Available online at www.sciencedirect.com

ScienceDirect

Information Sciences 177 (2007) 84–110

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Advances in type-2 fuzzy sets and systems

Jerry M. Mendel *

Signal and Image Processing Institute, Department of Electrical Engineering, University of Southern California, 3740 McClintock Avenue, Los Angeles, CA 90089-2564, United States

Received 10 January 2006; received in revised form 30 April 2006; accepted 7 May 2006

Abstract

In this state-of-the-art paper, important advances that have been made during the past five years for both general and interval type-2 fuzzy sets and systems are described. Interest in type-2 subjects is worldwide and touches on a broad range of applications and many interesting theoretical topics. The main focus of this paper is on the theoretical topics, with descriptions of what they are, what has been accomplished, and what remains to be done.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Type-2 fuzzy sets; Type-2 fuzzy systems; Interval type-2 fuzzy sets; Interval type-2 fuzzy logic systems; Centroid; KM algorithms; Computing with words; Fuzzy weighted average

1. Introduction

Type-2 fuzzy sets (T2 FS), which were introduced by Zadeh in [98], are now very well established and (as shall be demonstrated in this paper) are gaining more and more in popularity. In [51] we find answers to the following:

1. *Why did it take so long for the concept of a T2 FS to emerge?* It seems that science moves in progressive ways where one theory is eventually replaced or supplemented by another, and then another. In school we learn about determinism before randomness. Learning about type-1 (T1) FSs before T2 FSs fits a similar learning model. So, from this point of view it was very natural for fuzzyites to develop T1 FSs as far as possible. Only by doing so was it really possible later to see the shortcomings of such FSs when one tries to use them to model words or to apply them to situations where uncertainties abound.
2. *Why didn't T2 FSs immediately become popular?* Although Zadeh introduced T2 FSs in 1975, very little was published about them until the mid-to late nineties. Until then they were studied by only a relatively small number of people, including: [13,14,19,20,63,64,66,86]. Recall that in the 1970s people were first learning

* Tel.: +1 213 740 4445; fax: +1 213 740 4651.
E-mail address: jmmprof@comcast.net

0020-0255/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2006.05.003

what to do with T1 FSs, e.g. fuzzy logic control. Bypassing those experiences would have been unnatural. Once it was clear what could be done with T1 FSs, it was only natural for people to then look at more challenging problems. This is where we are today.

3. *Why do we believe that by using T2 FSs we will outperform the use of T1 FSs?* T2 FSs are described by membership functions (MFs) that are characterized by more parameters than are MFs for T1 FSs. Hence, T2 FSs provide us with more design degrees of freedom; so using T2 FSs has the potential to outperform using T1 FSs, especially when we are in uncertain environments. Note that, at present, *there is no theory that guarantees that a T2 FS will always do this*.

One sign of a vibrant field is its applications. Here we categorize the applications that have appeared in the literature for T2 fuzzy sets and systems since 2001. For applications prior to that year, see [47, pp. 13–14].

> *Approximation:* [61] (TSK/steel strip temperature); *Clustering:* [26] (*C* spherical shells algorithm), [72] (fuzzy C-means); *Control:* [40] (marine and traction diesel engines), [75] (integrated development platform), [4] (evolutionary computing/NL dynamic plants), [36] (buck DC–DC converters), [17] (tracking mobile objects/robotic soccer games), [93] (liquid-level), [81] (proportional control), [25] (autonomous mobile robots), [24] (autonomous mobile robots/hierarchical), [44] (adaptive control of nonlinear plants); *Databases:* [67] (summarization); *Decision making:* [70] (variation in human decision making); *Embedded agents:* [10,11] (ambient intelligent environments); *Health care:* [27] (clinical diagnosis), [9] (differential diagnosis), [92] (nursing assessment); *Hidden Markov models:* [103] (phoneme recognition); *Neural networks:* [73] (fuzzy perceptron); *Noise cancellation:* [3] (adaptive noise cancellation); *Pattern classification:* [74] (fuzzy *k*-nearest neighbor); *Quality Control:* [45] (sound speakers); *Spatial query:* [71] (spatial objects); *Wireless communications:* [35] (wireless sensors/power on-off control), [77] (wireless sensor network lifetime analysis).

This paper focuses on advances in T2 fuzzy sets and systems since the year 2001, because earlier works are already well documented, e.g. [47]. The focus is on theoretical and computational issues. While some issues have been resolved, many new ones have been exposed, so T2 is a very fertile field for research.

Up until 2001, there was a very heavy emphasis on interval T2 FSs (IT2 FSs) and FLSs (IT2 FLSs), primarily because of their computational tractability. This emphasis has continued; however, interests have also turned towards more general kinds of T2 FSs and systems. Both T2 paths are covered in this paper. Section 2 covers topics about general T2 FSs and FLSs, and Section 3 covers topics about IT2 FSs and IT2 FLSs. Section 4 covers the fuzzy weighted average; Section 5 covers computing with words; and, Section 6 provides our conclusions.

It is assumed that the reader has some familiarity with T2 fuzzy sets and systems. For a relatively simple introduction to the former, see [54], and for the latter, see [47,55].

2. General T2 FSs and FLSs

In Section 2.1 we begin by presenting a Representation Theorem for a T2 FS. It is one of the most useful results in T2 FS theory because it can be used to derive many things that are associated with that theory, both old and new, in a simple and straightforward manner. Unfortunately, is not useful for computation; hence, the latter needs to be approached from other viewpoints. As for T1 FSs, the fundamental computations for T2 FSs are union, intersection and complement, and how to compute them, as well as attendant difficulties in such computations, are discussed in Section 2.2. One of the major applications for T2 FSs is a rule-based FLS, namely a T2 FLS, which is overviewed in Section 2.3. The major new calculation in a T2 FLS is called *type-reduction*; it maps a T2 FS into a T1 Fs, after which it is a simple matter to defuzzify the T1 FS in order to obtain a number at the output of the T2 FLS. Type-reduction, which is a major bottleneck for a T2 FLS, is overviewed in Section 2.4, and new ways for computing it are mentioned.

Zadeh [99–101] has introduced the *computing with words* (CWW) paradigm. Because words mean different things to different people, Mendel [50,51] has argued that words must be modeled using T2 FSs when computers interact with people and the interactions use FSs. In order to map from T2 FS word models back into a word, one will need the concept of similarity of T2 FSs, which is discussed in Section 2.5.

2.1. Representation theorem for a T2 FS

Mendel and John [54] have presented the following new representation for a T2 FS.

Theorem 1 (Representation theorem). *Assume that primary variable x is sampled at N values, $x_1, x_2, \ldots, x_N$, and at each of these values its primary memberships u_i are sampled at M_i values, $u_{i1}, u_{i2}, \ldots, u_{iM_i}$. Let $\widetilde{A}_e^j$ denote the jth T2 embedded set*[1] *for T2 FS $\widetilde{A}$, i.e.,*

$$\widetilde{A}_e^j \equiv \{(x_i, (u_i^j, f_{x_i}(u_i^j))), u_i^j \in \{u_{ik}, k = 1, \ldots, M_i\}, i = 1, \ldots, N\} \tag{1}$$

in which $f_{x_i}(u_i^j)$ is the secondary grade at u_i^j. Note that $\widetilde{A}_e^j$ can also be expressed as

$$\widetilde{A}_e^j = \sum_{i=1}^{N} [f_{x_i}(u_i^j)/u_i^j]/x_i, \quad u_i^j \in \{u_{ik}, k = 1, \ldots, M_i\} \tag{2}$$

Then $\widetilde{A}$ can be represented as the union of its T2 embedded sets, i.e.,

$$\widetilde{A} = \sum_{j=1}^{n_A} \widetilde{A}_e^j \tag{3}$$

$$n_A = \prod_{i=1}^{N} M_i \tag{4}$$

This representation of a T2 FS, in terms of much simpler T2 FSs, the embedded T2 FSs, is very useful for deriving theoretical results, as we explain later in this paper; however, it is not recommended for computational purposes, because it would require the explicit enumeration of the n_A embedded T2 FSs and n_A can be astronomical.

2.2. Operations on general T2 FSs

Consider two T2 FSs sets[2] $\widetilde{A}$ and $\widetilde{B}$, i.e.,

$$\widetilde{A} = \int_X \mu_{\tilde{A}}(x)/x = \int_X \left[\int_{J_x^u} f_x(u)/u\right]/x, \qquad J_x^u = \{(x,u) : u \in [\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]\} \subseteq [0,1] \tag{5}$$

and

$$\widetilde{B} = \int_X \mu_{\tilde{B}}(x)/x = \int_X \left[\int_{J_x^w} g_x(w)/w\right]/x, \qquad J_x^w = \{(x,w) : w \in [\underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{B}}(x)]\} \subseteq [0,1] \tag{6}$$

It is well-known [63] that the union of $\widetilde{A}$ and $\widetilde{B}$ is another T2 FS whose MF can be computed from:

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \int_{u\in J_x^u} \int_{w\in J_x^w} f_x(u)\bigstar g_x(w)/(u \vee w) = \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x), \quad x \in X \tag{7}$$

where $\sqcup$ denotes the *join* operation. The use of the notation $\mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x)$ to indicate the join between the secondary MFs $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ is, of course, a *shorthand notation* for the operations in the middle of (7). What (7) says is that to perform the join between two secondary MFs, $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, $v = u \vee w$ must be performed between every possible pair of primary memberships u and w, such that $u \in J_x^u$ and $w \in J_x^w$ and that the secondary grade of $\mu_{\tilde{A}\cup\tilde{B}}(x)$ must be computed as the t-norm operation between the corresponding secondary grades of $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, $f_x(u)$ and $g_x(w)$, respectively. Note that at each value of x the join involves T1 FSs (i.e., secondary MFs), and that the join must be computed for $\forall x \in X$. If more than one combination

[1] An embedded T2 FS is a T2 FS that has only one primary membership at each x_i. It is also called a *wavy slice* [54].
[2] Note that (5) means $\widetilde{A} : X \to \{[a,b] : 0 \leqslant a \leqslant b \leqslant 1\}$.

of u and w gives the same point $u \vee w$, then in the join we keep the one with the largest membership grade. Usually, the maximum t-conorm is used, as suggested in [98,63].

In general, evaluating the join is difficult to do for arbitrary T2 FSs. Karnik and Mendel [28] have shown that for n convex and normal T1 FSs, $F_1, \ldots, F_n$, characterized by MFs $f_1(\theta), \ldots, f_n(\theta)$, respectively, where $f_1(v_1) = \cdots = f_n(v_n) = 1$, and the $f_i(\theta)$ are re-ordered so that $v_1 \leqslant v_2 \leqslant \cdots \leqslant v_n$, the MF of $\sqcup_{i=1}^n F_i$, using maximum t-conorm and either minimum or product t-norm, can be expressed as

$$\mu_{\sqcup_{i=1}^n F_i}(\theta) = \begin{cases} T_{i=1}^n f_i(\theta) & \theta < v_1 \\ T_{i=1}^k f_i(\theta) & v_k \leqslant \theta \leqslant v_{k+1} \quad 1 \leqslant k \leqslant n-1 \\ \vee_{i=1}^n f_i(\theta) & \theta > v_n \end{cases} \tag{8}$$

Unfortunately, this formula is not so easy to use. Coupland and John [7] have taken a very new and novel approach to computing the join. Because their approach is also used for computing the meet, we describe it later in this section.

It is also well known [63] that the intersection of $\widetilde{A}$ and $\widetilde{B}$ is another T2 FS whose MF can be computed from:

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \int_{u\in J_x^u} \int_{w\in J_x^w} f_x(u) \bigstar g_x(w)/u \wedge w = \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x), \quad x \in X \tag{9}$$

where $\sqcap$ denotes the *meet* operation. The use of the notation $\mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x)$ to indicate the meet between the secondary MFs $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ is another *shorthand notation*, but this time for the operations in the middle of (9). What (9) says is that to perform the meet between two secondary MFs, $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, $v = u \wedge w$ must be performed between every possible pair of primary memberships u and w, such that $u \in J_x^u$ and $w \in J_x^w$, and the secondary grade of $\mu_{\tilde{A}\cap\tilde{B}}(x)$ must be computed as the t-norm operation between the corresponding secondary grades of $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, $f_x(u)$ and $g_x(w)$, respectively. This must be done for $\forall x \in X$. If more than one combination of u and w gives the same point $u \wedge w$, then in the meet (just as in the join) we keep the one with the largest membership grade.

Note that in (9) there are two t-norms, $\bigstar$ and $\wedge$. Although they are usually chosen to be the same, they do not have to be. See [78] for very interesting discussions about this.

In general, evaluating the meet is also difficult to do for arbitrary T2 FSs, especially when the product t-norm is used. Karnik and Mendel [28] have also shown that for n convex and normal T1 FSs, $F_1, \ldots, F_n$, characterized by MFs $f_1(\theta), \ldots, f_n(\theta)$, respectively, where $f_1(v_1) = \cdots = f_n(v_n) = 1$, and the $f_i(\theta)$ are re-ordered so that $v_1 \leqslant v_2 \leqslant \cdots \leqslant v_n$, the MF of $\sqcap_{i=1}^n F_i$ using maximum t-conorm and minimum t-norm can be expressed as

$$\mu_{\sqcap_{i=1}^n F_i}(\theta) = \begin{cases} \vee_{i=1}^n f_i(\theta) & \theta < v_1 \\ \wedge_{i=1}^k f_i(\theta) & v_k \leqslant \theta \leqslant v_{k+1} \quad 1 \leqslant k \leqslant n-1 \\ \wedge_{i=1}^n f_i(\theta) & \theta > v_n \end{cases} \tag{10}$$

Unfortunately, it is still difficult to use (10). To-date, no formula that is similar to (10) exits for the product t-norm, which is unfortunate because many applications use product t-norm. When all MFs are Gaussian, then Karnik and Mendel [28] have an approximation for computing the meet under product t-norm, one that leads to another Gausssian MF, so that the approximate meet is "reproducing", and can be expanded in multi-argument form.

The usual derivations of (7) and (9) utilize Zadeh's extension principle. Mendel and John [54] show how (7) and (9) can easily be derived without having to use the extension principle, when $\widetilde{A}$ and $\widetilde{B}$ are represented as in Theorem 1. These derivations were the first theoretical uses of the new Representation Theorem.

Coupland and John [6,7] have shown how to compute the join and meet of $\widetilde{A}$ and $\widetilde{B}$ using methods from computational geometry (e.g., a modified Weiler–Atherton clipping algorithm, and a Bentley–Ottman plane sweep algorithm). Their approach is based on modeling a secondary MF geometrically as a [7] "set of connected straight line segments that need not be equally-spaced across the domain", and is limited so far to the minimum t-norm and the maximum t-conorm. They distinguish between a partially discrete T2 FS and a discrete T2 FS. A *partially discrete T2 FS* is one whose primary variable is discrete (sampled) but whose secondary MFs are continuous, whereas a *discrete T2 FS* is one whose primary variable and secondary MFs are discrete (sampled). Based on extensive simulations of a two-rule FLS in which each rule has two-antecedents, and the secondary MFs are discretized into 10 points, and are also described by two line segments, Coupland and John obtain over a ‘‘four and a half fold increase in inferencing speed’’. They also state ‘‘... that for any T2 FLS with secondary MFs with five or more discretizations, using a partially discrete model would give a faster and more accurate system’’. This approach to computing operations for general T2 FSs looks very promising and is continuing.

