An Interval Type-2 Fuzzy Logic System Cannot Be Implemented by Traditional Type-1 Fuzzy Logic Systems

Dongrui Wu, *Member, IEEE*
Industrial Artificial Intelligence Laboratory, GE Global Research, Niskayuna, NY 12309 USA
E-mail: wud@ge.com

***Abstract*—Interval type-2 fuzzy logic systems (IT2 FLSs) have been successfully employed in many applications, and frequently they outperform their type-1 (T1) counterparts. The reasons are still under exploration. This paper points out two fundamental differences between IT2 FLSs and traditional T1 FLSs: 1) *Novelty*, meaning that the upper and lower membership functions of the same IT2 FS may be used simultaneously in computing each bound of the type-reduced interval; and, 2) *adaptiveness*, meaning that the embedded T1 fuzzy sets used to compute the bounds of the type-reduced interval change as input changes. Adaptiveness has been explained in our previous studies; however, this is the first time that novelty is discovered. Based on these fundamental differences, we show that an IT2 FLS with Karnik-Mendel type-reducer cannot be implemented by traditional T1 FLSs. We also examine the reasonableness of five approximations to the Karnik-Mendel algorithms using these two criteria. None of them can simultaneously capture novelty and adaptiveness. The results in this paper will help researchers understand the fundamentals of IT2 FLSs.**

***Index Terms*—Interval type-2 fuzzy logic system, type-1 fuzzy logic system, type-reduction, Karnik-Mendel algorithms**

I. Introduction

Interval type-2 fuzzy logic systems (IT2 FLSs) [16] have been gaining popularity rapidly in the last decade. Many applications, particularly those in control [7]–[9], [11]–[13], [22], [29], [32], [33], image processing [2], [20], [21], and speech recognition [15], [36], have demonstrated that IT2 FLSs can outperform their type-1 (T1) counterparts.

However, one concern of IT2 FLSs is their computational cost, as there is no closed-form formula for computing the output, and usually we rely on the iterative Karnik-Mendel (KM) algorithms [10], [16]. There has been several approaches to expedite this process, which can be grouped into four categories:

1) More efficient implementations of the KM algorithms, e.g., the Enhanced KM algorithms [18], [24], [25], a fast recursive method for computing the generalized centroid of IT2 FSs [14], an iterative algorithm with stop condition for computing the generalized centroid[1] [6], and an iterative algorithm with stop condition and new initialization [**?**].
2) Approximations of the KM algorithms, e.g., the Uncertainty Bounds method [35], which approximates the interval output of the KM algorithms, the equivalent membership grade methods [19], [30], which find an embedded T1 fuzzy set (FS) to replace each IT2 FS, and the linear combination methods [1], [4], which use a linear combination of several T1 FLSs to approximate the IT2 FLS.
3) Alternatives to the KM algorithms, e.g., Coupland and John [3] used the join operation to combine the fired rule consequents into a new IT2 FS and then computed its geometric centroid as the output. Note that this method only works when the rule consequents are IT2 FSs.
4) Simplified IT2 FLS structures, e.g., based on the observation [29], [32] that an IT2 fuzzy logic controller tends to have a smoother control surface in the region around the steady state (both the error and the change of error approach 0), Wu and Tan [28], [33] proposed a structure to use IT2 FSs only for this critical region and T1 FSs for the rest. The simplified IT2 FLS achieved similar performance as traditional IT2 FLSs, but the computational cost was greatly reduced.

This paper focuses on the second category. Our previous studies [26], [27] have shown that the approximations to the KM algorithms are different from the original KM algorithms in that, when the lower membership functions of the IT2 FSs do not cover the input domain completely, the original KM algorithms may have jump discontinuities[2], whereas the approximations do not. This paper takes a closer look at these approximations: Do they capture the essential characteristics of the original KM algorithms? Particularly, can an IT2 FLS with KM type-reducer be accurately implemented by a traditional T1 FLS or a linear combination of several traditional T1 FLSs?

The rest of this paper is organized as follows: Section II presents some background materials on IT2 FSs and FLSs, and two examples to illustrate the operations of an IT2 FLS. Section III points out two fundamental differences between IT2 FLSs and T1 FLSs, and shows that IT2 FLSs with KM

[1] Though the latter two algorithms [6], [14] were initially proposed for computing the generalized centroid of IT2 FSs, they can also be used in type-reduction of IT2 FLSs.

[2] A function $f(x)$ has a *jump discontinuity* at c if $f(c)$ is defined but $\lim_{x\to c^+} f(x) \neq \lim_{x\to c^-} f(x)$, i.e., both $f(c)$ and $f(c+\delta)$ are defined, but $f(c+\delta)$ does not approach $f(c)$ as δ approaches 0.

type-reducer cannot be implemented by T1 FLSs. Section IV introduces five existing approximations to the KM type-reducer and examines their reasonableness using novelty and adaptiveness. Finally, Section V draws conclusions.

II. Interval Type-2 Fuzzy Sets and Systems

A. Interval Type-2 Fuzzy Sets (IT2 FSs)

Definition 1: A *type-1 fuzzy set* X is comprised of a *domain* D_X of real numbers (also called the *universe of discourse* of X) together with a *membership function* (MF) $\mu_X : D_X \to [0,1]$, i.e.,

$$X = \int_{D_X} \mu_X(x)/x. \tag{1}$$

Here $\int$ denotes the collection of all points $x \in D_X$ with associated *membership grade* $\mu_X(x)$. □

Definition 2: [16], [17] An IT2 FS $\tilde{X}$ is characterized by its MF $\mu_{\tilde{X}}(x,u)$, i.e.,

$$\begin{aligned}\widetilde{X} &= \int_{x\in D_{\tilde{X}}} \int_{u\in J_x\subseteq[0,1]} \mu_{\tilde{X}}(x,u)/(x,u) \\ &= \int_{x\in D_{\tilde{X}}} \int_{u\in J_x\subseteq[0,1]} 1/(x,u) \\ &= \int_{x\in D_{\tilde{X}}} \left[\int_{u\in J_x\subseteq[0,1]} 1/u\right] \Big/ x \end{aligned} \tag{2}$$

where x, called the *primary variable*, has domain $D_{\tilde{X}}$; $u \in [0,1]$, called the *secondary variable*, has domain $J_x \subseteq [0,1]$ at each $x \in D_{\tilde{X}}$; J_x is also called the *support of the secondary MF*; and, the amplitude of $\mu_{\tilde{X}}(x,u)$, called a *secondary grade* of $\tilde{X}$, equals 1 for $\forall x \in D_{\tilde{X}}$ and $\forall u \in J_x \subseteq [0,1]$. □

An example of an IT2 FS, $\tilde{X}$, is shown in Fig. 1. Observe that unlike a T1 FS, whose membership for each x is a number, the membership of an IT2 FS is an interval. Observe also that an IT2 FS is bounded from the above and below by two T1 FSs, $\overline{X}$ and $\underline{X}$, which are called *upper membership function* (UMF) and *lower membership function* (LMF), respectively. The area between $\overline{X}$ and $\underline{X}$ is the *footprint of uncertainty*. An *embedded T1 FS* is any T1 FS within the footprint of uncertainty. $\underline{X}$ and $\overline{X}$ are two such sets.

