

## Research Article

# A SVR Learning Based Sensor Placement Approach for Nonlinear Spatially Distributed Systems

Xian-xia Zhang,<sup>1</sup> Zhi-qiang Fu,<sup>1</sup> Wei-lu Shan,<sup>1</sup> Bing Wang,<sup>1</sup> and Tao Zou<sup>2</sup>

<sup>1</sup>Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronics and Automation, Shanghai University, Shanghai 200072, China

<sup>2</sup>Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

Correspondence should be addressed to Xian-xia Zhang; [xianxia\\_zh@shu.edu.cn](mailto:xianxia_zh@shu.edu.cn)

Received 29 August 2016; Accepted 27 September 2016

Academic Editor: Wu Deng

Copyright © 2016 Xian-xia Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many industrial processes are inherently distributed in space and time and are called spatially distributed dynamical systems (SDDSs). Sensor placement affects capturing the spatial distribution and then becomes crucial issue to model or control an SDDS. In this study, a new data-driven based sensor placement method is developed. SVR algorithm is innovatively used to extract the characteristics of spatial distribution from a spatiotemporal data set. The support vectors learned by SVR represent the crucial spatial data structure in the spatiotemporal data set, which can be employed to determine optimal sensor location and sensor number. A systematic sensor placement design scheme in three steps (data collection, SVR learning, and sensor locating) is developed for an easy implementation. Finally, effectiveness of the proposed sensor placement scheme is validated on two spatiotemporal 3D fuzzy controlled spatially distributed systems.

## 1. Introduction

Many industrial processes are inherently distributed in space and time, such as fluid flow process, spray deposition process, heat exchange process, and snap curing process. These systems are usually called spatially distributed dynamical systems (SDDSs) or distributed parameter systems (DPSs) [1]. Traditionally, the spatially distributed nature is ignored, and then an SDDS is simplified into a lumped parameter system (LPS). However, its performance will deteriorate if the system dynamics significantly vary with space [2]. To satisfy tighter product quality requirements, the spatial nature should be considered in the modeling and control. Consequently, the spatiotemporal modeling and control for SDDSs have become the highlight in the modern modeling and control theory.

On the research of SDDS, since sensor placement affects capturing the spatial distribution, it becomes one of key issues to influence the performance of spatiotemporal modeling or control. Sensor placement is a difficult problem, particularly in nonlinear and multivariable case [3].

In the past several decades, sensor placement has been studied and can be classified into two categories, that is, model-based method and data-driven based method. For the first category, the sensor placement is dependent on an accurate mathematical model. Some methods aim at achieving better parameter estimation with different criteria, for example, scalar criteria of the covariance matrix [4] and scalar measure of performance defined on the Fisher information matrix [5]. Some methods intend to improve state estimation or state observer with possible criteria, such as observability measures [6], convergence properties of observer [7], and cost function related to Gramian observability matrix [8]. Some methods integrate optimal actuator and sensor placement with nonlinear output feedback control [9–11] and improve control performance, where the penalty on the response of the closed-loop system is used for cost function of actuator placement and the control action, and the estimation error in the closed-loop infinite-dimensional system is used for cost function of sensor placement.

For the second category, only data information is used for sensor placement. Wouwer et al. [12] proposed the determinant of Fisher information matrix formed by sensitivity

functions for parameter estimation and the Gram determinant formed by sensor responses for observer estimation. Zamproga et al. [13] selected the most suitable secondary process variable as soft sensor inputs for batch distillation by exploiting the properties of principal component analysis (PCA) on the sensitivity matrix. Tongpadungroda et al. [14] identified the optimal position of an applied load by GA under the criterion related to PCA. Zhang et al. [15] proposed a sensor placement method for spatiotemporal 3D fuzzy control system based on spatial constrained fuzzy  $c$ -means algorithm. Compared with the model-based method, the data-driven method is very promising in practical engineering applications since no mathematical model is used.

In this study, a new data-driven based sensor placement method is proposed. Support vector regression (SVR) algorithm is used to extract the main characteristics of spatial distribution from a spatiotemporal data set, which can be directly used for the sensor placement. The support vectors learned by SVR represent the crucial spatial data structure hidden in the spatiotemporal data set, which can be employed to determine the optimal sensor location and sensor number. A systematic sensor placement design scheme is developed for an easy implementation, which consists of data collection, SVR learning, and sensor locating. The effectiveness of the proposed sensor placement scheme is validated on two spatiotemporal 3D fuzzy controlled nonlinear spatially distributed systems.

This paper is organized as follows. Preliminaries about nonlinear spatially distributed systems and SVR algorithm are described in Section 2. In Section 3, the SVR learning based sensor placement method is presented in detail. Two illustrative examples are given in Section 4. Finally, Section 5 summarizes the conclusion.

## 2. Preliminaries

**2.1. Nonlinear Spatially Distributed Dynamical Systems.** Many industrial processes exhibit highly nonlinear behavior and strong spatial variations. They usually can be represented by nonlinear partial differential equations. Some processes have strong convection characteristics [16], some have strong diffusion phenomena [17], and some have both convection and diffusion characteristics [18]. These systems are usually installed with multiple spatially distributed actuators and multiple sensors and give rise to nonlinear control problems that involve the regulation of highly distributed control variables using these spatially distributed actuators and sensors.

In the following subsections, two examples of SDDSs are presented and illustrated.

**2.1.1. A Catalytic Packed-Bed Reactor.** The catalytic packed-bed reactor [17, 19] in Figure 1 is a typical spatiotemporal dynamic system. A reaction of the form  $C \rightarrow D$  takes place on the catalyst. The reaction is endothermic and a jacket is used to heat the reactor. The dimensionless mathematical model

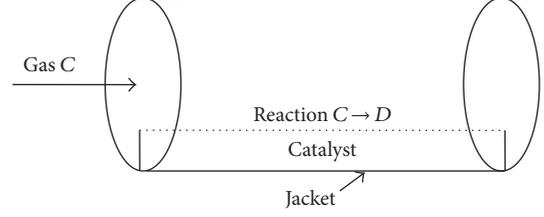


FIGURE 1: Sketch of a catalytic packed-bed reactor.

that describes this nonlinear tubular chemical reactor is given as follows:

$$\begin{aligned} \varepsilon_p \frac{\partial T_g(z, t)}{\partial t} &= -\frac{\partial T_g(z, t)}{\partial z} + \alpha_c (T_s(z, t) - T_g(z, t)) \\ &\quad - \alpha_g (T_g(z, t) - u(t)), \\ \frac{\partial T_s(z, t)}{\partial t} &= \frac{\partial^2 T_s(z, t)}{\partial z^2} + B_0 \exp\left(\frac{\gamma T_s(z, t)}{1 + T_s(z, t)}\right) \\ &\quad - \beta_c (T_s(z, t) - T_g(z, t)) \\ &\quad - \beta_p (T_s(z, t) - b(z)u(t)) \end{aligned} \quad (1)$$

subject to the following boundary conditions:

$$\begin{aligned} z = 0, \\ T_g(z, t) &= 0, \\ \frac{\partial T_s(z, t)}{\partial z} &= 0, \\ z = 1, \\ \frac{\partial T_s(z, t)}{\partial z} &= 0, \end{aligned} \quad (2)$$

where  $T_g(z, t)$  and  $T_s(z, t)$  denote the dimensionless temperature of gas and catalyst, respectively, which are spatially dependent on  $z \in [0, 1]$ ,  $b(z)u(t)$  denotes the spatiotemporal heating source with the distribution  $b(z)$  and the manipulated input  $u(t)$ , and  $u(t)$  denotes the dimensionless temperature of jacket. The values of the process parameters are given as follows:

$$\begin{aligned} \varepsilon_p &= 0.01, \\ \gamma &= 21.14, \\ \beta_c &= 1.0, \\ \beta_p &= 15.62, \\ B_0 &= -0.003, \\ \alpha_c &= 0.5, \\ \alpha_g &= 0.5. \end{aligned} \quad (3)$$