When secondary MFs are triangular (an interesting compromise between interval secondary MFs and general secondary MFs, and one that is also considered by Coupland and John [7]) then Starczewski [78,79] has shown that ‘‘extended *t*-norms[3] of triangular fuzzy truth values may be approximated by triangular fuzzy truth values as well’’. One of the most interesting aspects of Starczewski’s [79] approach is it ‘‘... reduces calculations of extended *t*-norms (a similar approach can be rearranged for *s*-norms) to computing only the three characteristic functions: principal, upper and lower. Arbitrary traditional *t*-norms (or *s*-norms) can be used to calculate these functions. A tremendously useful feature of this approach is that the resultant MF preserves triangular shapes of the two arguments, and this way the approximate *t*-norms can be expanded to multi-argument form. Moreover, for each triangular fuzzy membership grade only three parameters have to be stored and processed by [a] FLS, instead of tabularized functions as in the general approach’’. Another very useful feature of this approach is that formulas are given for the operations, so that explicit derivative formulas can be obtained if a triangular T2 FLS is designed (optimized) using a method that requires such derivatives (e.g., steepest descent). Starczewski’s results also seem very promising and are continuing. Some additional work by him for Gaussian T2 FSs is in [80].

In a (singleton) T2 FLS the meet may only have to be computed at a single value of x, namely $x = x'$.

Definition. In general,

$$\widetilde{A} = \{((x,u), \mu_{\widetilde{A}}(x,u)) | \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{11}$$

in which $\mu_{\widetilde{A}}(x,u)$ is the T2 MF of $\widetilde{A}$. By a *type*-2 *fuzzy singleton*, we mean a T2 FS for which

$$\mu_{\widetilde{A}}(x,u) = \begin{cases} 1/1 & x = x' \\ 1/0 & \forall x \neq x' \end{cases} \tag{12}$$

In (12), 1/1 (1/0) means that at $x = x'$, when its primary variable $u = 1$ ($u = 0$), the associated secondary grade equals 1. At all other values of u the secondary grade equals 0; hence, by convention, such secondary grades are not shown.

Example. The meet between a T2 singleton, $\widetilde{A}$, and a normal[4] T2 FS, $\widetilde{B}$, under minimum and product *t*-norms is a widely used operation in a T2 FLS (Section 2.3), and is (for a derivation, see [47, p. 222]):

$$\mu_{\widetilde{A}}(x) \sqcap \mu_{\widetilde{B}}(x) = \begin{cases} \mu_{\widetilde{B}}(x') & x = x' \\ 1/0 & \forall x \neq x' \end{cases} \tag{13}$$

Observe that the meet between a type-2 singleton, $\widetilde{A}$, and a T2 FS, $\widetilde{B}$, sifts out a specific vertical slice of $\mu_{\widetilde{B}}(x,u)$, namely $\mu_{\widetilde{B}}(x')$, the secondary MF at $x = x'$.

Because it is very easy to compute the complement of $\widetilde{A}$, we do not discuss it here.

Finally, Walker and Walker [87–89] have many interesting and important mathematical results about join and meet within the framework of the algebra of truth values of T2 FSs. How these results can be applied is worth exploring.

[3] An extended *t*-norm is a *t*-norm obtained by applying the Extension Principle to a T1 *t*-norm. It was the basis for Mizumoto and Tanaka’s [63] works.

[4] This means the secondary MFs of $\widetilde{B}$ reach the value 1.

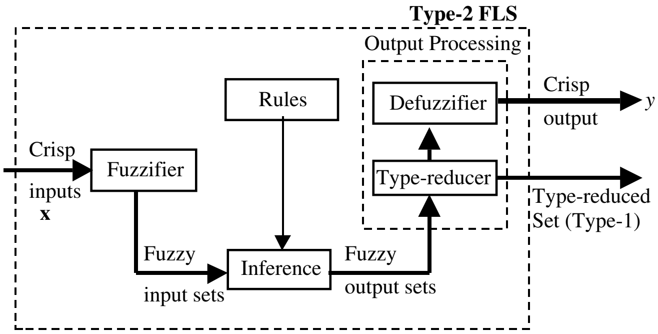

Fig. 1. Type-2 FLS.

2.3. Type-2 FLS

A general T2 FLS is depicted in Fig. 1. It is very similar to a T1 FLS, the major structural difference being that the defuzzifier block of a T1 FLS is replaced by the *output processing* block in a T2 FLS. That block consists of *type-reduction* (TR) followed by *defuzzification*. We will have a lot to say about TR in Sections 2.4 and 3.5.

Consider a T2 FLS having p inputs $x_1 \in X_1, \ldots, x_p \in X_p$, one output $y \in Y$, and M rules, where the lth rule has the form

$$R^l: \ \text{IF } x_1 \text{is } \widetilde{F}_1^l \text{ and } \cdots \text{ and } x_p \text{ is } \widetilde{F}_p^l, \text{ THEN } y \text{ is } \widetilde{G}^l \quad l = 1, \ldots, M \tag{14}$$

This rule represents a T2 relation between the input space $X_1 \times \cdots \times X_p$, and the output space, Y, of the T2 FLS. Each rule in (14) is interpreted as a T2 fuzzy implication. With reference to (14), let $\widetilde{F}_1^l \times \cdots \times \widetilde{F}_p^l = \widetilde{A}^l$; then, as is well known, (14) can be re-expressed as

$$R^l \ : \ \widetilde{F}_1^l \times \cdots \times \widetilde{F}_p^l \to \widetilde{G}^l = \widetilde{A}^l \to \widetilde{G}^l, \quad l = 1, \ldots, M \tag{15}$$

R^l is described by the MF $\mu_{R^l}(\mathbf{x}, y) = \mu_{R^l}(x_1, \ldots, x_p, y)$, where[5]

$$\mu_{R^l}(\mathbf{x}, y) = \mu_{\widetilde{A}^l \to \widetilde{G}^l}(\mathbf{x}, y) = [\sqcap_{i=1}^p \mu_{\widetilde{F}_i^l}(x_i)] \sqcap \mu_{\widetilde{G}^l}(y) \tag{16}$$

Most generally, the p-dimensional input to R^l is given by the T2 FS $\widetilde{A}_\mathbf{x}$ whose MF is

$$\mu_{\widetilde{A}_\mathbf{x}}(\mathbf{x}) = \sqcap_{i=1}^p \mu_{\widetilde{X}_i}(x_i) \tag{17}$$

where $\widetilde{X}_i$ $(i = 1, \ldots, p)$ are the labels of the FSs describing the inputs. Each rule R^l determines a T2 FS $\widetilde{B}^l = \widetilde{A}_\mathbf{x} \circ R^l$ such that

$$\mu_{\widetilde{B}^l}(y) = \mu_{\widetilde{A}_\mathbf{x} \circ R^l}(y) = \sqcup_{\mathbf{x} \in \mathbf{X}}[\mu_{\widetilde{A}_\mathbf{x}}(\mathbf{x}) \sqcap \mu_{R^l}(\mathbf{x}, y)], \quad \forall y \in Y \ l = 1, \ldots, M \tag{18}$$

This equation is the input–output relation in Fig. 1 between the T2 FS that excites one rule in the inference engine and the T2 FS at the output of that engine. Substituting (16) and (17) into (18), it is straightforward to show $(l = 1, \ldots, M)$:

$$\mu_{\widetilde{B}^l}(y) = \mu_{\widetilde{G}^l}(y) \sqcap \left\{ [\sqcup_{x_1 \in X_1} \mu_{\widetilde{X}_1}(x_1) \sqcap \mu_{\widetilde{F}_1^l}(x_1)] \sqcap \cdots \sqcap [\sqcup_{x_p \in X_p} \mu_{\widetilde{X}_p}(x_p) \sqcap \mu_{\widetilde{F}_p^l}(x_p)] \right\}, \quad \forall y \in Y \tag{19}$$

To-date, only the product or minimum t-norms have been used for the meet, and as we have discussed in Section 2.2, it is very difficult to compute the meet for general T2 FSs.

As in the T1 case, fired rule sets are combined either before or as part of output processing, in the latter case during type-reduction. We return to this later in Section 2.4.

Referring to Fig. 1, in general, the *fuzzifier* maps a crisp point $\mathbf{x} = (x_1, \ldots, x_p)^\mathrm{T} \in X_1 \times X_2 \times \cdots \times X_p \equiv \mathbf{X}$ into a T2 FS $\widetilde{A}_\mathbf{x}$ in $\mathbf{X}$. Here we focus on a major simplification of (19) as a result of *singleton fuzzification*,

[5] Derivations of (16)–(20) can be found, e.g. in [47, Chapter 10].

which is the only case described in this paper, because non-singleton fuzzification may be at present too complicated for general T2 FLSs.

For singleton fuzzification, the join operations in (19) are very easy to evaluate because each $\mu_{\tilde{X}_i}(x_i)$ is nonzero only at one point, $x_i = x_i'$; hence (for minimum and product t-norms), applying (13) to (19), we find:

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{G}^l}(y) \sqcap [\sqcap_{i=1}^{p} \mu_{\tilde{F}_i^l}(x_i')], \quad \forall y \in Y \tag{20}$$

The term in the bracket on the last line of (20) is referred to as the *firing set*, i.e.

$$\text{Firing Set} = \sqcap_{i=1}^{p} \mu_{\tilde{F}_i^l}(x_i') \tag{21}$$

Because $\mu_{\tilde{F}_i^l}(x_i')$ is a T1 FS, the firing set is the meet of p T1 FSs.

Note that $\mu_{\tilde{B}^l}(y)$ depends upon $\mathbf{x} = \mathbf{x}'$, although this dependence is not shown explicitly in the notation $\mu_{\tilde{B}^l}(y)$; however, when $\mathbf{x}'$ changes $\mu_{\tilde{B}^l}(y)$ changes as well. In general, computing $\mu_{\tilde{B}^l}(y)$ is very difficult because, as we have discussed in Section 2.2, it is still very difficult to compute the meet of general T2 FSs. Hopefully, the new approaches for computing the meet, that were discussed in Section 2.2, will lead to a practical computation of $\mu_{\tilde{B}^l}(y)$.

2.4. Type-reduction for general T2 FSs

Referring to Fig. 1, we see that the outputs of the inference engine are type-reduced and then defuzzified. A type-reducer combines all fired-rule output sets in some way (just like a T1 defuzzifier combines the T1 rule output sets), which leads to a T1 FS that is called a *type-reduced (TR) set*. Karnik and Mendel [29] have proposed five kinds of TR. Here we briefly review two of them (each of these methods is quite different) because how to compute the TR set for general T2 FSs is in general quite difficult. We need to understand why that is so and what new options have recently become available for making these computations more practical.

Centroid TR: To begin, all the fired rule-output T2 FSs, $\widetilde{B}^l$, are combined by finding their union, i.e.

$$\bigcup_{l=1}^{M} \widetilde{B}^l \equiv \widetilde{B} \tag{22}$$

where

$$\mu_{\tilde{B}}(y) = \sqcup_{l=1}^{M} \mu_{\tilde{B}^l}(y) \quad \forall y \in Y \tag{23}$$

in which $\mu_{\tilde{B}^l}(y)$ is the secondary MF for the lth rule, and $\mu_{\tilde{B}^l}(y)$ is given by (20). Centroid TR calculates the centroid of $\widetilde{B}$.

As another application of the Representation Theorem, the centroid TR set, $Y_c(\mathbf{x})$, is simply the union of the centroids of all the embedded T2 FSs of $\widetilde{B}$. Until very recently, the only way to compute $Y_c(\mathbf{x})$ was to use the following procedure. For each $\mathbf{x} = \mathbf{x}'$:

1. Compute $\mu_{\tilde{B}}(y)$ using (23). This is possible because $\mu_{\tilde{B}^l}(y)$ $(l = 1, \ldots, M)$ will already have been computed for all $y \in Y$, as in (20).
2. Discretize the y-domain into N points $y_1, \ldots, y_N$.
3. Discretize each $J_{y_i}(\mathbf{x}')$ (the primary memberships of $\mu_{\tilde{B}}(y)$ at y_i) into a suitable number of points, say M_i $(i = 1, \ldots, N)$. Let $\theta_i(\mathbf{x}') \in J_{y_i}(\mathbf{x}')$.
4. Enumerate all the embedded T1 sets[6] of $\widetilde{B}$; there will be $\prod_{i=1}^{N} M_i$ of them.
5. Compute the centroid of each enumerated embedded T1 set and assign it a membership grade equal to the t-norm of the secondary grades corresponding to that enumerated embedded T1 set.

[6] An embedded T1 set is the domain for an embedded T2 FS (see e.g., (34)).

Mathematically, this means

$$Y_{\mathrm{c}}(\mathbf{x}') = \{(\zeta_k, (T_{i=1}^{N} f_{y_i}(\theta_i(\mathbf{x}')))_k)\}_{k=1}^{\prod_{i=1}^{N} M_i} \tag{24a}$$

$$\zeta_k = \left(\frac{\sum_{i=1}^{N} y_i \theta_i(\mathbf{x}')}{\sum_{i=1}^{N} \theta_i(\mathbf{x}')}\right)_k \tag{24b}$$

Note that if two or more embedded T1 sets have the same centroid, we keep the one with the largest value of $T_{i=1}^{N} f_{y_i}(\theta_i(\mathbf{x}'))$. Note, also, that in (24a) we must use the minimum t-norm, due to technical reasons that are explained in [20,47].

This procedure is not very practical because it requires $\prod_{i=1}^{N} M_i$ centroid calculations, and this number will in general be astronomical.

A recently-developed much more practical method for computing $Y_{\mathrm{c}}(\mathbf{x})$, that is based on the fuzzy weighted average (FWA), is described in Section 4. This method uses α-cuts, an α-cut decomposition theorem [30] and the KM algorithms (see Section 3.3.3) that were originally developed for computing the centroid of an interval T2 FS.

Another recent approach for computing $Y_{\mathrm{c}}(\mathbf{x})$, based on randomly sampling embedded T2 FSs and computing their centroids [21] gives rise to a significant reduction in the time or resources needed to perform type-reduction. Greenwood et al. have examples (for four different primary MFs and different discretizations) that demonstrate the number of embedded sets [randomly] selected only marginally affects the defuzzified value. Excellent results have been obtained for as few as 10 randomly selected embedded T2 FSs. That such a small number of randomly chosen embedded T2 FSs can lead to such good results is a surprising result that awaits a theoretical explanation. Such an explanation is under study.