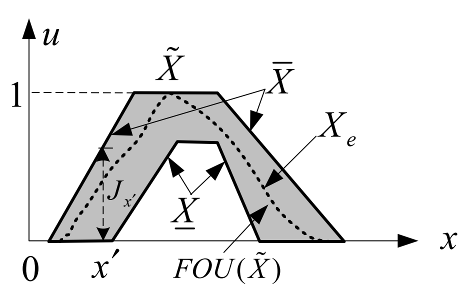

Fig. 1. An IT2 FS. $\underline{X}$ (the LMF), $\overline{X}$ (the UMF), and the dotted curve are three embedded T1 FSs.

B. Interval Type-2 Fuzzy Logic Systems (IT2 FLSs)

Fig. 2 shows the schematic diagram of an IT2 FLS. It is similar to its T1 counterpart, the major difference being that at least one of the FSs in the rule base is an IT2 FS. Hence, the outputs of the inference engine are IT2 FSs, and a type-reducer [10], [16] is needed to convert them into a T1 FS before defuzzification can be carried out.

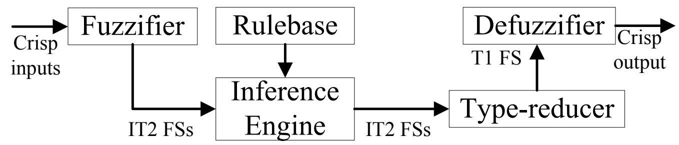

Fig. 2. An IT2 FLS.

In practice the computations in an IT2 FLS can be significantly simplified. Consider the rulebase of an IT2 FLS consisting of N rules assuming the following form:

$$R^n\text{:}\quad \text{IF } x_1 \text{ is } \widetilde{X}_1^n \text{ and } \cdots \text{ and } x_I \text{ is } \widetilde{X}_I^n, \quad \text{THEN } y \text{ is } Y^n \qquad n = 1, 2, ..., N$$

where $\widetilde{X}_i^n$ $(i = 1, \ldots, I)$ are IT2 FSs, and $Y^n = [\underline{y}^n, \overline{y}^n]$ is an interval, which can be understood as the centroid [10], [16] of a consequent IT2 FS[3], or the simplest TSK model. In many applications we use $\underline{y}^n = \overline{y}^n$, i.e., each rule consequent is a crisp number.

For an input vector $\mathbf{x}' = (x_1', x_2', ..., x_I')$, typical computations in an IT2 FLS involve the following steps:

1) Compute the membership of x_i' on each X_i^n, $[\mu_{\underline{X}_i^n}(x_i'), \mu_{\overline{X}_i^n}(x_i')]$, $i = 1, 2, ..., I$, $n = 1, 2, ..., N$.
2) Compute the firing interval of the n^{th} rule, $F^n(\mathbf{x}')$:

$$\begin{aligned}F^n(\mathbf{x}') &= [\mu_{\underline{X}_1^n}(x_1') \times \cdots \times \mu_{\underline{X}_I^n}(x_I'), \\ &\qquad \mu_{\overline{X}_1^n}(x_1') \times \cdots \times \mu_{\overline{X}_I^n}(x_I')] \\ &\equiv [\underline{f}^n, \overline{f}^n], \quad n = 1, ..., N\end{aligned} \tag{3}$$

3) Perform type-reduction to combine $F^n(\mathbf{x}')$ and the corresponding rule consequents. There are many such methods [16]. The most commonly used one is the center-of-sets type-reducer:

$$Y_{cos}(\mathbf{x}') = \bigcup_{\substack{f^n \in F^n(\mathbf{x}') \\ y^n \in Y^n}} \frac{\sum_{n=1}^{N} f^n y^n}{\sum_{n=1}^{N} f^n} = [y_l, \ y_r] \tag{4}$$

It has been shown that [16], [18], [25]:

$$\begin{aligned}y_l &= \min_{k\in[1,N-1]} \frac{\sum_{n=1}^{k} \overline{f}^n \underline{y}^n + \sum_{n=k+1}^{N} \underline{f}^n \underline{y}^n}{\sum_{n=1}^{k} \overline{f}^n + \sum_{n=k+1}^{N} \underline{f}^n} \\ &\equiv \frac{\sum_{n=1}^{L} \overline{f}^n \underline{y}^n + \sum_{n=L+1}^{N} \underline{f}^n \underline{y}^n}{\sum_{n=1}^{L} \overline{f}^n + \sum_{n=L+1}^{N} \underline{f}^n}\end{aligned} \tag{5}$$

[3]The rule consequents can be IT2 FSs; however, when the popular center-of-sets type-reduction method [16] is used, these consequent IT2 FSs are replaced by their centroids in the computation; so, it is more convenient to represent the rule consequents as intervals directly.

$$y_r = \max_{k\in[1,N-1]} \frac{\sum_{n=1}^{k} \underline{f}^n \overline{y}^n + \sum_{n=k+1}^{N} \overline{f}^n \overline{y}^n}{\sum_{n=1}^{k} \underline{f}^n + \sum_{n=k+1}^{N} \overline{f}^n}$$
$$\equiv \frac{\sum_{n=1}^{R} \underline{f}^n \overline{y}^n + \sum_{n=R+1}^{N} \overline{f}^n \overline{y}^n}{\sum_{n=1}^{R} \underline{f}^n + \sum_{n=R+1}^{N} \overline{f}^n} \tag{6}$$

where the *switch points* L and R are determined by

$$\underline{y}^L \leqslant y_l \leqslant \underline{y}^{L+1} \tag{7}$$
$$\overline{y}^R \leqslant y_r \leqslant \overline{y}^{R+1} \tag{8}$$

and $\{\underline{y}^n\}_{n=1,\ldots,N}$ and $\{\overline{y}^n\}_{n=1,\ldots,N}$ have been sorted in ascending order, respectively.

