

On the Fundamental Differences Between Interval Type-2 and Type-1 Fuzzy Logic Controllers

Dongrui Wu, *Member, IEEE*

Abstract—Interval type-2 fuzzy logic controllers (IT2 FLCs) have recently been attracting a lot of research attention. Many reported results have shown that IT2 FLCs are better able to handle uncertainties than their type-1 (T1) counterparts. A challenging question is the following: *What are the fundamental differences between IT2 and T1 FLCs?* Once the fundamental differences are clear, we can better understand the advantages of IT2 FLCs and, hence, make better use of them. This paper explains two fundamental differences between IT2 and T1 FLCs: 1) *Adaptiveness*, meaning that the embedded T1 fuzzy sets used to compute the bounds of the type-reduced interval change as input changes; and 2) *Novelty*, meaning that the upper and lower membership functions of the same IT2 fuzzy set may be used simultaneously in computing each bound of the type-reduced interval. T1 FLCs do not have these properties; thus, a T1 FLC cannot implement the complex control surface of an IT2 FLC given the same rulebase. We also present several methods to visualize and analyze the effects of these two fundamental differences, including the control surface, the P-map, the equivalent generalized T1 fuzzy sets, and the equivalent PI gains. Finally, we examine five alternative type reducers for IT2 FLCs and explain why they do not capture the fundamentals of IT2 FLCs.

Index Terms—Continuity, equivalent generalized type-1 fuzzy sets (EGT1FSs), equivalent PI gains, interval type-2 fuzzy logic controller (IT2 FLC), P-map.

I. INTRODUCTION

INTERVAL type-2 fuzzy logic controllers (IT2 FLCs) have recently been attracting a lot of research attention. Many reported results have shown that IT2 FLCs are better able to handle uncertainties than their type-1 (T1) counterparts [8], [21], [31], [60], [61]. For example, Hagrass [20] implemented a hierarchical IT2 FLC for different types of mobile robots navigating in indoor and outdoor environments. It outperformed a T1 FLC and had significantly fewer rules. Wu and Tan [57], [60], [61] showed through both simulation and experiments that IT2 FLCs are better able to cope with modeling uncertainties, and hence, IT2 FLCs optimized from simulations are more likely to perform well on the actual plant than T1 FLCs.

A challenging question is the following: *What are the fundamental differences between IT2 and T1 FLCs?* Once the fundamental differences are clear, we can better understand the

advantages of IT2 FLCs and, hence, make better use of them. In the literature, there has been considerable effort on answering this challenging and fundamental question. Some important arguments are the following.

- 1) *An IT2 fuzzy set (FS) can better model intrapersonal¹ and interpersonal² uncertainties*, which are intrinsic to natural language, because the membership grade of an IT2 FS is an interval instead of a crisp number in a T1 FS. Mendel [33] also showed that IT2 FS is a scientifically correct model for modeling linguistic uncertainties, whereas T1 FS is not.
- 2) *Using IT2 FSs to represent the FLC inputs and outputs will result in the reduction of the rulebase when compared with using T1 FSs* [21], [31], as the ability of the footprint of uncertainty (FOU) to represent more uncertainties enables one to cover the input/output domains with fewer FSs. This makes it easier to construct the rulebase using expert knowledge, as well as increase robustness [57], [60], [61].
- 3) *An IT2 FLC can give a smoother control surface than its T1 counterpart, especially in the region around the steady state* (for a proportional-integral (PI) controller, this means both the error and the change of error approach 0) [23], [57], [60], [61]. Wu and Tan [62] (see also Section VII of this paper) showed that when the baseline T1 FLC implements a linear PI control law and the IT2 FSs of an IT2 FLC are obtained from symmetrical perturbations of the T1 FSs, the resulting IT2 FLC implements a variable gain PI controller around the steady state. These gains are smaller than the PI gains of the baseline T1 FLC, which result in a smoother control surface around the steady state. The PI gains of the IT2 FLC also change with the inputs, which cannot be achieved by the baseline T1 FLC.
- 4) *IT2 FLCs are more adaptive and they can realize more complex input–output relationships which cannot be achieved by T1 FLCs*. Karnik and Mendel [26] pointed out that an IT2 fuzzy logic system can be thought of as a collection of many different embedded T1 fuzzy logic systems. Wu and Tan [59] (see also Section VI of this paper) proposed a systematic method to identify the *equivalent*

¹According to Mendel [33], intrapersonal uncertainty describes “*the uncertainty a person has about the word.*” It is also explicitly pointed out by psychologists Wallsten and Budescu [46] as “*except in very special cases, all representations are vague to some degree in the minds of the originators and in the minds of the receivers,*” and they suggest to model it by a T1 FS.

²According to Mendel [33], interpersonal uncertainty describes “*the uncertainty that a group of people have about the word,*” i.e., “*words mean different things to different people.*” It is also explicitly pointed out by psychologists Wallsten and Budescu [46] as “*different individuals use diverse expressions to describe identical situations and understand the same phrases differently when hearing or reading them.*”

Manuscript received June 17, 2011; revised September 28, 2011 and December 23, 2011; accepted January 5, 2012. Date of publication February 3, 2012; date of current version October 2, 2012.

The author is with the Machine Learning Laboratory, GE Global Research, Niskayuna, NY 12309 USA (e-mail: wud@ge.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TFUZZ.2012.2186818

generalized T1 FSs (EGT1FSs) that can be used to replace the FOU. They showed that the EGT1FSs are significantly different from traditional T1 FSs, and there are different EGT1FSs for different inputs. Du and Ying [12], and Nie and Tan [38], also showed that a symmetrical IT2 fuzzy-PI (or the corresponding PD) controller, which is obtained from a baseline T1 PI FLC, partitions the input domain into many small regions, and in each region it is equivalent to a nonlinear PI controller with variable gains. The control law of the IT2 FLC in each small region is much more complex than that of the baseline T1 FLC, and, hence, it can realize more complex input–output relationship that cannot be achieved by a T1 FLC using the same rulebase.

- 5) *IT2 FLCs have a novelty that does not exist in traditional T1 FLCs.* Wu [49] (see also Section III of this paper) showed that, in an IT2 FLC, *different* membership grades from the same IT2 FS can be used in different rules, whereas, for traditional T1 FLC, the *same* membership grade from the same T1 FS is always used in different rules. This again implies that an IT2 FLC is more complex than a T1 FLC and it cannot be implemented by a T1 FLC using the same rulebase.

This paper summarizes some recent research results on understanding the fundamental differences between IT2 and T1 FLCs. It explains why *adaptiveness* and *novelty* are two fundamental differences and proposes several methods to visualize and analyze the effects of these two differences.

