New Fuzzy Inference System Using a Kernel-based Method

Jongcheol Kim*, Sangchul Won**, and Yasuo Suga*

*Graduate School of Science and Technology, Keio University, Yokohama, Japan
(Tel : +81-45-563-1141; E-mail: suga@mech.keio.ac.kr)

**Division of Electrical and Computer Engineering, Pohang University of Science and Technology, Pohang, Korea
(Tel: +82-54-279-5066; Email:won@postech.ac.kr)

Abstract: In this paper, we proposes a new fuzzy inference system for modeling nonlinear systems given input and output data. In the suggested fuzzy inference system, the number of fuzzy rules and parameter values of membership functions are automatically decided by using the kernel-based method. The kernel-based method individually performs linear transformation and kernel mapping. Linear transformation projects input space into linearly transformed input space. Kernel mapping projects linearly transformed input space into high dimensional feature space. The structure of the proposed fuzzy inference system is equal to a Takagi-Sugeno fuzzy model whose input variables are weighted linear combinations of input variables. In addition, the number of fuzzy rules can be reduced under the condition of optimizing a given criterion by adjusting linear transformation matrix and parameter values of kernel functions using the gradient descent method. Once a structure is selected, coefficients in consequent part are determined by the least square method. Simulated result illustrates the effectiveness of the proposed technique.

Keywords: Fuzzy inference system, kernel-based method, linear transformation, gradient descent method

1. INTRODUCTION

A Fuzzy Inference System (FIS) has been shown powerful capability for the modeling of nonlinear systems [1] [2]. FISs can be directly obtained either from human experts using knowledge experiments or learning machine methods using numeric data. For complex and uncertain systems, FISs based only on human experts may not lead to sufficient accuracy. Because of this reason, neuro-fuzzy modeling which acquires knowledge from a set of input-output data has been actively investigated [3]. The important concerns of neuro-fuzzy modeling for the real system are how to determine proper the number of fuzzy rules and parameter values of membership functions. To decide the number of fuzzy rules from data is associated with how to partition input space. There are usually two kinds of group methods. The first one involves conventional partition. Though this partition methods have good motivations, those have disadvantages which include the curse of input dimensionality [5], the exponential increase in the number of rules [6], unpredictable completeness [7] and computation cost [9]. The other one is a clustering method. In clustering techniques, the number of clusters must be known in advance [10], or previously settled grid points of grid lines can be candidates for cluster centers [4].

Recently kernel methods have popularly developed in classification and regression. Kernel techniques offer an alternative solution by mapping the data into high dimensional feature space to increase the computational power. Particularly, Support Vector Machine (SVM)[11] has been used in order to automatically find the number of network nodes or fuzzy rules based on given error bound [13] [15] [17]. The Support Vector Neural Network (SVNN) is proposed to select the best structure of radial based function network for the given precision [13]. The SVM is suggested to improve the simplified fuzzy inference system for the fuzzy neural network [15]. The Support Vector Fuzzy Inference System is proposed to find the reduced number of rules using gradient descent method updating kernel parameters [17]. However, because the general support vector learning methodology is used in above all, they have computational complexity for solving the quadratic problem in optimization process and problem determining the type of kernel function corresponding with nonlinear system.

In this paper, to overcome these drawbacks, we propose a new fuzzy inference system using a kernel-based method. The linear transformation of input variables is used to solve problem determining the exact type of the kernel function. Therefore input variables of the proposed FIS become input variables of the Takagi-Sugeno (TS) fuzzy model which are weighted linear combinations of the input variables. The structure of fuzzy model is obtained using Feature Vector Selection (FVS) [16] algorithm based on the kernel method. Unlikely the SVM having computational complexity, the FVS performs a simple computation optimizing a given criterion into the feature space. The FVS algorithm is to select a basis of the data subspace in feature space. A basis of the data subspace is called a $feature\ vector$. Ultimately, this feature vector becomes the center of the membership function. Kernel functions mapping the linearly transformed data into feature space become membership functions. In addition, the number of fuzzy rules can be reduced under the condition of optimizing a given criterion by adjusting the linear transformation matrix and parameter values of kernel functions using the gradient descent method. Once a structure is selected, coefficients in consequent part of the modified TS fuzzy model are determined by the least square method. So we can automatically determine the fuzzy model using the iterative procedure which involves linear transformation, kernel mapping and FVS method under optimizing a given criterion.