In this application, the control target is to control the catalyst temperature  $T_s(z, t)$  throughout the reactor in order

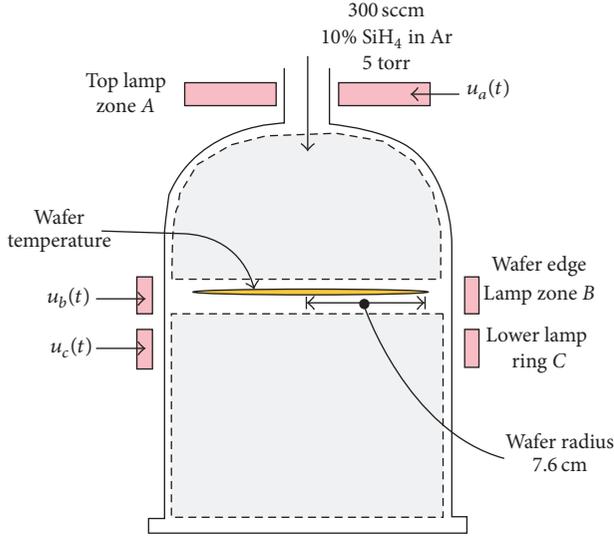


FIGURE 2: Structure of a RTCVD system.

to maintain a desired degree of reaction rate using the measurements of catalyst temperature from  $q$  sensing locations  $z' = [z_1 \ z_2 \ \dots \ z_q]$  and manipulating one spatially distributed heating source ( $b(z) = 1 - \cos(\pi z)$ ). The spatial reference profile is given as  $T_{sd}(z) = 0.42 - 0.2 \cos(\pi z)$ ,  $0 \leq z \leq 1$ , and then the entire spatial catalyst temperature should follow this reference.

**2.1.2. A Rapid Thermal Chemical Vapor Deposition System.** Consider a rapid thermal chemical vapor deposition (RTCVD) reactor with three-zone heating banks (Adomaitis 1995; Theodoropoulou et al., 1998). The sketch of the RTCVD system is depicted in Figure 2, where the lamp bank A mainly heats the total area of the wafer, the lamp bank B mainly heats the wafer edge, and the lamp bank C is used to coarsely adjust the wafer temperature average. When a wafer is being heated, it is rotated for azimuthal temperature uniformity. 10% silane feed in inert gas is inputted into the reactor from the top, and the silane is decomposed into silicon and hydrogen. A  $0.5 \mu\text{m}$  film of polysilicon is expected to deposit on the wafer when deposition temperature appears at near 800 K or higher. The temperature is controlled by operating the power to lamp banks A, B, and C. A dimensionless wafer thermal dynamics equation (Adomaitis 1995) over the wafer domain is given as follows:

$$\frac{\partial T_r'}{\partial t'} = \kappa_0 \left[ \left( \frac{1}{r'} \right) \frac{\partial T_r'}{\partial r'} + \frac{\partial^2 T_r'}{\partial r'^2} \right] + \sigma_0 (1 - T_r'^4) + \omega_r q_a(r') u_a + \omega_r q_b(r') u_b + \omega_r q_c(r') u_c \quad (4)$$

subject to the boundary condition

$$\begin{aligned} \frac{\partial T_r'}{\partial r'} &= \sigma_{ed} (1 - T_r'^4) + q_{ed} u_b \quad \text{when } r' = 1 \\ \frac{\partial T_r'}{\partial r'} &= 0 \quad \text{when } r' = 0, \end{aligned} \quad (5)$$

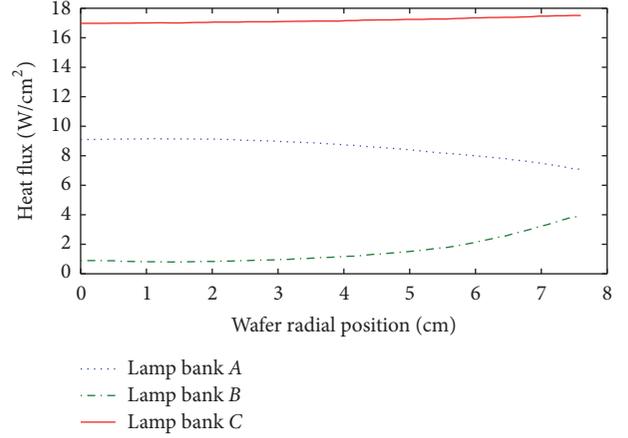


FIGURE 3: Radiation flux distribution of three-zone heating lamp banks.

where  $T_r' = T_r/T_{\text{amb}}$  denotes dimensionless wafer temperature,  $T_r$  denotes the actual wafer temperature, and  $T_{\text{amb}} = 300 \text{ K}$  denotes ambient temperature;  $r' = r/R_w$  denotes the dimensionless radial position of the wafer with  $r$  denoting the actual radial position of the wafer and  $R_w$  denoting the wafer radius ( $R_w = 7.6 \text{ cm}$ );  $t' = t/\tau$  denotes dimensionless time,  $t$  denotes the actual time, and  $\tau = 2.9 \text{ s}$  denotes residence time of upper chamber;  $q_a(r')$ ,  $q_b(r')$ , and  $q_c(r')$  denote the radiant energy flux on the radial position  $r'$  from the lamp banks A, B, and C to the wafer, respectively (the distributions are shown in Figure 3);  $u_a$ ,  $u_b$ , and  $u_c$  denote the percentage of the lamp power for the three-zone lamp banks. The values of the process parameters are given as follows:

$$\begin{aligned} \kappa_0 &= 0.0021, \\ \sigma_0 &= 0.0012, \\ \sigma_{ed} &= 0.0037, \\ q_{ed} &= 4.022, \\ \omega_r &= 0.0256. \end{aligned} \quad (6)$$

From (4) and (5), we can find that the wafer temperature is a variable varying with time and space. The concerned control problem is to control the wafer temperature throughout the wafer radius to arrive at the set temperature 1000 K quickly and uniformly using the measurements of wafer temperature from  $q$  sensing locations  $z' = [z_1 \ z_2 \ \dots \ z_q]$  and manipulating power percentage ( $u_a$ ,  $u_b$ , and  $u_c$ ) to three zones of lamp banks.