Center-of-sets TR: Each consequent T2 set, $\widetilde{G}^l$, is first replaced by its centroid, $C_{\tilde{G}^l}$ (which itself is a T1 set) and then a weighted average of these centroids is computed. The weight associated with the lth centroid is the (T1) firing set corresponding to the lth rule namely [see (21)] $\sqcap_{i=1}^{p} \mu_{\tilde{F}_i^l}(x_i') \equiv E_l(\mathbf{x}')$. Until very recently, the only way to compute the center-of-sets TR set, $Y_{\cos}(\mathbf{x}')$, was to use the following procedure. For each $\mathbf{x} = \mathbf{x}'$:

1. Discretize the output space Y into a suitable number of points, and compute the centroid $C_{\tilde{G}^l}$ of each consequent set on the discretized output space using the brute-force methodology that has just been described above. These consequent centroid sets can be computed ahead of time and stored for future use.
2. Compute the T1 firing set $E_l(\mathbf{x}') = \sqcap_{i=1}^{p} \mu_{\tilde{F}_i^l}(x_i')$ associated with the lth fired-rule consequent set.
3. Discretize the domain of each T1 FS $C_{\tilde{G}^l}$ into a suitable number of points, say N_l $(l = 1, \ldots, M)$.
4. Discretize the domain of each T1 FS $E_l(\mathbf{x}')$ into a suitable number of points, say M_l $(l = 1, \ldots, M)$.
5. Enumerate all the possible combinations $(d_1, \ldots, d_M, e_1(\mathbf{x}'), \ldots, e_M(\mathbf{x}'))$ such that $d_l \in C_{\tilde{G}^l}$ and $e_l(\mathbf{x}') \in E_l(\mathbf{x}')$. The total number of combinations will be $\prod_{l=1}^{M} M_l N_l$.
6. Compute the centroid $\sum_{l=1}^{M} d_l e_l(\mathbf{x}') / \sum_{l=1}^{M} e_l(\mathbf{x}')$ of each of the enumerated combinations and assign it a membership grade equal to the t-norm $T_{l=1}^{M} \mu_{C_{\tilde{G}^l}}(d_l) \bigstar T_{l=1}^{M} \mu_{E_l(\mathbf{x}')}(e_l(\mathbf{x}'))$.

Mathematically, this means

$$Y_{\cos}(\mathbf{x}') = \{(\xi_k, (T_{l=1}^{M} \mu_{C_{\tilde{G}^l}}(d_l) \bigstar T_{l=1}^{M} \mu_{E_l(\mathbf{x}')}(e_l(\mathbf{x}')))_k)\}_{k=1}^{\prod_{i=1}^{M} M_i N_i} \tag{25a}$$

$$\xi_k = \left(\frac{\sum_{l=1}^{M} d_l e_l(\mathbf{x}')}{\sum_{l=1}^{M} e_l(\mathbf{x}')}\right)_k \tag{25b}$$

Because there are exactly $MC_{\tilde{G}^l}$ and $E_l(\mathbf{x}')$, where M is the number of rules, we can use product or minimum t-norm in (25a). If two or more combinations of $(d_1, \ldots, d_M, e_1(\mathbf{x}'), \ldots, e_M(\mathbf{x}'))$ have the same centroid, we keep the one with the largest value of $T_{l=1}^{M} \mu_{C_{\tilde{G}^l}}(d_l) \bigstar T_{l=1}^{M} \mu_{E_l(\mathbf{x}')}(e_l(\mathbf{x}'))$. The expression for the center-of-sets TR set is also called a *generalized centroid*[7] (GC).

[7] A GC is a weighted average in which all numbers are T1 FSs. For discussions, see [29] or [47].

In Step 6 the weighted sum and t-norm operations in (25a) and (25b) have to be repeated $\prod_{l=1}^{M} M_l N_l$ times. Although this number is much less than that required for centroid TR, because usually $M \ll N$, $\prod_{l=1}^{M} M_l N_l$ can also be astronomical. Consequently, this procedure also is not very practical.

How to compute $Y_{\cos}(\mathbf{x})$ using the FWA is described in Section 4.

How the random sampling method [21] can be used for center-of-sets (and other kinds of) TR also remains to be explored. Note that a big difference between centroid TR and e.g., center-of-sets TR is that for the former the primary variable x can be discretized as finely as desired [i.e., N in (24) can be made as large as desired], whereas for the latter we do not have control over the number of rules M during the TR process. Whether or not this affects the random sampling method is an open question.

A recent unpublished work [79] that focuses on triangular secondary MFs includes an approximate KM-like procedure for TR. This is very important if such T2 FSs are to find practical use in applications of T2 FLSs.

2.5. Similarity of T2 FSs

The literature on similarity of T1 FSs is quite extensive, e.g. [5,8,91]. Mitchell [62] was the first to define a similarity measure for general T2 FSs. Here we summarize his similarity measure, but using the symbols of this paper.

Given two T2 FSs $\widetilde{A}$ and $\widetilde{B}$, for which (using the Representation Theorem) $\widetilde{A}$ is given by (3) and (2), and

$$\widetilde{B} = \sum_{k=1}^{n_B} \widetilde{B}_e^k \tag{26}$$

$$\widetilde{B}_e^k = \sum_{q=1}^{N} [g_{x_q}(w_q^k)/w_q^k]/x_q, \quad w_q^k \in \{w_{qn}, n = 1, \ldots, M_q\} \tag{27}$$

Mitchell's T2 similarity measure, $\widetilde{S}(\widetilde{A},\widetilde{B})$, is a weighted average of a T1 similarity measure S_{jk} that is computed for all of the $n_A n_B$ combinations of $\widetilde{A}$ and $\widetilde{B}$'s embedded T2 FSs (again, using the Representation Theorem), i.e.[8]

$$\widetilde{S}(\widetilde{A},\widetilde{B}) = \sum_{j=1}^{n_A} \sum_{k=1}^{n_B} S_{jk} \Lambda_{jk} \tag{28}$$

where S_{jk} can be any of the known T1 similarity measures, the summations are arithmetic, and

$$\Lambda_{jk} = \frac{T_{i=1}^{N} f_{x_i}(u_i^j) \times T_{q=1}^{N} g_{x_q}(w_q^k)}{\sum_{j=1}^{n_A} \sum_{k=1}^{n_B} T_{i=1}^{N} f_{x_i}(u_i^j) \times T_{q=1}^{N} g_{x_q}(w_q^k)} \tag{29}$$

How to calculate (28) and (29) other than by direct enumeration of all embedded T2 FSs is an open question.

For IT2 FSs, since all secondary MFs equal one, it is easy to show that $\Lambda_{jk} = 1/n_A n_B$, so that

$$\widetilde{S}(\widetilde{A},\widetilde{B}) = \frac{1}{n_A n_B} \sum_{j=1}^{n_A} \sum_{k=1}^{n_B} S_{jk} \tag{30}$$

Unfortunately, (30) still requires direct enumeration of all embedded T1 FSs so that each S_{jk} can be computed.

$\widetilde{S}(\widetilde{A},\widetilde{B})$ in (28) and (30) is a number. An alternative is to treat the summations in these equations as the union, in which case $\widetilde{S}(\widetilde{A},\widetilde{B})$ in (28) becomes a T1 FS, namely

$$\widetilde{S}(\widetilde{A},\widetilde{B}) = \bigcup_{j=1}^{n_A} \bigcup_{k=1}^{n_B} (\Lambda_{jk}/S_{jk}) \tag{31}$$

[8] Our statement of his $\widetilde{S}(\widetilde{A},\widetilde{B})$ is based on choosing the embedded T2 FSs as in Theorem 1, whereas his statement of $\widetilde{S}(\widetilde{A},\widetilde{B})$ is based on choosing the embedded T2 FSs randomly.

For an IT2 FS, $\widetilde{S}(\widetilde{A},\widetilde{B})$in (30) becomes an interval set, namely

$$\widetilde{S}(\widetilde{A},\widetilde{B}) = \bigcup_{j=1}^{n_A}\bigcup_{k=1}^{n_B}(1/S_{jk}) = \left[\min_{\forall j,k} S_{jk}, \max_{\forall j,k} S_{jk}\right] \equiv [s_l, s_r] \tag{32}$$

To obtain a single similarity number from either (31) or (32) is simple; just find their centroid. What is still very challenging is enumerating all of the embedded T2 FSs in order to obtain $\widetilde{S}(\widetilde{A},\widetilde{B})$ in (31), or computing s_l and s_r in order to obtain $\widetilde{S}(\widetilde{A},\widetilde{B})$ in (32).

As mentioned in Section 2.1, in the past, we have found the use of the Representation Theorem very good for theoretical developments but very poor for computation. The centroid (Section 3.3) is a good example of this situation. Just as a totally different approach was needed to compute the centroid of an IT2 FS, we conjecture that a totally different approach will be needed to compute $\widetilde{S}(\widetilde{A},\widetilde{B})$.

3. Interval T2 FSs and FLs

In Section 3.1 we begin by specializing the Representation Theorem to IT2 FSs, because it has been and continues to be widely used for such FSs. Connections between IT2 FSs and interval-valued fuzzy sets and non-stationary T1 FSs are described in Section 3.2. The centroid of an IT2 FS is very widely used. Advances about the centroid, including many of its properties, how to compute it using the KM algorithms (including its properties), and its bounds (which are in terms of geometric properties of an IT2 FS), are summarized in Section 3.3. IT2 FLSs are reviewed in Section 3.4, as a prelude to discussions about TR, and how it can be avoided, which are given in Section 3.5. For IT2 FLSs to become widely used in commercial products, hardware realizations of them must be developed. This is discussed in Section 3.6.

3.1. Representation for an IT2 FS

An IT2 FS $\widetilde{A}$ is completely described by its lower and upper MFs, $\underline{\mu}_{\tilde{A}}(x)$ and $\bar{\mu}_{\tilde{A}}(x)$, respectively. The footprint of uncertainty (FOU) of an IT2 FS is described in terms of these MFs, as

$$\mathrm{FOU}(\widetilde{A}) = \bigcup_{x\in X}[\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)] \tag{33}$$

If X is discrete, then (33) is modified to

$$\mathrm{FOU}(\widetilde{A}) = \bigcup_{x\in X}\{\underline{\mu}_{\tilde{A}}(x), \ldots, \bar{\mu}_{\tilde{A}}(x)\} \tag{34}$$

In (34) the ... notation means all of the embedded T1 FSs between the lower and upper MFs. Frequently, (33) and (34) are used interchangeably without any confusion. The specialization of Representation Theorem 1 to an IT2 FS is contained in the following:

Corollary 1 (Representation for an IT2 FS [55]). *For an IT2 FS, for which X and U are discrete, the domain of $\widetilde{A}$ is equal to the union of all of its embedded T1 FSs, so that $\widetilde{A}$ can be expressed as*

$$\widetilde{A} = 1/\mathrm{FOU}(\widetilde{A}) = 1\bigg/\bigcup_{j=1}^{n_A} A_e^j \tag{35}$$

where A_e^j is an embedded T1 FS (that acts as the domain for $\widetilde{A}_e^j$, $j = 1,\ldots,n_A$), n_A is given by (4), and

$$A_e^j = \bigcup_{i=1}^{N}(u_i^j/x_i) \quad u_i^j \in \{\underline{\mu}_{\tilde{A}}(x_i), \ldots, \bar{\mu}_{\tilde{A}}(x_i)\} \tag{36}$$

In (33) it is understood that the notation $1/\mathrm{FOU}(\widetilde{A})$ means putting a secondary grade of 1 at all elements in the $\mathrm{FOU}(\widetilde{A})$.

This representation of an IT2 FS, in terms of embedded T1 FSs, is very useful for deriving theoretical results, as we explain below; however, it is not recommended for computational purposes, because it would require the enumeration of the n_A embedded T1 FSs and n_A can be astronomical.

3.2. Interpretations of an IT2 FS

It turns out that an IT2 FS is the same as an interval-valued FS (IVFS) for which there is a very extensive literature (e.g., [98], [2, see the many references in this article]). These two seemingly different kinds of FSs were historically approached from very different starting points, which as we shall explain next has turned out to be a very good thing.

The IT2 FS has always been considered to be a special case of a general T2 FS; consequently, things that were developed for the latter were then specialized to the former. Embedded T2 FSs, the Representation Theorem, type-reduction and centroid all originated for a general T2 FS and were then specialized to an IT2 FS. They do not appear at all in the IVFS literature.

For a general T2 FS the new third dimension of its secondary-grades is very important, because it provides additional knowledge about the FS. When all of these grades are the same, as in an IT2 FS, then they convey no new information, and so for such a T2 FS the third dimension is superfluous. This is why an IT2 FS can be completely described by its two-dimensional FOU, as in (35). It is this recognition that lets one immediately connect an IT2 FS with an IVFS. The latter was always approached as two-dimensional.

Gorzalczany (e.g., [19,20]) must be acknowledged as a pioneer in the development of IVFSs. Some of his diagrams for these FSs look just like those given for a FOU, although he does not call them "FOU". One of the most comprehensive treatments of IVFSs is [2]. Some of Bustince's results about approximate reasoning are the same as those in [34] which focused on IT2 FLSs; however, Bustince was not constrained by Mamdani implication, as were Liang and Mendel; hence, there are many more results for a variety of Generalized Modus Ponen algorithms in his article.

In [2] there is also a section about the similarity of IVFSs. First Bustince defines a *normal interval-valued similarity measure* $S(A,B)$ between[9] two IVFSs A and B, as one that satisfies the following five properties: (1) $S(A,B) = S(B,A)$ for all $A, B \in \mathrm{IVFSs}(X)$; (2) $S(D, D_C) = [0,0]$ for all $D \in P(X)$, where D_C is the complement of D and $P(X)$ is the class of all crisp sets of X; (3) $S(C, C) = [1,1]$ for all $C \in \mathrm{IVFSs}(X)$; (4) for all $A, B, C \in \mathrm{IVFSs}(X)$, if $A \leqslant B \leqslant C$, then $S(A, B) \geqslant S(A, C)$ and $S(B, C) \geqslant S(A, C)$; and, (5) if $A, B \in \mathrm{IVFSs}(X)$, then $S(A, B) \in [0,1]$. Bustince then shows that

$$S(A,B) = [S_{\mathrm{L}}(A,B), S_{\mathrm{U}}(A,B)], \tag{37}$$

in which $S_{\mathrm{L}}(A, B)$ is a T1 similarity measure between the lower MFs of A and B, and $S_{\mathrm{U}}(A, B)$ is a T1 similarity measure between the upper MFs of A and B, satisfies the five properties. There are additional results, but they are beyond the scope of this paper. What the connection is between the interval similarity measure in (32) and Bustince's similarity measure in (37) is an open question.

Garibaldi, et al. [18] have introduced the concept of a *non-stationary T1 FS* and have proposed that it be compared with an IT2 FS. According to them: "A non-stationary FS $\dot{A}$ is characterized by a MF $\mu_{\dot{A}}(x,t)$, where $x \in X$, $\mu_{\dot{A}}(x,t) \in [0,1]$ and t is a free variable, *time*—the time at which the FS is instantiated, i.e. ...

$$\dot{A} = \int_{x\in X} \mu_{\dot{A}}(x,t)/x \tag{38}$$

... The three main alternative kinds of non-stationary [for a T1 MF] are variation in location, variation in slope and noise variation". For example, let c denote the center value of a T1 MF, and $c(t) = c + kf(t)$ denote a time-varying model for its variation, in which $f(t)$ is referred to by them as a "perturbation function", which may be random, hence the terminology "non-stationary FS".

