ICIC Express Letters ICIC International ©2009 ISSN 1881-803X
Volume **3**, Number **3(B)**, September **2009** pp. 695–700

DESIGN OF INTERVAL TYPE-2 FUZZY LOGIC SYSTEM USING SAMPLED DATA AND PRIOR KNOWLEDGE

Chengdong Li, Jianqiang Yi and Dongbin Zhao

Institute of Automation
Chinese Academy of Sciences
Beijing 100190, P. R. China
{ chengdong.li; jianqiang.yi; dongbin.zhao }@ia.ac.cn

Received March 2009; accepted May 2009

Abstract. *This paper tries to study how to design interval type-2 fuzzy logic systems (IT2FLSs) using both sampled data and prior knowledge. Sufficient conditions on the parameters of IT2FLSs are given to ensure that the prior knowledge of symmetry or odd symmetry, bounded range and monotonicity can be incorporated into IT2FLSs. And, the constrained least squares algorithm is adopted for the design of prior-knowledge-based IT2FLSs (PK-IT2FLSs). Simulation results demonstrate that, if uncertainties exist in the sampled data, then the PK-IT2FLS performs better than the other fuzzy logic systems designed without prior knowledge.*
Keywords: Prior knowledge, Type-2 fuzzy, Symmetry, Bounded range, Monotonicity

1. **Introduction.** In recent years, a number of extensions to traditional fuzzy logic system (type-1 fuzzy logic system: T1FLS) are attracting interest. One of the most widely used extensions is interval type-2 fuzzy logic system (IT2FLS) [1-5]. IT2FLSs not only have the merits of T1FLSs, but also can provide the capability to model high levels of uncertainties [1-3]. Generally speaking, compared with T1FLSs, their corresponding IT2FLSs can produce more complex input-output maps and give better performance, as IT2FLSs utilize interval type-2 fuzzy sets (IT2FSs) which can provide additional degrees of freedom [1-3].

Until now, IT2FLSs are always designed only using the information from sampled data. Sometimes, this data-driven design method for IT2FLSs can achieve satisfactory performance, but, when the information contained in the sampled data is insufficient to reflect system characteristics (this may be caused by small number of sampled data, high levels of uncertainties, etc), the precision and generalization ability of the designed IT2FLSs will be limited. In practical applications, wrong data and noisy data can not be avoided and it is quite difficult to obtain enough sampled data for some plants, hence, the information from sampled data is always insufficient.

On the other hand, although, in most cases, it is hard to obtain exact physical structure knowledge of some complex plants or systems, part of their physical knowledge can be observed easily, such as monotonicity, bounded range, symmetry or odd symmetry, etc. These prior knowledge can partly reflect the characteristics of the unknown plants or systems and offset the insufficiency of the information from sampled data. Recently, the topic on how to utilize prior knowledge has gained considerable concern from different research areas. Impressive results have been accomplished on how to incorporate prior knowledge into support vector machines [6], neural networks [7], T1FLSs [8, 9], etc.

However, to the authors' knowledge, there is no work concerning how to encode the prior knowledge into IT2FLSs. In this paper, we will study this issue and the prior knowledge of symmetry or odd symmetry, bounded range and monotonicity will be considered. First, we will present sufficient conditions on the antecedent IT2FSs and the consequent interval weights of IT2FLSs to ensure that the prior knowledge can be incorporated. Then, we will show how to design prior-knowledge-based IT2FLSs using the constrained least squares algorithm. At last, we will give a simulation to show the usefulness of the prior knowledge and the advantages of the prior-knowledge-based IT2FLSs under noisy circumstances.

2. **Interval Type-2 Fuzzy Logic system.** In this section, we will introduce the inference process of IT2FLS [1-5] briefly.

Suppose that the rule base of an IT2FLS has M fuzzy rules, each of which has the following form

$$R^k: \; x_1 = \widetilde{A}_1^k, \; x_2 = \widetilde{A}_2^k, \; ..., \; x_p = \widetilde{A}_p^k \; \rightarrow y = [\underline{w}^k, \overline{w}^k] \tag{1}$$

where $k = 1, 2, ..., M$, p is the number of the input variables in the antecedent part, $\widetilde{A}_i^k (i = 1, 2, ..., p, k = 1, 2, ..., M)$ are IT2FSs of the IF-part, and $[\underline{w}^k, \overline{w}^k]$s are consequent interval weights of the THEN-part.

