International Journal of Innovative
Computing, Information and Control
Volume **7**, Number **3**, March **2011**

ICIC International ©2011 ISSN 1349-4198
pp. 1133–1144

ENCODING PRIOR KNOWLEDGE INTO DATA DRIVEN DESIGN OF INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS

Chengdong Li[1,2], Jianqiang Yi[2] and Tiechao Wang[2]

[1]School of Information and Electrical Engineering
Shandong Jianzhu University
Fengming Road, Lingang Development Zone, Jinan 250101, P. R. China
chengdong.li@ia.ac.cn

[2]Institute of Automation
Chinese Academy of Sciences
Beijing 100190, P. R. China

Received November 2009; revised March 2010

Abstract. *In system identification or modeling problems, interval type-2 fuzzy logic systems (IT2FLSs), which have obvious advantages for handling different sources of uncertainties, are usually constructed only using the information from sample data. This paper tries to utilize the information from both sample data and prior knowledge to design IT2FLSs to compensate the insufficiency of the information from single knowledge source. First, sufficient conditions on the antecedent and consequent parameters of IT2FLSs are given to ensure that the prior knowledge can be incorporated into IT2FLSs and three kinds of prior knowledge – bounded range, symmetry (odd and even) and monotonicity (increasing and decreasing) – are explored. Then, design of IT2FLSs using the information from both sample data and prior knowledge is transformed to the constrained least squares optimization problem. At last, to show the superiority of the proposed method, simulations and comparisons are made.*

Keywords: Information fusion, Type-2 fuzzy, Least squares algorithm, Prior knowledge

1. **Introduction.** Recently, a number of extensions to classical fuzzy logic systems (type-1 fuzzy logic systems: T1FLSs) have been attracting interest. One of the most widely used extensions is the interval type-2 fuzzy logic system (IT2FLS) [1-10]. IT2FLSs have obvious advantages for handling different sources of uncertainties, reducing the number of fuzzy rules and weakening noisy disturbance, etc., as IT2FLSs utilize interval type-2 fuzzy sets (IT2FSs) which can provide additional degrees of freedom and have more parameters than type-1 fuzzy sets (T1FSs) in T1FLSs [1,2,6]. Due to these merits, IT2FLSs have found lots of applications, for example, time-series forecasting [2], control of autonomous mobile robots [3] and direct model reference control [8].

Until now, IT2FLSs are always constructed only using the information from the single knowledge source – sample data (training data). Sometimes, satisfactory performance of IT2FLSs can be achieved using this data-driven design method, but, in general, only sample data is not enough to provide sufficient information for system identification, especially when the sample data is not informative enough or is noisy.

One way to compensate this weakness is to incorporate prior knowledge into IT2FLSs. Although, in most cases, it is hard to obtain exact physical structure knowledge of some complex systems, part of their physical properties can be observed easily, such as monotonicity, bounded range, symmetry, etc. Such prior knowledge can partly reflect the characteristics of the unknown systems and compensate the insufficiency of the information from sample data. Recently, this topic has gained considerable concern from different

research areas. Some work has been done to incorporate prior knowledge into support vector machines [11,12], neural networks [13-15], T1FLSs [16-19], etc.

But, to the authors' knowledge, there is no work concerning on how to make full use of the information from multiple knowledge sources (sample data and prior knowledge) to compensate the insufficiency of the information from single knowledge source in the design of IT2FLSs. In this paper, we will study this issue and three kinds of prior knowledge – bounded range, symmetry (odd and even) and monotonicity (increasing and decreasing) – will be considered. First, we will present sufficient conditions on the parameters of IT2FLSs to ensure that the prior knowledge can be encoded. Then, we will show how to utilize the constrained least squares algorithms to incorporate the information from both sample data and prior knowledge into the design of IT2FLSs. At last, we will give simulations and comparisons to show the superiority of the proposed method.

2. Interval Type-2 Fuzzy Logic Systems.

In this section, we will introduce the inference process of IT2FLSs [1-10] briefly. At first, let us give the definition of trapezoid IT2FSs. Without detailed specification, in this study, all IT2FSs are the trapezoid IT2FSs.

2.1. Trapezoid IT2FS.

Figure 1 shows a trapezoid IT2FS and a triangular IT2FS. The triangular IT2FS is a special case of the trapezoid IT2FSs. A trapezoid IT2FS $\widetilde{A}$ can be described by its lower and upper membership functions $\underline{\mu}_{\widetilde{A}}(x,\boldsymbol{\theta})$ and $\overline{\mu}_{\widetilde{A}}(x,\boldsymbol{\theta})$ as

$$\underline{\mu}_{\widetilde{A}}(x,\boldsymbol{\theta}) = \begin{cases} h_2\frac{x-a_2}{b_2-a_2}, & a_2 < x \le b_2, \\ h_2, & b_2 < x \le c_2, \\ h_2\frac{d_2-x}{d_2-c_2}, & c_2 < x \le d_2, \\ 0, & \text{else}, \end{cases} \quad \text{and} \quad \overline{\mu}_{\widetilde{A}}(x,\boldsymbol{\theta}) = \begin{cases} h_1\frac{x-a_1}{b_1-a_1}, & a_1 < x \le b_1, \\ h_1, & b_1 < x \le c_1, \\ h_1\frac{d_1-x}{d_1-c_1}, & c_1 < x \le d_1, \\ 0, & \text{else}, \end{cases}$$

where $a_1 \le b_1 \le c_1 \le d_1$, $a_2 \le b_2 \le c_2 \le d_2$, $a_1 \le a_2$, $d_1 \ge d_2$, $h_2 \le h_1 \le 1$ and $\boldsymbol{\theta} = (a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, h_1, h_2)$ is the parameter vector of all the parameters in $\widetilde{A}$.

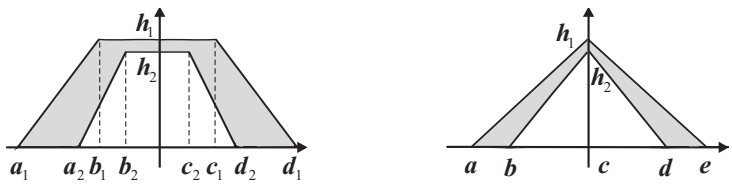

FIGURE 1. Examples of trapezoid IT2FSs

2.2. IT2FLS.

Consider the following complete rule base of an IT2FLS

$$R^{j_1j_2\cdots j_p} : x_1 = \widetilde{A}_1^{j_1},\ x_2 = \widetilde{A}_2^{j_2}, \ldots, x_p = \widetilde{A}_p^{j_p}, \to y = \left[\underline{w}^{j_1j_2\cdots j_p}, \overline{w}^{j_1j_2\cdots j_p}\right]$$

where p is the number of the input variables, $\widetilde{A}_i^{j_i}$ $(i = 1, 2, \ldots, p,\ j_i = 1, 2, \ldots, N_i)$ are IT2FSs of the IF-part and $[\underline{w}^{j_1j_2\cdots j_p},\ \overline{w}^{j_1j_2\cdots j_p}]$s are consequent interval weights of the THEN-part. There are totally $\prod_{i=1}^{p} N_i$ fuzzy rules in this complete rule base.