y_l and y_r can be computed by the KM algorithms [10], [16] or Enhanced KM (EKM) algorithms [25]. The main idea of the KM algorithms is to find the switch points for y_l and y_r. Take y_l as an example. y_l is the minimum of $Y_{cos}(\mathbf{x}')$. Since $\underline{y}^n$ increases from the left to the right along the horizontal axis of Fig. 3(a), we should choose a large weight (upper bound of the firing interval) for $\underline{y}^n$ on the left and a small weight (lower bound of the firing interval) for $\underline{y}^n$ on the right. The KM algorithm for y_l finds the switch point L. For $n \leqslant L$, the upper bounds of the firing intervals are used to calculate y_l; for $n > L$, the lower bounds are used. This ensures y_l is the minimum.

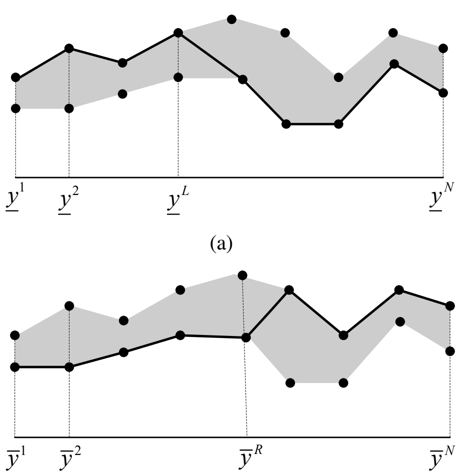

Fig. 3. Illustration of the switch points in computing y_l and y_r. (a) Computing y_l: switch from the upper bounds of the firing intervals to the lower bounds; (b) Computing y_r: switch from the lower bounds of the firing intervals to the upper bounds.

4) Compute the defuzzified output as:

$$y = \frac{y_l + y_r}{2}. \tag{9}$$

C. Examples of IT2 FLS

In this subsection, the mathematical operations in an IT2 FLS, introduced above, are illustrated using two examples.

Consider an IT2 FLS with two inputs (x_1 and x_2) and one output (y). Each input domain consists of two IT2 FSs, shown as the shaded areas in Fig. 4. The rulebase consists of the following four rules:

$$R^1: \text{ IF } x_1 \text{ is } \tilde{X}_{11} \text{ and } x_2 \text{ is } \tilde{X}_{21}, \text{ THEN } y \text{ is } Y^1.$$
$$R^2: \text{ IF } x_1 \text{ is } \tilde{X}_{11} \text{ and } x_2 \text{ is } \tilde{X}_{22}, \text{ THEN } y \text{ is } Y^2.$$
$$R^3: \text{ IF } x_1 \text{ is } \tilde{X}_{12} \text{ and } x_2 \text{ is } \tilde{X}_{21}, \text{ THEN } y \text{ is } Y^3.$$
$$R^4: \text{ IF } x_1 \text{ is } \tilde{X}_{12} \text{ and } x_2 \text{ is } \tilde{X}_{22}, \text{ THEN } y \text{ is } Y^4.$$

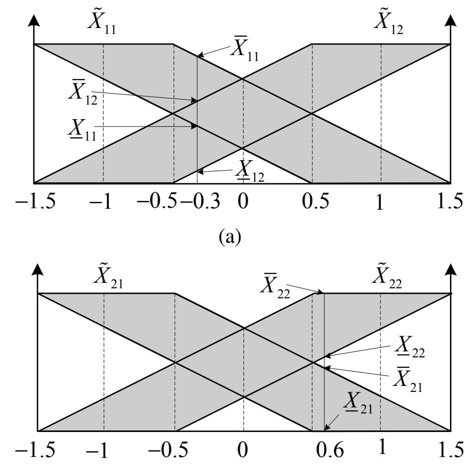

Fig. 4. MFs of the IT2 FLS. (a) Input MFs of x_1; (b) Input MFs of x_2.

The corresponding rule consequents are given in Table I.

TABLE I
RULEBASE AND CONSEQUENTS OF THE IT2 FLS.

x_1 \ x_2	$\tilde{X}_{21}$	$\tilde{X}_{22}$
$\tilde{X}_{11}$	$Y^1 = [\underline{y}^1, \overline{y}^1] = [-1, -0.9]$	$Y^2 = [\underline{y}^2, \overline{y}^2] = [-0.6, -0.4]$
$\tilde{X}_{12}$	$Y^3 = [\underline{y}^3, \overline{y}^3] = [0.4, 0.6]$	$Y^4 = [\underline{y}^4, \overline{y}^4] = [0.9, 1]$

Example 1: Consider an input vector $\mathbf{x}' = (x_1', x_2') = (-0.3, 0.6)$, as shown in Fig. 4. The firing intervals of the four IT2 FSs are:

$$[\mu_{\underline{X}_{11}}(x_1'), \mu_{\overline{X}_{11}}(x_1')] = [0.4,\ 0.9] \tag{10}$$
$$[\mu_{\underline{X}_{12}}(x_1'), \mu_{\overline{X}_{12}}(x_1')] = [0.1,\ 0.6] \tag{11}$$
$$[\mu_{\underline{X}_{21}}(x_2'), \mu_{\overline{X}_{21}}(x_2')] = [0,\ 0.45] \tag{12}$$
$$[\mu_{\underline{X}_{22}}(x_2'), \mu_{\overline{X}_{22}}(x_2')] = [0.55,\ 1] \tag{13}$$

The firing intervals of the four rules are shown in Table II. From the KM algorithms, we find that $L = 1$ and $R = 3$. So,

$$y_l = \frac{\overline{f}^1 \underline{y}^1 + \underline{f}^2 \underline{y}^2 + \underline{f}^3 \underline{y}^3 + \underline{f}^4 \underline{y}^4}{\overline{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4} \tag{14}$$
$$= \frac{0.405 \times (-1) + 0.22 \times (-0.6) + 0 \times 0.4 + 0.055 \times 0.9}{0.405 + 0.22 + 0 + 0.055}$$
$$= -0.7169$$
$$y_r = \frac{\underline{f}^1 \overline{y}^1 + \underline{f}^2 \overline{y}^2 + \underline{f}^3 \overline{y}^3 + \overline{f}^4 \overline{y}^4}{\underline{f}^1 + \underline{f}^2 + \underline{f}^3 + \overline{f}^4}$$

$$= \frac{0 \times (-0.9) + 0.22 \times (-0.4) + 0 \times 0.6 + 0.6 \times 1}{0 + 0.22 + 0 + 0.6}$$
$$= 0.6244$$