The rest of this paper is organized as follows: Section II introduces background materials on IT2 FSs and FLCs, and shows two numerical examples on IT2 FLCs. Section III explains adaptiveness and novelty, two fundamental differences between IT2 and T1 FLCs. Sections IV–VI introduce the control surface, the P-map, and the EGT1FSs, respectively, which are three methods to visualize the effect of the two fundamental differences. Section VII analyzes the effect of the two fundamental differences by deriving the equivalent PI gains for a special IT2 FLC near the steady state. Section VIII examines five alternative type-reduction strategies against adaptiveness and novelty, and explains why they do not capture the fundamentals of IT2 FLCs. Finally, Section IX draws conclusions.

II. INTERVAL TYPE-2 FUZZY SETS AND CONTROLLERS

A. Interval Type-2 Fuzzy Sets

T1 FS theory was first introduced by Zadeh [66] in 1965 and has been successfully applied in many areas, including modeling and control [6], [47], [65], data mining [22], [40], [69], time-series prediction [27], [29], [45], linguistic summarization [24], [54], [64], computing with words [41], [42], [68], etc.

Definition 1: A T1 FS X is comprised of a domain D_X of real numbers (also called the *universe of discourse* of X) together with a *membership function* (MF) $\mu_X : D_X \rightarrow [0, 1]$, i.e.,

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

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

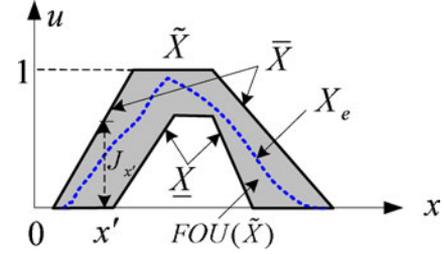


Fig. 1. IT2 FS. \underline{X} (the LMF), \bar{X} (the UMF), and X_e are three embedded T1 FSs.

Despite having a name that carries the connotation of uncertainty, research has shown that there are limitations in the ability of T1 FSs to model and minimize the effect of uncertainties [20], [21], [31], [60]. This is because a T1 FS is certain in the sense that its membership grades are crisp values. Recently, type-2 FSs [67], which are characterized by MFs that are themselves fuzzy, have been attracting much interest. IT2 FSs³ [31], a special case of type-2 FSs, are currently the most widely used for their reduced computational cost, and are also the focus of this paper.

Definition 2 [31], [34]: An IT2 FS \tilde{X} is characterized by its MF $\mu_{\tilde{X}}(x, u)$, i.e.

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

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

The connection between IT2 FSs notations and mathematical notation and terminology of functions on space has been established in [1].

An example of an IT2 FS, \tilde{X} , is shown in Fig. 1. Observe that unlike a T1 FS, whose membership grade for each x is a number, the membership of an IT2 FS is an interval. Observe also that an IT2 FS is bounded from above and below by two T1 FSs \bar{X} and \underline{X} , which are called *upper MF* (UMF) and *lower MF* (LMF), respectively. The area between \bar{X} and \underline{X} is the FOU. An *embedded* T1 FS⁴ is any T1 FS within the FOU. \underline{X} and \bar{X} are two such sets.

³IT2 FSs have also been called *interval-valued* FSs in the literature [7], [15], [44]. They can also be mapped into *intuitionistic* FSs [4]. Deschrijver and Kerre [11] had a comprehensive study on the relationships among some important extensions of T1 FSs, including interval-valued FSs, intuitionistic FSs, interval-valued intuitionistic FSs [3], and *L*-FSs [14].

⁴According to the Mendel–John representation theorem [34], an embedded T1 FS can be subnormal and nonconvex. Recently, there have been arguments that only convex and normal T1 FSs [48], or only T1 FSs assuming a particular shape [2], [13], should be considered as embedded T1 FSs.

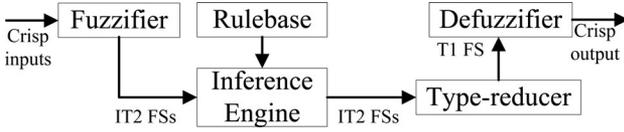


Fig. 2. Schematic diagram of an IT2 FLC.

B. Interval Type-2 Fuzzy Logic Controllers

Fig. 2 shows the schematic diagram of an IT2 FLC. It is similar to its T1 counterpart; the major difference being that at least one of the FSs in the rulebase is an IT2 FS. Hence, the outputs of the inference engine are IT2 FSs, and a type reducer [26], [31] is needed to convert them into a T1 FS before defuzzification can be carried out.

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

$$\tilde{R}^n: \text{IF } x_1 \text{ is } \tilde{X}_1^n \text{ and } \dots \text{ and } x_I \text{ is } \tilde{X}_I^n, \text{ THEN } y \text{ is } Y^n$$

where \tilde{X}_i^n ($i = 1, \dots, I; n = 1, 2, \dots, N$) are IT2 FSs, and $Y^n = [\underline{y}^n, \bar{y}^n]$ is an interval, which can be understood as the centroid [25], [31] of a consequent IT2 FS,⁵ or the simplest Takagi–Sugeno–Kang (TSK) model. In many applications [57], [60], [61], we use $\underline{y}^n = \bar{y}^n$, i.e., each rule consequent is represented by a crisp number.

For an input vector $\mathbf{x}' = (x'_1, x'_2, \dots, x'_I)$, typical computations in an IT2 FLC involve the following steps:

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

$$\begin{aligned} F^n(\mathbf{x}') &= [\mu_{\underline{X}_1^n}(x'_1) \times \dots \times \mu_{\underline{X}_I^n}(x'_I), \\ &\quad \mu_{\bar{X}_1^n}(x'_1) \times \dots \times \mu_{\bar{X}_I^n}(x'_I)] \\ &\equiv [\underline{f}^n, \bar{f}^n], \quad n = 1, \dots, N. \end{aligned} \quad (1)$$

Note that the *minimum* t -norm may also be used in (1). However, this paper focuses only on the *product* t -norm.

- 3) Perform type reduction to combine $F^n(\mathbf{x}')$ and the corresponding rule consequents. There are many such methods [31], [51], [52]. The most commonly used one is the center-of-sets type reducer [31], which is derived from the extension principle [66]

$$Y_{\text{cos}}(\mathbf{x}') = \bigcup_{\substack{f^n \in F^n(\mathbf{x}') \\ y^n \in Y^n}} \frac{\sum_{n=1}^N f^n y^n}{\sum_{n=1}^N f^n} = [y_l, y_r]. \quad (2)$$

It has been shown that [31], [35], [53]

$$y_l = \min_{k \in [1, N-1]} \frac{\sum_{n=1}^k \bar{f}^n \underline{y}^n + \sum_{n=k+1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^k \bar{f}^n + \sum_{n=k+1}^N \underline{f}^n}$$

⁵The rule consequents can be IT2 FSs; however, when the popular center-of-sets type-reduction method [31] is used, these consequent IT2 FSs are replaced by their centroids in the computation; therefore, it is more convenient to represent the rule consequents as intervals directly.