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

$$F^k(\boldsymbol{x}) = [\underline{f}^k(\boldsymbol{x}), \overline{f}^k(\boldsymbol{x})] \tag{2}$$

where

$$\underline{f}^k(\boldsymbol{x}) = \underline{\mu}_{\widetilde{A}_1^k}(x_1) \star \underline{\mu}_{\widetilde{A}_2^k}(x_2) \star ... \star \underline{\mu}_{\widetilde{A}_p^k}(x_p) \tag{3}$$

$$\overline{f}^k(\boldsymbol{x}) = \overline{\mu}_{\widetilde{A}_1^k}(x_1) \star \overline{\mu}_{\widetilde{A}_2^k}(x_2) \star ... \star \overline{\mu}_{\widetilde{A}_p^k}(x_p) \tag{4}$$

in which $\underline{\mu}(), \overline{\mu}()$ denote the grades of the lower and upper membership functions of IT2FSs, and $\star$ denotes minimum or product t-norm.

Then, using the center-of-sets (COS) type-reducer [1-3] and the center average defuzzifier, the crisp output of the IT2FLS can be computed as

$$y(\boldsymbol{x}) = \frac{1}{2}(y_l(\boldsymbol{x}) + y_r(\boldsymbol{x})) \tag{5}$$

where $y_l(\boldsymbol{x})$ and $y_r(\boldsymbol{x})$ are the left and right end points of the type-reduced interval set and can be expressed as

$$y_l(\boldsymbol{x}) = \min \left\{ \frac{\sum_{k=1}^M f^k w^k}{\sum_{k=1}^M f^k} \middle| f^k \in F^k(\boldsymbol{x}), w^k \in [\underline{w}^k, \overline{w}^k] \right\} \tag{6}$$

$$= \frac{\sum_{k=1}^M [\underline{\delta}^k \overline{f}^k(\boldsymbol{x}) + (1 - \underline{\delta}^k)\underline{f}^k(\boldsymbol{x})]\underline{w}^k}{\sum_{k=1}^M [\underline{\delta}^k \overline{f}^k(\boldsymbol{x}) + (1 - \underline{\delta}^k)\underline{f}^k(\boldsymbol{x})]} \tag{7}$$

$$y_r(\boldsymbol{x}) = \max \left\{ \frac{\sum_{k=1}^M f^k w^k}{\sum_{k=1}^M f^k} \middle| f^k \in F^k(\boldsymbol{x}), w^k \in [\underline{w}^k, \overline{w}^k] \right\} \tag{8}$$

$$= \frac{\sum_{k=1}^M [\overline{\delta}^k \underline{f}^k(\boldsymbol{x}) + (1 - \overline{\delta}^k)\overline{f}^k(\boldsymbol{x})]\overline{w}^k}{\sum_{k=1}^M [\overline{\delta}^k \underline{f}^k(\boldsymbol{x}) + (1 - \overline{\delta}^k)\overline{f}^k(\boldsymbol{x})]} \tag{9}$$

in which $\underline{\delta}^k$ and $\overline{\delta}^k$ can be determined in $\{0, 1\}$ by Karnik-Mendel algorithms [1-3].

3. **Parameter Conditions of Prior-Knowledge-Based IT2FLS.** In this section, we will present sufficient conditions on the parameters of IT2FLSs to ensure that the prior knowledge can be incorporated. The prior knowledge of odd symmetry or symmetry, bounded range, and monotonicity (increasing or decreasing) will be considered. First, let us consider the prior knowledge of odd symmetry and symmetry.