Finally, the crisp output of the IT2 FLS, y, is:

$$y = \frac{y_l + y_r}{2} = \frac{-0.7169 + 0.6244}{2} = -0.0463. \quad \square$$

Example 2: Consider another input vector $\mathbf{x}' = (x_1', x_2') = (0.3, 0.6)$, as shown in Fig. 5. The firing intervals of the four IT2 FSs are:

$$[\mu_{\underline{X}_{11}}(x_1'), \mu_{\overline{X}_{11}}(x_1')] = [0.1,\ 0.6] \tag{15}$$
$$[\mu_{\underline{X}_{12}}(x_1'), \mu_{\overline{X}_{12}}(x_1')] = [0.4,\ 0.9] \tag{16}$$
$$[\mu_{\underline{X}_{21}}(x_2'), \mu_{\overline{X}_{21}}(x_2')] = [0,\ 0.45] \tag{17}$$
$$[\mu_{\underline{X}_{22}}(x_2'), \mu_{\overline{X}_{22}}(x_2')] = [0.55,\ 1] \tag{18}$$

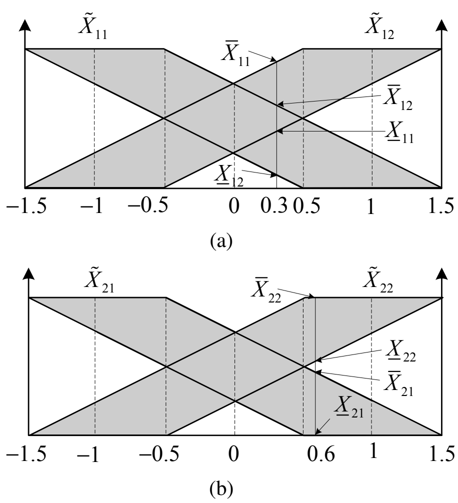

Fig. 5. Inputs in Example 2.

The firing intervals of the four rules are shown in Table III. From the KM algorithms, we find that $L = 2$ and $R = 3$. So,

$$y_l = \frac{\overline{f}^1 \underline{y}^1 + \overline{f}^2 \underline{y}^2 + \underline{f}^3 \underline{y}^3 + \underline{f}^4 \underline{y}^4}{\overline{f}^1 + \overline{f}^2 + \underline{f}^3 + \underline{f}^4} \tag{19}$$
$$= \frac{0.27 \times (-1) + 0.6 \times (-0.6) + 0 \times 0.4 + 0.22 \times 0.9}{0.27 + 0.6 + 0 + 0.22}$$
$$= -0.3963$$
$$y_r = \frac{\underline{f}^1 \overline{y}^1 + \underline{f}^2 \overline{y}^2 + \underline{f}^3 \overline{y}^3 + \overline{f}^4 \overline{y}^4}{\underline{f}^1 + \underline{f}^2 + \underline{f}^3 + \overline{f}^4}$$
$$= \frac{0 \times (-0.9) + 0.055 \times (-0.4) + 0 \times 0.6 + 0.9 \times 1}{0 + 0.055 + 0 + 0.9}$$
$$= 0.9194$$

Finally, the crisp output of the IT2 FLS, y, is:

$$y = \frac{y_l + y_r}{2} = \frac{-0.3963 + 0.9194}{2} = 0.2615. \quad \square$$

III. Can an IT2 FLS Be Implemented by Traditional T1 FLSs?

Observe from (9), and also the two examples, that the output of an IT2 FLS is the average of two "T1 FLSs". However, these two "T1 FLSs" are fundamentally different from traditional T1 FLSs, for the following reasons:

1) **Novelty:** Take y_l in (14) as an example. The firing levels of the four rules are $\overline{f}^1$, $\underline{f}^2$, $\underline{f}^3$ and $\underline{f}^4$, respectively, which are computed from different lower and upper MFs, as shown in the first part of Table IV, and Fig. 6(a). Observe that both the upper and lower MFs of $\tilde{X}_{11}$ are used in computing y_l, and they are used in different rules: The UMF of $\tilde{X}_{11}$ is used in computing $\overline{f}^1$, the firing level of Rule R^1, whereas the LMF of $\tilde{X}_{11}$ is used in computing $\underline{f}^2$, the firing level of Rule R^2. Similarly, the upper and lower MFs of $\tilde{X}_{21}$ are used simultaneously in different rules for computing y_l. Observe also from the second part of Table IV and Fig. 6(b) that the upper and lower MFs of $\tilde{X}_{21}$ and $\tilde{X}_{22}$ are used simultaneously in different rules for computing y_l. This novelty is impossible for a traditional T1 FLS, where the same MFs are always used in computing the firing levels of all rules.
2) **Adaptiveness:** Comparing the two parts of Table IV, and the two sub-figures in Fig. 6, we observe that when the input (x_1', x_2') changes from $(-0.3, 0.6)$ to $(0.3, 0.6)$, different embedded T1 FSs are used in computing the four firing levels and hence y_l, which has been discovered in [31]. This adaptiveness is again impossible for a traditional T1 FLS.

TABLE IV

The embedded T1 FSs from which the four firing levels in (14) and (19) are obtained.

		$\tilde{X}_{11}$		$\tilde{X}_{12}$		$\tilde{X}_{21}$		$\tilde{X}_{22}$	
		UMF	LMF	UMF	LMF	UMF	LMF	UMF	LMF
Equation (14) $(x_1', x_2') = (-0.3, 0.6)$	$\overline{f}^1$	√				√			
	$\underline{f}^2$		√						√
	$\underline{f}^3$				√		√		
	$\underline{f}^4$				√				√
Equation (19) $(x_1', x_2') = (0.3, 0.6)$	$\overline{f}^1$	√				√			
	$\overline{f}^2$	√						√	
	$\underline{f}^3$				√		√		
	$\underline{f}^4$				√				√

We consider the novelty and adaptiveness as two fundamental differences between IT2 FLSs and T1 FLSs. Though they are illustrated by specific examples, the conclusion can be extended to arbitrary IT2 FLSs.