When $f(t)$ is a known deterministic function, then $\mu_{\dot{A}}(x,t)$ can be lower and upper bounded, in which case there is a direct connection between $\dot{A}$ and an IT2 FS. On the other hand, when $f(t)$ is random (a random process), then $\dot{A}$ is a *random fuzzy set* [39]. Such FSs, that are very different from fuzzy random variables (e.g., [1,65]), can be treated as nonlinear transformations of random processes. If the distribution function can be computed for the now random $\mu_{\dot{A}}(x,t)$, then perhaps lower and upper probability bounds can be established for each value of a primary variable. These bounds might then be somehow related to the lower and upper MFs of an IT2 FS. How to carry out such calculations remains to be explored.

[9] In this paragraph A and B denote IVFSs.

3.3. Centroid of an IT2 FS

The centroid of an IT2 FS, developed by Karnik and Mendel [29], has turned out to be a fundamental concept not only for an IT2 FS and IT2 FLSs but also for general T2 FSs. It provides a measure of uncertainty for an IT2 FS. In this section, we review the definition of the centroid, summarize its properties, provide the KM algorithms for its computation, summarize properties of the KM algorithms, show how the KM algorithms can be used to compute the generalized centroid (a computation that is needed when center-of-sets TR is used, and also for the FWA), and provide bounds for the centroid that are explicit functions of the geometry of a FOU.

3.3.1. Definition

Using the Representation Theorem for an IT2 FS $\widetilde{A}$, we define its centroid, $C_{\tilde{A}}$, as the collection of the centroids of all of its embedded IT2 FSs. From (35) we see that this means we need to compute the centroids of all of the n_A embedded T1 FSs contained within $\mathrm{FOU}(\widetilde{A})$. The results of doing this will be a collection of n_A numbers, and these numbers will have both a smallest and largest element, $c_l(\widetilde{A}) \equiv c_l$ and $c_r(\widetilde{A}) \equiv c_r$, respectively. That such numbers exist is because the centroid of each of the embedded T1 FSs is a finite number. Associated with each of these numbers will be a membership grade of 1, because the secondary grades of an IT2 FS are all equal to 1. This means

$$C_{\tilde{A}} = 1/\{c_l, \ldots, c_r\} \tag{39}$$

where[10]

$$c_l = \min_{\forall\theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{A}}(x_i)]} \left(\sum_{i=1}^{N} x_i\theta_i \Big/ \sum_{i=1}^{N} \theta_i \right) \tag{40}$$

$$c_r = \max_{\forall\theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{A}}(x_i)]} \left(\sum_{i=1}^{N} x_i\theta_i \Big/ \sum_{i=1}^{N} \theta_i \right) \tag{41}$$

and

$$x_1 < x_2 < \cdots < x_N \tag{42}$$

The latter is true because x_i are sampled values of the primary variable; x_1 denotes the left-hand (smallest) sampled value and x_N denotes the right-hand (largest) sampled value.[11]

How to compute c_l and c_r will be explained below in Section 3.3.3.

3.3.2. Properties of the centroid

Since the introduction of the centroid its properties have been studied by Mendel and Wu [60], Liu and Mendel [38] and Mendel [52]. Here we state the properties without proof, since they provide insights about the centroid and can also greatly simplify the computation of the centroid.

Property 1 [60]. *Let $\widetilde{A}$ be an IT2 FS defined on X, and $\widetilde{A}'$ be $\widetilde{A}$ shifted by*[12] *Δm along X, i.e. $\underline{\mu}_{\tilde{A}'}(x) = \underline{\mu}_{\tilde{A}}(x - \Delta m)$ and $\bar{\mu}_{\tilde{A}'}(x) = \bar{\mu}_{\tilde{A}'}(x - \Delta m)$. Then the centroid of $\widetilde{A}'$, $[c_l(\widetilde{A}'), c_r(\widetilde{A}')]$, is the same as the centroid of $\widetilde{A}$, $[c_l(\widetilde{A}), c_r(\widetilde{A})]$, shifted by Δm, i.e. $c_l(\widetilde{A}') = c_l(\widetilde{A}) + \Delta m$ and $c_r(\widetilde{A}') = c_r(\widetilde{A}) + \Delta m$.*

This property lets us relocate $\mathrm{FOU}(\widetilde{A})$ to a more convenient place for the actual computations of c_l and c_r, and demonstrates that it is only the shape of $\mathrm{FOU}(\widetilde{A})$ that effects the centroid and not where that shape resides on the axis of the primary variable.

---

[10] When discretizations of the primary variable and primary membership approach zero, $\{c_l, \ldots, c_r\} \to [c_l, c_r]$, an interval set. In the literature about the centroid, it is customary to see (39) written as $C_{\tilde{A}} = [c_l, c_r]$.

[11] If Gaussian MFs are used, then in theory $x_1 \to -\infty$ and $x_N \to \infty$; but, in practice when truncations are used x_1 and x_N are again finite numbers.

[12] Δm may be positive or negative.

Property 2 [60]

(a) *If* $\mathrm{FOU}(\widetilde{A})$ *is amplitude scaled by* λ, *where* $0<\lambda<1$, *meaning that* $\mathrm{FOU}(\widetilde{A}) \to \lambda\mathrm{FOU}(\widetilde{A})$ [*i.e.*, $\bar{\mu}_{\tilde{A}'}(x) = \lambda\bar{\mu}_{\tilde{A}}(x)$ *and* $\underline{\mu}_{\tilde{A}'}(x) = \lambda\underline{\mu}_{\tilde{A}}(x)$] *then the centroid is FOU scale-invariant.*

(b) *If* $\mathrm{FOU}(\widetilde{A})$ *is shifted vertically by a constant amount* [*i.e.*, $\bar{\mu}_{\tilde{A}'}(x) = \bar{\mu}_{\tilde{A}}(x)+\delta$ *and* $\underline{\mu}_{\tilde{A}'}(x) = \underline{\mu}_{\tilde{A}}(x)+\delta$] *then the centroid is* not *vertically shift-invariant.*

(c) *If the primary variable* x *is uniformly scaled to* x/γ, *where* $\gamma>0$ [*i.e.*, $\bar{\mu}_{\tilde{A}'}(x) = \bar{\mu}_{\tilde{A}}(x/\lambda)$ *and* $\underline{\mu}_{\tilde{A}'}(x) = \underline{\mu}_{\tilde{A}}(x/\lambda)$] *then the centroid scales by* γ *to* $\gamma[c_l(\widetilde{A}), c_r(\widetilde{A})]$.

When compared with Property 1 , Property 2 alerts us to the fact that the centroid is not always invariant.

Property 3 [60]. *If the primary variable* (x) *is bounded, i.e.* $x \in [x_1, x_N]$, *then* $c_l(\widetilde{A}) \geqslant x_1$ *and* $c_r(\widetilde{A}) \leqslant x_N$.

Although we cannot compute the centroid in closed form, this property provides bounds for the centroid that are available from a priori knowledge of the domain of the primary variable.

Property 4 [60]. *If* $\mathrm{LMF}(\widetilde{A})$ *is entirely on the primary-variable* (x) *axis, and* $x \in [x_1, x_N]$, *then the centroid does not depend upon the shape of* $\mathrm{FOU}(\widetilde{A})$ *and, as the sampling approaches zero, it equals* $[x_1, x_N]$.

Any FOU whose LMF is entirely on the primary-variable axis is said to be *completely filled-in*. This property demonstrates that for such a FOU its UMF plays no role in determining the centroid.

Property 5 [52]. *If* $\widetilde{A}$ *is symmetrical about primary variable* x *at* $x = m$, *then the centroid of such an IT2 FS is symmetrical about* $x = m$, *and the average value (i.e., the defuzzified value) of all the elements in the centroid equals* m.

Note that by using Property 5, we only have to compute c_l, because $c_r(\widetilde{A}) = 2m - c_l(\widetilde{A})$. This represents a 50% savings in computation. Note, also, that if one is planning only to use the defuzzified value of a symmetrical IT2 FS, then to carry out T2 computations is a wasted effort, because the same results could have been obtained by using T1 calculations. Fortunately (see [52] for a complete discussion), in an IT2 FLS in which two or more rules are fired this sort of symmetry is not encountered. It is encountered only when one rule is fired.

Property 6 [60]. *If* $\widetilde{A}$ *is symmetrical about* $m \in X$, *then* $c_l(\widetilde{A}) \leqslant m$ *and* $c_r(\widetilde{A}) \geqslant m$.

This property can be combined with Property 3 to provide lower and upper bounds for $c_l(\widetilde{A})$ and $c_r(\widetilde{A})$. Tighter bounds will be presented below in Section 3.3.5.

Property 7 [29]. *It is true that*

$$c_l(\widetilde{A}) \equiv c_l = \frac{\sum_{i=1}^{k_L} x_i \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k_L+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k_L} \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k_L+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)} \tag{43}$$

$$c_r(\widetilde{A}) \equiv c_r = \frac{\sum_{i=1}^{k_R} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k_R+1}^{n} x_i \bar{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k_R} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k_R+1}^{n} \bar{\mu}_{\tilde{A}}(x_i)} \tag{44}$$

where k_L *and* k_R *are switch points that are computed using the KM algorithms, which are described in Section* 3.3.3.

This property provides us with formulas to express $c_l(\widetilde{A})$ and $c_r(\widetilde{A})$, even when k_L and k_R are not known a priori.

For our next three properties, we let

$$c_l(k) \equiv \frac{\sum_{i=1}^{k} x_i \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k} \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)} \tag{45}$$

Property 8 (Location of c_l [38]). *When $k = k_L$, then $c_l(k_L) = c_l$ and $x_{k_L} \leqslant c_l < x_{k_L+1}$.*

The continuous version of this property [60] is rather amazing, and is $x_{k_L} = c_l$, which means that the left-end switch point equals the left end-point of the centroid.

Property 9 (Location of $c_l(k)$ in relation to the line $y = x_k$ [38]). *It is true that $c_l(k) > x_k$ when $x_k < c_l$, and $c_l(k) < x_k$ when $x_k > c_l$.*

This property does not imply $c_l(k)$ is monotonic on either side of c_l; but, it does demonstrate that $c_l(k)$ cannot be above the line $y = x_k$ to the right of c_l.

Property 10 (Monotonicity of $c_l(k)$ [38]). *It is true that $c_l(k-1) \geqslant c_l(k)$ when $x_k < c_l$, and $c_l(k+1) \geqslant c_l(k)$ when $x_k > c_l$.*

This property shows that $c_l(k)$ is a monotonic function (but not a strictly-monotonic function) on both sides of the minimum c_l.

Properties very similar to Properties 8–10 also exist for c_r. Simply replace $c_l(k)$, c_l and k_L by $c_r(k)$, c_r and k_U, respectively.

3.3.3. Computing the centroid using the KM algorithms

The centroid of an IT2 FS cannot be computed in closed form.[13] Instead iterative algorithms developed by Karnik and Mendel [29], now known as the *KM algorithms*, are used. We provide statements of these algorithms next.

KM Algorithm for $c_l(\tilde{A})$: The five steps of this algorithm are (1) In (40), initialize θ_i by setting $\theta_i = [\underline{\mu}_{\tilde{A}}(x_i) + \bar{\mu}_{\tilde{A}}(x_i)]/2,\ i = 1,\ldots,N$, and then compute $c' = c(\theta_1,\ldots,\theta_N) = \sum_{i=1}^{N} x_i\theta_i / \sum_{i=1}^{N}\theta_i$. (2) Find k $(1 \leqslant k \leqslant N-1)$ such that $x_k \leqslant c' \leqslant x_{k+1}$. (3) Set $\theta_i = \bar{\mu}_{\tilde{A}}(x_i)$ when $i \leqslant k$, and $\theta_i = \underline{\mu}_{\tilde{A}}(x_i)$ when $i \geqslant k+1$, and then compute $c_l(k)$ in (45). (4) Check if $c_l(k) = c'$. If yes, stop and set $c_l(k) = c_l$ and call k k_L. If no, go to Step 5. (5) Set $c' = c_l(k)$ and go to Step 2.

KM Algorithm for $c_r(\tilde{A})$: Steps 1 and 2 are the same as in the previous algorithm, but they are for (41); (3) Set $\theta_i = \underline{\mu}_{\tilde{A}}(x_i)$ when $i \leqslant k$, and $\theta_i = \bar{\mu}_{\tilde{A}}(x_i)$ when $i \geqslant k+1$, and then compute

$$c_r(k) = \frac{\sum_{i=1}^{k} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} x_i \bar{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \bar{\mu}_{\tilde{A}}(x_i)} \tag{46}$$

(4) Check if $c_r(k) = c'$. If yes, stop and set $c_r(k) = c_r$ and call k k_R. If no, go to Step 5. (5) Set $c' = c_r(k)$ and go to Step 2.

Mendel and Liu [56] have proven that the KM algorithms are monotonically convergent and that they converge super-exponentially fast. Both properties are highly desirable for iterative algorithms and explain why in practice the KM algorithms have been observed to converge very fast, thereby making them very practical to use. Prior to these results, the only available convergence statement for them was very pessimistic [29] namely convergence occurs in *at most* N iterations where N equals the number of sampled values of the primary variable; as N increases this bound becomes very uninformative.

3.3.4. Generalized centroid

In the following weighted average

$$y(z_1,\ldots,z_N,w_1,\ldots,w_N) = \frac{\sum_{l=1}^{N} z_l w_l}{\sum_{l=1}^{N} w_l} \tag{47}$$

if each z_l is replaced by an interval set $z_i \in [a_i, b_i]$ and each w_l is also replaced by an interval set $w_i \in [c_i, d_i]$, then (47) is called a *generalized centroid* (GC) and GC is the interval set $GC = [y_l, y_r]$. The GC is used to perform center-of-sets TR, and is also used to compute the FWA. Because z_i only appears in the numerator of (47), it is easy to show that

[13] One exception to this is the centroid of a fuzzy granule, for which it is possible to compute formulas for both c_l and c_r [60].

$$y_l = \min_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{N} a_i w_i \middle/ \sum_{i=1}^{N} w_i \right) \tag{48}$$

$$y_r = \max_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{N} b_i w_i \middle/ \sum_{i=1}^{N} w_i \right) \tag{49}$$

Comparing (48) with (40), we see that y_l can be computed using the KM algorithm stated above for c_l in which x_i is replaced by a_i, and $\underline{\mu}_{\tilde{A}}(x_i)$ and $\bar{\mu}_{\tilde{A}}(x_i)$ are replaced by c_i and d_i, respectively. Similarly, comparing (49) with (41), we see that y_r can be computed using the KM algorithm stated above for c_r in which x_i is replaced by b_i, and again $\underline{\mu}_{\tilde{A}}(x_i)$ and $\bar{\mu}_{\tilde{A}}(x_i)$ are replaced by c_i and d_i, respectively.