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

Theorem 3.1. *An IT2FLS is odd symmetric, i.e. $y(\boldsymbol{x}) = -y(-\boldsymbol{x})$, if the fuzzy rules in its rule base satisfy the following conditions:*

1) the same number of fuzzy rules (denoted as $R^{k_1}, R^{k_2}, ..., R^{k_t}$, and $R^{\tilde{k}_1}, R^{\tilde{k}_2}, ..., R^{\tilde{k}_t}$, respectively) can be fired when $\boldsymbol{x}$ and $-\boldsymbol{x}$ are input to the IT2FLS;

2) the firing intervals satisfy that $[\underline{f}^{k_i}(\boldsymbol{x}), \overline{f}^{k_i}(\boldsymbol{x})] = [\underline{f}^{\tilde{k}_i}(-\boldsymbol{x}), \overline{f}^{\tilde{k}_i}(-\boldsymbol{x})]$, $i = 1, 2, ..., t$;

3) the consequent interval weights satisfy that $[\underline{w}^{k_i}, \overline{w}^{k_i}] = [-\overline{w}^{\tilde{k}_i}, -\underline{w}^{\tilde{k}_i}]$, $i = 1, 2, ..., t$.

Proof: If the above conditions 1) 2) and 3) are satisfied, then, from (6) and (8), we have

$$\begin{aligned}
y_l(\boldsymbol{x}) &= \min\left\{ \frac{\sum_{s=1}^{t} f^{k_s} w^{k_s}}{\sum_{s=1}^{t} f^{k_s}} \middle| f^{k_s} \in [\underline{f}^{k_s}(\boldsymbol{x}), \overline{f}^{k_s}(\boldsymbol{x})], w^{k_s} \in [\underline{w}^{k_s}, \overline{w}^{k_s}] \right\} \\
&= -\max\left\{ \frac{\sum_{s=1}^{t} f^{k_s}(-w^{k_s})}{\sum_{s=1}^{t} f^{k_s}} \middle| f^{k_s} \in [\underline{f}^{k_s}(\boldsymbol{x}), \overline{f}^{k_s}(\boldsymbol{x})], -w^{k_s} \in [-\overline{w}^{k_s}, -\underline{w}^{k_s}] \right\} \\
&= -\max\left\{ \frac{\sum_{s=1}^{t} f^{\tilde{k}_s} w^{\tilde{k}_s}}{\sum_{s=1}^{t} f^{\tilde{k}_s}} \middle| f^{\tilde{k}_s} \in [\underline{f}^{\tilde{k}_s}(-\boldsymbol{x}), \overline{f}^{\tilde{k}_s}(-\boldsymbol{x})], w^{\tilde{k}_s} \in [\underline{w}^{\tilde{k}_s}, \overline{w}^{\tilde{k}_s}] \right\} \\
&= -y_r(-\boldsymbol{x}).
\end{aligned} \tag{10}$$

In the same way, $y_r(\boldsymbol{x}) = -y_l(-\boldsymbol{x})$.

Therefore,

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

Theorem 3.2. *An IT2FLS is symmetric, i.e. $y(\boldsymbol{x}) = y(-\boldsymbol{x})$, if the fuzzy rules in its rule base satisfy the following conditions:*

1) the same number of fuzzy rules (denoted as $R^{k_1}, R^{k_2}, ..., R^{k_t}$, and $R^{\tilde{k}_1}, R^{\tilde{k}_2}, ..., R^{\tilde{k}_t}$, respectively) can be fired when $\boldsymbol{x}$ and $-\boldsymbol{x}$ are input to the IT2FLS;

2) the firing intervals satisfy that $[\underline{f}^{k_i}(\boldsymbol{x}), \overline{f}^{k_i}(\boldsymbol{x})] = [\underline{f}^{\tilde{k}_i}(-\boldsymbol{x}), \overline{f}^{\tilde{k}_i}(-\boldsymbol{x})]$, $i = 1, 2, ..., t$;

3) the consequent interval weights satisfy that $[\underline{w}^{k_i}, \overline{w}^{k_i}] = [\underline{w}^{\tilde{k}_i}, \overline{w}^{\tilde{k}_i}]$, $i = 1, 2, ..., t$.

Proof: This theorem can be proved in the similar way as the previous theorem. But, note that, in the proof process of this theorem, $y_l(\boldsymbol{x}) = y_l(-\boldsymbol{x})$ and $y_r(\boldsymbol{x}) = y_r(-\boldsymbol{x})$. $\square$

From Theorem 3.1, it is easy to show that, if the input of the odd symmetric IT2FLS is $\mathbf{0}$, then the output of the odd symmetric IT2FLS is also 0. This property always needs to be satisfied when IT2FLSs are used as fuzzy controllers for control problems.