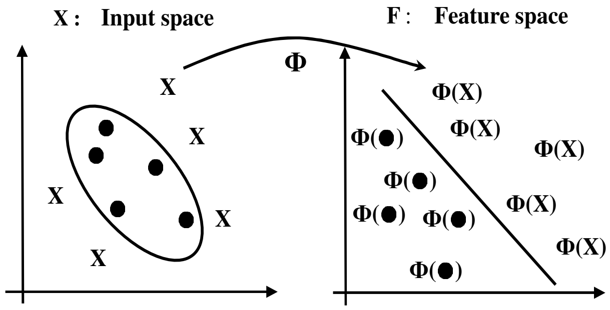

Fig. 1. Nonlinear mapping.

The rest of this paper is organized as follows. General kernel method and FVS algorithm are presented in Sect. 2. The structure and learning algorithm of the new FIS using a kernel-based method are given in Sect. 3. Simulated result of the proposed FIS is illustrated by example involving benchmark nonlinear system in Sect. 4. Conclusion is given in Sect. 5.

2. PRELIMINARIES

In this section, the reviews of the general kernel method and the FVS algorithm are presented in order to understand a kernel-based method.

2.1. Kernel method

Kernel method performs a nonlinear mapping which projects input space into high dimensional feature space. Generally, preprocessing step in learning machine contains representation of given input-output data [14]:

$$\mathbf{x} = (x_1, ..., x_n) \;\mapsto\; \Phi(\mathbf{x}) = (\Phi_1(\mathbf{x}), ..., \Phi_n(\mathbf{x})).$$

This step is equivalent to mapping the input space X into a new space, $F = \{\Phi(\mathbf{x})|\mathbf{x} \in X\}$. Projecting the given data into hypothesis space can not only increase computational power in learning machine but also supply various methods for extracting relevant information through new representation of data. The quantities introduced to describe the data are called *features*. The work of selecting the best suitable representation is known as the *feature selection*. The space X is referred to as the *input space*, while $F = \{\Phi(\mathbf{x})|\mathbf{x} \in X\}$ is called the *feature space* [14].

Figure 1 shows an example of the nonlinear mapping which projects the training data from input space to a higher-dimensional feature space via Φ. In input space, data can not be separated by linear function, instead of being able to be in the feature space. Now, we will present the definition and the characteristic of the kernel.

Definition 21: [14] A kernel is a function K, such that for all $\mathbf{x},\ \mathbf{z} \in X$

$$K(\mathbf{x},\mathbf{z}) \quad = \quad < \Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) > \tag{1}$$

where Φ is a mapping from X to an (inner product) feature space F.

Following Mercer's theorem provides a characterization when a function $K(\mathbf{x},\mathbf{z})$ is a kernel.

Theorem 21: [14] Let X be a compact subset of $\mathbb{R}^n$. Suppose K is a continuous symmetric function such that the integral operator $T_k : L_2(X) \to L_2(X)$,

$$T_k f(\cdot) := \int_X K(\cdot, \mathbf{x}) f(\mathbf{x}) d\mathbf{x} \tag{2}$$

is positive. That is

$$\int_{X\times X} K(\mathbf{x},\mathbf{z}) f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z} \geq 0, \tag{3}$$

for all $f \in L_2(X)$. Then we can expand $K(\mathbf{x},\mathbf{z})$ in a uniformly convergent series (on $X \times X$) in terms of T_k's eigen-functions $\Phi_j \in L_2(X)$, normalized in such a way that $\|\Phi_j\|_{L_2} = 1$, and positive associated eigenvalue $\lambda_j \geq 0$,

$$K(\mathbf{x},\mathbf{z}) = \sum_{j=1}^{\infty} \lambda_j \Phi_j(\mathbf{x}) \Phi_j(\mathbf{z}). \tag{4}$$

From these definition and theorem, we can summary kernel function as follows,

$$K(\mathbf{x},\mathbf{z}) =< \Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) >= \sum_{i=1}^{\infty} \lambda_i \Phi_i(\mathbf{x}) \Phi_i(\mathbf{z}). \tag{5}$$

An example in [11] gives brief understanding.

Example (Quadratic feature in $\mathbb{R}^2$) Consider the map $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ with

$$\Phi(\mathbf{x}) = \Phi(x_1, x_2) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2), \tag{6}$$

where x_1 and $x_2 \in \mathbb{R}^2$, for instance, the polynomial kernel $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})^d$. For $d = 2$, and $\mathbf{x},\mathbf{y} \in \mathbb{R}^2$, we have

$$\begin{aligned} (\mathbf{x}\cdot\mathbf{y})^2 &= \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right)^2, \\ &= \left(\begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix} \cdot \begin{pmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{pmatrix} \right), \\ &= (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})). \end{aligned} \tag{7}$$

Table 1 shows some kernel functions commonly used.