**2.2. SVR Algorithm.** Support vector regression (SVR) is a learning technique that originated from theoretical foundations of the statistical learning theory [20]. Over the past decade, SVR has become one of the most popular regression techniques and has been employed in a variety of applications. It performs a new inductive principle (structural risk minimization) for learning from finite training data sets and chooses a model of the right complexity from a large number

of candidate models (learning machines) to describe training data pairs.

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. SVR builds a function as follows:

$$\begin{aligned} f(x, w) : R^s &\longrightarrow R \\ x &\longmapsto \langle w, \psi(x) \rangle_{\wp} + b, \end{aligned} \quad (7)$$

where  $b$  is a bias term and  $\psi(\cdot) : R^s \rightarrow \wp$  is an application mapping the space of factors into a feature space  $\wp$ .

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

$$\min_{w, b} = \frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^q [f(x_i, w) - y_i]_{\varepsilon} \right), \quad (8)$$

where  $C$  is a design parameter chosen by the user, which determines the trade-off between the complexity of  $f(x, w)$

and the approximate error, and  $[\cdot]_{\varepsilon}$  is the  $\varepsilon$ -insensitive loss function which is defined as follows:

$$\begin{aligned} [f(x, w) - y]_{\varepsilon} \\ = \begin{cases} 0 & \text{if } |f(x, w) - y| \leq \varepsilon \\ |f(x, w) - y| - \varepsilon & \text{otherwise.} \end{cases} \end{aligned} \quad (9)$$

In practice,  $\wp, w \in \wp$  and  $\psi(x) \in \wp$  are deduced from a kernel  $K(\cdot, \cdot) : R^s \times R^s \rightarrow R$  involved in the computation of a scalar product:

$$w = \sum_i w_i K(x_i, \cdot),$$

$$\psi(x) = K(x_i, \cdot), \quad (10)$$

$$\langle w, \psi(x) \rangle_{\wp} = \sum_i w_i K(x_i, x).$$

The above optimization problem can be solved in a dual space. By introducing the Lagrange multipliers  $\alpha_i$  ( $\alpha_i^*$ ), the primal optimization problem can be formulated in its dual form as follows:

$$\begin{aligned} \max_{\alpha_i, \alpha_i^*} & \left\{ -\frac{1}{2} \sum_{i=1}^q \sum_{j=1}^q (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) K(x_i, x_j) - \varepsilon \sum_{i=1}^q (\alpha_i^* + \alpha_i) + \sum_{i=1}^q (\alpha_i^* - \alpha_i) y_i \right\} \\ \text{subject to} & \sum_{j=1}^q \alpha_j^* = \sum_{i=1}^q \alpha_i \quad 0 \leq \alpha_i^* \leq C, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, q. \end{aligned} \quad (11)$$

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

$$\begin{aligned} w &= \sum_{i=1}^q (\alpha_i^* - \alpha_i) \psi(x_i) \\ b &= \frac{1}{q} \left( \sum_{i=1}^q (y_i - \langle w, \psi(x_i) \rangle) \right). \end{aligned} \quad (12)$$

Then, the best regression hypersurface is given by

$$\begin{aligned} f(x, w) &= \sum_{i=1}^q (\alpha_i^* - \alpha_i) K(x_i, x) + b \\ &= \sum_{i \in \text{SV}} (\alpha_i^* - \alpha_i) K(x_i, x) + b. \end{aligned} \quad (13)$$

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

### 3. SVR Learning Based Sensor Placement Scheme

**3.1. Design Methodology.** Consider a nonlinear SDDS in Figure 4,  $\mathbf{u}(t) \in R^L$  is the temporal input, and  $y(z, t) \in R$  is the spatiotemporal output, where  $t$  is the time variable,  $z \in \bar{Z}$  is the spatial variable, and  $\bar{Z}$  is the spatial domain. Here suppose the system is controlled by the  $L$  actuators with implemental temporal signal  $\mathbf{u}(t) = [u_1(t) \ u_2(t) \ \dots \ u_L(t)]$  and certain spatial distribution. The output is measured at the  $p$  spatial locations  $z_1, z_2, \dots, z_p$  ( $p \geq 1$ ); therefore, let  $\bar{Z} = [z_1, z_2, \dots, z_p]$  and let  $y(\bar{Z}, t) = [y(z_1, t), y(z_2, t), \dots, y(z_p, t)]$  be spatial sensing output. Let  $\mathbf{S} = [S_1 \ S_2 \ \dots \ S_p]$  be a collected data set, where  $S_j = [X_j, z_j]$ ,  $X_j = [x_{j1}, \dots, x_{jN}]$  is the collected measurement data from the  $j$ th ( $1 \leq j \leq p$ ) sensing location in the time interval  $[t_0, t_1]$ , and  $x_{js} = y(z_j, s)$  is the output measurement at the  $s$ th sampling period,  $1 \leq s \leq N$ . The sensor placement problem is to learn optimal sensor locations from the spatiotemporal data set  $\mathbf{S}$ .

The design methodology can be depicted in Figure 4.  $\mathbf{S}$  is taken as the training set.  $\varepsilon$ -SVR is used to find support vectors and produce a function so that all training patterns  $X$  have

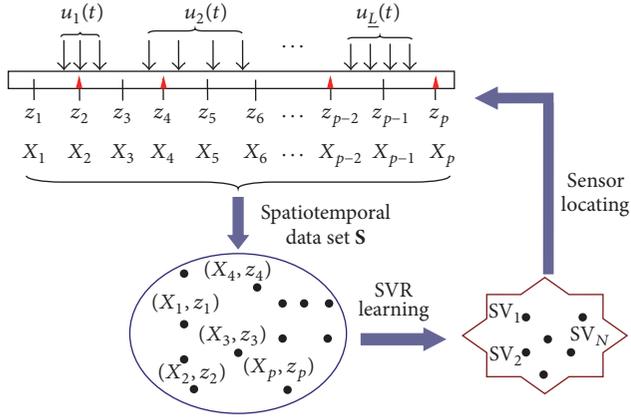


FIGURE 4: Methodology of SVR learning based sensor placement.

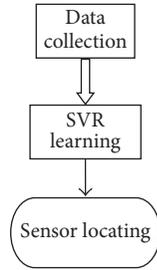


FIGURE 5: Systematic design for sensor placement.

a maximum deviation  $\varepsilon$  from the target values  $z_i$  and have a maximum margin. The support vectors, as crucial data points that represent principle spatial structures, are employed to determine the optimal sensor location and sensor number.

**3.2. Systematic Design Scheme.** The systematic sensor placement design scheme is composed of three steps shown in Figure 5, including data collection, SVR learning, and sensor locating. Firstly, a set of data is collected as the fundamental data for sensor locating. To acquire sufficiently rich information from the system, the system should be excited by adding persistently exciting perturbation signals. Secondly, SVR is used to extract crucial data points from the collected data set. Thirdly, the sensor location is determined in terms of the extracted crucial data.