Once a crisp input $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^{\mathrm{T}}$ is applied to the IT2FLS, through the singleton fuzzifier, the interval firing strength of the rule $R^{j_1j_2\cdots j_p}$ can be obtained as

$$F^{j_1j_2\cdots j_p}(\boldsymbol{x}, \Theta_a) = \left[\underline{f}^{j_1j_2\cdots j_p}(\boldsymbol{x}, \Theta_a), \overline{f}^{j_1j_2\cdots j_p}(\boldsymbol{x}, \Theta_a)\right] = \left[\mathcal{T}_{i=1}^{p}\underline{\mu}_{\widetilde{A}_i^{j_i}}(x_i, \boldsymbol{\theta}_i^{j_i}),\ \mathcal{T}_{i=1}^{p}\overline{\mu}_{\widetilde{A}_i^{j_i}}(x_i, \boldsymbol{\theta}_i^{j_i})\right],$$

where Θ_a is a vector of all the parameters in the antecedent part of the IT2FLS, and $\mathcal{T}$ denotes minimum or product t-norm.

Then, using the center-of-sets type-reducer [1,2,4,6] and the center average defuzzifier, the crisp output of the IT2FLS can be computed as

$$y(\boldsymbol{x},\Theta)=\frac{1}{2}(y_l(\boldsymbol{x},\Theta)+y_r(\boldsymbol{x},\Theta)),$$

where $\Theta=(\Theta_a,\Theta_c)^{\mathrm{T}}$ in which Θ_c is the parameter vector of all the parameters ($\underline{w}^{j_1j_2\cdots j_p}$ and $\overline{w}^{j_1j_2\cdots j_p}$) in the consequent part of the IT2FLS; $y_l(\boldsymbol{x},\Theta)$ and $y_r(\boldsymbol{x},\Theta)$ are the left and right end points of the type-reduced interval set and can be expressed as

$$y_l(\boldsymbol{x},\Theta)=\min\left\{\frac{\sum_{j_1=1}^{N_1}\cdots\sum_{j_p=1}^{N_p}f^{j_1\cdots j_p}w^{j_1\cdots j_p}}{\sum_{j_1=1}^{N_1}\cdots\sum_{j_p=1}^{N_p}f^{j_1\cdots j_p}}\middle|f^{j_1\cdots j_p}\in F^{j_1\cdots j_p}(\boldsymbol{x},\Theta_a),w^{j_1\cdots j_p}\in[\underline{w}^{j_1\cdots j_p},\overline{w}^{j_1\cdots j_p}]\right\},$$

$$y_r(\boldsymbol{x},\Theta)=\max\left\{\frac{\sum_{j_1=1}^{N_1}\cdots\sum_{j_p=1}^{N_p}f^{j_1\cdots j_p}w^{j_1\cdots j_p}}{\sum_{j_1=1}^{N_1}\cdots\sum_{j_p=1}^{N_p}f^{j_1\cdots j_p}}\middle|f^{j_1\cdots j_p}\in F^{j_1\cdots j_p}(\boldsymbol{x},\Theta_a),w^{j_1\cdots j_p}\in[\underline{w}^{j_1\cdots j_p},\overline{w}^{j_1\cdots j_p}]\right\}.$$

Mendel et al. have given the iterative Karnik-Mendel algorithm based formulas for the computation of $y_l(\boldsymbol{x},\Theta)$ and $y_r(\boldsymbol{x},\Theta)$. For more detail about this topic, see [1,2,4,6].

3. Parameter Conditions.

In this section, we will present sufficient conditions on the parameters of IT2FLSs to ensure that the prior knowledge of bounded range, symmetry (odd and even) and monotonicity (increasing and decreasing) can be encoded. First, let us consider the prior knowledge of bounded range.

3.1. Prior knowledge of bounded range.

Notice that the bounded range of an IT2FLS automatically implies the bounded-input-bounded-output (BIBO) stability of the IT2FLS, which is usually required in many real-world applications. For the prior knowledge of bounded range, we have the following results for IT2FLSs:

Theorem 3.1. *The output $y(\boldsymbol{x},\Theta)$ of an IT2FLS falls in the bounded range $[\underline{b},\ \overline{b}]$, i.e., $\underline{b}\le y(\boldsymbol{x},\Theta)\le\overline{b}$, if its consequent parameters satisfy that:*

$$\min_{\substack{j_i=1,\cdots,N_i\\ i=1,\cdots,p}}\{\underline{w}^{j_1\cdots j_p}\}\ge\underline{b}\quad\textit{and}\quad\max_{\substack{j_i=1,\cdots,N_i\\ i=1,\cdots,p}}\{\overline{w}^{j_1\cdots j_p}\}\le\overline{b}.$$

Proof: $\forall f^{j_1\cdots j_p}\in F^{j_1\cdots j_p}(\boldsymbol{x},\Theta_a)$ and $\forall w^{j_1\cdots j_p}\in[\underline{w}^{j_1\cdots j_p},\overline{w}^{j_1\cdots j_p}]$, we have

$$\frac{\sum_{j_1=1}^{N_1}\cdots\sum_{j_p=1}^{N_p}f^{j_1\cdots j_p}w^{j_1\cdots j_p}}{\sum_{j_1=1}^{N_1}\cdots\sum_{j_p=1}^{N_p}f^{j_1\cdots j_p}}\ge\min_{\substack{j_i=1,\cdots,N_i\\ i=1,\cdots,p}}\{w^{j_1\cdots j_p}\}\ge\min_{\substack{j_i=1,\cdots,N_i\\ i=1,\cdots,p}}\{\underline{w}^{j_1\cdots j_p}\}.$$

Thus,

$$y_l(\boldsymbol{x},\Theta)=\min\left\{\frac{\sum_{j_1=1}^{N_1}\cdots\sum_{j_p=1}^{N_p}f^{j_1\cdots j_p}w^{j_1\cdots j_p}}{\sum_{j_1=1}^{N_1}\cdots\sum_{j_p=1}^{N_p}f^{j_1\cdots j_p}}\middle|f^{j_1\cdots j_p}\in F^{j_1\cdots j_p}(\boldsymbol{x},\Theta_a),w^{j_1\cdots j_p}\in[\underline{w}^{j_1\cdots j_p},\overline{w}^{j_1\cdots j_p}]\right\}$$
$$\ge\min_{\substack{j_i=1,\cdots,N_i\\ i=1,\cdots,p}}\{\underline{w}^{j_1\cdots j_p}\}.$$

Similarly, we can prove that $y_r(\boldsymbol{x},\Theta)\le\max_{\substack{j_i=1,\cdots,N_i\\ i=1,\cdots,p}}\{\overline{w}^{j_1\cdots j_p}\}$.

Therefore, if $\min_{\substack{j_i=1,\cdots,N_i\\ i=1,\cdots,p}}\{\underline{w}^{j_1\cdots j_p}\}\ge\underline{b}$ and $\max_{\substack{j_i=1,\cdots,N_i\\ i=1,\cdots,p}}\{\overline{w}^{j_1\cdots j_p}\}\le\overline{b}$, then,

$$\underline{b}\le y(\boldsymbol{x},\Theta)=\frac{1}{2}(y_l(\boldsymbol{x},\Theta)+y_r(\boldsymbol{x},\Theta))\le\overline{b}.\qquad\square$$

Remark 3.1. *For the prior knowledge of bounded range, we only need to constrain the consequent parameters Θ_c and there is no need to constrain the antecedent parameters. As a result, in this case, there is no requirement on how to partition each input domain.*

3.2. **Prior knowledge of symmetry (odd and even).** For many applications, especially control problems, the fuzzy systems designed for them should be odd symmetric or even symmetric. The following two theorems will show how to constrain the parameters of IT2FLSs to incorporate the prior knowledge of odd symmetry and even symmetry.