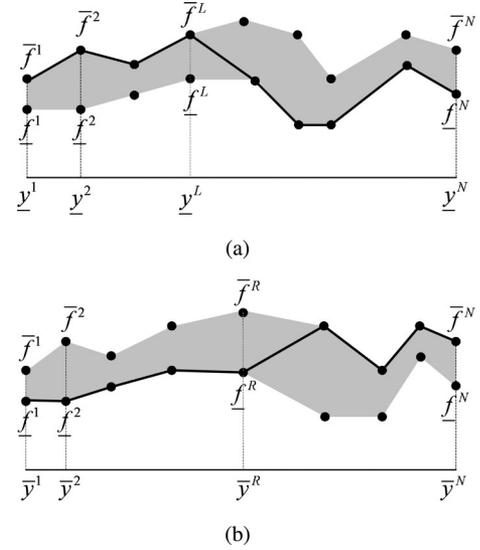


Fig. 3. Switch points in computing y_l and y_r . (a) Computing y_l : Switch from the upper bounds of the firing intervals to the lower bounds. (b) Computing y_r : Switch from the lower bounds of the firing intervals to the upper bounds.

$$\equiv \frac{\sum_{n=1}^L \bar{f}^n \underline{y}^n + \sum_{n=L+1}^N \underline{f}^n \underline{y}^n}{\sum_{n=1}^L \bar{f}^n + \sum_{n=L+1}^N \underline{f}^n} \quad (3)$$

$$\begin{aligned} y_r &= \max_{k \in [1, N-1]} \frac{\sum_{n=1}^k \underline{f}^n \bar{y}^n + \sum_{n=k+1}^N \bar{f}^n \bar{y}^n}{\sum_{n=1}^k \underline{f}^n + \sum_{n=k+1}^N \bar{f}^n} \\ &\equiv \frac{\sum_{n=1}^R \underline{f}^n \bar{y}^n + \sum_{n=R+1}^N \bar{f}^n \bar{y}^n}{\sum_{n=1}^R \underline{f}^n + \sum_{n=R+1}^N \bar{f}^n} \end{aligned} \quad (4)$$

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

$$\begin{aligned} \underline{y}^L &\leq y_l \leq \underline{y}^{L+1} \\ \bar{y}^R &\leq y_r \leq \bar{y}^{R+1} \end{aligned}$$

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

y_l and y_r can be computed by the Karnik–Mendel (KM) algorithms [26], [31] or their many variants [53], [56]. The main idea of the KM algorithms is to find the switch points for y_l and y_r . Take y_l as an example. y_l is the minimum of $Y_{\text{cos}}(\mathbf{x}')$. Since \underline{y}^n increases from the left to the right along the horizontal axis of Fig. 3(a), we should choose a large weight (upper bound of the firing interval) for \underline{y}^n on the left and a small weight (lower bound of the firing interval) for \underline{y}^n on the right. The KM algorithm for y_l finds the switch point L . For $n \leq L$, the upper bounds of the firing intervals are used to calculate y_l ; for $n > L$, the lower bounds are used. This ensures that y_l is the minimum.

- 4) Compute the defuzzified output as

$$y = \frac{y_l + y_r}{2}. \quad (5)$$

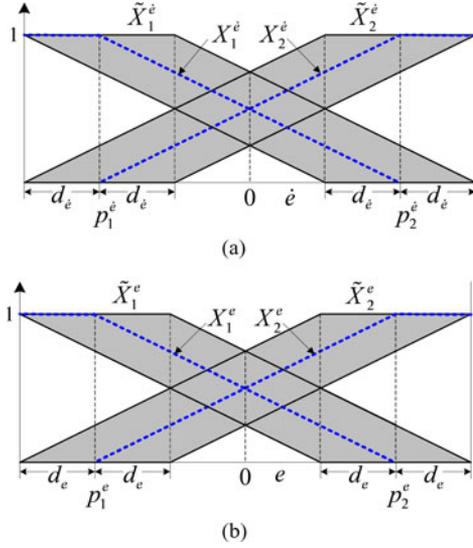


Fig. 4. MFs of FLC_1 and \widetilde{FLC}_1 . (a) MFs in the \dot{e} domain. (b) MFs in the e domain.

C. Two Pairs of Type-1 and Interval Type-2 Fuzzy Logic Controllers

A linear PI control law is usually implemented as

$$\dot{u} = K_P \cdot \dot{e} + K_I \cdot e \quad (6)$$

where \dot{u} is the *change of the control signal*, e is the *feedback error*, \dot{e} is the *change of error*, and K_P and K_I are the *proportional and integral gains*, respectively. A T1 FLC with rulebase

If \dot{e} is $X_i^{\dot{e}}$ and e is X_j^e , then \dot{u} is y_{ij} .

where $i = 1, \dots, N$ and $j = 1, \dots, M$ implements the linear PI controller in (6) if [36]:

- 1) triangular T1 FSs are used for input MFs $X_i^{\dot{e}}$ and X_j^e , and they are constructed in such a way that, for any input, the firing levels of all MFs add to 1; and,
- 2) the consequents of the rules are crisp numbers defined as

$$y_{ij} = K_P \cdot p_i^{\dot{e}} + K_I \cdot p_j^e \quad (7)$$

where $p_i^{\dot{e}}$ and p_j^e are located at the apexes of the antecedent triangular T1 FSs.

The following two pairs of T1 and IT2 PI FLCs are used throughout this paper.

- 1) FLC_1 and \widetilde{FLC}_1 : The first pair of T1 and IT2 PI FLCs, FLC_1 and \widetilde{FLC}_1 , are shown in Fig. 4, where the T1 FLC, FLC_1 , is shown as the bold dashed lines. Its four rules are

- $$\begin{aligned} R^1 &: \text{ IF } \dot{e} \text{ is } X_1^{\dot{e}} \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y_{11} \\ R^2 &: \text{ IF } \dot{e} \text{ is } X_1^{\dot{e}} \text{ and } e \text{ is } X_2^e, \text{ THEN } \dot{u} \text{ is } y_{12} \\ R^3 &: \text{ IF } \dot{e} \text{ is } X_2^{\dot{e}} \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y_{21} \\ R^4 &: \text{ IF } \dot{e} \text{ is } X_2^{\dot{e}} \text{ and } e \text{ is } X_2^e, \text{ THEN } \dot{u} \text{ is } y_{22} \end{aligned}$$

where y_{11} , y_{12} , y_{21} , and y_{22} are defined in (7). When $\dot{e} \in [p_1^{\dot{e}}, p_2^{\dot{e}}]$ and $e \in [p_1^e, p_2^e]$, FLC_1 implements the linear PI control law in (6).