**(1) Data Collection.** The idea data should contain sufficiently rich information of dynamic characteristics. To acquire the sufficiently rich data, one possible method is to add persistently exciting perturbation signals to the input  $\mathbf{u}(t)$ . In this study, we use white noise and the pseudorandom multilevel signals (PRMSs) with maximal length [21] as the input perturbation signals for two application examples, respectively. When input perturbation signals are sequentially added to the input  $\mathbf{u}(t)$ , the system output is collected. The collected data and its corresponding spatial location consist of fundamental data set  $\mathbf{S}$  for the subsequent support vector learning.

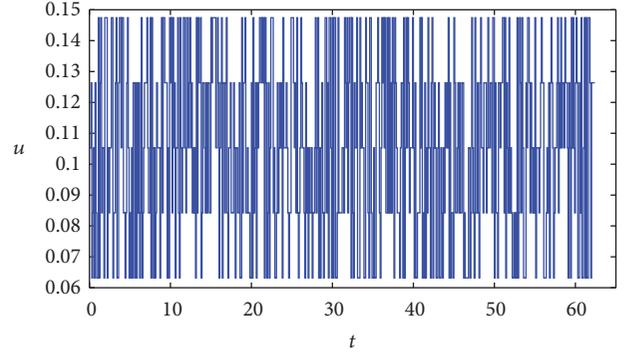


FIGURE 6: Perturbation signal generated by PRQS.

**(2) SVR Learning.** Before the learning, Gaussian kernel function with its spread  $\sigma$  is selected and two learning parameters  $C$  and  $\varepsilon$  are required to be set. We use  $K$ -Fold Cross-Validation method to acquire  $C$ ,  $\varepsilon$ , and  $\sigma$ . Root mean-squared error (RMSE) is employed as quantitative performance criteria for test, which is defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^p (z_k^* - z_k)^2}{p}}, \quad (14)$$

where  $z_k^*$  denotes the predicted spatial point after learning,  $z_k$  denotes the practical spatial point, and  $p$  denotes the number of sensors.

**(3) Sensor Locating.** After the second step, we can obtain some data points that represent the spatial structure implied in the data set. In this step, we can easily determine the sensor locations, namely, the spatial locations in these support vectors. The number of support vectors is equal to the number of sensors.

## 4. Case Studies

In this section, we take the two practical applications described in Section 2 as examples. The mathematical model (1) and (2) in the catalytic packed-bed reactor and the mathematical model (4) and (5) in the RTCVD system are only for the process simulation for evaluation of the sensor placement scheme. The method of lines [22] is used to simulate the two models.

**4.1. The Catalytic Packed-Bed Reactor.** The pseudorandom quinary signal (PRQS) with maximum length of 624 is used as the perturbation signal. The parameters of PRQS are chosen as follows: the lowest and uppermost values of input are 0.1474 and 0.0632, respectively, the number of the levels is 5, the sampling time is 0.1, the length of the period is 624, and the minimum switching time (clock period) is 0.1. The perturbation signal generated by PRQS is depicted in Figure 6.

The space domain of the system is discretized uniformly into 81 points; that is,  $\bar{Z} = [z_1, z_2, \dots, z_{81}]$ , the sampling period is set as 0.1, and the time interval of simulation is

TABLE 1: Performance index comparisons (SVR learning based sensing locations versus evenly distributed locations).

Performance index	Sensor number with evenly distributed locations								SVR learning based sensor number (sensor location)
	2	3	4	5	6	7	8	9	8 {0.0625, 0.275, 0.475, 0.5375, 0.575, 0.7625, 0.8125, 1}
SSE ( $\times 10^{-2}$ )	3.332	1.73	1.74	1.74	1.74	1.74	1.75	1.74	1.70
IAE ( $\times 10^{-1}$ )	3.382	2.355	2.362	2.366	2.36	2.364	2.365	2.362	2.341
ITAE ( $\times 10^{-1}$ )	10.751	5.715	5.749	5.761	5.754	5.753	5.771	5.754	5.615

[0, 62.3]. We collect the catalytic temperature measurements from the system over the space domain and constitute the data set  $\mathbf{S} = [S_1 \ S_2 \ \dots \ S_{81}]$ , where  $S_j = [X_j, z_j]$ ,  $X_j = [x_{j1}, \dots, x_{j624}]$  ( $1 \leq j \leq p$ ),  $x_{js} = y(z_j, s)$  is the catalytic temperature measurement at the  $s$ th sampling period from the  $j$ th sensing location,  $1 \leq s \leq 624$ .

Then, the SVR is used to learn support vectors from the data set  $\mathbf{S}$ . As for model selection problem of the SVR, we employ 5-Fold Cross-Validation method to select proper  $C$  from  $[2^2, 2^{15}]$  with step-size set as 10,  $\varepsilon$  from  $[2^{-10}, 2^2]$  with step-size set as 0.01, and  $\sigma$  from  $\{\sigma_0/100, \sigma_0/95, \dots, \sigma_0/(n+5), \sigma_0/n, \sigma_0/(n-5), \dots, \sigma_0/10, \sigma_0/5, \sigma_0\}$ , where  $\sigma_0$  is set as 10% of the length of the input interval of  $y(\bar{Z}, t)$  denoted by

$$\sigma_0 = \max_{1 \leq j \leq p} (\sigma_j) = \max_{1 \leq j \leq p} \left\{ \left( \frac{y^{\max}(z_j) - y^{\min}(z_j)}{10} \right) \right\}, \quad (15)$$

where  $y^{\max}(z_j)$  and  $y^{\min}(z_j)$  are the maximum and the minimum bound value of temperature measurements from the  $j$ th spatial input variable, respectively. In this application,  $\sigma_0 = 0.03969$ .

After 5-Fold Cross-Validation, we can find a group of best parameters; that is,  $C = 4096$ ,  $\varepsilon = 0.174$ , and  $\sigma = \sigma_0/19 = 0.00209$ . The learned support vectors are  $\mathbf{S} = \{S_6, S_{23}, S_{39}, S_{44}, S_{47}, S_{62}, S_{66}, S_{81}\}$ . Therefore, the optimal sensor locations are  $\{z_6, z_{23}, z_{39}, z_{44}, z_{47}, z_{62}, z_{66}, z_{81}\}$ ; that is,  $\{0.0625, 0.275, 0.475, 0.5375, 0.575, 0.7625, 0.8125, 1\}$ . At the same time, we can find the optimal sensor number is eight.

The effectiveness of the proposed sensor locating method is validated on the 3D fuzzy controlled catalytic packed-bed reactor. The design of 3D FLC is given in Appendices A and B. We carried out two different sensor placement schemes on the 3D fuzzy controlled catalytic packed-bed reactor. One is that two to nine sensors are evenly distributed in the space domain, respectively. The other is the proposed sensor locating scheme. The control performance comparison of the two schemes is given in Table 1, where SSE, IAE, and ITAE [23] stand for steady-state error, integral of the absolute error, and integral of time multiplied by absolute error for spatiotemporal dynamic systems, respectively. Graph comparison is given in Figure 7, where eight sensors in (a) are placed in the proposed optimal locations, while eight sensors in (b) are evenly placed. In terms of Table 1 and Figure 7, we can find that the proposed sensor placement scheme has improved the control performance.