2.2. Feature vector selection

The FVS [16] is based on kernel method. The FVS technique is to select feature vector being a basis of data subspace and capturing the structure of the entire data into feature space F.

The FVS for estimating the mapping $\hat{\phi}_i$ of any vector x_i is as follows:

$$\hat{\phi}_i = \Phi_S \cdot \mathbf{a}_i, \tag{8}$$

where the mapping of each vector x_i is noted $\phi(x_i) = \phi_i$ for $1 \leq i \leq M$, the selected vectors x_{s_j} into feature space F is noted $\phi(x_{s_j}) = \phi_{S_j}$ for $1 \leq j \leq L$, $\Phi_S = \{\phi_{S_i}, ..., \phi_{S_L}\}$ is the matrix of the selected vectors $S = \{x_{s_1}, ..., x_{s_L}\}$ into F and $\mathbf{a}_i = [a_i^1, ..., a_i^L]^T$ is the associated weight vector.

The feature vector is obtained from process finding the weights vector $\mathbf{a}_i$. The weights vector is given by minimizing the following normalized Euclidean distance in feature space.

$$\delta_i = \frac{\|\phi_i - \hat{\phi}_i\|^2}{\|\phi_i\|^2}. \tag{9}$$

Table 1. Kernel function and type.

Kernel Function	Type
$K(\mathbf{x},\mathbf{y}) \;=\; ((\mathbf{x}\cdot\mathbf{y})+1)^d$	Polynomial of degree d
$K(\mathbf{x},\mathbf{y}) \;=\; exp(-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2})$	Gaussian RBF
$K(\mathbf{x},\mathbf{y}) \;=\; exp(-\frac{\lvert\mathbf{x}-\mathbf{y}\rvert}{2\sigma^2})$	Exponential RBF
$K(\mathbf{x},\mathbf{y}) \;=\; tanh(a(\mathbf{x}\cdot\mathbf{y})-b)$	Multi-layer perceptron
$K(\mathbf{x},\mathbf{y}) \;=\; \frac{sin(N+\frac{1}{2})(\mathbf{x}-\mathbf{y})}{sin(\frac{1}{2}(\mathbf{x}-\mathbf{y}))}$	Fourier series

The minimum of Eq. (9) for a given S can be expressed over all vector as follows:

$$\min_{S} \sum_{x_i \in X} \left(1 - \frac{K_{si}^t K_{ss}^{-1} K_{si}}{K_{ii}}\right) \tag{10}$$

where $K_{ss} =< \Phi_S \cdot \Phi_S >$ is a kernel function which is the dot product of the selected vectors, $K_{si} =< \Phi_S \cdot \phi_i >$ is a kernel function which is the dot product of between x_i and the selected vectors and $K_{ii} =< \phi_i \cdot \phi_i >$ is a kernel function which is the dot product of x_i.
The fitness function is defined as follows:

$$J_S = \frac{1}{M} \sum_{x_i \in X} \left(\frac{K_{si}^t K_{ss}^{-1} K_{si}}{K_{ii}}\right). \tag{11}$$

Thus Eq. (11) can be written by

$$\max_{S} \; J_S, \tag{12}$$

where $\max_S J_S$ is a value between 0 and 1 for $x_i \in S$.
The FVS algorithm is an iterative process which performs sequential forward selection until the fitness reaches a given value.

3. NEW FUZZY INFERENCE SYSTEM USING A KERNEL-BASED METHOD

This section describes the structure and learning algorithm of a new fuzzy inference system using a kernel-based method.

3.1. The structure of the FIS using a kernel-based method

The kernel-based method is that linear transformation is added to kernel mapping in order to solve the problem selecting the type of kernel function related to nonlinear system, Thus input variables of the proposed FIS become input variables of the TS fuzzy model which are weighted linear combinations of original input variables.
Suppose we have given input and output data

$$(\mathbf{x}_1, \mathbf{y}_1), \; ..., \; (\mathbf{x}_l, \mathbf{y}_l) \tag{13}$$

where $\mathbf{x}_i$=$[x_1^i, x_1^i, ..., x_D^i]^T (i = 1, 2, ..., l)$ is original input variable and Y=$[\mathbf{y}_1, ..., \mathbf{y}_l]^T$ is output variable. The proposed TS fuzzy model with fuzzy if-then rules can be represented by Eq. (14).