3.3.5. Centroid bounds

We have seen that an IT2 FS is characterized by its FOU, which in turn is characterized by its upper and lower MFs. Wu and Mendel [95] showed that the centroid provides a measure of the uncertainty for an IT2 FS. Intuitively, we anticipate that geometric properties about the FOU, such as its area and the center of gravities (centroids) of its upper and lower MFs, will be associated with the amount of uncertainty in such a T2 FS. Recently, Mendel and Wu [57] demonstrated that this intuition is correct. They quantified uncertainty bounds for the centroid of both symmetric and non-symmetric IT2 FSs with respect to such geometric properties. Using these results, it is possible to formulate and solve *forward problems*, i.e. to go from parametric IT2 FS models to data with associated uncertainty bounds. Here we only state results for a symmetrical FOU.

The geometric properties that we shall make use of, for a FOU that is symmetric about m, are A_{UMF}, the area under the upper MF; A_{LMF}, the area under the lower MF; A_{FOU}, the area of the FOU, where

$$A_{\mathrm{FOU}} = A_{\mathrm{UMF}} - A_{\mathrm{LMF}} = 2\int_{m}^{\infty} [\bar{\mu}(x) - \underline{\mu}(x)]\,\mathrm{d}x; \tag{50}$$

and, $c_{\mathrm{HFOU}}(\widetilde{A})$, the centroid of half of $\mathrm{FOU}(\widetilde{A})$, where

$$c_{\mathrm{HFOU}}(\widetilde{A}) = \frac{\int_m^\infty x[\bar{\mu}(x) - \underline{\mu}(x)]\,\mathrm{d}x}{\int_m^\infty [\bar{\mu}(x) - \underline{\mu}(x)]\,\mathrm{d}x} = \frac{\int_m^\infty x[\bar{\mu}(x) - \underline{\mu}(x)]\,\mathrm{d}x}{A_{\mathrm{FOU}}/2}. \tag{51}$$

Theorem 2 [57]. *Let* $[x_1, x_N]$ *be the primary domain of* $\widetilde{A}$. *Then the end-points*, c_l *and* c_r, *for the centroid of a symmetric IT2 FS*, $\widetilde{A}$, *are bounded from below and above by* (*Fig.* 2).[14]

$$\max(x_1, \underline{c}_l(\widetilde{A})) \leqslant c_l(\widetilde{A}) \leqslant \bar{c}_l(\widetilde{A}) \tag{52}$$

$$\underline{c}_r(\widetilde{A}) \leqslant c_r(\widetilde{A}) \leqslant \min(\bar{c}_r(\widetilde{A}), x_N) \tag{53}$$

where

$$\underline{c}_r(\widetilde{A}) = m + [c_{\mathrm{HFOU}}(\widetilde{A}) - m]\frac{A_{\mathrm{FOU}}}{A_{\mathrm{UMF}} + A_{\mathrm{LMF}}} \tag{54}$$

$$\bar{c}_r(\widetilde{A}) = m + [c_{\mathrm{HFOU}}(\widetilde{A}) - m]\frac{A_{\mathrm{FOU}}}{2A_{\mathrm{LMF}}} \tag{55}$$

$$\bar{c}_l(\widetilde{A}) = 2m - \underline{c}_r(\widetilde{A}) \tag{56}$$

$$\underline{c}_l(\widetilde{A}) = 2m - \bar{c}_r(\widetilde{A}) \tag{57}$$

We return to the use of these bounds in Section 5 where they will be used to solve inverse problems, i.e. going from interval data about a word to a FOU for that word.

[14] In general, $-\infty < x_1 < x_N < \infty$, e.g. if the primary MF is Gaussian, then its associated FOU extends to $\pm\infty$, in which case, $\max(x_1, \underline{c}_l(\widetilde{A})) = \underline{c}_l(\widetilde{A})$ and $\min(\bar{c}_r(\widetilde{A}), x_N) = \bar{c}_r(\widetilde{A})$. For most other FOUs, x_1 and x_N are finite numbers, and we need to use the more complete bounds in (52) and (53).

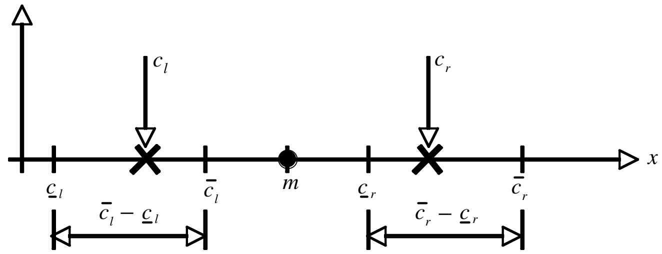

Fig. 2. End-points (X) of the centroid $[c_l, c_r]$ of $\tilde{A}$ for a FOU that is symmetrical about m, and the lower and upper bounds (|) for the two end-points.

3.4. Interval T2 FLSs

To-date, because of the computational complexity of using a general T2 FS in a T2 FLS, most people only use an IT2 FS, the result being an IT2 FLS. Unfortunately, there is a heavy educational burden on the practitioner even to using an IT2 FLS. This burden has to do with first learning general T2 FS mathematics, and then specializing it to an IT2 FS. In retrospect, requiring a person to use T2 FS mathematics represents a barrier to the use of an IT2 FLS. Mendel et al. [55] demonstrate that it is unnecessary to take the route from general T2 FS to IT2 FS, and that all of the results that are needed to implement an IT2 FLS can be obtained using T1 FS mathematics. Their paper is a very simple way for someone who is new to the field of an IT2 FLS to get into it very quickly.

When all T2 FSs are IT2 FSs, then the firing set (21) and fired-rule output sets (20) are IT2 FSs, and this simplifies all computations enormously. More specifically, for an interval singleton T2 FLS [34]: (a) the firing set becomes a *firing interval*, i.e.

$$\sqcap_{i=1}^{p}\mu_{\tilde{F}_i^l}(x_i') \equiv F^l(\mathbf{x}') = [\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')] \equiv [\underline{f}^l, \bar{f}^l] \tag{58}$$

where

$$\underline{f}^l(\mathbf{x}') = \underline{\mu}_{\tilde{F}_1^l}(x_1') \bigstar \cdots \bigstar \underline{\mu}_{\tilde{F}_p^l}(x_p') \tag{59}$$

$$\bar{f}^l(\mathbf{x}') = \bar{\mu}_{\tilde{F}_1^l}(x_1') \bigstar \cdots \bigstar \bar{\mu}_{\tilde{F}_p^l}(x_p') \tag{60}$$

(b) The rule R^l fired output consequent set, $\mu_{\tilde{B}^l}(y)$ in (20), is the IT2 FS

$$\mu_{\tilde{B}^l}(y) = \int_{b^l \in [\underline{f}^l \bigstar \underline{\mu}_{\tilde{G}^l}(y), \bar{f}^l \bigstar \bar{\mu}_{\tilde{G}^l}(y)]} 1/b^l, \quad y \in Y \tag{61}$$

where $\underline{\mu}_{\tilde{G}^l}(y)$ and $\bar{\mu}_{\tilde{G}^l}(y)$ are the lower and upper membership grades of $\mu_{\tilde{G}^l}(y)$.

These results have been very widely used in all applications of IT2 FLSs; however, there still may be a problem to use an IT2 FLS for real-time applications, because of the computational bottleneck of TR. Various approaches have been reported on for bypassing TR, some of which are summarized next.

3.5. Type-reduction and bypassing it for IT2 FLSs

When Karnik and Mendel [29] introduced TR they required adherence to the following:

Design requirement: If all sources of uncertainty disappear then a T2 FLS must reduce to a T1 FLS.

This seems like a very reasonable requirement; however, because TR may be a computational bottleneck even for an IT2 FLS, we need to question whether or not TR is really needed.

While it is true that when all sources of uncertainty disappear, Karnik and Mendel's TR methods reduce to their T1 defuzzification counterparts, thereby ensuring that their IT2 FLS reduces to a T1 FLS, this does not necessarily mean that TR is the only way for this design requirement to be met. For example, suppose that we have computed the union of fired rule output sets, to obtain[15] $\mu_{\tilde{B}}(y)$ in (23), i.e. [34,47] $(N \leqslant M)$

$$\mu_{\tilde{B}}(y) = \int_{b\in[[\underline{f}^1\star\underline{\mu}_{\tilde{G}^1}(y)]\vee\cdots\vee[\underline{f}^N\star\underline{\mu}_{\tilde{G}^N}(y)],[\bar{f}^1\star\bar{\mu}_{\tilde{G}^1}(y)]\vee\cdots\vee[\bar{f}^N\star\bar{\mu}_{\tilde{G}^N}(y)]]} 1/b, \quad y \in Y$$

Let $\mathrm{COG}(\underline{\mu}_{\tilde{B}}(y))$ and $\mathrm{COG}(\bar{\mu}_{\tilde{B}}(y))$ denote the centroids of the LMF and UMF, respectively, of $\mu_{\tilde{B}}(y)$. Then, we can define the output of an IT2 FLS, $y(\mathbf{x})$, as

$$y(\mathbf{x}) \equiv w\mathrm{COG}(\underline{\mu}_{\tilde{B}}(y)) + (1-w)\mathrm{COG}(\bar{\mu}_{\tilde{B}}(y)) \tag{62}$$

where weight $w \in [0,1]$ and can be tuned during a design procedure. Observe that, when all sources of uncertainty disappear, so that $\underline{\mu}_{\tilde{B}}(y) = \bar{\mu}_{\tilde{B}}(y) = \mu_B(y)$, (62) reduces to

$$y(\mathbf{x}) = \mathrm{COG}(\mu_B(y)) \tag{63}$$

as required by the above Design Requirement. Similar results can be obtained for other kinds of defuzzifiers (e.g., center-of-sets defuzzification), and we leave them to the reader.

Niewiadomski et al. [67] have defined four other kinds of TR, $\mathrm{TR}_{opt}(\widetilde{B})$, $\mathrm{TR}_{pes}(\widetilde{B})$, $\mathrm{TR}_{re}(\widetilde{B})$ and $\mathrm{TR}_{rew}(\widetilde{B})$, where the lower indices mean: *opt—optimistic*, *pes—pessimistic*, *re—realistic*, and *rew—realistic-weighted*, and:

$$\mathrm{TR}_{opt}(\widetilde{B}) = \bar{\mu}_{\tilde{B}}(y) \quad \forall y \in Y \tag{64}$$

$$\mathrm{TR}_{pes}(\widetilde{B}) = \underline{\mu}_{\tilde{B}}(y) \quad \forall y \in Y \tag{65}$$

$$\mathrm{TR}_{re}(\widetilde{B}) = \frac{1}{2}[\underline{\mu}_{\tilde{B}}(y) + \bar{\mu}_{\tilde{B}}(y)] \quad \forall y \in Y \tag{66}$$

$$\mathrm{TR}_{rew}(\widetilde{B}) = w\underline{\mu}_{\tilde{B}}(y) + (1-w)\bar{\mu}_{\tilde{B}}(y) \quad \forall y \in Y \tag{67}$$

Each of the functions in (64)–(67) is a T1 FS. Observe that when all sources of uncertainty disappear, so that $\underline{\mu}_{\tilde{B}}(y) = \bar{\mu}_{\tilde{B}}(y) = \mu_B(y)$, then

$$\mathrm{TR}_{opt}(\widetilde{B}) = \mathrm{TR}_{pes}(\widetilde{B}) = \mathrm{TR}_{re}(\widetilde{B}) = \mathrm{TR}_{rew}(\widetilde{B}) = \mu_B(y) \quad \forall y \in Y \tag{68}$$

also as required by the above Design Requirement.

Observe also that (62) and (67) look similar, in that they both involve lower and upper MFs of $\mu_{\tilde{B}}(y)$, but they are conceptually quite different, i.e. (62) by-passes TR completely, whereas (67) does not. Using (67), we could of course compute $y(\mathbf{x})$ as, e.g. $y(\mathbf{x}) = \mathrm{COG}(\mathrm{TR}_{rew}(\widetilde{B}))$.

Gorzalczany [19,20] introduced the following interesting function of[16] $\underline{\mu}_{\tilde{B}}(y)$ and $\bar{\mu}_{\tilde{B}}(y)$:

$$f(y) \equiv \frac{1}{2}[\underline{\mu}_{\tilde{B}}(y) + \bar{\mu}_{\tilde{B}}(y)] \times \{1 - [\bar{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]\} \quad \forall y \in Y \tag{69}$$

in which he calls $[\bar{\mu}_{\tilde{B}}(y) - \underline{\mu}_{\tilde{B}}(y)]$ the "bandwidth" of $\mu_{\tilde{B}}(y)$. He then suggests two ways to compute $y(\mathbf{x})$ from $f(y)$:

$$y_1(\mathbf{x}) = \arg\max_{\forall y \in Y} f(y) \tag{70}$$

$$y_2(\mathbf{x}) = \mathrm{median}\ (f(y)) \tag{71}$$

Observe that when all sources of uncertainty disappear, so that $\underline{\mu}_{\tilde{B}}(y) = \bar{\mu}_{\tilde{B}}(y) = \mu_B(y)$, then

$$f(y) = \mu_B(y) \quad \forall y \in Y, \tag{72}$$

again, as required.

So now it seems that we have numerous ways to bypass TR and obtain a defuzzified output, all of which satisfy the basic requirement stated above. Which of these ways is best in some sense is an open question.

[15] An explicit formula for $\mu_{\tilde{B}}(y)$ can be found in [34] and [47, Eq. (10-30)].

[16] We are explaining his function using our notation, which is quite different from his notation.

There is still a problem. For random systems, we usually require both the mean and standard deviation. The latter is extremely important because it provides a measure of dispersion about the mean, and the larger (smaller) the random uncertainty, the larger (smaller) the dispersion. Mendel [47, pp. 8–9] argues that we should not expect less of a FLS, i.e. if we view the defuzzified output of a T2 FLS as analogous to the mean, we also need a measure of dispersion about the defuzzified output, something that is analogous to standard deviation. The Karnik-Mendel TR set provides such an uncertainty measure, but all of the above approaches do not (yet) lead to such a measure. How to obtain a measure (or measures) of uncertainty for $\mathrm{TR}_{opt}(\widetilde{B})$, $\mathrm{TR}_{pes}(\widetilde{B})$, $\mathrm{TR}_{re}(\widetilde{B})$, $\mathrm{TR}_{rew}(\widetilde{B})$, and $f(y)$ is an open question. If such a measure will require numerical integration (e.g., like an integrated-squared error), then the computational cost for doing this should be compared against the computational cost for TR performed using the KM algorithms.

An interesting alternative to inventing new kinds of TR has been provided by Wu and Mendel [95]. In their approach, they replace TR with lower and upper bounds—*uncertainty bounds*—for the end-points of the TR set, and those bounds, which are optimal in a minimax sense, can be computed without having to perform TR.[17] Because these uncertainty bounds have been used in some other approaches to bypassing TR (described below after (87)), we state them next.