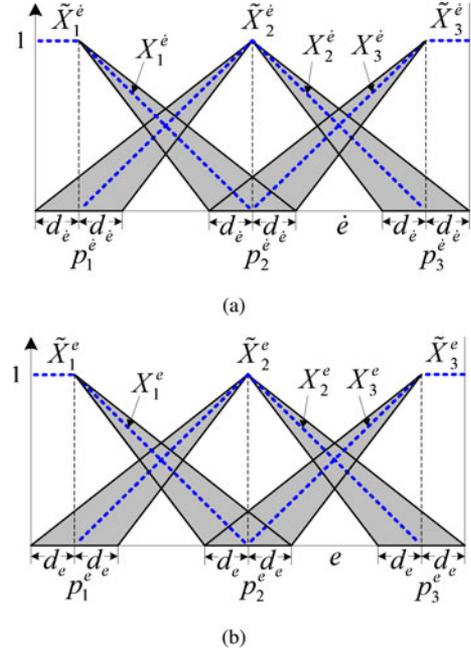


Fig. 5. MFs of FLC_2 and \widetilde{FLC}_2 . (a) MFs in the \dot{e} domain. (b) MFs in the e domain.

An IT2 fuzzy PI controller, \widetilde{FLC}_1 , is constructed by blurring the T1 FSs of FLC_1 to IT2 FSs,⁶ as shown in Fig. 4. For simplicity, symmetrical FOUs⁷ are used in this paper, and their sizes are determined by d_e and d_e in Fig. 4. The rulebase of \widetilde{FLC}_1 is

- $$\begin{aligned} \widetilde{R}^1 &: \text{ IF } \dot{e} \text{ is } \widetilde{X}_1^{\dot{e}} \text{ and } e \text{ is } \widetilde{X}_1^e, \text{ THEN } \dot{u} \text{ is } y_{11} \\ \widetilde{R}^2 &: \text{ IF } \dot{e} \text{ is } \widetilde{X}_1^{\dot{e}} \text{ and } e \text{ is } \widetilde{X}_2^e, \text{ THEN } \dot{u} \text{ is } y_{12} \\ \widetilde{R}^3 &: \text{ IF } \dot{e} \text{ is } \widetilde{X}_2^{\dot{e}} \text{ and } e \text{ is } \widetilde{X}_1^e, \text{ THEN } \dot{u} \text{ is } y_{21} \\ \widetilde{R}^4 &: \text{ IF } \dot{e} \text{ is } \widetilde{X}_2^{\dot{e}} \text{ and } e \text{ is } \widetilde{X}_2^e, \text{ THEN } \dot{u} \text{ is } y_{22} \end{aligned}$$

where $\widetilde{X}_i^{\dot{e}}$ and \widetilde{X}_j^e are IT2 FSs obtained by blurring $X_i^{\dot{e}}$ and X_j^e , respectively, and y_{ij} are the same as those in FLC_1 .

- 2) FLC_2 and \widetilde{FLC}_2 : The second pair of PI FLCs are shown in Fig. 5, where the T1 FLC, FLC_2 , is shown as the bold dashed lines. Its nine rules are

- $$\begin{aligned} R^1 &: \text{ IF } \dot{e} \text{ is } X_1^{\dot{e}} \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y_{11} \\ R^2 &: \text{ IF } \dot{e} \text{ is } X_1^{\dot{e}} \text{ and } e \text{ is } X_2^e, \text{ THEN } \dot{u} \text{ is } y_{12} \\ R^3 &: \text{ IF } \dot{e} \text{ is } X_1^{\dot{e}} \text{ and } e \text{ is } X_3^e, \text{ THEN } \dot{u} \text{ is } y_{13} \\ R^4 &: \text{ IF } \dot{e} \text{ is } X_2^{\dot{e}} \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y_{21} \\ &\vdots \\ R^9 &: \text{ IF } \dot{e} \text{ is } X_3^{\dot{e}} \text{ and } e \text{ is } X_3^e, \text{ THEN } \dot{u} \text{ is } y_{33} \end{aligned}$$

⁶An IT2 FLC can also be constructed from scratch without using a baseline T1 FLC [60], [61]. This paper uses a baseline T1 FLC for comparison purposes.

⁷Except for the results in Section VII, which are specific to a very special IT2 FLC with symmetrical FOU, the methods, observations, and conclusions presented in this paper also hold for IT2 FLCs with arbitrary FOU.

TABLE I
RULE CONSEQUENTS OF FLC_1 AND \widetilde{FLC}_1

e	X_1^e (\widetilde{X}_1^e)	X_2^e (\widetilde{X}_2^e)
X_1^e (\widetilde{X}_1^e)	$y^1 \equiv y_{11} = K_P \cdot p_1^e + K_I \cdot p_1^e = -2.2923$	$y^2 \equiv y_{12} = K_P \cdot p_1^e + K_I \cdot p_2^e = -1.8797$
X_2^e (\widetilde{X}_2^e)	$y^3 \equiv y_{21} = K_P \cdot p_2^e + K_I \cdot p_1^e = 1.8797$	$y^4 \equiv y_{22} = K_P \cdot p_2^e + K_I \cdot p_2^e = 2.2923$

where $y_{11} - y_{33}$ are defined in (7). When $e \in [p_1^e, p_3^e]$ and $e \in [p_1^e, p_3^e]$, FLC_2 implements the linear PI control law in (6).

An IT2 fuzzy PI controller, \widetilde{FLC}_2 , is constructed by blurring the T1 FSs of FLC_2 to IT2 FSs, as shown in Fig. 5. Note that, again, symmetrical FOUs are used; however, the shape of the FOUs is different from those in \widetilde{FLC}_1 . The rulebase of \widetilde{FLC}_2 is

- \widetilde{R}^1 : IF \dot{e} is $\widetilde{X}_1^{\dot{e}}$ and e is \widetilde{X}_1^e , THEN \dot{u} is y_{11}
- \widetilde{R}^2 : IF \dot{e} is $\widetilde{X}_1^{\dot{e}}$ and e is \widetilde{X}_2^e , THEN \dot{u} is y_{12}
- \widetilde{R}^3 : IF \dot{e} is $\widetilde{X}_2^{\dot{e}}$ and e is \widetilde{X}_1^e , THEN \dot{u} is y_{13}
- \widetilde{R}^4 : IF \dot{e} is $\widetilde{X}_2^{\dot{e}}$ and e is \widetilde{X}_2^e , THEN \dot{u} is y_{21}
- \vdots
- \widetilde{R}^9 : IF \dot{e} is $\widetilde{X}_3^{\dot{e}}$ and e is \widetilde{X}_3^e , THEN \dot{u} is y_{33}

where $\widetilde{X}_i^{\dot{e}}$ and \widetilde{X}_j^e are IT2 FSs obtained by blurring $X_i^{\dot{e}}$ and X_j^e , respectively, and y_{ij} are the same as those in FLC_2 .

D. Examples

In this section, the mathematical operations in an IT2 FLC, which are introduced in Section II-B, are illustrated using two numerical examples, which will be revisited in Section III.

For simplicity, FLC_1 and \widetilde{FLC}_1 , in Fig. 4, are used. Additionally, we use $p_1^e = p_1^e = -1$, $p_2^e = p_2^e = 1$, $d_e = d_e = 0.5$, $K_P = 2.086$, and $K_I = 0.2063$, which are the same as those used in [62]. The corresponding rule consequents are given in Table I.