Theorem 3.2. *An IT2FLS is odd symmetric, i.e.,* $y(\boldsymbol{x},\Theta) = -y(-\boldsymbol{x},\Theta)$*, if the following conditions are satisfied:*

1) $\forall i \in \{1,2,\ldots,p\}$*, the input domain of* x_i *is partitioned symmetrically around 0 by trapezoid IT2FSs* $\widetilde{A}_i^1, \widetilde{A}_i^2, \ldots, \widetilde{A}_i^{N_i}$ *as shown in Figure 2, where* N_i *is an odd number.*

2) The consequent parameters of the rules $R^{j_1\cdots j_p}$ *and* $R^{(N_1+1-j_1)\cdots(N_p+1-j_p)}$ *satisfy that* $\left[\underline{w}^{j_1\cdots j_p}, \overline{w}^{j_1\cdots j_p}\right] = \left[-\overline{w}^{(N_1+1-j_1)\cdots(N_p+1-j_p)}, -\underline{w}^{(N_1+1-j_1)\cdots(N_p+1-j_p)}\right]$*, where* $j_i = 1,2,\ldots,\frac{N_i+1}{2}$.

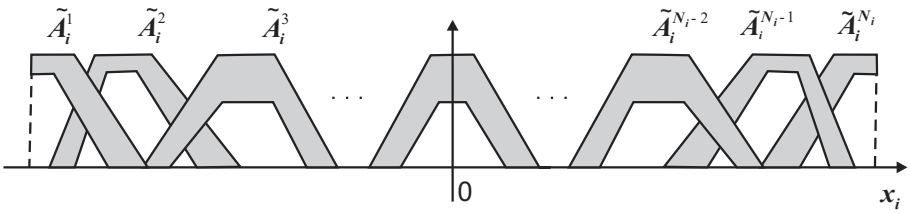

FIGURE 2. Interval type-2 fuzzy partition with N_i trapezoid IT2FSs

Proof: Suppose that t fuzzy rules, which are denoted as $R^{j_1^1\cdots j_p^1}, \ldots, R^{j_1^t\cdots j_p^t}$, can be fired when $\boldsymbol{x}$ is given to the IT2FLS.

As the rule base is complete, and each input domain is partitioned symmetrically around 0, hence, when $-\boldsymbol{x}$ is given to the IT2FLS, the fuzzy rules $R^{(N_1+1-j_1^1)\cdots(N_p+1-j_p^1)}$, $\ldots, R^{(N_1+1-j_1^t)\cdots(N_p+1-j_p^t)}$ can be fired.

Also, from the conditions above, it is obvious that, for $s = 1,2,\ldots,t$

$$F^{j_1^s\cdots j_p^s}(\boldsymbol{x},\Theta_a) = F^{(N_1+1-j_1^s)\cdots(N_p+1-j_p^s)}(-\boldsymbol{x},\Theta_a),$$
$$\left[\underline{w}^{j_1^s\cdots j_p^s}, \overline{w}^{j_1^s\cdots j_p^s}\right] = \left[-\overline{w}^{(N_1+1-j_1^s)\cdots(N_p+1-j_p^s)}, -\underline{w}^{(N_1+1-j_1^s)\cdots(N_p+1-j_p^s)}\right].$$

Hence,

$$\begin{aligned}
y_l(\boldsymbol{x},\Theta) &= \min\left\{\frac{\sum_{s=1}^t f^s w^s}{\sum_{s=1}^t f^s}\middle| f^s \in F^{j_1^s\cdots j_p^s}(\boldsymbol{x},\Theta_a), w^s \in \left[\underline{w}^{j_1^s\cdots j_p^s}, \overline{w}^{j_1^s\cdots j_p^s}\right]\right\} \\
&= -\max\left\{\frac{\sum_{s=1}^t f^s(-w^s)}{\sum_{s=1}^t f^s}\middle| f^s \in F^{j_1^s\cdots j_p^s}(\boldsymbol{x},\Theta_a), -w^s \in \left[-\overline{w}^{j_1^s\cdots j_p^s}, -\underline{w}^{j_1^s\cdots j_p^s}\right]\right\} \\
&= -\max\left\{\frac{\sum_{s=1}^t \tilde{f}^s \tilde{w}^s}{\sum_{s=1}^t \tilde{f}^s}\middle| \tilde{f}^s \in F^{(N_1+1-j_1^s)\cdots(N_p+1-j_p^s)}(-\boldsymbol{x},\Theta_a),\right. \\
&\qquad\qquad \left. \tilde{w}^s \in \left[\underline{w}^{(N_1+1-j_1^s)\cdots(N_p+1-j_p^s)}, \overline{w}^{(N_1+1-j_1^s)\cdots(N_p+1-j_p^s)}\right]\right\} \\
&= -y_r(-\boldsymbol{x},\Theta)
\end{aligned}$$

In the similar way, we can prove that $y_r(\boldsymbol{x},\Theta) = -y_l(-\boldsymbol{x},\Theta)$.

Therefore,

$$y(\boldsymbol{x},\Theta) = \frac{y_l(\boldsymbol{x},\Theta) + y_r(\boldsymbol{x},\Theta)}{2} = \frac{-y_r(-\boldsymbol{x},\Theta) - y_l(-\boldsymbol{x},\Theta)}{2} = -y(-\boldsymbol{x},\Theta). \qquad \square$$

Similarly, for the prior knowledge of even symmetry, we have following results:

Theorem 3.3. *An IT2FLS is even symmetric, i.e.,* $y(\boldsymbol{x},\Theta)=y(-\boldsymbol{x},\Theta)$*, if the following conditions are satisfied:*

1) $\forall i\in\{1,2,\ldots,p\}$*, the input domain of* x_i *is partitioned symmetrically around 0 by trapezoid IT2FSs* $\widetilde{A}_i^1,\widetilde{A}_i^2,\ldots,\widetilde{A}_i^{N_i}$ *as shown in Figure 2, where* N_i *is an odd number.*

2) The consequent parameters of the rules $R^{j_1\cdots j_p}$ *and* $R^{(N_1+1-j_1)\cdots(N_p+1-j_p)}$ *satisfy that* $\left[\underline{w}^{j_1\cdots j_p},\overline{w}^{j_1\cdots j_p}\right]=\left[\underline{w}^{(N_1+1-j_1)\cdots(N_p+1-j_p)},\overline{w}^{(N_1+1-j_1)\cdots(N_p+1-j_p)}\right]$*, where* $j_i=1,2,\ldots,(N_i+1)/2$*; especially, the consequent parameters of the rule* $R^{\frac{N_1+1}{2}\cdots\frac{N_p+1}{2}}$ *satisfies that* $\underline{w}^{\frac{N_1+1}{2}\cdots\frac{N_p+1}{2}}=\overline{w}^{\frac{N_1+1}{2}\cdots\frac{N_p+1}{2}}$.