$$\begin{aligned}
R_1 \;\; &: \;\; \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}_{11}^*) \text{ and ... } \bar{x}_D \text{ is } K(\bar{x}_D, \bar{x}_{1D}^*), \\
&\;\;\;\; \text{Then } f_1 = a_{10} + a_{11}\bar{x}_1 + ... + a_{1D}\bar{x}_D \\
R_2 \;\; &: \;\; \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}_{21}^*) \text{ and ... } \bar{x}_D \text{ is } K(\bar{x}_D, \bar{x}_{2D}^*), \\
&\;\;\;\; \text{Then } f_2 = a_{20} + a_{21}\bar{x}_1 + ... + a_{2D}\bar{x}_D \\
&\;\;\;\; \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
R_n \;\; &: \;\; \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}_{n1}^*) \text{ and ... } \bar{x}_D \text{ is } K(\bar{x}_D, \bar{x}_{nD}^*), \\
&\;\;\;\; \text{Then } f_n = a_{n0} + a_{n1}\bar{x}_1 + ... + a_{nD}\bar{x}_D,
\end{aligned} \tag{14}$$

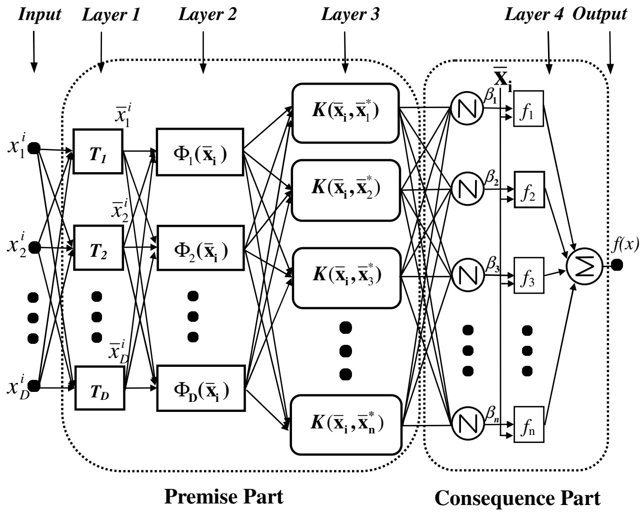

Fig. 2. The structure of the proposed fuzzy inference system.

where n is the number of fuzzy rules, D is the dimension of input variables, $\bar{x}_j (j = 1, 2, ..., D)$ is a linearly transformed input variable, f_i is a local output variable, $K(\bar{x}_j, \bar{x}_{ij}^*)(i = 1, 2, ..., n, j = 1, 2, ..., D)$ is a fuzzy set and $a_{ij}(i = 1, 2, ..., n, j = 0, 1, ..., D)$ is a consequent parameter.
Linearly transformed input variables are defined as follows:

$$\begin{bmatrix} \bar{x}_1^i \\ \bar{x}_2^i \\ \vdots \\ \bar{x}_D^i \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1D} \\ t_{21} & t_{22} & \dots & t_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ t_{D1} & t_{D2} & \dots & t_{DD} \end{bmatrix} \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_D^i \end{bmatrix} \tag{15}$$

where $\overline{\mathbf{x}}_i$=$[\bar{x}_1^i, \bar{x}_2^i, ..., \bar{x}_D^i]^T (i$=$1, 2, ..., l)$ is a linearly transformed input variable, and T_i=$[t_{i1}, t_{i2}, ..., t_{iD}](i = 1, 2, ..., D)$ is the ith transformed direction unit vector of the original input space.
Now, we describe the structure of FIS using a kernel-based method. It consists of four layers as shown in Fig. 2.
The four layers involved in the proposed FIS are as follows:
Layer 1: Input space is projected into a linearly transformed input space by a linearly transformation matrix.

$$\overline{\mathbf{x}}_i = T\mathbf{x}_i, \quad i = 1, 2, ..., l \tag{16}$$

where T=$[T_1, T_2, ..., T_D]^T$ is a linear transformation matrix.