To begin, four centroids (also called *boundary T1 FLSs*) are defined, all of which can be computed once $\underline{f}^l$ and $\bar{f}^l$ $(l = 1, \ldots, M)$ have been computed. In these centroids, y_l^i and y_r^i are the left- and right-end points of the centroid of the ith consequent IT2 FS. These consequent centroids only have to be computed (and stored) one time after the IT2 FLS has been designed, since they do not depend upon the input to the FLS. Note, e.g. that in (73) {LMF, left} refers to the fact that this centroid only uses lower MFs of the firing interval and left-end point values of the consequent set centroid.

$$\{\text{LMFs, left}\}: \quad y_l^{(0)}(\mathbf{x}) = \sum_{i=1}^{M} \underline{f}^i y_l^i \Big/ \sum_{i=1}^{M} \underline{f}^i \tag{73}$$

$$\{\text{LMFs, right}\}: \quad y_r^{(M)}(\mathbf{x}) = \sum_{i=1}^{M} \underline{f}^i y_r^i \Big/ \sum_{i=1}^{M} \underline{f}^i \tag{74}$$

$$\{\text{UMFs, left}\}: \quad y_l^{(M)}(\mathbf{x}) = \sum_{i=1}^{M} \bar{f}^i y_l^i \Big/ \sum_{i=1}^{M} \bar{f}^i \tag{75}$$

$$\{\text{UMFs, right}\}: \quad y_r^{(0)}(\mathbf{x}) = \sum_{i=1}^{M} \bar{f}^i y_r^i \Big/ \sum_{i=1}^{M} \bar{f}^i \tag{76}$$

Uncertainty bounds are provided in the following:

Theorem 3 (Minimax uncertainty bounds [95]). *The end-points $y_l(\mathbf{x})$ and $y_r(\mathbf{x})$ of the TR set of an IT2 FLS for the input $\mathbf{x}$, are bounded from below and above by $\underline{y}_l(\mathbf{x}) \leqslant y_l(\mathbf{x}) \leqslant \bar{y}_l(\mathbf{x})$ and $\underline{y}_r(\mathbf{x}) \leqslant y_r(\mathbf{x}) \leqslant \bar{y}_r(\mathbf{x})$, where*:

$$\bar{y}_l(\mathbf{x}) = \min\left\{y_l^{(0)}(\mathbf{x}), y_l^{(M)}(\mathbf{x})\right\} \tag{77}$$

$$\underline{y}_r(\mathbf{x}) = \max\left\{y_r^{(0)}(\mathbf{x}), y_r^{(M)}(\mathbf{x})\right\} \tag{78}$$

$$\underline{y}_l(\mathbf{x}) = \bar{y}_l(\mathbf{x}) - \left[\frac{\sum_{i=1}^{M}(\bar{f}^i - \underline{f}^i)}{\sum_{i=1}^{M}\bar{f}^i \sum_{i=1}^{M}\underline{f}^i} \times \frac{\sum_{i=1}^{M}\underline{f}^i(y_l^i - y_l^1)\sum_{i=1}^{M}\bar{f}^i(y_l^M - y_l^i)}{\sum_{i=1}^{M}\underline{f}^i(y_l^i - y_l^1) + \sum_{i=1}^{M}\bar{f}^i(y_l^M - y_l^i)}\right] \tag{79}$$

$$\bar{y}_r(\mathbf{x}) = \underline{y}_r(\mathbf{x}) + \left[\frac{\sum_{i=1}^{M}(\bar{f}^i - \underline{f}^i)}{\sum_{i=1}^{M}\bar{f}^i \sum_{i=1}^{M}\underline{f}^i} \times \frac{\sum_{i=1}^{M}\bar{f}^i(y_r^i - y_r^1)\sum_{i=1}^{M}\underline{f}^i(y_r^M - y_r^i)}{\sum_{i=1}^{M}\bar{f}^i(y_r^i - y_r^1) + \sum_{i=1}^{M}\underline{f}^i(y_r^M - y_r^i)}\right] \tag{80}$$

Observe that the four bounds in (77)–(80) can be computed without having to perform TR. Wu and Mendel then approximate the TR set, as

[17] In [95] there are detailed derivations of the uncertainty bounds for center-of-sets TR (because it handles non-symmetrical shoulder MFs better than do other kinds of TR); however, these results are also applicable to other kinds of TR, as explained in [95, Table V].

$$[y_l(\mathbf{x}), y_r(\mathbf{x})] \approx [(\underline{y}_l(\mathbf{x}) + \bar{y}_l(\mathbf{x}))/2, (\underline{y}_r(\mathbf{x}) + \bar{y}_r(\mathbf{x}))/2] \tag{81}$$

and compute the output of the IT2 FLS as

$$y(\mathbf{x}) = \frac{1}{2}[(\underline{y}_l(\mathbf{x}) + \bar{y}_l(\mathbf{x}))/2 + (\underline{y}_r(\mathbf{x}) + \bar{y}_r(\mathbf{x}))/2] \tag{82}$$

(instead of as $(y_l(\mathbf{x}) + y_r(\mathbf{x}))/2$). So, *by using the uncertainty bounds, they obtain both an approximate TR set as well as a defuzzified output*. They show that

Corollary 2 [95]. *The difference $\delta(\mathbf{x})$ between the defuzzified outputs of the TR set and its approximation set for the input $\boldsymbol{x}$, which is defined as*

$$\delta(\mathbf{x}) = \left| \frac{y_l(\mathbf{x}) + y_r(\mathbf{x})}{2} - \frac{1}{2}\left[\frac{\underline{y}_l(\mathbf{x}) + \bar{y}_l(\mathbf{x})}{2} + \frac{\underline{y}_r(\mathbf{x}) + \bar{y}_r(\mathbf{x})}{2}\right]\right| \tag{83}$$

is bounded from above as

$$\delta(\mathbf{x}) \leqslant \frac{1}{4}[(\bar{y}_l(\mathbf{x}) - \underline{y}_l(\mathbf{x})) + (\bar{y}_r(\mathbf{x}) - \underline{y}_r(\mathbf{x}))] \tag{84}$$

The next theorem helps to justify bypassing TR during the operational stage of an IT2 FLS:

Theorem 4 [95]. *For a group of input–output data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ and an IT2 FLS, let the risk function (i.e., the sample mean of the squared error), R_{TR}, associated with the TR set $[y_l(\mathbf{x}), y_r(\mathbf{x})]$ be given as*

$$R_{\mathrm{TR}} = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - \frac{y_l(\mathbf{x}_i) + y_r(\mathbf{x}_i)}{2}\right]^2 \tag{85}$$

and the risk function, R_{APP}, associated with its approximation set $[(\underline{y}_l(\mathbf{x}) + \bar{y}_l(\mathbf{x}))/2, (\underline{y}_r(\mathbf{x}) + \bar{y}_r(\mathbf{x}))/2]$, be given as

$$R_{\mathrm{APP}} = \frac{1}{N}\sum_{i=1}^{N}\left\{y_i - \frac{1}{2}\left[\frac{\underline{y}_l(\mathbf{x}_i) + \bar{y}_l(\mathbf{x}_i)}{2} + \frac{\underline{y}_r(\mathbf{x}_i) + \bar{y}_r(\mathbf{x}_i)}{2}\right]\right\}^2 \tag{86}$$

Then

$$\left|\sqrt{R_{\mathrm{TR}}} - \sqrt{R_{\mathrm{APP}}}\right| \leqslant \frac{1}{4}\sqrt{\frac{1}{N}\sum_{i=1}^{N}[(\bar{y}_l(\mathbf{x}_i) - \underline{y}_l(\mathbf{x}_i)) + (\bar{y}_r(\mathbf{x}_i) - \underline{y}_r(\mathbf{x}_i))]^2} \tag{87}$$

Beginning with Theorem 3 and Corollary 2, three approaches for eliminating TR have been proposed:

1. Wu and Mendel [95] use TR during the design of the IT2 FLS but then run the IT2 FLS using (77)–(82). During the design, instead of only using R_{TR}, they use a weighted average between R_{TR} and the $N\delta^2(\mathbf{x}_i)$, $i = 1, \ldots, N$. In this way, they trade off some RMSE with not having to perform TR. The drawback to this approach is that TR is still performed during the design step.
2. Lynch et al. [40] abandon TR completely. They replace all of the IT2 FLS computations with those in (77)–(82), i.e. (77)–(82) are their IT2 FLS. Their design of this IT2 FLS is then based on minimizing R_{APP}. They found that the errors incurred by doing this are extremely small. This is a very clever approach.
3. Melgerejo et al. [42,43] and Melgerejo and Penha-Reyes [41] have designed a VLSI IT2 FLS chip (see Section 3.6). It is also based on (77)–(82), and the Wu–Mendel design and implementation approach; however, it could also be based on the Lynch et al. approach. Because (77)–(82) are hard-wired, the design stage can be done very quickly.

Wu and Tan [94] have approached the elimination of TR from a very different point of view. To begin, they define *equivalent type-1 sets* (*ET1S*) as "the collection of type-1 sets that can be used in place of the FOUs in a type 2 FLS". They state: "The key idea behind the proposed type-reducer is to view a T2 set as being equivalent to a collection of ET1Ss. TR is then simplified to finding the ET1S corresponding to a particular input. More specifically, the type-reducer needs to identify the *equivalent T1 membership grade* (f_{eq}) for each interval firing-strength. Once the *equivalent T1 membership grade* has been deduced, the firing set of a T2 FS reduces to a crisp value and a traditional fuzzy inference engine and defuzzifier can be employed to find the output of the T2 FLS. In summary, the proposed TR procedure is applied before the inference engine ...". Even though they refer to this as a "proposed type-reducer", it is not actually TR (as they have noted) because it does not lead to a TR interval; hence, "it does not allow the uncertainties to flow to the inference engine ...". It is, in effect, a new kind of defuzzification strategy that is able to bypass TR. They claim that this approach is very fast and the "... computational load is reduced because the inference engine behaves like the one in a T1 FLS". So far, they have demonstrated this ET1S methodology for a two-input PI controller. Whether or not this methodology generalizes to more complicated systems is an open question.

3.6. Hardware realization of an IT2 FLS

One of the reasons that T1 FLSs became so widely used is that in the 1980s Takagi developed the so-called *fuzzy chip*. This made it possible to implement T1 FL in hardware, which in turn led to commercialization of FL products. Recently, Melgarejo et al. [42,43] and Melgarejo and Penha-Reyes [41] have proposed a hardware architecture for an already-designed IT2 FLS, one that uses the Wu–Mendel uncertainty bounds as TR [(77)–(82)] instead of the KM algorithm for TR, and also the minimum *t*-norm. According to [41], " Distributed arithmetic is proposed for efficiently computing on hardware the type-reduction stage ... [and there is a] reconfigurable rule base, which is implemented in FPGA (Field Programmable Gate Array) technology. Results show that the processor performs more than 30 million of type-2 fuzzy inferences per second".

4. The fuzzy weighted average (FWA) and TR

The FWA is a weighted average involving T1 FSs. It has been studied in multiple criteria decision making [12,15,22,23,33,37] and is useful as an aggregation (fusion) method in situations where [see (88)] decisions (x_i) and expert weights (w_i) are modeled as T1 FSs, or where the decisions are modeled as either crisp numbers or interval sets, and the expert weights are still modeled as T1 FSs.

Sometimes the same or similar problem is solved in different settings. This is the case for computing the FWA and the GC of IT2 FSs, which is why we are discussing it in this paper. Consider the following *weighted average*:

$$y = \sum_{i=1}^{n} w_i x_i \Big/ \sum_{i=1}^{n} w_i = f(w_1, \ldots, w_n, x_1, \ldots, x_n) \tag{88}$$

In (88), w_i are *weights* that act upon *attributes* (indicators, features, decisions, etc.), x_i. In the FWA, $\forall x_i$ are T1 fuzzy numbers, i.e. each x_i is described by the MF of a T1 FS, $\mu_{X_i}(x_i)$, that must be pre-specified, and $\forall w_i$ are also T1 fuzzy numbers, i.e. each w_i is described by the MF of a T1 FS, $\mu_{W_i}(w_i)$, that must also be pre-specified. In (88), y is a T1 FS, with MF $\mu_Y(y)$, but there is no known closed-form formula for computing $\mu_Y(y)$. Instead, α-cuts, an α-cut Decomposition Theorem [30] of a T1 FS, and a variety of algorithms can be used to compute $\mu_Y(y)$ (e.g., [23,37,33]).

When α-cuts are used to compute the FWA, the complete range of the membership [0,1] of the fuzzy numbers is discretized into the following finite number of m α-cuts, $\alpha_1, \ldots, \alpha_m$, where the degree of accuracy depends on the number of α-cuts, i.e. m. For each α_j, the corresponding intervals for X_i in x_i and W_i in w_i are found. The end-points of the intervals of x_i and w_i ($i = 1, \ldots, n$) are denoted by $[a_i(\alpha_j), b_i(\alpha_j)]$ and $[c_i(\alpha_j), d_i(\alpha_j)]$, respectively. Using these α-cuts, one computes the corresponding α-cut of y, namely $y(\alpha_j)$, i.e.

$$y(\alpha_j) = [f_L^*(\alpha_j), f_R^*(\alpha_j)] \tag{89}$$

$\mu_Y(y)$ is then computed using $y(\alpha_1), \ldots, y(\alpha_m)$ as follows. Following [30], let $I_{\alpha_{jY}}(y)$ denote the indicator function:

$$I_{\alpha_{jY}}(y) = \begin{cases} 1 & \forall y \in [f_L^*(\alpha_j), f_R^*(\alpha_j)] \\ 0 & \forall y \notin [f_L^*(\alpha_j), f_R^*(\alpha_j)] \end{cases} \tag{90}$$

Then

$$\mu_Y(y) = \sup_{\forall \alpha_j \in [0,1] (j=1,\ldots,m)} \alpha_j I_{\alpha_{jY}}(y) \tag{91}$$

Liou and Wang [37] were the first to observe that since the x_i appear only in the numerator of (88), only the smallest values of the x_i are used to find the smallest value of (88), and only the largest values of the x_i are used to find the largest value of (88); hence,

$$f_L^*(\alpha_j) = \min_{\substack{\forall x_i \in [a_i(\alpha_j), b_i(\alpha_j)] \\ \forall w_i \in [c_i(\alpha_j), d_i(\alpha_j)]}} f(w_1, \ldots, w_n, x_1, \ldots, x_n | \alpha_j) = \min_{\forall w_i \in [c_i(\alpha_j), d_i(\alpha_j)]} f(w_1, \ldots, w_n, a_1, \ldots, a_n | \alpha_j) \tag{92}$$

where

$$f(w_1, \ldots, w_n, a_1, \ldots, a_n | \alpha_j) = \sum_{i=1}^{n} w_i(\alpha_j) a_i(\alpha_j) \Bigg/ \sum_{i=1}^{n} w_i(\alpha_j) \tag{93}$$

and

$$f_R^*(\alpha_j) = \max_{\substack{\forall x_i \in [a_i(\alpha_j), b_i(\alpha_j)] \\ \forall w_i \in [c_i(\alpha_j), d_i(\alpha_j)]}} f(w_1, \ldots, w_n, x_1, \ldots, x_n | \alpha_j) = \max_{\forall w_i \in [c_i(\alpha_j), d_i(\alpha_j)]} f(w_1, \ldots, w_n, b_1, \ldots, b_n | \alpha_j) \tag{94}$$

where

$$f(w_1, \ldots, w_n, b_1, \ldots, b_n | \alpha_j) = \sum_{i=1}^{n} w_i(\alpha_j) b_i(\alpha_j) \Bigg/ \sum_{i=1}^{n} w_i(\alpha_j) \tag{95}$$