Remark 3.2. *The first condition in Theorems 3.2 and 3.3 means that:* $\forall i\in\{1,2,\cdots p\}$,

1) If $j_i\in\{1,2,\ldots,(N_i-1)/2\}$*, then* $\forall\theta\in\{a_1,b_1,c_1,d_1,a_2,\ b_2,c_2,d_2\}$, $\theta_i^{j_i}=-\theta_i^{N_i+1-j_i}$ *and* $\forall\theta\in\{h_1,h_2\}$, $\theta_i^{j_i}=\theta_i^{N_i+1-j_i}$.

2) If $j_i=(N_i+1)/2$*, then* $a_{1,i}^{\frac{N_i+1}{2}}=-d_{1,i}^{\frac{N_i+1}{2}}$, $a_{2,i}^{\frac{N_i+1}{2}}=-d_{2,i}^{\frac{N_i+1}{2}}$, $b_{1,i}^{\frac{N_i+1}{2}}=-c_{1,i}^{\frac{N_i+1}{2}}$ *and* $b_{2,i}^{\frac{N_i+1}{2}}=-c_{2,i}^{\frac{N_i+1}{2}}$.

This is the constraint on the antecedent parameters of IT2FLSs to ensure that the prior knowledge of odd symmetry and even symmetry can be encoded.

3.3. **Prior knowledge of monotonicity (increasing and decreasing).** The monotonicity is a common kind of prior knowledge [20]. For example, consider the water heating system [18] and the coupled-tank liquid-level system [21]. The outputs of both systems will change monotonically with respect to their inputs.

For simplicity, here, we only give useful results about the single-input monotonically increasing and monotonically decreasing IT2FLSs.

Theorem 3.4. *Assume that the input domain* $U=[\underline{u},\overline{u}]$ *is partitioned by* N *trapezoid IT2FSs* $\widetilde{A}^1,\widetilde{A}^2,\ldots,\widetilde{A}^N$*, where* $\forall j\in\{1,2,\ldots,N\}$, $\overline{\mu}_{\widetilde{A}^j}(x)=\overline{\mu}_{\widetilde{A}^j}\left(x,a_1^j,b_1^j,c_1^j,d_1^j,h_1^j\right)$, $\underline{\mu}_{\widetilde{A}^j}(x)=\underline{\mu}_{\widetilde{A}^j}\left(x,a_2^j,b_2^j,c_2^j,d_2^j,h_2^j\right)$*. Then, the IT2FLS is monotonically increasing, if the following conditions are satisfied:*

1) IT2FSs $\widetilde{A}^1,\widetilde{A}^2,\ldots,\widetilde{A}^N$ *form fuzzy partition as shown in Figure 3 where* $a_1^1=b_1^1=a_2^1=b_2^1=\underline{u}$, $c_1^N=d_1^N=c_2^N=d_2^N=\overline{u}$, $b_1^j\le a_1^{j+1}$, $d_1^j\le c_1^{j+1}$, $d_2^j>a_2^{j+1}$ *and* $d_1^{j-1}\le a_1^{j+1}$.

2) The consequent parameters satisfy that $\underline{w}^1\le\underline{w}^2\le\ldots\le\underline{w}^N$ *and* $\overline{w}^1\le\overline{w}^2\le\ldots\le\overline{w}^N$.

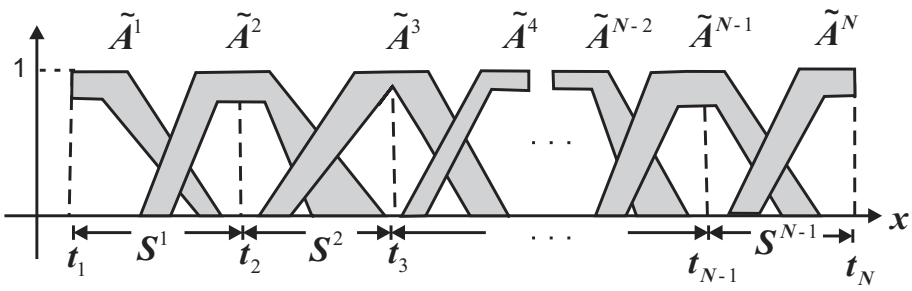

FIGURE 3. Fuzzy partition with N trapezoid IT2FSs

Proof: Denote Ω_j $(j=1,\ldots,N)$ as

$$\Omega_j=\begin{cases}\{\underline{u}\}, & j=1\\ U\setminus\left[\left(\underline{u},d_1^{j-1}\right)\cup\left(a_1^{j+1},\overline{u}\right)\right], & 1<j<N\\ \{\overline{u}\}, & j=N\end{cases}$$

As $d_1^{j-1}\le a_1^{j+1}$, hence, $\Omega_j\neq\emptyset$.

For any point $t_j \in \Omega_j$ $(j = 1, \ldots, N)$, the input domain $U = [\underline{u}, \overline{u}]$ can be partitioned into $N-1$ areas as shown in Figure 3, i.e., $U = \cup_{j=1}^{N-1} S^j$, where $S^j = [t_j, t_{j+1}]$.

The first condition assures that the trapezoid IT2FSs $\widetilde{A}^1$, $\widetilde{A}^2, \ldots, \widetilde{A}^N$ are arranged in the ascending order, and, at each point, no more than two IT2FSs can be fired. Hence, from (55) and (56) in [1], we can obtain that:

If $x \in S_j$ $(j = 1, 2, \ldots, N-1)$, then

$$y_l(x, \Theta) = \frac{\overline{f}^j(x, \Theta_a)\underline{w}^j + \underline{f}^{j+1}(x, \Theta_a)\underline{w}^{j+1}}{\overline{f}^j(x, \Theta_a) + \underline{f}^{j+1}(x, \Theta_a)}, \tag{1}$$

$$y_r(x, \Theta) = \frac{\underline{f}^j(x, \Theta_a)\overline{w}^j + \overline{f}^{j+1}(x, \Theta_a)\overline{w}^{j+1}}{\underline{f}^j(x, \Theta_a) + \overline{f}^{j+1}(x, \Theta_a)}, \tag{2}$$

where $\underline{f}^j(x, \Theta_a) = \underline{\mu}_{\widetilde{A}^j}(x)$, $\overline{f}^j(x, \Theta_a) = \overline{\mu}_{\widetilde{A}^j}(x)$. For short, we use $\underline{f}^j(x)$, $\overline{f}^j(x)$ to replace $\underline{f}^j(x, \Theta_a)$ and $\overline{f}^j(x, \Theta_a)$ below.

From (1) and (2), we can see that, if $x \in S^j$, then

$$\underline{w}^j \le y_l(x, \Theta) \le \underline{w}^{j+1} \quad \text{and} \quad \overline{w}^j \le y_r(x, \Theta) \le \overline{w}^{j+1}.$$

Hence, if $x \in S^j$, then

$$\frac{\underline{w}^j + \overline{w}^j}{2} \le y(x, \Theta) \le \frac{\underline{w}^{j+1} + \overline{w}^{j+1}}{2}. \tag{3}$$

$\forall x$, $x' \in U$, suppose that $x \le x'$ and $x \in S^j$, $x' \in S^{j'}$.