3.2. **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 3.3. *The output $y(\boldsymbol{x})$ of an IT2FLS falls in the bounded range $B = [\underline{b}, \overline{b}]$, if its consequent interval weights satisfy that $\min\limits_{j=1,\ldots,M}\{\underline{w}^j\} \geq \underline{b}$ and $\max\limits_{j=1,\ldots,M}\{\overline{w}^j\} \leq \overline{b}$.*

Proof: Note that

$$\min_{j=1,\ldots,M}\{\underline{w}^j\} \leq \frac{\sum_{k=1}^{M} f^k \underline{w}^k}{\sum_{k=1}^{M} f^k} \leq \frac{\sum_{k=1}^{M} f^k w^k}{\sum_{k=1}^{M} f^k} \leq \frac{\sum_{k=1}^{M} f^k \overline{w}^k}{\sum_{k=1}^{M} f^k} \leq \max_{j=1,\ldots,M}\{\overline{w}^j\} \tag{11}$$

where $f^k \in F^k(\boldsymbol{x}), w^k \in [\underline{w}^k, \overline{w}^k]$.

From (5) (6) and (8), it is obvious that

$$\min_{j=1,\dots,M}\{\underline{w}^j\} \le y_l(\boldsymbol{x}) \le y_r(\boldsymbol{x}) \le \max_{j=1,\dots,M}\{\overline{w}^j\}. \tag{12}$$

If the conditions in this theorem hold, then

$$\underline{b} \le \min_{j=1,\dots,M}\{\underline{w}^j\} \le y(\boldsymbol{x}) \le \max_{j=1,\dots,M}\{\overline{w}^j\} \le \bar{b}. \tag{13}$$

Hence, this theorem holds. □

3.3. **Prior knowledge of monotonicity.** The monotonicity between the input and output is one particular, but common, type prior knowledge. In [10], we have addressed how to incorporate the monotonicity property into IT2FLSs. And, we have also presented sufficient monotonicity conditions on the parameters of IT2FLSs. For simplicity, here, we only give useful results about the single-input monotonically increasing IT2FLS.

Theorem 3.4. [10] *Assume that the input domain U is partitioned by M pseudotrapezoid IT2FSs for a single-input IT2FLS, then, the IT2FLS is monotonically increasing, if the pseudotrapezoid IT2FSs $\widetilde{A}^1$, $\widetilde{A}^2$, ..., $\widetilde{A}^M$ form fuzzy partition as shown in Figure 1, and its consequent interval weights satisfy that $\underline{w}^1 \le \underline{w}^2 \le \dots \le \underline{w}^M$ and $\overline{w}^1 \le \overline{w}^2 \le \dots \le \overline{w}^M$.*

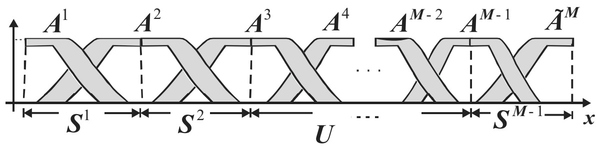

FIGURE 1. Fuzzy partition with M pseudotrapezoid IT2FSs

In this section, parameter constraints of IT2FLSs have been studied to ensure that the aforementioned prior knowledge can be incorporated. In the following section, we will show how to design prior-knowledge-based IT2FLSs using these results.

4. **Design of Prior-Knowledge-Based IT2FLS.** For simplicity, in this work, we only consider the design of prior-knowledge-based single-input IT2FLSs using both sampled data and the aforementioned prior knowledge.

To obtain a satisfactory prior-knowledge-based IT2FLS, two steps are needed. The first step is to set up the IT2FSs in the antecedent part of the IT2FLS, and the second step is to optimize the consequent interval weights under the constraints in Theorem 3.1-3.4. The first step can be accomplished by partitioning the input domains intuitively or through clustering algorithms. In the following, we mainly focus on the second step.