**4.2. The RTCVD System.** To acquire sufficiently rich information from the system, white noise signals, whose values are

less than 10% steady state, are added as perturbation signals to the manipulated variables  $u_a$ ,  $u_b$ , and  $u_c$ . Therefore, the manipulated variables with perturbation signals are given as follows:

$$\begin{aligned} u_{a\_excite}(t) &= 0.2028 + 0.1 * 0.2028 * \text{rand}() \\ &\quad * \text{sgn}(\text{rand}() - 0.5), \\ u_{b\_excite}(t) &= 0.1008 + 0.1 * 0.1008 * \text{rand}() \\ &\quad * \text{sgn}(\text{rand}() - 0.5), \\ u_{c\_excite}(t) &= 0.2245 + 0.1 * 0.2245 * \text{rand}() \\ &\quad * \text{sgn}(\text{rand}() - 0.5). \end{aligned} \quad (16)$$

In this application, the sample interval is 0.5 s, and the simulation duration is 100 s. The evolution profiles of manipulated variables are shown in Figure 8, and the measured wafer temperature  $y(z, t)$  is shown in Figure 9.

The space domain of the system is discretized uniformly into 61 points; that is,  $\bar{Z} = [z_1, z_2, \dots, z_{61}]$ . We collect the wafer temperature measurements from the system over the wafer radius and constitute the data set  $\mathbf{S} = [S_1 \ S_2 \ \dots \ S_{61}]$ , where  $S_j = [X_j, z_j]$ ,  $X_j = [x_{j1}, \dots, x_{j200}]$  ( $1 \leq j \leq p$ ), and  $x_{js} = y(z_j, s)$  is the wafer temperature measurement at the  $s$ th sampling period from the  $j$ th sensing location,  $1 \leq s \leq 200$ .

Then, the SVR is used to learn support vectors from the data set  $\mathbf{S}$ . As for model selection problem of the SVR, we employ 5-Fold Cross-Validation method to select proper  $C$  from  $[2^2, 2^{15}]$  with step-size set as 10,  $\varepsilon$  from  $[2^{-10}, 2^2]$  with step-size set as 0.01, and  $\sigma$  from  $\{\sigma_0/130, \sigma_0/125, \dots, \sigma_0/(n+5), \sigma_0/n, \sigma_0/(n-5), \dots, \sigma_0/10, \sigma_0/5, \sigma_0\}$ , where  $\sigma_0$  is the initial set as in (15). In this application,  $\sigma_0 = 1.2816$ .

Using 5-Fold Cross-Validation, we can find a group of best parameters; that is,  $C = 7000$ ,  $\varepsilon = 0.146$ , and  $\sigma = \sigma_0/50 = 0.00256$ . The learned support vectors are  $\mathbf{S} = \{S_5, S_{13}, S_{27}, S_{39}, S_{49}, S_{61}\}$ . Therefore, the optimal sensor locations are  $\{z_5, z_{13}, z_{27}, z_{39}, z_{49}, z_{61}\}$ ; that is,  $\{0.507, 1.52, 3.29, 4.81, 6.08, 7.60\}$ . At the same time, we can find the optimal sensor number is six.

The effectiveness of the proposed sensor placement method is validated on the space decomposition and coordination based 3D fuzzy controlled RTCVD system. The design of space decomposition and coordination based 3D FLC is given in Appendix C. We carried out two different sensor placement schemes on the 3D fuzzy controlled RTCVD system. One is that three to eleven sensors are evenly distributed in the space domain, respectively. The other is the proposed sensor placement scheme. The control performance



TABLE 2: Performance index comparisons (SVR learning based sensing locations versus evenly distributed locations).

Performance index	Sensor number with evenly distributed locations								SVR learning based sensor number (sensor location)
	3	4	5	6	7	8	9	10	6 {0.507, 1.52, 3.29, 4.81, 6.08, 7.60}
Max nonuniformity (K)	3.48	84.58	4.46	1.013	6.687	3.905	2.795	5.993	0.8620
SSE	8.352	56.34	5.49	4.37	18.42	7.77	6.038	10.16	4.2195
IAE ( $\times 10^4$ )	3.327	3.64	3.285	3.268	3.313	3.276	3.271	3.285	3.268
ITAE ( $\times 10^5$ )	1.764	4.00	1.51	1.473	1.737	1.505	1.485	1.593	1.469

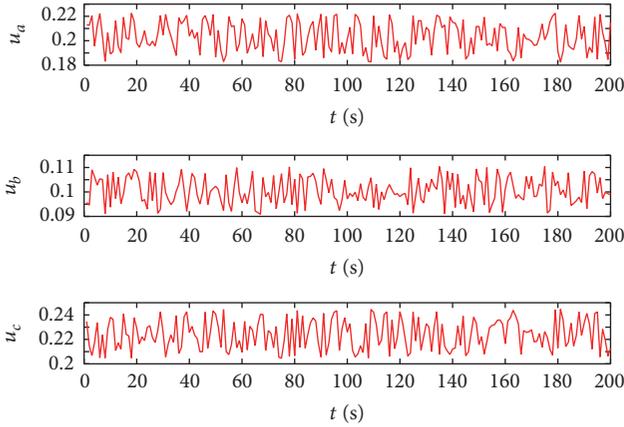
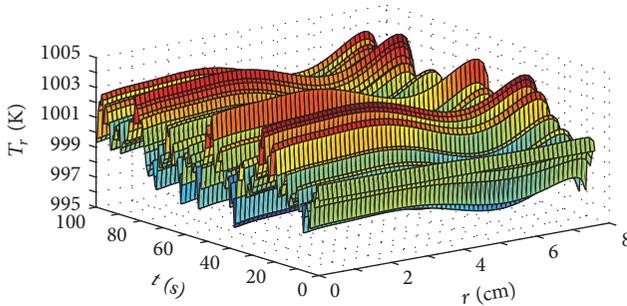
FIGURE 8: Evolution profiles of manipulated variables  $u_a$ ,  $u_b$ , and  $u_c$  with perturbed signals.

FIGURE 9: The measured wafer temperature with perturbed signals.

decomposed three subsystems are given as follows:  $u_a$  and sensing locations  $\{1/9, 2/9, 3/9, 4/9\}$  for subsystem 1,  $u_b$  and sensing locations  $\{3/9, 4/9, 5/9, 6/9, 7/9\}$  for subsystem 2, and  $u_c$  and sensing locations  $\{5/9, 6/9, 7/9, 8/9, 1\}$  for subsystem 3. In terms of Table 2 and Figure 10, we can find that the proposed sensor locating scheme has improved the control performance in the RTCVD system.

**4.3. Discussions.** According to the experiment results in the two examples, we can find that the proposed sensor placement scheme outperforms all evenly distributed sensor placement scheme. The results validate the effectiveness of the proposed sensor placement method in terms of the spatiotemporal 3D fuzzy control system.

## 5. Conclusion

The sensor placement for an unknown nonlinear SDDS is necessary for various applications (e.g., modeling and control design). In this paper, a data-based SVR learning based sensor placement method was proposed. The systematic design method is divided into three steps. Firstly, sufficiently rich data information is generated by adding persistently exciting perturbation signals and is collected as fundamental data for sensor placement. Secondly, SVR is used to extract the characteristics of spatial distribution that is expressed as support vectors. Thirdly, the learned support vectors are employed to determine optimal sensor location and sensor number. Simulations were presented to illustrate the effectiveness of this sensor placement method and its potential for a wide range of SDDSs.