If $j < j'$, from (3) and condition 2) in this theorem, we can see that $y(x, \Theta) \le y(x', \Theta)$;

If $j = j'$, then we have following results:

a) From (1), $\forall x \le x'$

$$y_l(x', \Theta) - y_l(x, \Theta) = \frac{\left[\underline{f}^{j+1}(x')\overline{f}^j(x) - \overline{f}^j(x')\underline{f}^{j+1}(x)\right](\underline{w}^{j+1} - \underline{w}^j)}{\left(\overline{f}^j(x') + \underline{f}^{j+1}(x')\right)\left(\overline{f}^j(x) + \underline{f}^{j+1}(x)\right)}$$

As $\underline{f}^{j+1}(.)$ is monotonically increasing and $\overline{f}^j(.)$ is monotonically decreasing in S^j, hence

$$\underline{f}^{j+1}(x') \ge \underline{f}^{j+1}(x) \quad \text{and} \quad \overline{f}^j(x) \ge \overline{f}^j(x').$$

This implies that

$$\underline{f}^{j+1}(x')\overline{f}^j(x) - \overline{f}^j(x')\underline{f}^{j+1}(x) \ge 0.$$

Therefore, $y_l(x', \Theta) \ge y_l(x, \Theta)$ in S^j.

b) In the similar way, $\forall x \le x' \in S^j$, $y_r(x', \Theta) \ge y_r(x, \Theta)$.

From a) and b), if both x' and x are in S^j, then $y(x', \Theta) \ge y(x, \Theta)$.

Thus, from the discussion above, we can conclude that this theorem holds. □

Similar results can be obtained for the prior knowledge of decreasing monotonicity.

Theorem 3.5. *Assume that the input domain $U = [\underline{u}, \overline{u}]$ is partitioned by N trapezoid IT2FSs $\widetilde{A}^1, \widetilde{A}^2, \ldots, \widetilde{A}^N$, where $\forall j \in \{1, 2, \ldots, N\}$, $\overline{\mu}_{\widetilde{A}^j}(x) = \overline{\mu}_{\widetilde{A}^j}\left(x, a_1^j, b_1^j, c_1^j, d_1^j, h_1^j\right)$, $\underline{\mu}_{\widetilde{A}^j}(x) = \underline{\mu}_{\widetilde{A}^j}\left(x, a_2^j, b_2^j, c_2^j, d_2^j, h_2^j\right)$. Then, the IT2FLS is monotonically decreasing, if the following conditions are satisfied:*

1) IT2FSs $\widetilde{A}^1, \widetilde{A}^2, \ldots, \widetilde{A}^N$ form fuzzy partition as shown in Figure 3 where $a_1^1 = b_1^1 = a_2^1 = b_2^1 = \underline{u}$, $c_1^N = d_1^N = c_2^N = d_2^N = \overline{u}$, $b_1^j \le a_1^{j+1}$, $d_1^j \le c_1^{j+1}$, $d_2^j > a_2^{j+1}$ and $d_1^{j-1} \le a_1^{j+1}$.

2) The consequent parameters satisfy that $\underline{w}^1 \ge \underline{w}^2 \ge \ldots \ge \underline{w}^N$ and $\overline{w}^1 \ge \overline{w}^2 \ge \ldots \ge \overline{w}^N$.

In this section, sufficient parameter constraints of IT2FLSs have been studied to ensure the aforementioned prior knowledge. In the following section, we will show how to design IT2FLSs using the information from both sample data and prior knowledge.

4. **Design of IT2FLSs Using Both Sample Data and Prior Knowledge.** In the design of IT2FLSs using the information from both sample data and prior knowledge, different kinds of prior knowledge are used to constrain the antecedent and consequent parameters of IT2FLSs, and sample data is utilized to train and optimize these parameters further. For simplicity, in this study, we only consider the design of the single-input IT2FLSs using both sample data and prior knowledge.

4.1. **Problem formulation.** Suppose that there are M input-output sample data (x^1, y^1), $(x^2, y^2), \ldots, (x^M, y^M)$. And, the training criteria is chosen to minimize the following squared error function:

$$E = \frac{1}{2} \sum\nolimits_{i=1}^{M} |y(x^i, \Theta) - y^i|^2, \tag{4}$$

where $y(x^i, \Theta)$ is the output of the IT2FLS, and Θ is the parameter vector of all the antecedent and consequent parameters of the IT2FLS.

In Section 3, we have studied how to transform different kinds of prior knowledge to the constraints of the antecedent and consequent parameters of IT2FLSs. These constraints can be rewritten abstractly as $\Theta \in \Omega$, where Ω represents the constrained feasible parameter space.

Therefore, design of the IT2FLSs using the information from both sample data and prior knowledge can be seen as the following optimization problem

$$\begin{cases} \min\limits_{\Theta} \frac{1}{2} \sum_{i=1}^{M} |y(x^i, \Theta) - y^i|^2 \\ \text{subject to} \quad \Theta \in \Omega. \end{cases} \tag{5}$$

As the outputs of the IT2FLSs are nonlinear with respect to the parameters of their antecedent IT2FSs, this optimization problem needs to be solved using constrained nonlinear optimization algorithms, e.g., genetic algorithms, particle swarm algorithms.

4.2. **Problem transformation.** Although different optimization algorithms can be utilized to solve the optimization problem in (5), in this work, we adopt another strategy, which determines and optimizes the antecedent parameters and consequent parameters separately. In this strategy, two steps are needed. The first step is to set up the membership functions of the IT2FSs in the antecedent part of the IT2FLSs, and the second step is to optimize the consequent interval weights under the constraints on these parameters. The first step can be accomplished by partitioning the input domains intuitively or by the clustering algorithms. How to use different clustering algorithms to obtain optimal or suboptimal interval type-2 fuzzy partitions, which satisfy the constraints on the antecedent membership functions in Theorems 3.1 – 3.5, will be one of our research directions in the near future. In this study, we mainly focus on the second step.

Below, we will first show that the output of the IT2FLS is linear with its consequent parameters, and then, we will demonstrate that the optimization problem (5) can be changed to a constrained least squares optimization problem.

We suppose that the fuzzy partition of the input domain satisfies that no more than two IT2FSs can be fired at each point as shown in Figure 3, and the rule base of the single-input IT2FLS is written as

$$\{R^j : \ x = \widetilde{A}^j \ \rightarrow \ y = [\underline{w}^j, \overline{w}^j]\}_{j=1}^{N}. \tag{6}$$