Example 1: Consider an input vector $\mathbf{x}' = (e', \dot{e}') = (-0.3, -0.6)$, as shown in Fig. 6. The firing levels of the four T1 FSs of FLC_1 are

$$\begin{aligned} \mu_{X_1^e}(e') &= 0.65, & \mu_{X_2^e}(e') &= 0.35 \\ \mu_{X_1^{\dot{e}}}(e') &= 0.8, & \mu_{X_2^{\dot{e}}}(e') &= 0.2. \end{aligned}$$

The firing levels of its four rules are shown in Table II. The output of FLC_1 is

$$\dot{u} = \frac{f^1 y^1 + f^2 y^2 + f^3 y^3 + f^4 y^4}{f^1 + f^2 + f^3 + f^4} = -1.3135.$$

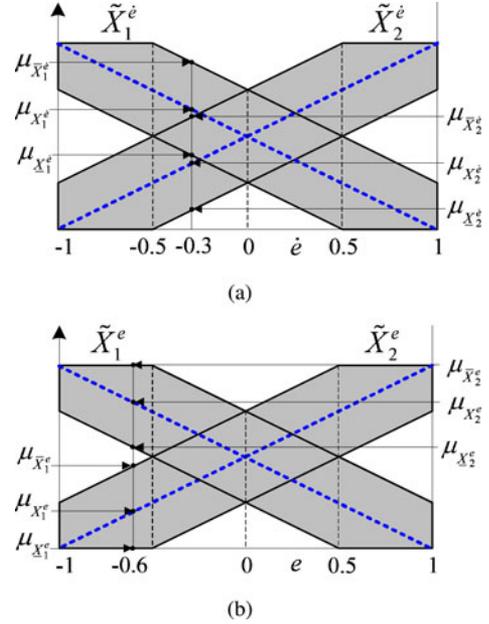


Fig. 6. Firing levels of FLC_1 , and firing intervals of \widetilde{FLC}_1 , when $\mathbf{x}' = (e', \dot{e}') = (-0.3, -0.6)$.

TABLE II
FIRING LEVELS OF THE FOUR RULES OF FLC_1 IN EXAMPLE 1

Rule No.:	Firing Level	→	Rule Consequent
R^1 :	$f^1 = \mu_{X_1^e}(e') \cdot \mu_{X_1^{\dot{e}}}(e') = 0.65 \times 0.8 = 0.52$	→	$y^1 = -2.2923$
R^2 :	$f^2 = \mu_{X_1^e}(e') \cdot \mu_{X_2^{\dot{e}}}(e') = 0.65 \times 0.2 = 0.13$	→	$y^2 = -1.8797$
R^3 :	$f^3 = \mu_{X_2^e}(e') \cdot \mu_{X_1^{\dot{e}}}(e') = 0.35 \times 0.8 = 0.28$	→	$y^3 = 1.8797$
R^4 :	$f^4 = \mu_{X_2^e}(e') \cdot \mu_{X_2^{\dot{e}}}(e') = 0.35 \times 0.2 = 0.07$	→	$y^4 = 2.2923$

For \widetilde{FLC}_1 , the firing intervals of the four IT2 FSs are

$$\begin{aligned} [\mu_{X_1^e}(e'), \mu_{\overline{X}_1^e}(e')] &= [0.4, 0.9] \\ [\mu_{X_2^e}(e'), \mu_{\overline{X}_2^e}(e')] &= [0.1, 0.6] \\ [\mu_{X_1^{\dot{e}}}(e'), \mu_{\overline{X}_1^{\dot{e}}}(e')] &= [0.55, 1] \\ [\mu_{X_2^{\dot{e}}}(e'), \mu_{\overline{X}_2^{\dot{e}}}(e')] &= [0, 0.45]. \end{aligned}$$

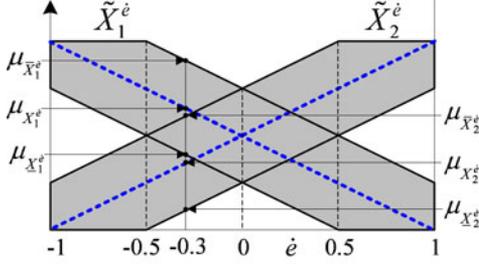
The firing intervals of the four rules are shown in Table III. From the KM algorithms, we find that $L = 1$ and $R = 2$. Therefore, y_l and y_r are computed by (8) and (9), shown at the bottom of the page.

$$y_l = \frac{\overline{f}^1 y^1 + \overline{f}^2 y^2 + \overline{f}^3 y^3 + \overline{f}^4 y^4}{\overline{f}^1 + \overline{f}^2 + \overline{f}^3 + \overline{f}^4} = \frac{0.9 \times (-2.2923) + 0 \times (-1.8797) + 0.055 \times 1.8797 + 0 \times 2.2923}{0.9 + 0 + 0.055 + 0} = -2.2685 \quad (8)$$

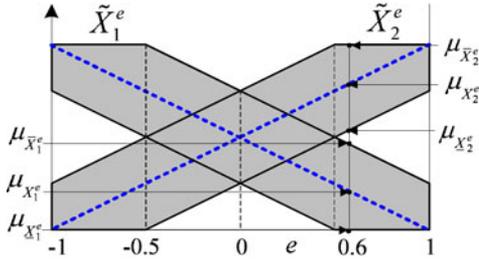
$$y_r = \frac{\underline{f}^1 y^1 + \underline{f}^2 y^2 + \underline{f}^3 y^3 + \underline{f}^4 y^4}{\underline{f}^1 + \underline{f}^2 + \underline{f}^3 + \underline{f}^4} = \frac{0.22 \times (-2.2923) + 0 \times (-1.8797) + 0.6 \times 1.8797 + 0.27 \times 2.2923}{0.22 + 0 + 0.6 + 0.27} = 0.8132 \quad (9)$$