## Appendix

### A. Introduction of 3D FLC

Three-dimensional fuzzy logic controller (3D FLC) [23, 24] is a novel rule-based fuzzy logic controller developed for SDDSs. It is based on a 3D fuzzy set (see Figure 11) for the spatial information and is based on a 3D inference engine to deal with spatial information. The central idea of the 3D FLC is to emulate the human operators' knowledge or expert experience to control a temperature field from the point of view of overall space domain. Similar to the traditional FLC, the 3D FLC still consists of fuzzification, rule inference, and defuzzification as shown in Figure 12. Due to its unique 3D nature, some detailed operations of the 3D FLC are different from the traditional one for spatial information expression, processing, and compression. The basic concept and basic components of the 3D FLC are introduced briefly as follows (refer to [23] for the detailed explanation).

(1) *Spatial Input Variable and 3D Fuzzy Set.* Spatial input variable, as an extension of the traditional input variable, is the function of the spatial coordinates and represents that the input information comes from the overall space domain. In actual application, finite point sensors can be used for measurement; therefore, the input information of spatial input variable comes from the sensors located in the space domain.

The 3D fuzzy set is an extension of the traditional fuzzy set by adding three coordinates for the spatial information. As shown in Figure 11, the 3D fuzzy set has three coordinates: one

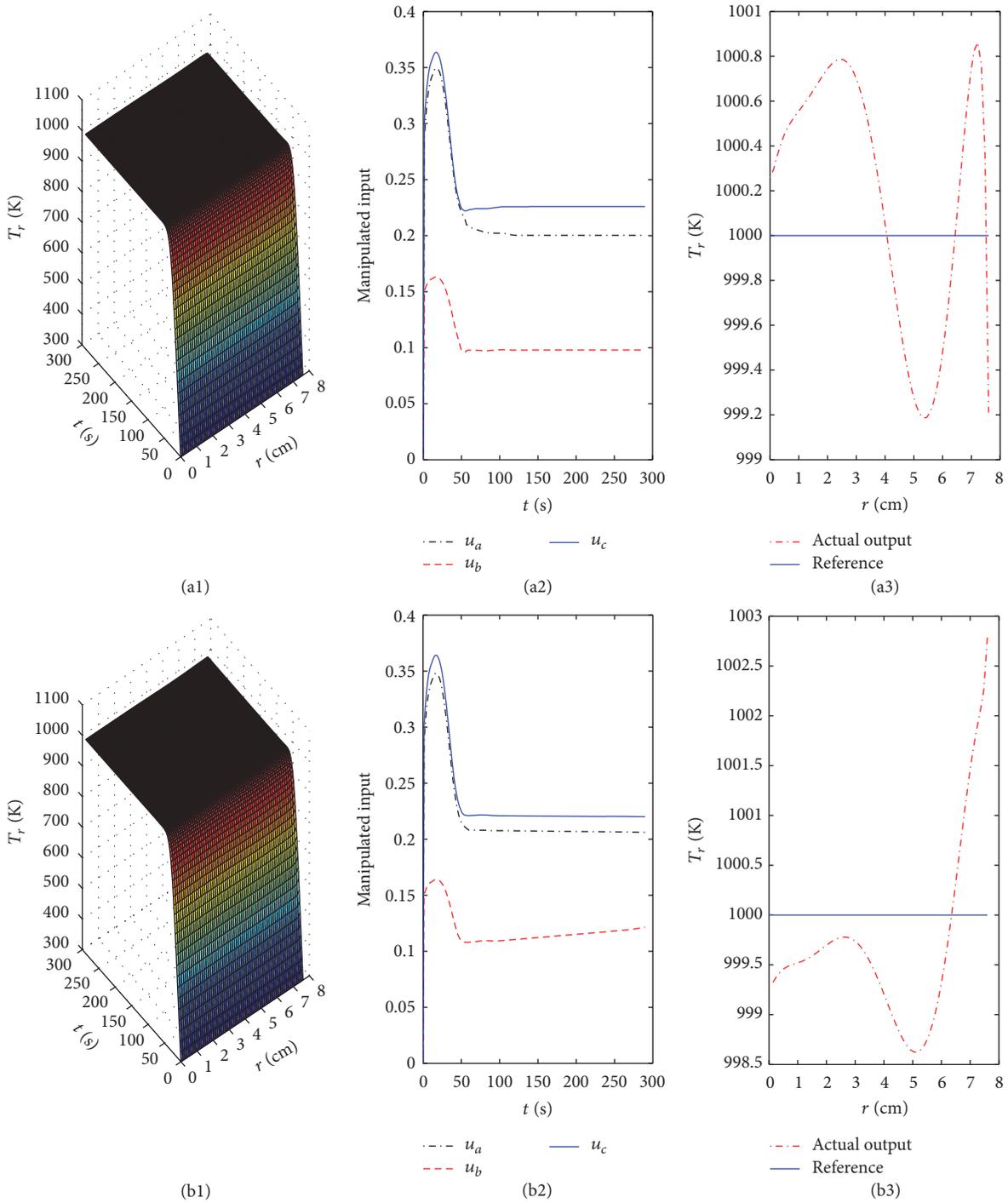


FIGURE 10: Control performance comparisons with different sensor placement schemes: (a1)–(a3) using proposed sensor locations (six optimal locations); (b1)–(b3) using nine evenly distributed sensor locations. (\*1) denotes wafer temperature evolution profile, (\*2) denotes three input signals, and (\*3) denotes wafer temperature under steady state, where \* may be (a) or (b).

is for the universe of discourse of the variable, another is for the spatial information, and the third is for the membership degree. If finite sensors are used, this 3D fuzzy set can be considered as the assembly of the traditional 2D fuzzy sets at each sensing location.

(2) *Fuzzification*. The fuzzification involving the spatial dimension will map crisp spatial inputs into 3D fuzzy inputs. There are two types of fuzzifiers: singleton fuzzifier and non-singleton fuzzifier, whose detailed definitions are given in [23]. If finite sensors are used, this 3D fuzzification can be

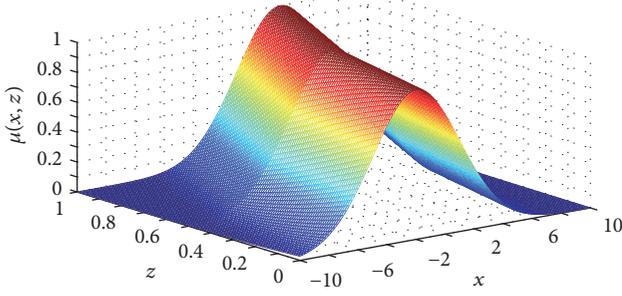


FIGURE 11: A three-dimensional fuzzy set.

considered as the assembly of the traditional 2D fuzzification at each sensing location.