From Theorem 3.3 in [22], we know that the crisp output of the single-input IT2FLS in (6) is equal to the average of the outputs of two T1FLSs (called Lower T1FLS and Upper T1FLS). The rules of the two T1FLSs are [22]:

$$\text{Lower T1FLS}: \left\{ R_L^j: \ x = A_L^j \to \ y = \underline{w}^j \ \right\}_{j=1}^N,$$
$$\text{Upper T1FLS}: \left\{ R_U^j: \ x = A_U^j \to \ y = \overline{w}^j \ \right\}_{j=1}^N,$$

where $A_L^1, \ldots, A_L^N$, $A_U^1, \ldots, A_U^N$ are type-1 fuzzy sets. Their membership functions can be determined by

$$\mu_{A_L^j}(x) = \begin{cases} \underline{\mu}_{\widetilde{A}^j}(x), & x \in S^{j-1}, \underline{w}^{j-1} \le \underline{w}^j, \\ \overline{\mu}_{\widetilde{A}^j}(x), & x \in S^{j-1}, \underline{w}^{j-1} > \underline{w}^j, \\ \overline{\mu}_{\widetilde{A}^j}(x), & x \in S^j, \underline{w}^j \le \underline{w}^{j+1}, \\ \underline{\mu}_{\widetilde{A}^j}(x), & x \in S^j, \underline{w}^j > \underline{w}^{j+1}. \end{cases} \quad \mu_{A_U^j}(x) = \begin{cases} \overline{\mu}_{\widetilde{A}^j}(x), & x \in S^{j-1}, \overline{w}^{j-1} \le \overline{w}^j, \\ \underline{\mu}_{\widetilde{A}^j}(x), & x \in S^{j-1}, \overline{w}^{j-1} > \overline{w}^j, \\ \underline{\mu}_{\widetilde{A}^j}(x), & x \in S^j, \overline{w}^j \le \overline{w}^{j+1}, \\ \overline{\mu}_{\widetilde{A}^j}(x), & x \in S^j, \overline{w}^j > \overline{w}^{j+1}. \end{cases}$$

Therefore, the crisp output of the single-input IT2FLS can be computed as

$$y(x, \Theta_c) = \frac{1}{2}\left[\frac{\sum_{i=1}^N \mu_{A_L^i}(x)\underline{w}^i}{\sum_{i=1}^N \mu_{A_L^i}(x)} + \frac{\sum_{i=1}^N \mu_{A_U^i}(x)\overline{w}^i}{\sum_{i=1}^N \mu_{A_U^i}(x)} \right] = [\phi_1(x), \ldots, \phi_{2N}(x)]\Theta_c, \tag{7}$$

where

$$\Theta_c = \left[\underline{w}^1, \ldots, \underline{w}^N, \overline{w}^1, \ldots, \overline{w}^N\right]^{\mathrm{T}},$$
$$\phi_i(x) = \begin{cases} \frac{1}{2}\frac{\mu_{A_L^i}(x)}{\sum_{i=1}^N \mu_{A_L^i}(x)}, & i = 1, \ldots, N, \\ \frac{1}{2}\frac{\mu_{A_U^{i-N}}(x)}{\sum_{i=1}^N \mu_{A_U^{i-N}}(x)}, & i = N+1, \ldots, 2N. \end{cases}$$

From (7), we can see that the output of the IT2FLS is linear with its consequent parameters. And, the training criteria (4) can be rewritten as

$$E = \frac{1}{2}\sum_{i=1}^M |y(x^i, \Theta_c) - y^i|^2 = \frac{1}{2}(\Phi\Theta_c - \boldsymbol{y})^{\mathrm{T}}(\Phi\Theta_c - \boldsymbol{y}),$$

where

$$\boldsymbol{y} = [y^1, y^2, \ldots, y^M]^{\mathrm{T}},$$
$$\Phi = \begin{bmatrix} \phi_1(x^1) & \phi_2(x^1) & \cdots & \phi_{2N}(x^1) \\ \phi_1(x^2) & \phi_2(x^2) & \cdots & \phi_{2N}(x^2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(x^M) & \phi_2(x^M) & \cdots & \phi_{2N}(x^M) \end{bmatrix} \in \mathbb{R}^{M \times 2N}.$$

As we only consider to optimize the consequent parameters, therefore, the abstract constraint $\Theta \in \Omega$ in (5) can be rewritten as $\Theta_c \in \Omega_c$. Below, we will write the constraints on the consequent parameters in Theorems 3.1 – 3.5 into the forms of linear-inequality constraints $\mathbf{C}\Theta_c \le \boldsymbol{b}$ and/or linear-equality constraints $\mathbf{C}_{eq}\Theta_c = \boldsymbol{b}_{eq}$.

1) Linear constraint for bounded range

$$\begin{bmatrix} -\mathbf{I}_N & 0 \\ 0 & \mathbf{I}_N \end{bmatrix} \Theta_c \le \begin{bmatrix} -\underline{\boldsymbol{b}} \\ \overline{\boldsymbol{b}} \end{bmatrix},$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix, $\underline{\boldsymbol{b}} = [\underbrace{\underline{b}, \underline{b}, \ldots, \underline{b}}_{N}]^{\mathrm{T}}$ and $\overline{\boldsymbol{b}} = [\underbrace{\overline{b}, \overline{b}, \ldots, \overline{b}}_{N}]^{\mathrm{T}}$.

2) Linear constraint for odd symmetry

$$\left[\ \mathbf{I}_N, \quad R(\mathbf{I}_N) \ \right] \Theta_c = \mathbf{0},$$

where $R(\mathbf{I}_N)$ denotes the $N \times N$ matrix obtained by rotating $\mathbf{I}_N$ 90 degrees clockwisely.

3) Linear constraint for even symmetry

$$\begin{bmatrix} \mathbf{I}_{\frac{N-1}{2}} & \mathbf{0}_{\frac{N-1}{2}\times 1} & -R(\mathbf{I}_{\frac{N-1}{2}}) & \mathbf{0}_{\frac{N-1}{2}} & \mathbf{0}_{\frac{N-1}{2}\times 1} & \mathbf{0}_{\frac{N-1}{2}} \\ \mathbf{0}_{1\times\frac{N-1}{2}} & 1 & \mathbf{0}_{1\times\frac{N-1}{2}} & \mathbf{0}_{1\times\frac{N-1}{2}} & -1 & \mathbf{0}_{1\times\frac{N-1}{2}} \\ \mathbf{0}_{\frac{N-1}{2}} & \mathbf{0}_{\frac{N-1}{2}\times 1} & \mathbf{0}_{\frac{N-1}{2}} & \mathbf{I}_{\frac{N-1}{2}} & \mathbf{0}_{\frac{N-1}{2}\times 1} & -R(\mathbf{I}_{\frac{N-1}{2}}) \end{bmatrix} \Theta_c = \mathbf{0}.$$

4) Linear constraint for increasing monotonicity

$$\begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix} \Theta_c \leq \mathbf{0},$$

where

$$\mathbf{V} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N}.$$

5) Linear constraint for decreasing monotonicity

$$\begin{bmatrix} -\mathbf{V} & \mathbf{0} \\ \mathbf{0} & -\mathbf{V} \end{bmatrix} \Theta_c \leq \mathbf{0}.$$

From the discussions above, we can observe that design of the IT2FLSs using both sample data and prior knowledge can be transformed to the following constrained least squares optimization problem

$$\begin{cases} \min\limits_{\Theta_c} (\Phi\Theta_c - \boldsymbol{y})^{\mathrm{T}}(\Phi\Theta_c - \boldsymbol{y}) \\ \text{subject to} \quad \mathbf{C}\Theta_c \leq \boldsymbol{b} \quad \text{and/or} \quad \mathbf{C}_{eq}\Theta_c = \boldsymbol{b}_{eq}. \end{cases}$$

5. Simulations and Comparisons.

5.1. Problem description. To show the usefulness of the prior knowledge, let's consider the following nonlinear function:

$$g(x) = (|x|/3)^2 \tanh(|x|),$$

where $x \in U = [-3, 3]$. Obviously, this function is bounded in [0, 1], even symmetric, monotonically decreasing in $[-3, 0]$ and monotonically increasing in [0, 3].