Theorem 1: y_l in (5) cannot be implemented by a traditional T1 FLS. $\square$

Proof: In this proof we make use of the following two facts:

TABLE II
FIRING INTERVALS OF THE FOUR RULES IN EXAMPLE 1.

Rule No.:	Firing Interval	$\rightarrow$	Rule Consequent
R^1 :	$[\underline{f}^1, \overline{f}^1] = [\mu_{\underline{X}_{11}}(x_1') \cdot \mu_{\underline{X}_{21}}(x_2'), \mu_{\overline{X}_{11}}(x_1') \cdot \mu_{\overline{X}_{21}}(x_2')]$ $= [0.4 \times 0, 0.9 \times 0.45] = [0,\ 0.405]$	$\rightarrow$	$[\underline{y}^1, \overline{y}^1] = [-1, -0.9]$
R^2 :	$[\underline{f}^2, \overline{f}^2] = [\mu_{\underline{X}_{11}}(x_1') \cdot \mu_{\underline{X}_{22}}(x_2'), \mu_{\overline{X}_{11}}(x_1') \cdot \mu_{\overline{X}_{22}}(x_2')]$ $= [0.4 \times 0.55, 0.9 \times 1] = [0.22,\ 0.9]$	$\rightarrow$	$[\underline{y}^2, \overline{y}^2] = [-0.6, -0.4]$
R^3 :	$[\underline{f}^3, \overline{f}^3] = [\mu_{\underline{X}_{12}}(x_1') \cdot \mu_{\underline{X}_{21}}(x_2'), \mu_{\overline{X}_{12}}(x_1') \cdot \mu_{\overline{X}_{21}}(x_2')]$ $= [0.1 \times 0, 0.6 \times 0.45] = [0,\ 0.27]$	$\rightarrow$	$[\underline{y}^3, \overline{y}^3] = [0.4, 0.6]$
R^4 :	$[\underline{f}^4, \overline{f}^4] = [\mu_{\underline{X}_{12}}(x_1') \cdot \mu_{\underline{X}_{22}}(x_2'), \mu_{\overline{X}_{12}}(x_1') \cdot \mu_{\overline{X}_{22}}(x_2')]$ $= [0.1 \times 0.55, 0.6 \times 1] = [0.055,\ 0.6]$	$\rightarrow$	$[\underline{y}^4, \overline{y}^4] = [0.9, 1]$

TABLE III
FIRING INTERVALS OF THE FOUR RULES IN EXAMPLE 2.

Rule No.:	Firing Interval	$\rightarrow$	Rule Consequent
R^1 :	$[\underline{f}^1, \overline{f}^1] = [\mu_{\underline{X}_{11}}(x_1') \cdot \mu_{\underline{X}_{21}}(x_2'), \mu_{\overline{X}_{11}}(x_1') \cdot \mu_{\overline{X}_{21}}(x_2')]$ $= [0.1 \times 0, 0.6 \times 0.45] = [0,\ 0.27]$	$\rightarrow$	$[\underline{y}^1, \overline{y}^1] = [-1, -0.9]$
R^2 :	$[\underline{f}^2, \overline{f}^2] = [\mu_{\underline{X}_{11}}(x_1') \cdot \mu_{\underline{X}_{22}}(x_2'), \mu_{\overline{X}_{11}}(x_1') \cdot \mu_{\overline{X}_{22}}(x_2')]$ $= [0.1 \times 0.55, 0.6 \times 1] = [0.055,\ 0.6]$	$\rightarrow$	$[\underline{y}^2, \overline{y}^2] = [-0.6, -0.4]$
R^3 :	$[\underline{f}^3, \overline{f}^3] = [\mu_{\underline{X}_{12}}(x_1') \cdot \mu_{\underline{X}_{21}}(x_2'), \mu_{\overline{X}_{12}}(x_1') \cdot \mu_{\overline{X}_{21}}(x_2')]$ $= [0.4 \times 0, 0.9 \times 0.45] = [0,\ 0.405]$	$\rightarrow$	$[\underline{y}^3, \overline{y}^3] = [0.4, 0.6]$
R^4 :	$[\underline{f}^4, \overline{f}^4] = [\mu_{\underline{X}_{12}}(x_1') \cdot \mu_{\underline{X}_{22}}(x_2'), \mu_{\overline{X}_{12}}(x_1') \cdot \mu_{\overline{X}_{22}}(x_2')]$ $= [0.4 \times 0.55, 0.9 \times 1] = [0.22,\ 0.9]$	$\rightarrow$	$[\underline{y}^4, \overline{y}^4] = [0.9, 1]$

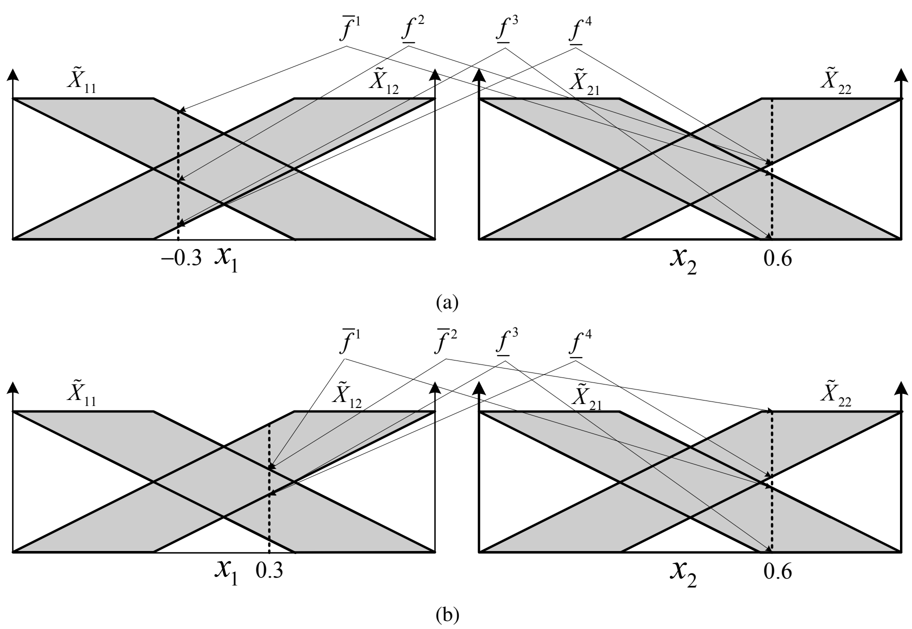

Fig. 6. The embedded T1 FSs used in (a) Equation (14), where $(x_1', x_2') = (-0.3, 0.6)$, and (b) Equation (19), where $(x_1', x_2') = (0.3, 0.6)$.