TABLE III
FIRING INTERVALS OF THE FOUR RULES OF \widetilde{FLC}_1 IN EXAMPLE 1

Rule No.:	Firing Interval	→	Rule Consequent
\widetilde{R}^1 :	$[\underline{f}^1, \overline{f}^1] = [\mu_{\widetilde{X}_1^e}(e') \cdot \mu_{\widetilde{X}_1^e}(e'), \mu_{\widetilde{X}_1^e}(e') \cdot \mu_{\widetilde{X}_1^e}(e')] = [0.4 \times 0.55, 0.9 \times 1] = [0.22, 0.9]$	→	$y^1 = -2.2923$
\widetilde{R}^2 :	$[\underline{f}^2, \overline{f}^2] = [\mu_{\widetilde{X}_1^e}(e') \cdot \mu_{\widetilde{X}_2^e}(e'), \mu_{\widetilde{X}_1^e}(e') \cdot \mu_{\widetilde{X}_2^e}(e')] = [0.4 \times 0, 0.9 \times 0.45] = [0, 0.405]$	→	$y^2 = -1.8797$
\widetilde{R}^3 :	$[\underline{f}^3, \overline{f}^3] = [\mu_{\widetilde{X}_2^e}(e') \cdot \mu_{\widetilde{X}_1^e}(e'), \mu_{\widetilde{X}_2^e}(e') \cdot \mu_{\widetilde{X}_1^e}(e')] = [0.1 \times 0.55, 0.6 \times 1] = [0.055, 0.6]$	→	$y^3 = 1.8797$
\widetilde{R}^4 :	$[\underline{f}^4, \overline{f}^4] = [\mu_{\widetilde{X}_2^e}(e') \cdot \mu_{\widetilde{X}_2^e}(e'), \mu_{\widetilde{X}_2^e}(e') \cdot \mu_{\widetilde{X}_2^e}(e')] = [0.1 \times 0, 0.6 \times 0.45] = [0, 0.27]$	→	$y^4 = 2.2923$



(a)



(b)

Fig. 7. Firing levels of FLC_1 , and firing intervals of \widetilde{FLC}_1 , when $\mathbf{x}' = (e', e') = (-0.3, 0.6)$.

Finally, the crisp output of \widetilde{FLC}_1 is

$$\dot{u} = \frac{y_l + y_r}{2} = \frac{-2.2685 + 0.8132}{2} = -0.7277.$$

Example 2: Consider another input vector $\mathbf{x}' = (e', e') = (-0.3, 0.6)$, as shown in Fig. 7. The firing levels of the four T1 FSs of FLC_1 are

$$\begin{aligned} \mu_{\widetilde{X}_1^e}(e') &= 0.65, & \mu_{\widetilde{X}_2^e}(e') &= 0.35 \\ \mu_{\widetilde{X}_1^e}(e) &= 0.2, & \mu_{\widetilde{X}_2^e}(e) &= 0.8. \end{aligned}$$

The firing levels of its four rules are shown in Table IV. The output of FLC_1 is

$$\dot{u} = \frac{f^1 y^1 + f^2 y^2 + f^3 y^3 + f^4 y^4}{f^1 + f^2 + f^3 + f^4} = 1.1897.$$

For \widetilde{FLC}_1 , the firing intervals of the four IT2 FSs are

$$[\mu_{\widetilde{X}_1^e}(e'), \mu_{\widetilde{X}_1^e}(e')] = [0.4, 0.9]$$

$$[\mu_{\widetilde{X}_2^e}(e'), \mu_{\widetilde{X}_2^e}(e')] = [0.1, 0.6]$$

$$[\mu_{\widetilde{X}_1^e}(e), \mu_{\widetilde{X}_1^e}(e)] = [0, 0.45]$$

$$[\mu_{\widetilde{X}_2^e}(e), \mu_{\widetilde{X}_2^e}(e)] = [0.55, 1].$$

The firing intervals of the four rules are shown in Table V. From the KM algorithms, we find that $L = 2$ and $R = 3$. Therefore, y_l and y_r are computed by (10) and (11), shown at the bottom of the page.

Finally, the crisp output of \widetilde{FLC}_1 is

$$\dot{u} = \frac{y_l + y_r}{2} = \frac{-0.9435 + 2.1816}{2} = 0.6191.$$

Observe from the aforementioned two examples that for the same input, IT2 and T1 FLCs give quite different outputs. The next section explains the fundamental reasons behind this difference.

III. FUNDAMENTAL DIFFERENCES BETWEEN INTERVAL TYPE-2 AND TYPE-I FUZZY LOGIC CONTROLLERS

Observe from (5), as well as Examples 1 and 2, that the output of an IT2 FLC is the average of two ‘‘T1 FLCs.’’ However, these two ‘‘T1 FLCs’’ are *fundamentally* different from traditional T1 FLCs, for the following reasons [49].

- 1) *Adaptiveness*: meaning that the embedded T1 FSs that are used to compute the bounds of the type-reduced interval change as input changes. Take y_l in (8) and (10) as an example. The firing levels of the four rules in (8) are \overline{f}^1 , \underline{f}^2 , \underline{f}^3 , and \underline{f}^4 , respectively, which are computed from different lower and upper MFs, as shown in the first part in Table VI and Fig. 8(a). The firing levels of the four rules in (10) are shown in the second part in Table VI and Fig. 8(b). Comparing the two parts in Table VI, and the two subfigures in Fig. 8, we can observe that when the input (e', e') changes from $(-0.3, -0.6)$ to $(-0.3, 0.6)$,

$$y_l = \frac{\overline{f}^1 y^1 + \overline{f}^2 y^2 + \underline{f}^3 y^3 + \underline{f}^4 y^4}{\overline{f}^1 + \overline{f}^2 + \underline{f}^3 + \underline{f}^4} = \frac{0.405 \times (-2.2923) + 0.9 \times (-1.8797) + 0 \times 1.8797 + 0.055 \times 2.2923}{0.405 + 0.9 + 0 + 0.055} = -0.9435 \quad (10)$$

$$y_r = \frac{\underline{f}^1 y^1 + \underline{f}^2 y^2 + \underline{f}^3 y^3 + \overline{f}^4 y^4}{\underline{f}^1 + \underline{f}^2 + \underline{f}^3 + \overline{f}^4} = \frac{0 \times (-2.2923) + 0.22 \times (-1.8797) + 0 \times 1.8797 + 0.6 \times 2.2923}{0 + 0.22 + 0 + 0.6} = 2.1816 \quad (11)$$

TABLE IV
FIRING LEVELS OF THE FOUR RULES OF FLC_1 IN EXAMPLE 2

Rule No.:	Firing Level	→	Rule Consequent
R^1 :	$f^1 = \mu_{X_1^e}(e') \cdot \mu_{X_1^e}(e') = 0.65 \times 0.2 = 0.13$	→	$y^1 = -2.2923$
R^2 :	$f^2 = \mu_{X_1^e}(e') \cdot \mu_{X_2^e}(e') = 0.65 \times 0.8 = 0.52$	→	$y^2 = -1.8797$
R^3 :	$f^3 = \mu_{X_2^e}(e') \cdot \mu_{X_1^e}(e') = 0.35 \times 0.2 = 0.07$	→	$y^3 = 1.8797$
R^4 :	$f^4 = \mu_{X_2^e}(e') \cdot \mu_{X_2^e}(e') = 0.35 \times 0.8 = 0.28$	→	$y^4 = 2.2923$