Comparing (93) and (95) with the weighted averages in (40) and (41), respectively, we see that they are exactly the same, provided we set $\theta_i = w_i(\alpha_j)$ and $x_i = a_i(\alpha_j)$ in (40), and $\theta_i = w_i(\alpha_j)$ and $x_i = b_i(\alpha_j)$ in (41). Consequently, the KM algorithms can be used to compute $f_L^*(\alpha_j)$ and $f_R^*(\alpha_j)$. Recently, Liu and Mendel [38] have compared this approach to computing the FWA with Lee and Park's Efficient FWA (EFWA) Algorithm [33] because prior to their work the EFWA was the fastest way to compute the FWA. Their simulations, in which $a_i(\alpha_j)$, $b_i(\alpha_j)$, $c_i(\alpha_j)$, and $d_i(\alpha_j)$ were chosen randomly, all according to the same probability distribution (e.g., uniform, exponential, normal, etc.), and n was varied from 2 to 200, demonstrated that for each α-cut, convergence occurred (to within a 97.5% confidence interval) in three iterations, regardless of how many terms were in the FWA (i.e., n), or how $a_i(\alpha_j)$, $b_i(\alpha_j)$, $c_i(\alpha_j)$, and $d_i(\alpha_j)$ were distributed, whereas the EFWA algorithm converged

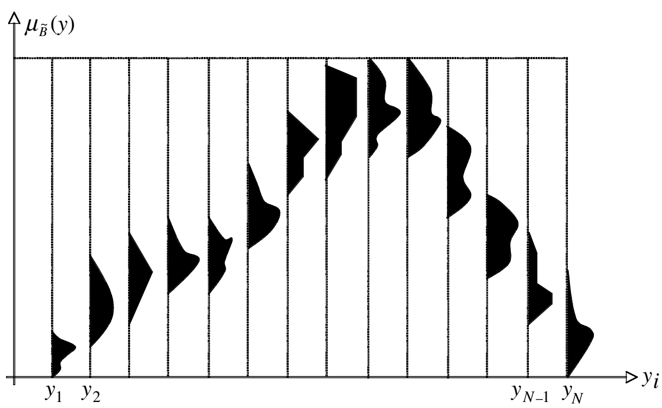

Fig. 3. Hypothetical combined rule output set for a general T2 FLS. At each y_i there is a different secondary MF (blackened figures) that come out of the page in the third dimension.

in $\ln n + c$ iterations, where c depended on the way in which $a_i(\alpha_j)$, $b_i(\alpha_j)$, $c_i(\alpha_j)$, and $d_i(\alpha_j)$ were distributed. So, a by-product of T2 works, the KM algorithms, can now be used to compute the solution to a non-T2 problem, the FWA.

The FWA can also be used to compute the TR set for general T2 FSs. Consider the general T2 FS depicted in Fig. 3, in which the filled-in figures along the primary-variable axis denote secondary MFs, which can all be different. Here we are not concerned with how this T2 FS was obtained; we are only interested in how to compute its TR set. Instead of using, e.g. $Y_c(\mathbf{x})$ as expressed in (24), we define the TR set as the weighted average of all of the sampled secondary MFs. This is a FWA in which x_i are crisp (they equal y_i) and w_i in (88) are the secondary MFs, which are T1 FSs. $y(\alpha_j)$ in (89) is the jth α-cut for the TR set, but now in (93) and (95), $a_i(\alpha_j) = b_i(\alpha_j) = y_i$. How to extend this discussion to other kinds of TR is straightforward and is left to the reader.

5. Computing with words

In 1996, Zadeh published his first seminal work on *computing with words*. Since that time entire books and other articles have been devoted to this important subject. A small sampling of these are [31,32,46,48,49,84,90,100–102]. Mendel [50] has proposed IT2 FSs as models for words, because words mean different things to different people, and so a fuzzy set model for words is needed that can capture a measure of their uncertainty.

In order to establish IT2 FS models for words, we need to collect data about words from a group of subjects. Next, two very different approaches for doing this are described. Both approaches map data collected from subjects into a parsimonious parametric model of a FOU, and illustrate the combining of fuzzy sets and statistics—*type-2 fuzzistics*.[18]

In one approach, called the *person MF approach* [53], we:

1. Collect person MF data (a person-MF is a FOU that a person provides on a prescribed scale for a primary-variable) that reflects both the intra- and inter-levels of uncertainties about a word, from a group of people;
2. Define an IT2 FS model for a word as a specific aggregation of all such person MFs; and
3. Mathematically model and approximate this aggregation.

This approach is based on six premises:

- P1. Uncertainty about a word is of two kinds: (a) *intra-uncertainty*, which is the uncertainty a person has about the word, and (b) *inter-uncertainty* which is the uncertainty that a group of people have about the word.
- P2. *Intra-uncertainty* about a word, A, can be modeled using an[19] *IT2 person FS*, $\widetilde{A}(p_j)$, where $j = 1, \ldots, n_A$. Such an IT2 FS is completely described by its person-FOU.
- P3. *Inter-uncertainty* about a word can be modeled by means of an equally weighted aggregation[19] of each person's word FS, $\widetilde{A}(p_j)$ $(j = 1, 2, \ldots, n_A)$.
- P4. A natural way to aggregate a group of subject's equally weighted word FSs is by the mathematical operation of the union.[19]
- P5. When person-MFs have been collected from a sufficient number of subjects, their union will be a filled-in word FS $\widetilde{A}_{FI}$ with associated FOU—the *filled-in* FOU—FOU$(\widetilde{A}_{FI})$. It is assumed that fill-in has occurred.

[18] This term was first coined in [50].

[19] Mendel [53] explains why an IT2 FS leads to a *first-order uncertainty* model and how difficult it would be to do otherwise; it explains equal versus non-equal weighting, and why equal weighting again leads to a first-order uncertainty model and how difficult it would be to do otherwise; and, it also explains other kinds of aggregation and why the union operation seems best.

P6. A *filled-in parametric model*, $\widehat{A}$, for $\widetilde{A}_{FI}$ is one that is described by two functions, a lower MF $\underline{\mu}_{\hat{A}}(x)$ and an upper MF $\bar{\mu}_{\hat{A}}(x)$. These functions have shapes that are chosen ahead of time (e.g., triangle, Gaussian, piecewise-linear, trapezoidal, etc.) and each shape is characterized by a small number of parameters, some or all of which may be shared by both $\underline{\mu}_{\hat{A}}(x)$ and $\bar{\mu}_{\hat{A}}(x)$. These parameters are fixed during some sort of design procedure.

Observe that mathematical IT2 FS models are only used at the very end of this approach.

Person MFs can only be collected from people who are already very knowledgeable about a FS, and therefore other techniques must be established for collecting word data from the vast majority.

In a second approach, called the *interval end-points approach* [53], we:

1. Collect interval end-point data about a word from a group of subjects;
2. Establish end-point statistics for the data; and
3. Map those statistics into a pre-specified parametric FOU.

This method is analogous in statistical modeling to first choosing the underlying probability distribution (i.e., data-generating model) and then fitting the parameters of that model using data and a meaningful design method, e.g. the method of maximum-likelihood. This method is based on the centroid bounds that were described in Theorem 2 (Section 3.3.5). The details of how to perform Step 3 are in [58] for symmetric FOUs and are under study for non-symmetrical FOUs [59].

We have mentioned all of this because we believe that IT2 FSs will play a very important role in Zadeh's paradigm of CWW, and because much more remains to be done to turn it into an operational reality.

We also wish to mention that Türksen and his students, beginning with [68,69], have written on the important problem of how to elicit information from a subject and how that information can then be mapped into the MF of a T1 FS.[20] In [84] there are some plots to show that the T1 MFs are lower and upper bounded. Although Türksen does not refer to these plots as "FOUs" for words, nor does he try to directly model this uncertainty using a T2 FS, the plots are good demonstrations (confirmations) that when MF data are collected from people, they lead to a FOU.

In a FL rule-based system, not only are there antecedent and consequent words, whose uncertainties have been modeled using T2 FSs, especially IT2 FSs, but there are also the connector words, AND/OR. Türksen and Yao [85] and Türksen (e.g., [82–84]) have argued that it is very important to model the uncertainties of these two connector words. His approach is based on[21] Shannon's [76] introduction of the disjunctive and conjunctive normal forms, the DNF and CNF. These forms are used in classical logic for facilitating proofs of some theorems, or as a tool to simplify complicated logical statements,[22] e.g., we could begin with a complicated statement that involves many AND's and OR's. Such complicated expressions can actually be simplified by replacing the AND's and OR's by their normal forms, since this leads to an expansion of the original complicated statement and the separate terms in the expansion are simpler than in the original statement.

Türksen et al. have fuzzified the CNF and DNF to fuzzy CNF (FCNF) and FDNF. Based on the FCNF and the FDNF each AND (intersection)/OR (union) operation leads to an interval set whose computation depends on the specific choices one makes for a *t*-norm or a *t*-conorm. So, even if antecedent and consequents are modeled as T1 FSs, when the AND/OR operations are modeled using the FDNF and FCNF operations, the result will be an IT2 FS. Things seem to get very complicated when there is a chain of AND/OR operations, as frequently occurs in many FL rules, and no results seem to have been published on how to propagate such a chain of normal forms.

Using FCNF and FDNF models for the AND/OR operations transforms a T1 FLS into a T2 FLS, and using them along with interval T2 FS models for rule-words transforms an IT2 FLS into an IT3 FLS [84]. How one actually computes a T3 FS and then goes from it to a number are un-researched questions.

[20] See [30] for references authored by other researchers on mapping data into T1 MF models.

[21] The author would like to thank Prof. George Klir for providing him with historical information about the DNF and CNF.

[22] The author would also like to thank Prof. Gandhi Puvadda, Dept. of Electrical Engineering, University of Southern California, who has been teaching logic design for decades, for sharing this knowledge with him.

Other approaches for modeling the uncertainties about the AND/OR connector words, such as[23] parametric *t*-norms (*t*-conorms), compensatory AND, and S-OWA operators, preserve all of the mathematics in the T1 domain. Additionally, when antecedents and consequents are modeled as IT2 FSs and compensatory AND and S-OWA models are used, then all of the mathematics remains in the IT2 FS domain [97]. Although there may be some performance advantage to modeling the AND/OR operations to account for their uncertainties, the additional computation required to implement such a system (it is non-trivial) must be justified.

When FSs are used in the CWW paradigm, then we have argued that, because words can mean different things to different people (and data collected from subjects supports this), we should model the antecedent and consequent words using IT2 FSs. An interesting question is "Do the AND/OR connector words also mean different things to different people?" Experimental psychologists (e.g., [16]) have conducted studies into these issues and they have arrived at some interesting conclusions, namely: (1) most people have no difficulty in understanding the AND connector, whereas (2) many people have difficulty in understanding the OR connector (because one must decide whether the OR is inclusive or exclusive). In fact, they have as much difficulty with OR as they do with logical implication. Based on these human-centered studies, it would seem that the CWW paradigm does not need to include an uncertainty model for the AND connector, but it may need to include an uncertainty model for the OR connector. Any uncertainty model should agree with a human's uncertainty about the basic words AND/OR because those are the words that they hear (read). As yet, there is no experimental evidence that logically equivalent models such as normal forms bear a relationship to what a person hears (reads). Which model is appropriate should somehow be tested on human subjects so that it will be in agreement with their internal uncertainty models for OR. This remains to be done.

6. Conclusions

Much research has been and is continuing to be done on both general and IT2 FSs. Works on these subjects are now occurring worldwide. Applications are broadening, and control applications, which were the original bread-and-butter ones for T1 FLSs, are now a major focus of attention for IT2 FLSs. Additionally, IT2 FSs are being used to implement Zadeh's [99] CWW paradigm; but, much more work needs to be done on this.

Only time will tell whether or not taking the step from a T1 FS to a T2 FS has made a significant difference. What is needed is a mathematical theory that establishes testable conditions for when a T2 FLS will outperform a T1 FLS. These conditions will, no doubt, depend upon how much uncertainty is present, and remain to be developed.

Finally, it is this author's opinion that any problem that has previously lent itself to T1 FSs, in which MFs are uncertain, is an excellent candidate for re-examination using T2 FSs.

Acknowledgement

The author wishes to thank the reviewers for their excellent suggestions that have been incorporated into this paper.

References

[1] J.J. Buckley, Fuzzy Probabilities: New Approaches and New Applications, Physica-Verlag, New York, 2003.
[2] H. Bustince, Indicator of inclusion grade for interval-valued fuzzy sets. Application to approximate reasoning based on interval-valued fuzzy sets, Int. J. Approx. Reas. 23 (2000) 137–209.
[3] O. Castillo, P. Melin, Adaptive noise cancellation using type-2 fuzzy logic and neural networks, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004.
[4] O. Castillo, G. Huesca, F. Valdez, Evolutionary computing for optimizing type-2 fuzzy systems in intelligent control of non-linear dynamic plants, in: Proceedings of North American Fuzzy Information Processing Society (NAFIPS), Ann Arbor, MI, June 2005, pp. 247–251.
[5] S.M. Chen, M.S. Yeh, P.Y. Hsiao, A comparison of similarity measures of fuzzy values, Fuzzy Sets Syst. 72 (1995) 79–89.

[23] There are many references about these operators. They can be found, e.g. in [96].