1) *Fact 1:* The rule firing levels used in the KM algorithms are the bounds of the firing intervals. For an upper bound, all involved embedded T1 FSs must be UMFs, and for a lower bound, all involved embedded T1 FSs must be LMFs. There is no mixture of UMFs and LMFs in computing the firing level of any rule.
2) *Fact 2:* $\overline{f}^1$ and $\underline{f}^N$ are always used for computing y_l in (5), though we are not sure about whether the upper or lower firing levels should be used for the rest of the rules. For $\overline{f}^1$, all involved embedded T1 FSs must be UMFs. For $\underline{f}^N$, all involved embedded T1 FSs must be LMFs.

We consider two cases separately:

1) *Rules R^1 and R^N share at least one IT2 FS $\tilde{X}_i$.* In this case, according to Fact 2, for Rule R^1 $\overline{X}_i$ must be used, whereas for Rule R^N, $\underline{X}_i$ must be used. This novelty cannot be implemented by a traditional T1 FLS.
2) *Rules R^1 and R^N do not have any IT2 FS in common,* (e.g., for y_l in (14), R^1 involves $\tilde{X}_{11}$ and $\tilde{X}_{21}$, whereas R^4 involves $\tilde{X}_{12}$ and $\tilde{X}_{22}$). This case is more complicated than the previous one. We prove it by contradiction. Assume y_l in (5) can be implemented by a traditional T1 FLS, where the same T1 MFs are used in computing all firing levels, e.g., if the UMF of $\tilde{X}_{11}$ is used in computing the firing level of Rule R^1, it must also be used in computing the firing levels of all other rules involving $\tilde{X}_{11}$.
   In this second case, it is always possible to find a Rule R^k such that Rules R^1 and R^k share at least one common IT2 FS $\tilde{X}_i$, and Rules R^k and R^N share at least one common IT2 FS $\tilde{X}_j$ (e.g., for y_l in (14), Rules R^1 and R^2 share $\tilde{X}_{11}$, and Rules R^2 and R^4 share $\tilde{X}_{22}$). According to Fact 2, $\overline{X}_i$ must be used in Rule R^1 for computing $\overline{f}^1$. If y_l can be implemented by a traditional T1 FLS, then $\overline{X}_i$ must also be used in Rule R^k. According to Fact 1, $\overline{X}_j$ must also be used for Rule R^k. For a traditional T1 FLS, this means $\overline{X}_j$ must also be used in Rule R^N, which is a contradiction with Fact 2. So, again y_l in (5) cannot be implemented by a traditional T1 FLS. □

Theorem 2: y_r in (6) cannot be implemented by a traditional T1 FLS. □

The proof is very similar to that for Theorem 1, so it is omitted.

Base on Theorems 1 and 2, we can easily reach the following conclusion:

Theorem 3: An IT2 FLS using the KM type-reducer cannot be implemented by a traditional T1 FLS or a linear combination of several traditional T1 FLSs. □

Theorem 3 may help understand why IT2 FLSs can outperform T1 FLSs. There has been several attempts to answer this fundamental question. The earliest argument is that, because of the footprint of uncertainty, an IT2 FS has more degrees of freedom than a T1 FS (i.e., more parameters are needed to describe an IT2 FS than a T1 FS); hence, IT2 FLSs have the potential to outperform T1 FLSs. Wu and Tan [29], [32] and Jammeh et al. [9] have shown that an IT2 fuzzy logic controller can outperform its T1 counterpart because it gives a smoother control surface, especially in the region around the steady state (both the error and the change of error approach 0). Wu and Tan [31] also showed that an IT2 FS is equivalent to a group of T1 FSs (instead of one T1 FS), and which T1 FS should be used depends on the input to the IT2 FLS. Du and Ying [5] showed that an IT2 fuzzy-PI (or the corresponding PD) controller is equivalent to a nonlinear PI (or PD) controller with variable gains and control offset. Wu and Tan [34] further showed that when the baseline T1 FLS implements a linear PI control law and the IT2 FSs of an IT2 FLS are obtained from the symmetrical perturbation of T1 FSs, the IT2 FLS implements a variable gain PI controller near the origin, and the variable gains depend on the inputs. They also gave the closed-form solution of the equivalent PI gains near the origin. More recently, Wu [23] used P-map to visualize the difference between a T1 FLS and an IT2 FLS obtained from perturbing the T1 FLS. He showed that the perturbation results in globally smaller proportional gains, and hence the IT2 FLS is slower than the baseline T1 FLS. Theorem 3 suggests that an IT2 FLS can implement more complex mappings than traditional T1 FLSs: when there is no FOU, an IT2 FLS collapses to a T1 FLS; with FOU, an IT2 FLS can implement a mapping that cannot be obtained from a traditional T1 FLS.

IV. Reasonableness of the Five Approximations to the KM Type-Reducer

As mentioned in the Introduction, there are several approximations to the KM type-reducer. With the two fundamental differences between IT2 FLSs and T1 FLSs, and Theorems 1-3, in mind, it is interesting to examine their reasonableness. But first, these approximations are introduced in more details.

A. The Five Approximations

The five approximations [1], [4], [19], [30], [35] to the KM type-reducer are:

1) **The Uncertainty Bound (UB) Method:** The UB type-reducer, proposed by Wu and Mendel [35], computes the output of the IT2 FLS by (9), but,

$$y_l = \frac{\underline{y}_l + \overline{y}_l}{2} \tag{20}$$

$$y_r = \frac{\underline{y}_r + \overline{y}_r}{2} \tag{21}$$

where

$$\overline{y}_l = \min\{\underline{y}^{(0)}, \underline{y}^{(N)}\} \tag{22}$$

$$\underline{y}_r = \max\{\overline{y}^{(0)}, \overline{y}^{(N)}\} \tag{23}$$

$$\underline{y}_l = \overline{y}_l - \frac{\sum_{n=1}^N (\overline{f}^n - \underline{f}^n)}{\sum_{n=1}^N \overline{f}^n \sum_{n=1}^N \underline{f}^n} \frac{\sum_{n=1}^N \underline{f}^n (\underline{y}^n - \underline{y}_1) \sum_{n=1}^N \overline{f}^n (\underline{y}^n - \underline{y}^n)}{\sum_{n=1}^N \underline{f}^n (\underline{y}^n - \underline{y}^1) + \sum_{n=1}^N \overline{f}^n (\underline{y}^n - \underline{y}^n)} \tag{24}$$

$$\overline{y}_r = \underline{y}_r + \frac{\sum_{n=1}^N (\overline{f}^n - \underline{f}^n)}{\sum_{n=1}^N \overline{f}^n \sum_{n=1}^N \underline{f}^n}$$
$$\frac{\sum_{n=1}^N \overline{f}^n (\overline{y}^n - \overline{y}_1) \sum_{n=1}^N \underline{f}^n (\overline{y}^n - \overline{y}^n)}{\sum_{n=1}^N \overline{f}^n (\overline{y}^n - \overline{y}^1) + \sum_{n=1}^N \underline{f}^n (\overline{y}^n - \overline{y}^n)} \quad (25)$$

in which

$$\underline{y}^{(0)} = \frac{\sum_{n=1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^N \underline{f}^n} \quad (26)$$

$$\underline{y}^{(N)} = \frac{\sum_{n=1}^N \overline{f}^n \underline{y}^n}{\sum_{n=1}^N \overline{f}^n} \quad (27)$$

$$\overline{y}^{(N)} = \frac{\sum_{n=1}^N \underline{f}^n \overline{y}^n}{\sum_{n=1}^N \underline{f}^n} \quad (28)$$

$$\overline{y}^{(0)} = \frac{\sum_{n=1}^N \overline{f}^n \overline{y}^n}{\sum_{n=1}^N \overline{f}^n} \quad (29)$$

The UB method explicitly considers the case that the rule consequents are intervals $[\underline{y}^n, \overline{y}^n]$. For the other four methods introduced below, $\underline{y}^n = \overline{y}^n \equiv y^n$ is used.

2) **The Wu-Tan (WT) Method:** Wu and Tan [30] proposed a closed-form type-reduction and defuzzification method by making use of the equivalent T1 membership grades [31]. The basic idea is to first find an equivalent T1 membership grade $\mu_{X_i^n}(x_i)$ to replace each firing interval $[\mu_{\underline{X}_i^n}(x_i), \mu_{\overline{X}_i^n}(x_i)]$, i.e.,

$$\mu_{X_i^n}(x_i) = \mu_{\overline{X}_i^n}(x_i) - h_i^n(\mathbf{x})[\mu_{\overline{X}_i^n}(x_i) - \mu_{\underline{X}_i^n}(x_i)] \quad (30)$$

where $h_i^n(\mathbf{x})$ is a function of the inputs $\mathbf{x}$, and is different for different IT2 FSs. Then, the firing strengths of the rules become point (instead of interval) numbers computed from these $\mu_{X_i^n}(x_i)$, and the output of the IT2 FLS is then computed as

$$y = \frac{\sum_{n=1}^N f^n y^n}{\sum_{n=1}^N f^n}. \quad (31)$$

3) **The Nie-Tan (NT) Method:** Nie and Tan [19] proposed another closed-form type-reduction and defuzzification method, where the output of an IT2 FLS is computed as:

$$y = \frac{\sum_{n=1}^N (\underline{f}^n + \overline{f}^n) y^n}{\sum_{n=1}^N (\underline{f}^n + \overline{f}^n)}. \quad (32)$$

Observe that the NT method is a special case of the WT method when $h_i^n(\mathbf{x}) = 0.5$.

4) **The Du-Ying (DY) Method:** Du and Ying [4], [5] proposed a different closed-form type-reduction and defuzzification method. It first computes the crisp outputs obtained by all possible combinations of the lower and upper firing levels, i.e.,

$$y_m = \frac{\sum_{n=1}^N f^{n^*} y^n}{\sum_{n=1}^N f^{n^*}}, \quad m = 1, 2, ..., 2^N \quad (33)$$

where $f^{n^*} \in \{\underline{f}^n, \overline{f}^n\}$. The final defuzzified output is then computed as the average of all these 2^N y_m, i.e.,

$$y = \frac{1}{2^N} \sum_{m=1}^{2^N} y_m. \quad (34)$$

5) **The Begian-Melek-Mendel (BMM) Method:** Begian, Melek and Mendel [1] proposed another closed-form type-reduction and defuzzification method for TSK IT2 FLSs, i.e.,

$$y = \alpha \frac{\sum_{n=1}^N \underline{f}^n y^n}{\sum_{n=1}^N \underline{f}^n} + \beta \frac{\sum_{n=1}^N \overline{f}^n y^n}{\sum_{n=1}^N \overline{f}^n} \quad (35)$$

where α and β are adjustable coefficients.

B. Reasonableness of the Five Approximations

We check the reasonableness of the five approximations to the KM type-reducer using novelty, adaptiveness, and Theorem 3. The observations are:

1) The NT method, DY method, and BMM method use a traditional T1 FLS or a linear combination of several traditional T1 FLSs to approximate the KM type-reducer. According to Theorem 3, they cannot accurately duplicate the output of IT2 FLSs with KM type-reducer.
2) The WT method explicitly captures adaptiveness in the sense that the embedded T1 FSs used to construct the T1 FLS change as input changes. However, whether there exists a group of $h_i^n(\mathbf{x})$ to also capture the novelty is an open problem.
3) For the UB method, $\underline{y}_l$ and $\overline{y}_r$ involve complex combinations which cannot be decomposed into traditional T1 FLSs, thus the results in this paper cannot be directly applied to it. $\overline{y}_l$ and $\underline{y}_r$ exhibit limited adaptiveness as $\overline{y}_l$ can be chosen from $\underline{y}^{(0)}$ and $\underline{y}^{(N)}$, and $\underline{y}_r$ can be chosen from $\overline{y}^{(0)}$ and $\overline{y}^{(N)}$. However, these terms do not incorporate novelty.