From (5) (7) and (9),

$$y(x) = \boldsymbol{\phi}^{\mathrm{T}}(x)\boldsymbol{w} \tag{14}$$

where $\boldsymbol{w} = [\underline{w}^1, \dots, \underline{w}^M, \overline{w}^1, \dots, \overline{w}^M]^{\mathrm{T}}$, and $\boldsymbol{\phi}(x) = [\phi_1(x), \dots \phi_{2M}(x)]^{\mathrm{T}}$ in which

$$\phi_k(x) = \begin{cases} \frac{1}{2}\frac{\underline{\delta}^k \overline{f}^k(x) + (1-\underline{\delta}^k)\underline{f}^k(x)}{\sum_{k=1}^{M}[\underline{\delta}^k \overline{f}^k(x) + (1-\underline{\delta}^k)\underline{f}^k(x)]}, & k = 1, \dots, M, \\ \frac{1}{2}\frac{\overline{\delta}^{k-M}\underline{f}^{k-M}(x) + (1-\overline{\delta}^{k-M})\overline{f}^{k-M}(x)}{\sum_{k=M+1}^{2M}[\overline{\delta}^{k-M}\underline{f}^{k-M}(x) + (1-\overline{\delta}^{k-M})\overline{f}^{k-M}(x)]}, & k = M+1, \dots, 2M. \end{cases} \tag{15}$$

From (14), we can see that the output of the IT2FLS is linear with its consequent parameters.

Suppose that there are N input-output sampled data (x^1, y^1), (x^2, y^2), ..., (x^N, y^N). And, the training criteria is chosen to minimize the following square error function:

$$E = \sum_{i=1}^{N} |y(x^i) - y^i|^2 = \sum_{i=1}^{N} |\boldsymbol{\phi}^{\mathrm{T}}(x^i)\boldsymbol{w} - y^i|^2 = (\Phi\boldsymbol{w} - \boldsymbol{y})^{\mathrm{T}}(\Phi\boldsymbol{w} - \boldsymbol{y}) \tag{16}$$

where

$$\boldsymbol{y} = [y^1, y^2, ..., y^N]^{\mathrm{T}} \tag{17}$$

$$\Phi = \begin{bmatrix} \boldsymbol{\phi}^{\mathrm{T}}(x^1) \\ \boldsymbol{\phi}^{\mathrm{T}}(x^2) \\ \vdots \\ \boldsymbol{\phi}^{\mathrm{T}}(x^N) \end{bmatrix} = \begin{bmatrix} \phi_1(x^1) & \phi_2(x^1) & \cdots & \phi_{2M}(x^1) \\ \phi_1(x^2) & \phi_2(x^2) & \cdots & \phi_{2M}(x^2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(x^N) & \phi_2(x^N) & \cdots & \phi_{2M}(x^N) \end{bmatrix} \in R^{N \times 2M}. \tag{18}$$

Thus, if the prior knowledge are used, designing an IT2FLS by minimizing the square error function can be seen as solving the least squares (LS) optimization problem with constraints.

The constraints on the consequent parameters in Theorem 3.1 - 3.4 all can be expressed as linear-inequality constraints with the form $C\boldsymbol{w} \leq \boldsymbol{b}$ and/or $C_{eq}\boldsymbol{w} = \boldsymbol{b}_{eq}$. The detailed expressions of C, $\boldsymbol{b}$, C_{eq}, $\boldsymbol{b}_{eq}$ for the aforementioned prior knowledge can be obtained, but, here, we omit these expressions due to page limitation.

All in all, designing a prior-knowledge-based IT2FLS by minimizing the square error function can be transformed to solve the following optimization problem:

$$\min_{\boldsymbol{w}} \ (\Phi\boldsymbol{w} - \boldsymbol{y})^{\mathrm{T}}(\Phi\boldsymbol{w} - \boldsymbol{y}) \tag{19}$$

$$\text{subject to} \quad \begin{cases} C\boldsymbol{w} \leq \boldsymbol{b}, \\ C_{eq}\boldsymbol{w} = \boldsymbol{b}_{eq}. \end{cases} \tag{20}$$

This least squares (LS) optimization problem with linear-inequality constraints can be solved using the MATLAB function *lsqlin*.

5. **Simulations.** In this section, we will give a simulation to show the usefulness of the prior knowledge and the advantages of the prior-knowledge-based IT2FLS under noisy circumstances.