Layer 2: Linearly transformed input space is nonlinearly mapped into feature space by a map Φ.

$$\begin{aligned}
\overline{\mathbf{x}}_i \;&=\; (\bar{x}_1^i, ..., \bar{x}_D^i) \;\; \mapsto \\
\Phi(\overline{\mathbf{x}}_i) \;&=\; (\Phi_1(\overline{\mathbf{x}}_i), ..., \Phi_D(\overline{\mathbf{x}}_i)), i = 1, 2, ..., l.
\end{aligned} \tag{17}$$

Layer 3: Feature Vector (FV) is determined from a FVS algorithm a using kernel method. Kernel method is

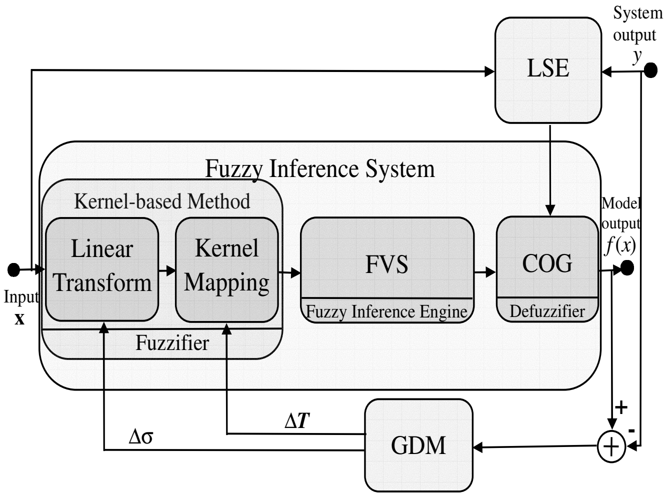

Fig. 3. The learning algorithm of the proposed FIS.

a dot product which is computed with the nonlinear mapped input $\Phi(\overline{\mathbf{x}}) = (\Phi(\overline{\mathbf{x}}_1), ..., \Phi(\overline{\mathbf{x}}_l))$ and feature vector $\Phi(\overline{\mathbf{x}}_i^*) = (\Phi_1(\overline{\mathbf{x}}_i^*), ..., \Phi_D(\overline{\mathbf{x}}_i^*))(i = 1, ..., n)$, where $\overline{\mathbf{x}}_i^* = [\bar{x}_{i1}^*, \bar{x}_{i1}^*, ..., \bar{x}_{iD}^*]^T$ is the subset of the input $\overline{\mathbf{x}}$. Dot product $\Phi(\overline{\mathbf{x}}) \cdot \Phi(\overline{\mathbf{x}}_i^*)$ corresponds to evaluating kernel function $K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*)$. The Gaussian kernel function with each variance σ_i is used as follows:

$$K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*) = exp\bigg(- \frac{(\overline{\mathbf{x}} - \overline{\mathbf{x}}_i^*)^2}{2\sigma_i^2} \bigg), i = 1, 2, ..., n \tag{18}$$

where $\overline{\mathbf{x}}_i^*$ is a FV, σ_i is called a kernel parameter and n is the number of FVs. This kernel function become a Gaussian membership function in the proposed FIS. $\overline{\mathbf{x}}_i^*$ and σ_i is respectively the center and the variance of the ith Gaussian membership function. FVS algorithm is a fuzzy inference engine determining the number of fuzzy rules in the FIS. The Layer 1 to 3 are related to the premise part of the FIS.

Layer 4: For the overall output of the fuzzy model constructed, defuzzification using the Center Of Gravity (COG) method is also performed.

$$\begin{aligned} f(\mathbf{x}) &= \frac{\sum_{i=1}^{n} K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*) \, f_i}{\sum_{j=1}^{n} K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_j^*)}, \\ &= \sum_{i=1}^{n} \beta_i \, (a_{i0} + a_{i1}\bar{x}_1 + \cdots + a_{iD}\bar{x}_D) \end{aligned} \tag{19}$$

where $\beta_i = \frac{K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*)}{\sum_{j=1}^{n} K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_j^*)}$ and $f_i = (a_{i0} + a_{i1}\bar{x}_1 + \cdots + a_{iD}\bar{x}_D)$. It is assumed that $K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_i^*) \geq 0$, $\sum_{j=1}^{n} K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_j^*) > 0$. Therefore, $0 \leq \beta_i \leq 1$, $(i = 1, 2, ..., n)$.
The Layer 4 connects with the consequence part of the FIS.

3.2. The learning algorithm of the FIS using a kernel-based method

The learning algorithm of the FIS using a kernel-based method is shown in Fig. 3. It can be achieved by the following iterative procedure.
Step 1: Assign the desired fitness and initialize the linear transformation matrix T and the kernel parameter σ_i.

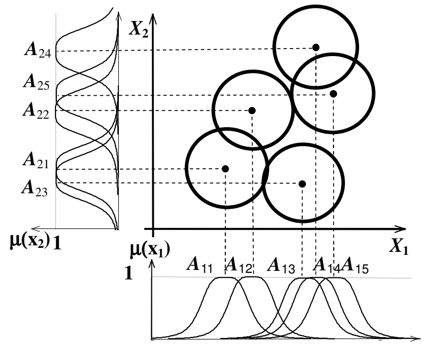

Fig. 4. The input space partitioning of the FVS method.