The noisy training data is generated by

$$\tilde{y}(x) = g(x) + \tilde{n},$$

where $\tilde{n}$ is the uniformly distributed additive noise in $[-b, b]$. In this simulation, three different levels of noisy disturbance are tested, where $b = 20\%$, 30% and 40%, respectively. In each noisy circumstance, we consider 5 cases. In these cases, the sizes of both the training data and the evaluation data are 30, 60, 100, 200, 300. In case i, we denote the training data set as $\mathfrak{T}_i = \left\{(x^1, \tilde{y}^1), (x^2, \tilde{y}^2), \ldots, \left(x^{K_i}, \tilde{y}^{K_i}\right)\right\}$. Furthermore, in each case, the noise-free data set is chosen as the evaluation data set $\mathfrak{D}_i = \left\{(x^1, g(x^1)), (x^2, g(x^2)), \ldots, \left(x^{K_i}, g(x^{K_i})\right)\right\}$ which can be used to check whether the trained model could follow the characteristics of the original data rather than the noisy data.

In our simulation, for comparison, we design four fuzzy logic systems (FLSs) to identify the nonlinear function: sample-Data-and-prior-Knowledge based IT2FLS (DK-IT2FLS), only sample-Data based IT2FLS (D-IT2FLS), sample-Data-and-prior-Knowledge based

T1FLS (DK-T1FLS), only sample-Data based T1FLS (D-T1FLS). To evaluate the approximation performance and the generalization performance of different FLSs, we consider the following two performance indexes – Root of the Mean Squared Errors (*RMSE*s) of the training data and evaluation data:

$$\delta_i = \left(\frac{1}{K_i} \sum\nolimits_{k=1}^{K_i} \left(\tilde{y}^k - \hat{y}\left(x^k\right) \right)^2 \right)^{\frac{1}{2}},$$

$$\sigma_i = \left(\frac{1}{K_i} \sum\nolimits_{k=1}^{K_i} \left(g(x^k) - \hat{y}\left(x^k\right) \right)^2 \right)^{\frac{1}{2}},$$

where $\hat{y}(x^k)$ is the output of FLSs for the input x^k. The first index δ_i can reflect the approximation ability of different FLSs for the training data in case i in each level of the noisy disturbance, while the second index σ_i can reflect the generalization characteristics of different FLSs for the original noise-free data in case i in each level of the noisy disturbance. For both indexes, the less the value is, the better the result is.

For each FLS, we use 7 fuzzy rules, the membership functions of whose antecedent parts are shown in Figure 4. Note that the fuzzy partitions in Figure 4 satisfy the conditions on fuzzy sets in Theorems 3.1 – 3.5. The left task is to tune the consequent weights with or without constraints.

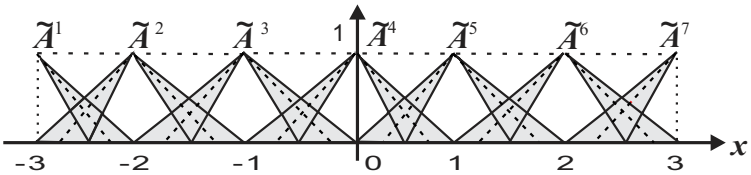

FIGURE 4. Type-1 (dotted line) and type-2 (gray area) fuzzy partitions

5.2. **Simulation results and comparisons.** As stated above, three different levels of noise are considered, and, under each noisy circumstance, five cases are tested. What is more, in each case, the data generation processes and the identification processes of the four FLSs are run 50 times. After that, the performance indexes in case i are calculated as the average of the corresponding values obtained in the 50 run times, i.e.,

$$\widetilde{\delta}_i = \frac{1}{T} \sum\nolimits_{j=1}^{T} \delta_i^j,$$

$$\widetilde{\sigma}_i = \frac{1}{T} \sum\nolimits_{j=1}^{T} \sigma_i^j,$$

where $T = 50$, and δ_i^j and σ_i^j are the first and second performance indexes obtained in the jth run in case i.

For the aforementioned three noise levels, the comparisons of the four FLSs with respect to the *RMSE*s for the training and evaluation data are shown in Figure 5. And, Figure 6 shows us an example of the identification result where 100 sample data are utilized to train the four FLSs and the noise distributes uniformly in [−30%, 30%].

As stated above, the *RMSE*s δ_i for training data can reflect the approximation abilities of different FLSs. From Figures 5(a) – 5(c), D-IT2FLS has the best approximation ability, and then, DK-IT2FLS, at last, D-T1FLS and DK-T1FLS. In general, according to the approximation performance in this simulation, the type-2 FLSs perform better than the type-1 FLSs; the reason for this is that the type-2 FLSs have more parameters and more freedoms than the type-1 FLSs. On the other hand, D-IT2FLS and D-T1FLS have better approximation performance compared with DK-IT2FLS and DK-T1FLS, as the