### (3) Rule Inference

(i) *Rule Base.* Rules represent control strategy and human experience. For a 3D FLC with two spatial input variables, the  $l$ th rule in the rule base can be expressed as follows:

$$\text{If } e(z) \text{ is } \overline{C}_1^l \text{ and } \Delta e(z) \text{ is } \overline{C}_2^l \text{ Then } \Delta u \text{ is } G^l, \quad (\text{A.1})$$

where  $e(z)$  and  $\Delta e(z)$  denote scaled spatial input variables for error and error in change, respectively;  $\overline{C}_1^l$  and  $\overline{C}_2^l$  denote 3D fuzzy sets;  $\Delta u$  denotes the incremental control output;  $G^l$  denotes a traditional fuzzy set.

(ii) *Inference Engine.* As the kernel of 3D FLC, the inference engine is able to process spatial information and realizes two main functions: one is for overall behavior capture from spatial domain and the other is for traditional fuzzy inference. These two functions are realized in three following operations: spatial information fusion, dimension reduction, and traditional inference operation as shown in Figure 12. The spatial information fusion operation will fuse information at each spatial point and ultimately form a spatial membership distribution for each fired rule. The dimension reduction operation is to compress the 3D spatial distribution information into 2D information for each fired rule. Different method can be designed for this operation according to different control and design requirement. In this paper, a centroid approach will be employed to represent the overall behavior of the spatial domain.

(4) *Defuzzification.* After the 3D inference operation, a traditional fuzzy output is produced. Then, the traditional defuzzifier can be used to yield a crisp control action. Numerous candidates can be chosen for engineering applications, such as maximum, mean-of-maxima, centroid, center-of-sums, height, modified height, and center-of-sets.

Generally speaking, once fuzzy set and rule base are designed, the 3D FLC is determined. To reduce the complexity to tune the controller, scaling factors can be added to the inputs and output.

## B. 3D FLC Design for the Catalytic Packed-Bed Reactor

The error of spatial catalyst temperature and its error change are taken as two spatial inputs for the 3D FLC; that is,  $e^*(z) = \{e_1^*, \dots, e_p^*\}$  and  $\Delta e^*(z) = \{\Delta e_1^*, \dots, \Delta e_p^*\}$ , where  $e_i^* = T_{sd}(z_i) - T_s(z_i, k)$  and  $\Delta e_i^* = e_i^*(k) - e_i^*(k-1)$ . Let  $ke_z = \{ke_1, \dots, ke_p\}$ ,  $kr_z = \{kr_1, \dots, kr_p\}$ , and  $k_{\Delta u}$  be the spatial scaling factors of  $e^*(z)$ ,  $\Delta e^*(z)$ , and the incremental control action  $\Delta u$ , respectively. Then, the scaled error  $e(z)$  and the change error  $\Delta e(z)$  are the direct spatial inputs of 3D FLC, where  $e(z) = \{e_1, \dots, e_p\}$  with  $e_i = ke_i e_i^*$  and  $\Delta e(z) = \{\Delta e_1, \dots, \Delta e_p\}$  with  $\Delta e_i = kr_i \Delta e_i^*$ .

(1) *3D Fuzzy Set.* Since 3D fuzzy set can be regarded as the assembly of traditional 2D fuzzy set from each sensing input, the design of 3D fuzzy set is transformed to the design of 2D fuzzy set. The 2D fuzzy set for each sensing input can be chosen as triangular shape as shown in Figure 13, where each input is classified into seven linguistic labels as positive large (PL), positive middle (PM), positive small (PS), zero (O), negative small (NS), negative middle (NM), and negative large (NL).

(2) *3D Fuzzification.* Since 3D fuzzification can be regarded as the assembly of the traditional 2D fuzzification at each sensing location, the design of 3D fuzzification is transformed to the design of traditional 2D fuzzification. In this study, singleton fuzzification is used.

(3) *3D Rule Base.* The linear control rule base is used; for instance, one of the rules is expressed as “if  $e(z)$  is  $\overline{PM}$  and  $\Delta e(z)$  is  $\overline{NB}$  then  $\Delta u$  is  $\overline{NS}$ ,” where  $\overline{PM}$  and  $\overline{NB}$  are 3D fuzzy sets, which are assembled by 2D fuzzy sets  $\overline{PM}$  and  $\overline{NB}$  at each sensing location;  $\Delta u$  is the incremental control action, whose fuzzy set is triangular in shape and classified into seven linguistic labels as shown in Figure 13;  $\overline{NS}$  is the 2D fuzzy set. And the rule weight is defaulted as unity.

(4) *3D Rule Inference.* The spatial  $t$ -norm in spatial information fusion operation is chosen as “minimum.” The centroid approach is used for the dimension reduction operation. In traditional inference operation, “minimum” and “maximum” are used for the  $t$ -norm in the intersection operation and for the  $t$ -conorm in the union operation, respectively.

(5) *Defuzzification.* The center-of-sets type defuzzifier is used.

(6) *Spatial Scaling Factors.* Scaling factor for each  $e_i^*$  is set to be 1.5, the scaling factor for each  $\Delta e_i^*$  is set to be 0.5, and the scaling factor for  $\Delta u$  is 1.0.

## C. The Design of Space Decomposition and Coordination Based 3D FLC for the RTCVD System

The RTCVD system is an SDDS with three control sources, where multiple control sources have the local influence feature on the space domain. Utilizing the concept of influence

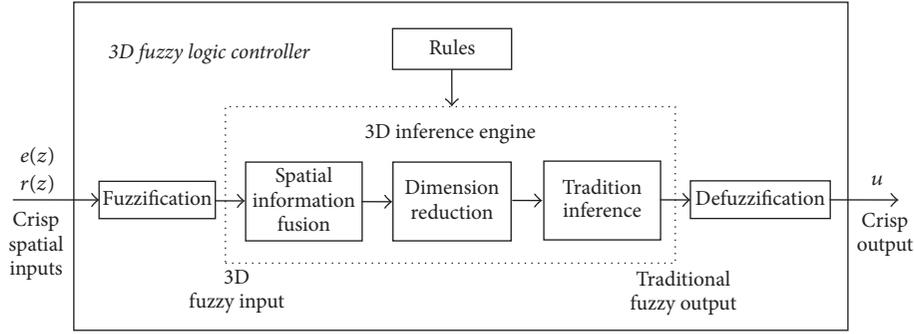
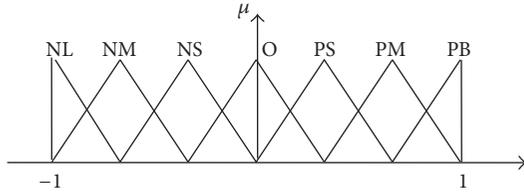


FIGURE 12: Basic structure of 3D FLC.


 FIGURE 13: Fuzzy sets for  $e_i$ ,  $\Delta e_i$ , and  $\Delta u$ .

degree [25], the space domain of the wafer along the radius can be partitioned into three subregions, and then the system is decomposed into three subsystems with one distributed control source.