TABLE V
FIRING INTERVALS OF THE FOUR RULES OF \widetilde{FLC}_1 IN EXAMPLE 2

Rule No.:	Firing Interval	→	Rule Consequent
\widetilde{R}^1 :	$[\underline{f}^1, \overline{f}^1] = [\mu_{X_1^e}(e') \cdot \mu_{X_1^e}(e'), \mu_{X_1^e}(e') \cdot \mu_{X_1^e}(e')] = [0.4 \times 0, 0.9 \times 0.45] = [0, 0.405]$	→	$y^1 = -2.2923$
\widetilde{R}^2 :	$[\underline{f}^2, \overline{f}^2] = [\mu_{X_1^e}(e') \cdot \mu_{X_2^e}(e'), \mu_{X_1^e}(e') \cdot \mu_{X_2^e}(e')] = [0.4 \times 0.55, 0.9 \times 1] = [0.22, 0.9]$	→	$y^2 = -1.8797$
\widetilde{R}^3 :	$[\underline{f}^3, \overline{f}^3] = [\mu_{X_2^e}(e') \cdot \mu_{X_1^e}(e'), \mu_{X_2^e}(e') \cdot \mu_{X_1^e}(e')] = [0.1 \times 0, 0.6 \times 0.45] = [0, 0.27]$	→	$y^3 = 1.8797$
\widetilde{R}^4 :	$[\underline{f}^4, \overline{f}^4] = [\mu_{X_2^e}(e') \cdot \mu_{X_2^e}(e'), \mu_{X_2^e}(e') \cdot \mu_{X_2^e}(e')] = [0.1 \times 0.55, 0.6 \times 1] = [0.055, 0.6]$	→	$y^4 = 2.2923$

TABLE VI
EMBEDDED T1 FSS FROM WHICH THE FOUR FIRING LEVELS IN (8) AND (10) ARE OBTAINED

		\widetilde{X}_1^e		\widetilde{X}_2^e		\widetilde{X}_1^e		\widetilde{X}_2^e	
		UMF	LMF	UMF	LMF	UMF	LMF	UMF	LMF
Equation (8) $(e', e') = (-0.3, -0.6)$	\overline{f}^1	✓				✓			
	\underline{f}^2		✓						✓
	\underline{f}^3				✓		✓		
	\overline{f}^4				✓				✓
Equation (10) $(e', e') = (-0.3, 0.6)$	\overline{f}^1	✓				✓			
	\underline{f}^2	✓						✓	
	\underline{f}^3				✓		✓		
	\overline{f}^4				✓				✓

Observe that \underline{f}^2 is used when $(e', e') = (-0.3, -0.6)$ and \overline{f}^2 is used when $(e', e') = (-0.3, 0.6)$; as a result, different embedded T1 FSS are used in Rule \widetilde{R}^2 when the input changes. Observe also that, when $(e', e') = (-0.3, -0.6)$, both the UMFs and LMFs of \widetilde{X}_1^e and \widetilde{X}_2^e are used in computing y_l , and when $(e', e') = (-0.3, 0.6)$ both the UMFs and LMFs of \widetilde{X}_1^e and \widetilde{X}_2^e are used in computing y_l .

different embedded T1 FSS of \widetilde{X}_1^e and \widetilde{X}_2^e are used in computing the firing levels for Rule \widetilde{R}^2 and, hence, y_l . This adaptiveness is impossible for a T1 FLC since it does not have such embedded T1 FSS.

- 2) *Novelty*: meaning that the UMF and LMF of the same IT2 FS may be used, simultaneously, in computing each bound of the type-reduced interval. Observe from the first part of Table VI, as well as Fig. 8(a), that both the upper and lower MFs of \widetilde{X}_1^e are used in computing y_l , and they are used in different rules: The UMF of \widetilde{X}_1^e is used in computing \overline{f}^1 , the firing level of Rule \widetilde{R}^1 , whereas the LMF of \widetilde{X}_1^e is used in computing \underline{f}^2 , the firing level of Rule \widetilde{R}^2 . Similarly, the upper and lower MFs of \widetilde{X}_2^e are used, simultaneously, in different rules for computing y_l . Observe also from the second part in Table VI and Fig. 8(b) that the upper and lower MFs of \widetilde{X}_1^e and \widetilde{X}_2^e are used, simultaneously, in different rules for computing y_l . This novelty is again impossible for a T1 FLC because it does not have embedded T1 FSS and the same MFs are always used in computing the firing levels of all rules.

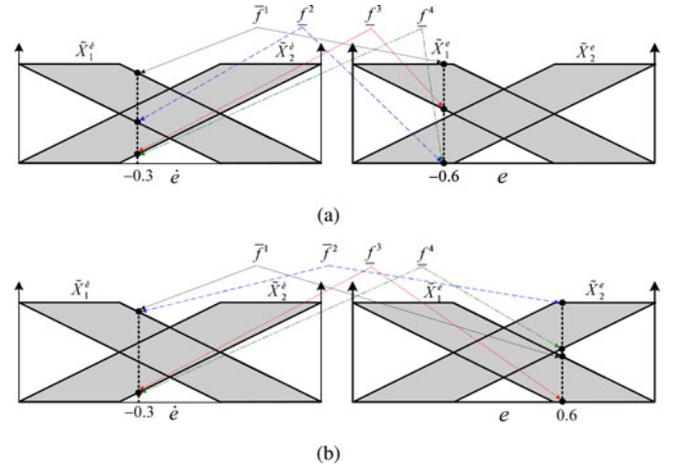


Fig. 8. Embedded T1 FSS used. (a) In (8) for computing y_l , where $(e', e') = (-0.3, -0.6)$ and the LMFs of \widetilde{X}_1^e and \widetilde{X}_2^e are used in computing the firing level \underline{f}_2 of Rule \widetilde{R}^2 . (b) In (10) for computing y_l , where $(e', e') = (-0.3, 0.6)$, and the UMFs of \widetilde{X}_1^e and \widetilde{X}_2^e are used in computing the firing level \overline{f}_2 of Rule \widetilde{R}^2 . Observe that in (a), both the UMFs and LMFs of \widetilde{X}_1^e and \widetilde{X}_2^e are used in computing y_l , and in (b), both the UMFs and LMFs of \widetilde{X}_1^e and \widetilde{X}_2^e are used in computing y_l .

We consider adaptiveness and novelty as two fundamental differences between IT2 and T1 FLCs. Although they are illustrated by specific numerical examples, they are fundamental to an arbitrary IT2 FLC.

Theorem 1: y_l in (3) cannot be implemented by a T1 FLC using the same rulebase.

The proof is given in Appendix A.

Theorem 2: y_r in (4) cannot be implemented by a T1 FLC using the same rulebase.

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

Based on Theorems 1 and 2, we can easily reach the following conclusion.

Theorem 3: An IT2 FLC using the KM type reducer cannot be implemented by a T1 FLC using the same rulebase.