V. Conclusions

In this paper, we have pointed out two fundamental differences between IT2 FLSs and traditional T1 FLSs: 1) Novelty, meaning that the UMF and LMF of the same IT2 FS may be used simultaneously in computing each bound of the type-reduced interval; and, 2) adaptiveness, meaning that the embedded T1 FSs used to compute the bounds of the type-reduced interval change as input changes. Adaptiveness has been discovered in our previous studies; however, this is the first time that novelty is brought to our attention. We have also showed that an IT2 FLS with KM type-reducer cannot be implemented by traditional T1 FLSs. Finally, we have also examined the reasonableness of five approximations to the KM algorithms using novelty and adaptiveness. None of them can simultaneously capture novelty and adaptiveness. The results in this paper will help researchers understand the fundamentals of IT2 FLSs.

REFERENCES

[1] M. Begian, W. Melek, and J. Mendel, "Stability analysis of type-2 fuzzy systems," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Hong Kong, June 2008, pp. 947–953.

[2] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, and J. Sanz, "Comment on: "image thresholding using type II fuzzy sets. importance of this method," *Pattern Recognition*, vol. 43, no. 9, pp. 3188–3192, 2010.

[3] S. Coupland and R. I. John, "Geometric type-1 and type-2 fuzzy logic systems," *IEEE Trans. on Fuzzy Systems*, vol. 15, no. 1, pp. 3–15, 2007.

[4] X. Du and H. Ying, "Control performance comparison between a type-2 fuzzy controller and a comparable conventional mamdani fuzzy controller," in *Proc. Annual Meeting of the North American Fuzzy Information Processing Society*, San Diego, CA, June 2007, pp. 100–105.

[5] ——, "Derivation and analysis of the analytical structures of the interval type-2 fuzzy-PI and PD controllers," *IEEE Trans. on Fuzzy Systems*, vol. 18, no. 4, pp. 802–814, 2010.

[6] K. Duran, H. Bernal, and M. Melgarejo, "Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set," in *Proc. Annual Meeting of the North American Fuzzy Information Processing Society*, New York, May 2008, pp. 1–5.

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

[8] ——, "Type-2 FLCs: A new generation of fuzzy controllers," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 30–43, 2007.

[9] E. A. Jammeh, M. Fleury, C. Wagner, H. Hagras, and M. Ghanbari, "Interval type-2 fuzzy logic congestion control for video streaming across IP networks," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 5, pp. 1123–1142, 2009.

[10] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, pp. 195–220, 2001.

[11] Q. Liang, N. N. Karnik, and J. M. Mendel, "Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 30, no. 3, pp. 329–339, 2000.

[12] Z. Liu, Y. Zhang, and Y. Wang, "A type-2 fuzzy switching control system for biped robots," *IEEE Trans. on Systems, Man, and Cybernetics–C*, vol. 37, no. 6, pp. 1202–1213, 2007.

[13] R. Martinez-Marroquin, O. Castillo, and L. T. Aguilar, "Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms," *Information Sciences*, vol. 179, no. 13, pp. 2158–2174, 2009.

[14] M. Melgarejo, "A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set," in *Proc. Annual Meeting of the North American Fuzzy Information Processing Society*, San Diego, CA, June 2007, pp. 190–194.

[15] P. Melin, J. Urias, D. Solano, M. Soto, M. Lopez, and O. Castillo, "Voice recognition with neural networks, type-2 fuzzy logic and genetic algorithms," *Journal of Engineering Letters*, vol. 13, no. 2, pp. 108–116, 2006.

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

[17] J. M. Mendel and R. I. John, "Type-2 fuzzy sets made simple," *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, 2002.

[18] J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*. Hoboken, NJ: Wiley-IEEE Press, 2010.

[19] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Hong Kong, June 2008, pp. 1425–1432.

[20] C.-M. Own, H.-H. Tsai, P.-T. Yu, and Y.-J. Lee, "Adaptive type-2 fuzzy median filter design for removal of impulse noise," *Imaging Science*, vol. 54, no. 1, pp. 3–18, 2006.

[21] F. C.-H. Rhee, "Uncertainty fuzzy clustering: Insights and recommendations," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 44–56, 2007.

[22] R. Sepulveda, O. Castillo, P. Melin, A. R. Diaz, and O. Montiel, "Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic," *Information Sciences*, vol. 177, no. 10, pp. 2023–2048, 2007.

[23] D. Wu, "P-map: An intuitive plot to visualize, understand, and compare variable-gain PI controllers," in *Proc. Int'l. Conf. on Autonomous and Intelligent Systems*, Burnaby, BC, Canada, June 2011.

[24] D. Wu and J. M. Mendel, "Enhanced Karnik-Mendel Algorithms for interval type-2 fuzzy sets and systems," in *Proc. Annual Meeting of the North American Fuzzy Information Processing Society*, San Diego, CA, June 2007, pp. 184–189.

[25] ——, "Enhanced Karnik-Mendel Algorithms," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.

[26] ——, "Examining the continuity of type-1 and interval type-2 fuzzy logic systems," in *Proc. IEEE World Congress on Computational Intelligence*, Barcelona, Spain, July 2010.

[27] ——, "On the continuity of type-1 and interval type-2 fuzzy logic systems," *IEEE Trans. on Fuzzy Systems*, vol. 19, no. 1, pp. 179–192, 2011.

[28] D. Wu and W. W. Tan, "A simplified architecture for type-2 FLSs and its application to nonlinear control," in *Proc. IEEE Conf. on Cybernetics and Intelligent Systems*, Singapore, Dec. 2004, pp. 485–490.

[29] ——, "A type-2 fuzzy logic controller for the liquid-level process," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, vol. 2, Budapest, Hungary, July 2004, pp. 953–958.

[30] ——, "Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Reno, NV, May 2005, pp. 353–358.

[31] ——, "Type-2 FLS modeling capability analysis," in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, Reno, NV, May 2005, pp. 242–247.

[32] ——, "Genetic learning and performance evaluation of type-2 fuzzy logic controllers," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 829–841, 2006.

[33] ——, "A simplified type-2 fuzzy controller for real-time control," *ISA Transactions*, vol. 15, no. 4, pp. 503–516, 2006.

[34] ——, "Interval type-2 fuzzy PI controllers: Why they are more robust," in *Proc. IEEE Int'l. Conf. on Granular Computing*, San Jose, CA, August 2010, pp. 802–807.

[35] H. Wu and J. M. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Trans. on Fuzzy Systems*, vol. 10, no. 5, pp. 622–639, 2002.

[36] J. Zeng and Z.-Q. Liu, "Type-2 fuzzy hidden Markov models and their applications to speech recognition," *IEEE Trans. on Fuzzy Systems*, vol. 14, no. 3, pp. 454–467, 2006.