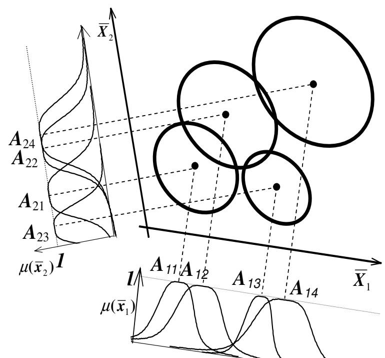

Fig. 5. The input space partitioning of the proposed FIS.

Step 2: Perform linear transformation in Eq.(16) in order to project input space into linearly transformed input space.

Step 3: Using the following FVS algorithm based on kernel mapping, find FVs $\overline{\mathbf{x}}_i^*$ that are the centers $\mathbf{c}_i$ of Gaussian membership functions.

$$\max_S J_S = \max_S \frac{1}{M} \sum_{\overline{\mathbf{x}}_i \in F} \bigg(\frac{K_{si}^t K_{ss}^{-1} K_{si}}{K_{ii}} \bigg). \tag{20}$$

Step 4: Using the following Least Square Estimation (LSE) method [1], estimate the parameter a_{ij} of the linear equation f_i in Eq. (19).
Let

$$A = \Big[\; a_{10} \; a_{11} \; ... \; a_{1D} \; ... \; a_{n0} \; a_{n1} \; ... \; a_{nD} \; \Big]^T, \tag{21}$$

$$W = \begin{bmatrix} \beta_1^1 & \beta_1^1\overline{x}_1^1 & ... & \beta_1^1\overline{x}_D^1 & ... & \beta_n^1 & \beta_n^1\overline{x}_1^1 & ... & \beta_n^1\overline{x}_D^1 \\ \beta_1^2 & \beta_1^2\overline{x}_1^2 & ... & \beta_1^2\overline{x}_D^2 & ... & \beta_n^2 & \beta_n^2\overline{x}_1^2 & ... & \beta_n^2\overline{x}_D^2 \\ \vdots & \vdots & ... & \vdots & ... & \vdots & \vdots & ... & \vdots \\ \beta_1^l & \beta_1^l\overline{x}_1^l & ... & \beta_1^l\overline{x}_D^l & ... & \beta_n^l & \beta_n^l\overline{x}_1^l & ... & \beta_n^l\overline{x}_D^l \end{bmatrix} \tag{22}$$

where $\beta_i^j = \frac{K(\overline{\mathbf{x}}_j, \overline{\mathbf{x}}_i^*)}{\sum_{k=1}^{n} K(\overline{\mathbf{x}}, \overline{\mathbf{x}}_k^*)}$. Thus fuzzy model output is $f(\mathbf{x}) = WA$.
If $(W^T W)$ is nonsingular, the parameter vector A is calculated by

$$A = (W^T W)^{-1} W^T Y. \tag{23}$$

Step 5: Using a Gradient Descent Method (GDM) [4], update the kernel parameter σ_i such that error is minimized.

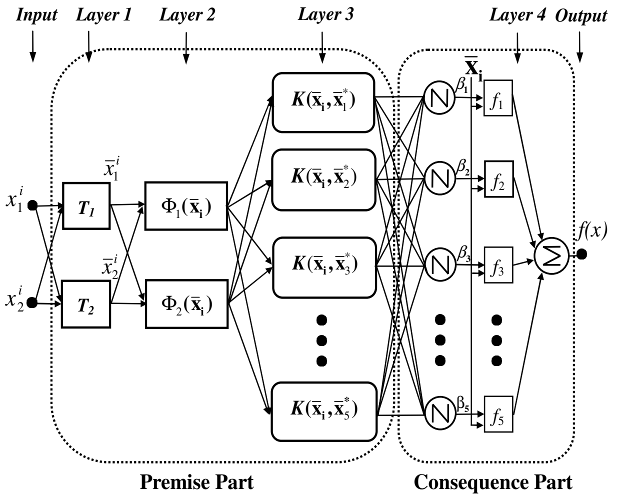

Fig. 6. The structure of the FIS for the modeling of $F(x_1, x_2)$.