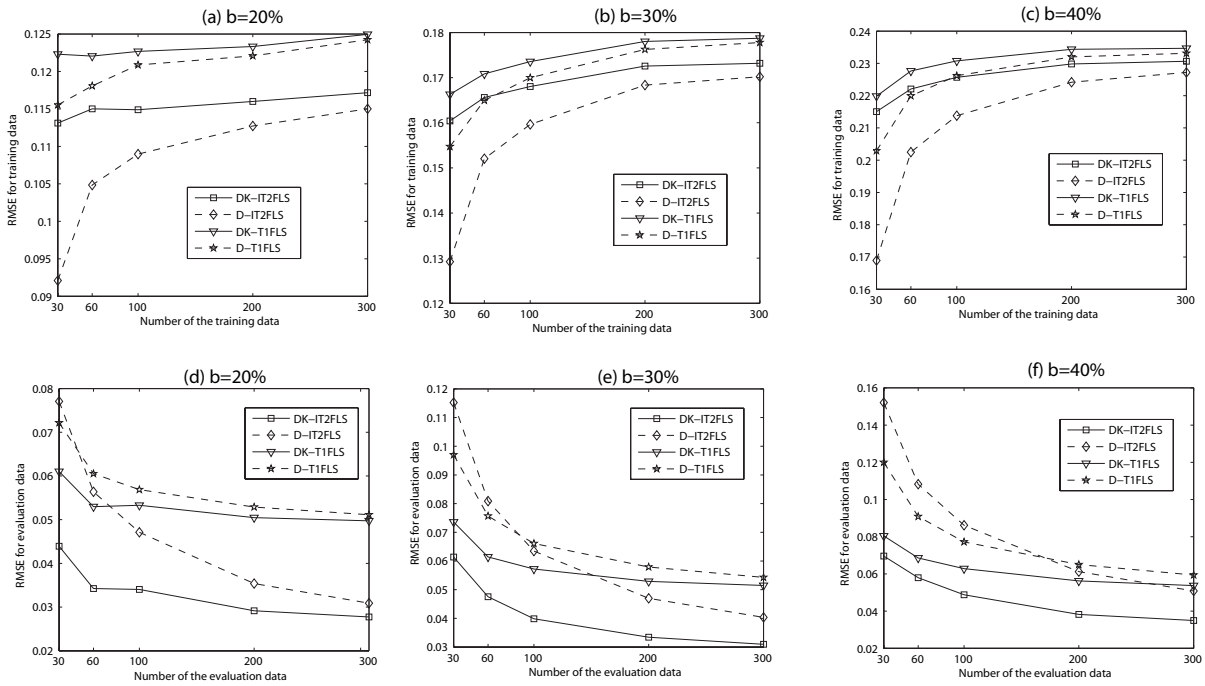

FIGURE 5. Performance comparisons of the four FLSs

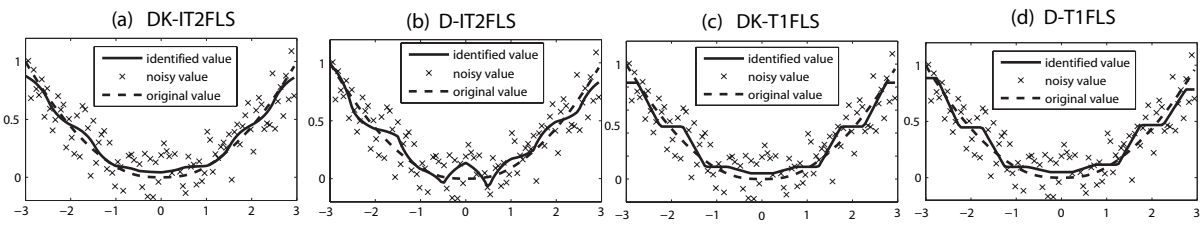

FIGURE 6. The identification results of the four FLSs

consequent weights of DK-IT2FLS and DK-T1FLS are constrained, and this makes their feasible parameter spaces smaller than those of D-IT2FLS and D-T1FLS.

Also, as stated above, the *RMSE*s σ_i for evaluation data can reflect the generalization abilities of different FLSs. From Figures 5(d) – 5(f), we can see that DK-IT2FLS and DK-T1FLS have better generalization ability than D-IT2FLS and D-T1FLS. This means that DK-IT2FLS and DK-T1FLS can follow the characteristics of the real system dynamics better than D-IT2FLS and D-T1FLS under noisy training circumstances.

In conclusion, the type-2 FLSs have better approximation ability than the type-1 FLSs, but the FLSs designed using both sample data and prior knowledge have better generalization characteristics. The reason for this is that prior knowledge can prevent impossible or implausible values, so that the designed FLSs will not be over fitted. Therefore, we can say that the prior knowledge can improve the performance of the IT2FLSs.

6. **Conclusions.** This study has presented how to design IT2FLSs using the information from both sampled data and prior knowledge. From the simulation results and comparisons, we can conclude that: 1) DK-IT2FLS is able to prevent impossible or implausible values which conflict with the prior knowledge; 2) the prior knowledge may improve the generalization ability of IT2FLSs. Here, we have just explored the prior knowledge of bounded range, symmetry (odd and even) and monotonicity (increasing and decreasing).

How to utilize the other prior knowledge, such as fixed points, stability, etc., will be one of our future research directions.

Acknowledgment. This work was partly supported by the NSFC Project under Grant Nos. 60975060 and 61074149, National 863 Program (No. 2007AA04Z239), China. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

REFERENCES

[1] Q. Liang and J. M. Mendel, Interval type-2 fuzzy logic systems: Theory and design, *IEEE Transactions on Fuzzy Systems*, vol.8, pp.535-550, 2000.

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

[3] H. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, *IEEE Transactions on Fuzzy Systems*, vol.12, pp.524-539, 2004.

[4] J. M. Mendel, R. I. John and F. Liu, Interval type-2 fuzzy sets made simple, *IEEE Transactions on Fuzzy Systems*, vol.14, pp.808-821, 2006.

[5] H. B. Mitchell, Ranking type-2 fuzzy numbers, *IEEE Transactions on Fuzzy Systems*, vol.14, pp.287-294, 2006.

[6] J. M. Mendel, Advances in type-2 fuzzy sets and systems, *Information Sciences*, vol.177, pp.84-110, 2007.

[7] C. Li, J. Yi and D. Zhao, Design of interval type-2 fuzzy logic system using sampled data and prior knowledge, *ICIC Express Letters*, vol.3, no.3(B), pp.695-700, 2009.

[8] M. A. Khanesar and M. Teshnehlab, Design of direct interval type-2 fuzzy model reference, *ICIC Express Letters*, vol.3, no.4(A), pp.975-981, 2009.

[9] O. Castillo and P. Melin, Intelligent systems with interval type-2 fuzzy logic, *International Journal of Innovative Computing, Information and Control*, vol.4, no.4, pp.771-783, 2008.

[10] J. M. Mendel and D. Wu, Perceptual reasoning for perceptual computing, *IEEE Transactions on Fuzzy Systems*, vol.16, pp.1550-1564, 2008.

[11] X. Y. Wu and R. Srihari, Incorporating prior knowledge with weighted margin support vector machines, *Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, pp.326-333, 2004.

[12] F. Lauer and G. Bloch, Incorporating prior knowledge in support vector machines for classification: A review, *Neurocomputing*, vol.71, pp.1578-1594, 2008.

[13] C. W. Chen and D. Z. Chen, Prior-knowledge-based feedforward network simulation of true boiling point curve of crude oil, *Computers and Chemistry*, vol.25, pp.541-550, 2001.

[14] S. Milanič, S. Strmčnik, D. Šel et al., Incorporating prior knowledge into artificial neural networks – An industrial case study, *Neuralcomputing*, vol.62, pp.131-151, 2004.

[15] W. H. Joerding and J. L. Meador, Encoding a priori information in feedforward networks, *Neural Networks*, vol.4, pp.847-856, 1991.

[16] J. Abonyi, R. Babuska, H. B. Verbruggen and F. Szeifert, Incorporating prior knowledge in fuzzy model identification, *International Journal of Systems Science*, vol.31, pp.657-667, 2000.

[17] P. F. van Lith, B. H. L. Betlem and B. Roffel, Combining prior knowledge with data driven modeling of a batch distillation column including start-up, *Computers and Chemical Engineering*, vol.27, pp.1021-1030, 2003.

[18] P. Lindskog and L. Ljung, Ensuring monotonic gain characteristics in estimated models by fuzzy model structures, *Automatica*, vol.36, pp.311-317, 2000.

[19] J. M. Won, S. Y. Park and J. S. Lee, Parameter conditions for monotonic Takagi-Sugeno-Kang fuzzy system, *Fuzzy Sets and System*, vol.132, pp.135-146, 2002.

[20] C. Li, J. Yi and D. Zhao, Analysis and design of monotonic type-2 fuzzy inference systems, *2009 IEEE International Conference on Fuzzy Systems*, Jeju Island, Korea, pp.1193-1198, 2009.

[21] C. Li, J. Yi and D. Zhao, Interval type-2 fuzzy neural network controller (IT2FNNC) and its application to a coupled-tank liquid-level control system, *Proc. of the 3rd International Conference on Innovative Computing, Information and Control*, Dalian, China, pp.508-511, 2008.

[22] C. Li and J. Yi, SIRMs based interval type-2 fuzzy inference systems: Properties and application, *International Journal of Innovative Computing, Information and Control*, vol.6, no.9, pp.4019-4028, 2010.