For each subsystem, a 3D FLC as in Appendix B is designed based on expert experience; in order to relieve the stronger couplings among subsystems, local coordination was added to the three 3D FLCs. Scaling factors for 3D FLCs and local coordination factors among them were given as follows:

$$\begin{aligned}
 k_e^A &= 0.0006, \\
 k_r^A &= 0.006, \\
 k_u^A &= 0.13, \\
 \lambda_{BA} &= -0.1, \\
 k_e^B &= 0.0006, \\
 k_r^B &= 0.006, \\
 k_u^B &= 0.07, \\
 \lambda_{AB} &= -0.2, \\
 \lambda_{CB} &= -0.05, \\
 k_e^C &= 0.0006, \\
 k_r^C &= 0.006, \\
 k_u^C &= 0.115, \\
 \lambda_{BC} &= -0.1,
 \end{aligned} \tag{C.1}$$

where  $k_e^i$ ,  $k_r^i$ , and  $k_u^i$  ( $i = A, B$ , or  $C$ ) denote the gains for spatial error, error in change, and output for the  $i$ th 3D FLC and  $\lambda_{jk}$  ( $j, k = A, B$ , or  $C$ ) denotes the coordination factor from the  $k$ th subsystem to the  $j$ th subsystem.

For detailed introduction, one can refer to [26].

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

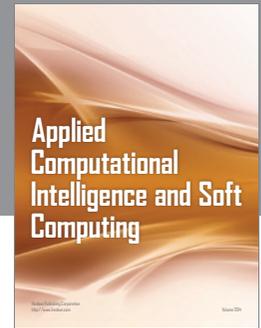
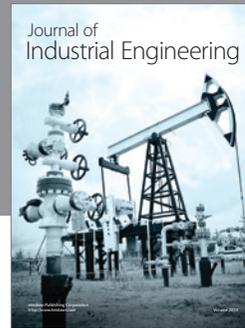
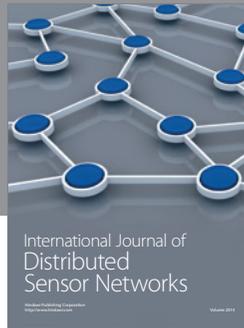
## Acknowledgments

This work was supported by the project from the National Science Foundation of China under Grants no. 61273182 and no. 31570998.

## References

- [1] P. D. Christofides, *Nonlinear and Robust Control of Partial Differential Equation Systems: Methods and Applications to Transport-reaction Processes*, Birkhäuser, Boston, Mass, USA, 2001.
- [2] C. K. Qi, H.-X. Li, S. Y. Li, X. Zhao, and F. Gao, "A fuzzy-based spatio-temporal multi-modeling for nonlinear distributed parameter processes," *Applied Soft Computing*, vol. 25, pp. 309–321, 2014.
- [3] H.-X. Li and C. K. Qi, "Modeling of distributed parameter systems for applications—a synthesized review from time-space separation," *Journal of Process Control*, vol. 20, no. 8, pp. 891–901, 2010.
- [4] A. Rensfelt, S. Mousavi, and M. Mossberg, "Optimal sensor locations for nonparametric identification of viscoelastic materials," *Automatica*, vol. 44, no. 1, pp. 28–38, 2008.
- [5] D. Ucinski, "Optimal sensor location for parameter estimation of distributed processes," *International Journal of Control*, vol. 73, no. 13, pp. 1235–1248, 2000.
- [6] A. Armaou and M. A. Demetriou, "Optimal actuator/sensor placement for linear parabolic PDEs using spatial H2 norm," *Chemical Engineering Science*, vol. 61, no. 22, pp. 7351–7367, 2006.
- [7] A. A. Alonso, I. G. Kevrekidis, J. R. Banga, and C. E. Frouzakis, "Optimal sensor location and reduced order observer design for distributed process systems," *Computers and Chemical Engineering*, vol. 28, no. 1-2, pp. 27–35, 2004.

- [8] I. Bruant, L. Gallimard, and S. Nikoukar, "Optimal piezoelectric actuator and sensor location for active vibration control, using genetic algorithm," *Journal of Sound and Vibration*, vol. 329, no. 10, pp. 1615–1635, 2010.
- [9] C. Antoniadis and P. D. Christofides, "Integrating nonlinear output feedback control and optimal actuator/sensor placement for transport-reaction processes," *Chemical Engineering Science*, vol. 56, no. 15, pp. 4517–4535, 2001.
- [10] C. Antoniadis and P. D. Christofides, "Integrated optimal actuator/sensor placement and robust control of uncertain transport-reaction processes," *Computers and Chemical Engineering*, vol. 26, no. 2, pp. 187–203, 2002.
- [11] Y. Lou and P. D. Christofides, "Optimal actuator/sensor placement for nonlinear control of the kuramoto-sivashinsky equation," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 5, pp. 737–745, 2003.
- [12] V. A. Wouwer, N. Point, S. Porteman, and M. Remy, "Approach to the selection of optimal sensor locations in distributed parameter systems," *Journal of Process Control*, vol. 10, no. 4, pp. 291–300, 2000.
- [13] E. Zamprogna, M. Barolo, and D. E. Seborg, "Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis," *Journal of Process Control*, vol. 15, no. 1, pp. 39–52, 2005.
- [14] P. Tongpadungroda, T. D. L. Rhysb, and P. N. Brett, "An approach to optimise the critical sensor locations in one-dimensional novel distributive tactile surface to maximise performance," *Sensors and Actuators, A: Physical*, vol. 105, no. 1, pp. 47–54, 2003.
- [15] X.-X. Zhang, H.-X. Li, and C.-K. Qi, "Spatially constrained fuzzy-clustering based sensor placement for spatio-temporal fuzzy-control system," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 946–957, 2010.
- [16] R. N. Silva, J. M. Lemos, and L. M. Rato, "Variable sampling adaptive control of a distributed collector solar field," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 5, pp. 765–772, 2003.
- [17] P. D. Christofides, "Robust control of parabolic PDE systems," *Chemical Engineering Science*, vol. 53, no. 16, pp. 2949–2965, 1998.
- [18] J. J. Winkin, D. Dochain, and P. Ligarius, "Dynamical analysis of distributed parameter tubular reactors," *Automatica*, vol. 36, no. 3, pp. 349–361, 2000.
- [19] K. A. Hoo and D. Zheng, "Low-order control-relevant models for a class of distributed parameter systems," *Chemical Engineering Science*, vol. 56, no. 23, pp. 6683–6710, 2001.
- [20] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [21] R. Haber and L. Keviczky, *Nonlinear System Identification—Input-Output Modeling Approach, Volume 1: Nonlinear System Parameter Identification*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.
- [22] W. E. Schiesser, *The Numerical Methods of Lines Integration of Partial Differential Equations*, Academic Press, San Diego, Calif, USA, 1991.
- [23] 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.
- [24] 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.
- [25] C. B. Kellogg and F. Zhao, "Influence-based model decomposition for reasoning about spatially distributed physical systems," *Artificial Intelligence*, vol. 130, no. 2, pp. 125–166, 2001.
- [26] X.-X. Zhang, S. Y. Li, and H.-X. Li, "Decomposition-coordination-based fuzzy logic control for spatially distributed systems," *Control and Decision*, vol. 23, no. 6, pp. 709–713, 2008.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