From the definition of the GDM,

$$\begin{aligned}\Delta\sigma_i &= -\eta_\sigma \nabla_{\sigma_i} E,\\ &= -2\eta_\sigma \sigma_i^{-3} \sum_{j=1}^{l} e_j \beta_i^j (f_i - y_j) \|\overline{\mathbf{x}}_j - \overline{\mathbf{x}}_i^*\|^2 \qquad (24)\end{aligned}$$

where η_σ is the learning late of σ_i, $e_j = f(\mathbf{x}_j) - y_j$ and $E = \sum_{j=1}^{l} e_j^2$.

Step 6: Also using the following GDM, update the linear transformation matrix T and go to step 2 until error and FVs are satisfied with given conditions.

$$\begin{aligned}\Delta T &= -\eta_T \nabla_T E,\\ &= -2\eta_T \sum_{j=1}^{l} e_j \overline{\mathbf{x}}_j \sum_{i=1}^{n} \beta_i^j,\\ &\quad \left[A_i + \|\overline{\mathbf{x}}_j - \overline{\mathbf{x}}_i^*\| \sigma_i^{-2} (y_j - f_i)\right] \qquad (25)\end{aligned}$$

where η_T is the learning late of T and $A_i = [a_{i1}, ..., a_{iD}]^T$. Figures 4 and 5 show input space partitioning methods of two-dimensional input space. The former describes the input space partitioning of the FVS method with the same Gaussian variance. The latter presents that of the proposed FIS using a kernel-based method with linear transformation and different variances. Compared with the former having five rules, Figure 5 with four rules shows that the number of fuzzy rules can be reduced as determining the appropriate linear transformation matrix and Gaussian variances using a GDM.

4. EXMAPLE

In this section, we show simulation result of the proposed FIS for the modeling of the typical nonlinear system.
To analyze the performance of the proposed FIS, the modeling error is defined by as following Root Mean Square Error (RMSE)

$$E = \sqrt{\frac{\sum_{k=1}^{N} (y_k - f(x_k))^2}{N}} \qquad (26)$$

Table 2. Parameter values of the FIS for modeling of $F(x_1, x_2)$.

Rule	Premise part		Consequent part
	c_i	σ_i	(a_{i0}, a_{i1}, a_{i2})
1	(2.4151, 2.4151)	3.3854	(270946,-18283, -10464)
2	(4.7757, 5.0076)	3.1297	(9966,-55,-373)
3	(1.2561, 4.5498)	3.0863	(-8620,623 ,484)
4	(4.3525, 1.5288)	3.1524	(-1059,1072,1133)
5	(1.2275, 1.5110)	3.3572	(-195313,-7695,-5205)

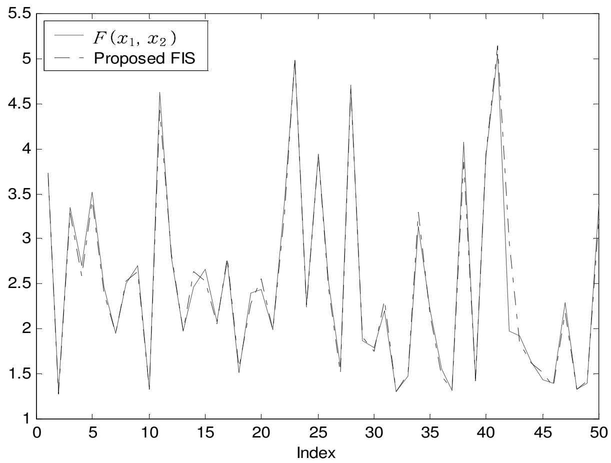

Fig. 7. The modeling result of $F(x_1, x_2)$.

where N is the number of data, y_k and $f(x_k)$ are respectively the system and the model output.

4.1. Example : modeling of 2-input nonlinear function

Consider the nonlinear function [2]

$$F(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5})^2. \qquad (27)$$

From input ranges $[1, 5] \times [1, 5]$ of Eq. (27), 50 training data pairs were obtained. The proposed FIS extracts the 5 FVs, so that it has 5 rules as follows,

$$\begin{aligned}R_i \;:\;& \text{If } \bar{x}_1 \text{ is } K(\bar{x}_1, \bar{x}_{i1}^*) \text{ and } \bar{x}_2 \text{ is } K(\bar{x}_2, \bar{x}_{i2}^*),\\ & \text{Then} f_i = a_{i0} + a_{i1}\bar{x}_1 + a_{i2}\bar{x}_2, i = 1, ..., 5. \quad (28)\end{aligned}$$

The structure of the FIS with 5 rules is shown in Fig. 6.
For given the fitness of $\max_S J_S = 0.992$ and the initial condition of $\sigma_i = 3.2$, the linear transformation matrix T, the center c_i and the variance σ_i of the Gaussian membership function in premise part and coefficients a_{ij} in consequence part are obtained through learning procedure. The linear transformation matrix is calculated as follows:

$$T = \begin{bmatrix} 0.9998 & 0.0001 \\ 0.0001 & 1.0002 \end{bmatrix}. \qquad (29)$$

The parameter values of premise and consequent parts are listed in Table 2. The method in the literature applied to the same function $F(x_1, x_2)$, and the results listed on the Table 3. Compared with the number of rules and modeling error

Table 3. Compared results of nonlinear function $F(x_1, x_2)$.