Consider the following function:

$$g(x) = 2 * \frac{\mathrm{e}^{1.2x} - \mathrm{e}^{-1.2x}}{\mathrm{e}^{1.2x} + \mathrm{e}^{-1.2x}} \tag{21}$$

where $x \in U = [-3, 3]$. It is obvious that this function is monotonically increasing, odd symmetric and bounded in [-2, 2].

We use 31 input-output data pairs to identify this function. The ith data pair is (x^i, y^i), where $y^i = g(x^i) + n^i$, in which n^i is the random noise uniformly distributed in $[-0.4, 0.4]$.

For comparison, we design 4 FLSs to identify the function: 1) prior-knowledge-based IT2FLS (PK-IT2FLS), 2) non-prior-knowledge-based IT2FLS (NPK-IT2FLS), 3) prior-knowledge-based T1FLS (PK-T1FLS), 4) non-prior-knowledge-based T1FLS (NPK-T1FLS). And, we use the following error index to evaluate the performances of the four FLSs:

$$\mathrm{EI} = \frac{1}{N} \sum_{i=1}^{N} |y(x^i) - g(x^i)|. \tag{22}$$

This index can reflect the generalization characteristics and robustness to noise of different FLSs. For this index, the less the value is, the better the result is.

For each FLS, we use 7 fuzzy rules, whose antecedent membership functions are shown in Figure 2. Considering the noisy disturbance, four simulations are run. In each simulation, after the consequent weights of the four FLSs being tuned by the MATLAB function *lsqlin*, the error indexes of the four FLSs are computed. The final results are shown in Table 1.

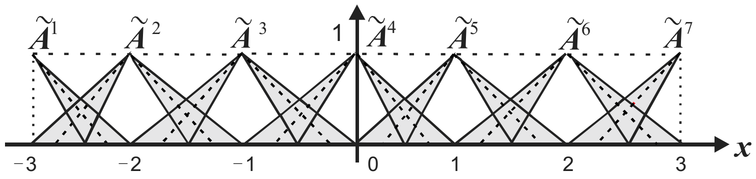

FIGURE 2. Type-1 and Type-2 fuzzy partitions of the input space $[-3, 3]$

TABLE 1. Performance of the four FLSs in different cases

Case	PK-IT2FLS	NPK-IT2FLS	PK-T1FLS	NPK-T1FLS
1	0.0598	0.1066	0.0873	0.1288
2	0.0614	0.1476	0.0906	0.1660
3	0.0792	0.1382	0.1012	0.1482
4	0.0769	0.1637	0.1201	0.1691
Average	0.0693	0.1390	0.0998	0.1530

From Table 1, we can observe that the PK-IT2FLS can realize the actual function best compared with the other three FLSs under noisy circumstances. This implies that prior knowledge and IT2FSs can help to improve the generalization ability of FLSs and make the FLSs more robust to noisy sampled data. Overall, if uncertainties (noise) exist in the sampled data, there is a need to incorporate prior knowledge into IT2FLSs to achieve better generalization performance.

6. **Conclusions.** This study has presented how to design IT2FLSs using both sampled data and prior knowledge. Simulation results and comparisons have shown us that the prior knowledge can help to improve the performance and generalization ability of IT2FLSs. Here, we have just explored the prior knowledge of symmetry or odd symmetry, bounded range and monotonicity. 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 Projects under Grant No.60621001, National 863 Program (No.2007AA04Z239), China.

REFERENCES

[1] Q. Liang and J. M. Mendel, Interval type-2 fuzzy logic systems: theory and design, *IEEE Transactions on Fuzzy Systems*, vol.8, no.5, pp.535-550, 2000.

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

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

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

[5] T. Wang, Y. Chen and S. Tong, Fuzzy reasoning models and algorithms on type-2 fuzzy sets, *Int. J. of Innovative Computing Information and Control*, vol.4, no.10, pp.2451-2460, 2008.

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

[7] 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.

[8] J. Abonyi, R. Babuska, H. B. Verbruggen and F. Szeifert, Incorporating prior knowledge in fuzzy model identification, *International Journal of Systems Science*, vol.31, no.5, pp.657-667, 2000.

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

[10] 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, 2009 (accepted).