[6] S. Coupland, R.I. John, A new and efficient method for the type-2 meet operation, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004, pp. 959–964.
[7] S. Coupland, R.I. John, Towards more efficient type-2 fuzzy logic systems, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 236–241.
[8] V. Cross, T. Sudkamp, Similarity and compatibility in fuzzy set theory, Studies in Fuzziness and Soft Computing, vol. 93, Physica-Verlag, Heidelberg, 2002.
[9] L. Di Lascio, A. Gisolfi, A. Nappi, Medical differential diagnosis through type-2 fuzzy sets, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 371–376.
[10] F. Doctor, H. Hagras, V. Callaghan, A type-2 fuzzy embedded agent for ubiquitous computing environment, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004.
[11] F. Doctor, H. Hagras, V. Callaghan, A type-2 fuzzy embedded agent to realise ambient intelligence in ubiquitous computing environments, Informat. Sci. 171 (2005) 309–334.
[12] W.M. Dong, F.S. Wong, Fuzzy weighted averages and implementation of the extension principle, Fuzzy Sets Syst. 21 (1987) 183–199.
[13] D. Dubois, H. Prade, Operations on fuzzy numbers, Int. J. Syst. Sci. 9 (1978) 613–626.
[14] D. Dubois, H. Prade, Operations in a fuzzy-valued logic, Informat. Cont. 43 (1979) 224–240.
[15] D. Dubois, H. Fargier, J. Fortin, A generalized vertex method for computing with fuzzy intervals, in: Proceedings of FUZZ IEEE 2004, Budapest, Hungary, 2004, pp. 541–546.
[16] J. St. B.T. Evans, S.E. Newstead, R.M.J. Byrne, Human Reasoning: The Psychology of Deduction, Lawrence Erlbaum Associates, Hove, UK, 1993.
[17] J. Figueroa, J. Posada, J. Soriano, M. Melgarejo, S. Rojas, A type-2 fuzzy controller for tracking mobile objects in the context of robotic soccer games, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 359–364.
[18] J.M. Garibaldi, S. Musikasuwan, T. Ozen, The association between non-stationary and interval type-2 fuzzy sets: a case study, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 224–229.
[19] M.B. Gorzalczany, Decision making in signal transmission problems with interval-valued fuzzy sets, Fuzzy Sets Syst. 23 (1987) 191–203.
[20] M.B. Gorzalczany, Interval-valued fuzzy controller based on verbal model of object, Fuzzy Sets Syst. 28 (1988) 45–53.
[21] S. Greenfield, R.I. John, S. Coupland, A novel sampling method for type-2 defuzzification, submitted for publication.
[22] Y.-Y. Guh, C.-C. Hon, E.S. Lee, Fuzzy weighted average: the linear programming approach via Charnes and Cooper's rule, Fuzzy Sets Syst. 117 (2001) 157–160.
[23] Y.-Y. Guh, C.-C. Hon, K.-M. Wang, E.S. Lee, Fuzzy weighted average: a max–min paired elimination method, Comput. Math. Appl. 32 (1996) 115–123.
[24] H. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Trans. Fuzzy Syst. 12 (4) (2004) 524–539.
[25] H. Hagras, A type-2 fuzzy logic controller for autonomous mobile robots, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004.
[26] C. Hwang, F.C.-H. Rhee, An interval type-2 fuzzy spherical shells algorithm, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004.
[27] R.I. John, P.R. Innocent, Modeling uncertainty in clinical diagnosis using fuzzy logic, IEEE Trans. Syst., Man, Cybernet. Part B: Cybernet. 35 (2005) 1340–1350.
[28] N.N. Karnik, J.M. Mendel, Operations on type-2 fuzzy sets, Fuzzy Sets Syst. 122 (2001) 327–348.
[29] N.N. Karnik, J.M. Mendel, Centroid of a type-2 fuzzy set, Informat. Sci. 132 (2001) 195–220.
[30] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, Upper Saddle River, NJ, 1995.
[31] J. Lawry, An alternative to computing with words, Int. J. Uncert. Fuzz. Know. Syst. 9 (Suppl.) (2001) 3–16.
[32] J. Lawry, J. Shanahan, A. Ralescu (Eds.), Modeling With Words, Lecture Notes in Artificial Intelligence 2873, Springer, New York, 2003.
[33] D.H. Lee, D. Park, An efficient algorithm for fuzzy weighted average, Fuzzy Sets Syst. 87 (1997) 39–45.
[34] Q. Liang, J.M. Mendel, Interval type-2 fuzzy logic systems: theory and design, IEEE Trans. Fuzzy Syst. 8 (2000) 535–550.
[35] Q. Liang, L. Wang, Sensed signal strength forecasting for wireless sensors using interval type-2 fuzzy logic system, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 25–30.
[36] P.-Z. Lin, C.-F. Hsu, T.-T. Lee, Type-2 fuzzy logic controller design for buck DC–DC converters, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 365–370.
[37] T.-S. Liou, M.-J.J. Wang, Fuzzy weighted average: an improved algorithm, Fuzzy Sets Syst. 49 (1992) 307–315.
[38] F. Liu, J.M. Mendel, Aggregation using the fuzzy weighted average, as computed by the KM algorithms, submitted for publication.
[39] L. Lushu, Random fuzzy sets and fuzzy martingales, Fuzzy Sets Syst. 69 (1995) 181–192.
[40] C. Lynch, H. Hagras, V. Callaghan, Embedded type-2 FLC for real-time speed control of marine and traction diesel engines, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 347–352.
[41] M.C.A. Melgarejo, C.A. Penha-Reyes, Hardware realization of an interval type-2 fuzzy inference system, IEEE Trans. Fuzzy Syst., in press.
[42] M.C.A. Melgarejo, A. Garcia, C.A. Penha-Reyes, Pro-two: a hardware based platform for real time type-2 fuzzy inference, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004.

[43] M.C.A. Melgarejo, C.A. Penha-Reyes, A. Garcia, Computational model and architectural proposal for a hardware type-2 fuzzy system, in: Proceedings of 2nd IASTED Conference Neural Network and Computational Intelligence, Grindewald, 2004, pp. 279–284.
[44] P. Melin, O. Castillo, A new method for adaptive model-based control of non-linear plants using type-2 fuzzy logic and neural networks, in: Proceedings of IEEE FUZZ Conference, St. Louis, MO, May 2003, pp. 420–425.
[45] P. Melin, O. Castillo, A new approach for quality control of sound speakers combining type-2 fuzzy logic and the fractal dimension, in: Proceedings of International Conference NAFIPS 2003, Chicago, USA, July 2003, pp. 20–25.
[46] J.M. Mendel, Computing with words, when words can mean different things to different people, in: Proceedings of Third International ICSC Symposium on Fuzzy Logic and Applications, Rochester Univ., Rochester, NY, June 1999.
[47] J.M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, Upper-Saddle River, NJ, 2001.
[48] J.M. Mendel, The perceptual computer: an architecture for computing with words, in: Proceedings of Modeling With Words Workshop in the Proceedings of FUZZ-IEEE, Melbourne, Australia, December 2001, pp. 35–38.
[49] J.M. Mendel, An architecture for making judgments using computing with words, Int. J. Appl. Math. Comput. Sci. 12 (3) (2002) 325–335.
[50] J.M. Mendel, Fuzzy sets for words: a new beginning, in: Proceedings of IEEE International Conference on Fuzzy Systems, St. Louis, MO, 2003, pp. 37–42.
[51] J.M. Mendel, Type-2 fuzzy sets: some questions and answers, IEEE Connect., Newslett. IEEE Neural Networks Soc. 1 (2003) 10–13.
[52] J.M. Mendel, On a 50% savings in the computation of the centroid of a symmetrical interval type-2 fuzzy set, Informat. Sci. 172 (2005) 417–430.
[53] J.M. Mendel, Computing with words and its relationships with fuzzistics, Informat. Sci., in press, doi:10.1016/j.ins.2006.06.008.
[54] J.M. Mendel, R.I. John, Type-2 fuzzy sets made simple, IEEE Trans. Fuzzy Syst. 10 (2002) 117–127. April 2002.
[55] J.M. Mendel, R.I. John, F. Liu, Interval type-2 fuzzy logic systems made simple, IEEE Trans. Fuzzy Syst., in press.
[56] J.M. Mendel, F. Liu, Super-exponential convergence of the Karnik–Mendel algorithms for computing the centroid of an interval type-2 fuzzy set, IEEE Trans. Fuzzy Syst., in press.
[57] J.M. Mendel, H. Wu, Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems, IEEE Trans. Fuzzy Syst., in press.
[58] J.M. Mendel, H. Wu, Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 2, inverse problems, IEEE Trans. Fuzzy Syst., in press.
[59] J.M. Mendel, H. Wu, Type-2 fuzzistics for non-symmetric interval type-2 fuzzy sets: forward problems, IEEE Trans. Fuzzy Syst., in press.
[60] J.M. Mendel, H. Wu, New results about the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule, Informat. Sci., in press, doi:10.1016/j.ins.2006.03.003.
[61] G.M. Mendez, O. Castillo, Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 230–235.
[62] H.B. Mitchell, Pattern recognition using type-II fuzzy sets, Informat. Sci. 170 (2005) 409–418.
[63] M. Mizumoto, K. Tanaka, Some properties of fuzzy sets of type-2, Informat. Cont. 31 (1976) 312–340.
[64] M. Mizumoto, K. Tanaka, Fuzzy sets of type-2 under algebraic product and algebraic sum, Fuzzy Sets Syst. 5 (1981) 277–290.
[65] B. Moller, M. Beere, Fuzzy Randomness: Uncertainty in Civil Engineering and Computational Mechanics, Springer, New York, 2004.
[66] J. Nieminen, J. On the algebraic structure of fuzzy sets of type-2, Kybernetica 13 (4) (1977).
[67] A. Niewiadomski, J. Kacprzyk, J. Ochelska, P.S. Szczepaniak, Interval-valued linguistic summaries of databases, Control & Cybernetics, Systems Research Institute, Polish Academy of Science, Warsaw, Poland, in press.
[68] A.M. Norwich, I.B. Türkşen, The fundamental measurement of fuzziness, in: R.R. Yager (Ed.), Fuzzy Sets and Possibility Theory, Pergamon Press, NY, 1982, pp. 49–60.
[69] A.M. Norwich, I.B. Türkşen, A model for the measurement of membership and the consequences of its empirical implementation, Fuzzy Sets Syst. 12 (1984) 1–25.
[70] T. Ozen, J, M. Garibaldi, S. Musikasuwan, Preliminary investigations into modeling the variation in human decision making, in: Proceedings of 10th Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU 2004), Perugia, Italy, July 2004, pp. 641–648.
[71] S. Rahimi, M. Cobb, A. Zhou, D. Ali, H. Yang, F.E. Petry, An inexact inferencing strategy for spatial objects with determined and indeterminate boundaries, in: Proceedings of IEEE FUZZ Conference, St. Louis, MO, 2003, pp. 778–783.
[72] F.C.-H. Rhee, C. Hwang, A type-2 fuzzy *c*-means clustering algorithm, in: Proceedings of IEEE FUZZ Conference, Melbourne, Australia, December 2001, pp. 1926–1929.
[73] F.C.-H. Rhee, C. Hwang, An interval type-2 fuzzy perceptron, in: Proceedings of IEEE FUZZ Conference, Honolulu, HI, May 2002.
[74] F.C.-H. Rhee, C. Hwang, An interval type-2 fuzzy *K*-nearest neighbor, in: Proceedings of IEEE FUZZ Conference, Honolulu, HI, May 2002, pp. 802–807.
[75] R. Sepulveda, O. Castillo, P. Melin, A. Rodriguez-Diaz, O. Montiel, Integrated development platform for intelligent control based on type-2 fuzzy logic, in: Proceedings of North American Fuzzy Information Processing Society (NAFIPS), Ann Arbor, MI, June, 2005, pp. 607–610.
[76] C. Shannon, A symbolic analysis of relay and switching circuits, Am. Inst. Elect. Engrs. 57 (1938).

[77] H. Shu, Q. Liang, Wireless sensor network lifetime analysis using interval type-2 fuzzy logic systems, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 19–24.
[78] J.T. Starczewski, Extended triangular norms, submitted for publication.
[79] J.T. Starczewski, A triangular type-2 fuzzy logic system, submitted for publication.
[80] J.T. Starczewski, Extended triangular norms on Gaussian fuzzy sets, in: Proceedings of EUSFLAT-LFA, Barcelona, Spain, September 2005, pp. 872–877.
[81] W.W. Tan, J. Lai, Development of a type-2 fuzzy proportional controller, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004.
[82] I.B. Türksen, Interval-valued fuzzy sets based on normal forms, Fuzzy Sets Syst. 20 (1986) 191–210.
[83] I.B. Türksen, Interval valued fuzzy sets and fuzzy connectives, J. Interval Comput. 4 (1993) 125–142.
[84] I.B. Türksen, Type-2 representation and reasoning for CWW, Fuzzy Sets Syst. 127 (2002) 17–36.
[85] I.B. Türksen, D.D.W. Yao, Representation of connectives in fuzzy reasoning: the view through normal forms, IEEE Trans. Syst. Man Cybernet. 14 (1984) 146–150.
[86] M. Wagenknecht, K. Hartmann, Application of fuzzy sets of type-2 to the solution of fuzzy equation systems, Fuzzy Sets Syst. 25 (1988) 183–190.
[87] C. Walker, E. Walker, The algebra of fuzzy truth values, Fuzzy Sets Syst., in press.
[88] C. Walker, E. Walker, Some general comments on fuzzy sets of type-2, presented at BISCSE 2005 Forging New Frontiers, 40th of Fuzzy Pioneers, BISC Special Event in Honor of Prof. Lotfi A. Zadeh, Univ. of California, Berkeley, CA, November 2–5, 2005.
[89] C. Walker, E. Walker, Automorphisms of the algebra of fuzzy truth values, in preparation.
[90] P.P. Wang (Ed.), Computing With Words, John Wiley & Sons Inc, New York, 2001.
[91] X. Wang, B. De Baets, E. Kerre, A comparative study of similarity measures, Fuzzy Sets Syst. 73 (1995) 259–268.
[92] K. Wills, R.I. John, S. Lake, Combining categories in nursing assessment using interval valued fuzzy sets, in: Proceedings of 10th Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU 2004), Perugia, Italy, July 2004.
[93] D. Wu, W.W. Tan, A type-2 fuzzy logic controller for the liquid-level process, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004, pp. 953–958.
[94] D. Wu, W.W. Tan, Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller, in: Proceedings of IEEE FUZZ Conference, Reno, NV, May 2005, pp. 353–358.
[95] H. Wu, J.M. Mendel, Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst. 10 (2002) 622–639.
[96] H. Wu, J.M. Mendel, On choosing models for linguistic connector words for Mamdani fuzzy logic systems, IEEE Trans. Fuzzy Syst. 12 (2004) 29–44.
[97] H. Wu, J.M. Mendel, Antecedent connector word models for Interval type-2 fuzzy logic system, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004.
[98] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning–1, Informat. Sci. 8 (1975) 199–249.
[99] L.A. Zadeh, Fuzzy logic = computing with words, IEEE Trans. Fuzzy Syst. 4 (1996) 103–111.
[100] L.A. Zadeh, From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions, IEEE Trans. Circ. Syst.—I: Fundam. Theory Appl. 4 (1999) 105–119.
[101] L.A. Zadeh, Towards a generalized theory of uncertainty (GTU)–an outline, Informat. Sci. 172 (2005) 1–40, June 2005.
[102] L.A. Zadeh, J. Kacprzyk (Eds.), Computing With Words in Information/Intelligent Systems 1 & 2, Physica-Verlag, New York, 1999.
[103] J. Zeng, Z.-Q. Liu, Interval type-2 fuzzy hidden markov models, in: Proceedings of IEEE FUZZ Conference, Budapest, Hungary, July 2004.