Type	Rules(or FVs)	RMSE
Sugeno and Yasukawa [2]	6	0.281
Gomez-Skarmeta et al. [8]	5	0.266
Chan et al. [13]	6	0.324
Baudat et al. [16]	6	0.333
Kim et al. [17]	5	0.171
Proposed FIS	5	0.164

of others, the proposed method gives the smallest modeling error. Figure 7 shows the modeling result of $F(x_1, x_2)$ using a kernel-based method.

5. CONCLUSION

In this paper, we have introduced a new fuzzy inference system using a kernel-based method. Our main concern is to determine the best structure of the TS fuzzy model for modeling nonlinear systems with measured input and output data. The number of rules and the parameter values of membership functions in the proposed FIS can be decided using an iterative FVS based on kernel-based method. The kernel-based method involves linear transform and kernel mapping. The linear transformation matrix and parameter values of kernel functions were adjusted using the gradient descent method. Coefficients in consequent part of the TS fuzzy model were determined by the least square method. Example showed the effectiveness of the proposed FIS for the modeling of nonlinear system.

References

[1] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications for modeling and control," *IEEE Trans. Syst., Man. Cybern.*, Vol. 15, No. 1, pp. 116-132, 1985.

[2] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. on Fuzzy Sys.*, Vol. 1, No. 1, pp. 7-31, 1993.

[3] J.R. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst., Man. Cybern.*, Vol. 23, No. 3, pp. 665-685, 1993.

[4] R.R. Yager and D.P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst., Man. Cybern.*, Vol. 24, No. 8, pp. 1279-1284, 1994.

[5] H. Ishibuchi, K. Nozaki, H. Tanaka, Y. Hosaka, and M. Matsuda, "Empirical study on learning in fuzzy systems by rice test analysis," *Fuzzy Sets and System*, Vol. 64, pp. 120-144, 1994.

[6] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and regression trees,* Wadsworth, Inc., Belmont, Califonia, 1984.

[7] L.X. Wang and J.M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man. Cybern.*, Vol. 22, No. 6, pp. 1414-1427, 1992.

[8] A.F. Gomez-Skarmeta, M.Delgado, M.A. Vila, "About the use of fuzzy clustering techniques for fuzzy model identification," *Fuzzy Sets and Systems*, Vol. 106, pp. 179-188, 1999.

[9] C.T. Sun, and J.S. Jang, "A neuro-fuzzy classifier and its applications," *Proc. IEEE Int. Conf. Fuzzy Syst.* San Francisco, Vol. 1, pp. 94-98, 1993.

[10] Y.H. Joo, H.S. Hwang and K.B. Kim, "Fuzzy system modeling by fuzzy partition and GA hybrid schemes," *Fuzzy Sets and System*, Vol. 86, pp. 279-288, 1997.

[11] V. Vapnik, *The Nature of Statistical Learning Theory*, New York : Spring-Verlag, 1995.

[12] S.R. Gunn, *Support Vector Machines for Classification and Regression*, Image Speech and Intelligent Systems Research Group, University of Southampton, 1998.

[13] W.C. Chan, W.C. Chan, K.C. Cheung and C.J. Hariss, "On the modeling of nonlinear dynamic systems using support vector neural networks," *Engineering Application of Artifical Intelligence*, Vol. 14, pp. 105-113, 2001.

[14] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[15] J.T. Jeng and T.T. Lee, "Support vector machines for the fuzzy neural networks," *IEEE SMC '99 Conference Proceedings*, Vol. 6, pp. 115-120, 1999.

[16] G. Baudat and F. Anouar, "Kernel-based methods and function approximation," *In International Joint Conference on Neural Networks* , pp. 1244-1249, 2001.

[17] J. Kim and S. Won, "New fuzzy inference system using a support vector machine," *Proc. 41th IEEE Conference of Decision and Control*, Las Vegas, USA, pp. 1349-1354, Dec., 2002.