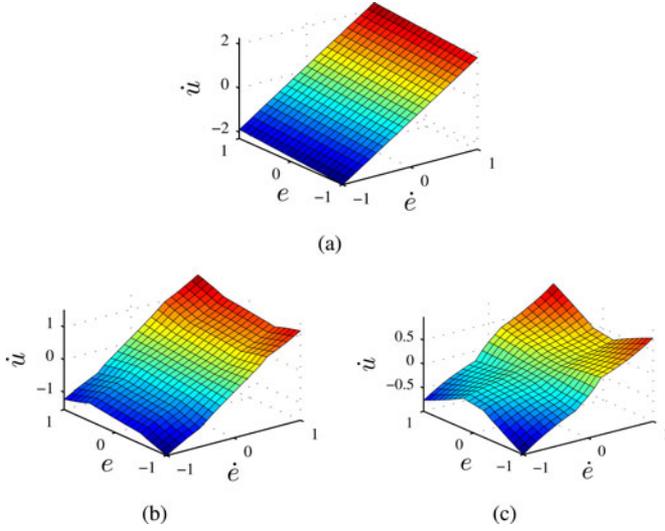


Fig. 9. Control surface of (a) FLC_1 , (b) \widetilde{FLC}_1 with $d_{\dot{e}} = d_e = 0.5$, and (c) \widetilde{FLC}_1 with $d_{\dot{e}} = d_e = 0.8$.

Theorem 3 is very helpful in understanding why IT2 FLCs may outperform T1 FLCs. It suggests that an IT2 FLC can implement a more complex control surface than a T1 FLC: When there is no FOU, an IT2 FLC collapses to a T1 FLC; with FOU, an IT2 FLC can implement a control surface that cannot be obtained from a T1 FLC using the same rulebase. Note that Theorem 3 does not conflict with the fact that T1 fuzzy logic systems are universal approximators [9], [28]: Being a universal approximator requires a T1 fuzzy logic system to have an arbitrarily large number of MFs, whereas, in this paper, we only consider IT2 and T1 FLCs with the same rulebase and a fixed (small) number of MFs.

Having pointed out the fundamental differences between IT2 and T1 FLCs, next, we will introduce several methods to visualize and analyze the effects of these differences.

IV. VISUALIZE THE DIFFERENCE BETWEEN INTERVAL TYPE-2 AND TYPE-1 FUZZY LOGIC CONTROLLERS: CONTROL SURFACE

The control surfaces of FLC_1 , \widetilde{FLC}_1 with $d_{\dot{e}} = d_e = 0.5$, and \widetilde{FLC}_1 with $d_{\dot{e}} = d_e = 0.8$ are shown in Fig. 9(a)–(c), respectively. The control surfaces of FLC_2 , \widetilde{FLC}_2 with $d_{\dot{e}} = d_e = 0.3$, and \widetilde{FLC}_2 with $d_{\dot{e}} = d_e = 0.6$ are shown in Fig. 10(a)–(c), respectively. Observe that the control surfaces of FLC_1 and FLC_2 are linear and monotonic, as they are constructed to be so; however, the control surfaces of \widetilde{FLC}_1 and \widetilde{FLC}_2 are non-linear, nonmonotonic, and more complex. As the area of FOU increases (i.e., as d_e and $d_{\dot{e}}$ increase), the control surfaces of the IT2 FLCs become more non-linear and more complex.

Interestingly, the control surface of \widetilde{FLC}_2 with $d_{\dot{e}} = d_e = 0.6$, which is shown in Fig. 10(c), has several *jump discontinuities*,⁸ e.g., when $\dot{e} = \pm 0.6$. This can never happen

⁸According to Wu and Mendel [55], a function $f(x)$ has a *jump discontinuity* at c if $f(c)$ is defined but $\lim_{x \rightarrow c^+} f(x) \neq \lim_{x \rightarrow c^-} f(x)$, i.e., both $f(c)$ and $f(c + \delta)$ are defined, but $f(c + \delta)$ does not approach $f(c)$ as δ approaches 0.

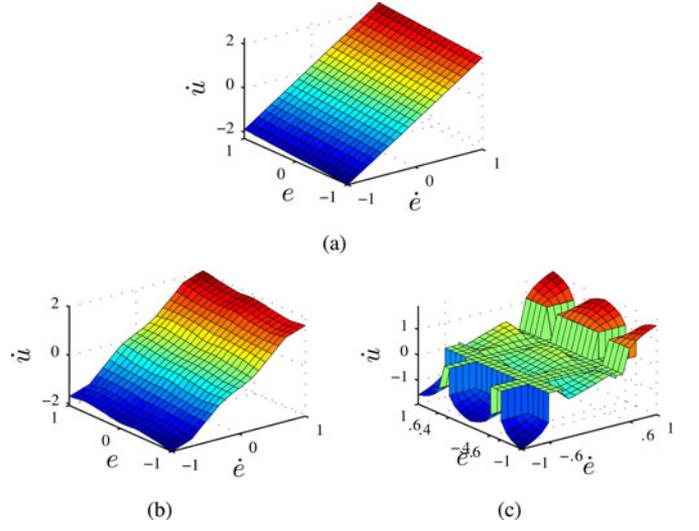


Fig. 10. Control surface (a) FLC_2 , (b) \widetilde{FLC}_2 with $d_{\dot{e}} = d_e = 0.3$, and (c) \widetilde{FLC}_2 with $d_{\dot{e}} = d_e = 0.6$.

in a T1 FLC. The reason is explained in a theorem (Theorem 5) and illustrated by an example by Wu and Mendel in [55]. We repeat their most important discovery ([55, Corollary 1]) in the following for the completeness of this paper. Note that it is applicable to any IT2 FLC with any number of IT2 FSs, whether the IT2 FSs are symmetrical or not.

Theorem 4: An IT2 FLC has a *jump discontinuity* at $\mathbf{x} = \{x_1, x_2, \dots, x_I\}$ if we have the following.

- 1) The input domains are fully covered by the UMFs.
- 2) There exists at least one x_i , $i \in [1, I]$ not covered by the LMFs.
- 3) All rules have different consequents.

The input IT2 FSs for \widetilde{FLC}_2 when $d_{\dot{e}} = d_e = 0.6$ are shown in Fig. 11. Clearly, \dot{e} and e domains are fully covered by the UMFs; however, there exist intervals that are not covered by the LMFs (see the intervals marked by red thick lines, e.g., $\dot{e} \in [-0.6, -0.4] \cup [0.4, 0.6]$). As all nine rules have different consequents, according to Theorem 4, there are jump discontinuities.

Because many times people want to design continuous IT2 FLCs, Wu and Mendel [55] also proposed the following guidelines to design continuous IT2 FLCs.

- 1) To guarantee a continuous control surface regardless of which type reduction and defuzzification method is used, Gaussian IT2 FSs should be employed.
- 2) When triangular and/or trapezoidal IT2 FSs are used, to guarantee a continuous control surface, the LMFs should cover every input domain. This implies that the UMFs must also cover every input domain.

The aforementioned Guideline 2 implies that the design of a continuous IT2 FLC using triangular and/or trapezoidal IT2 FSs is a constrained problem, and the covering constraint must be considered at the beginning of the design.

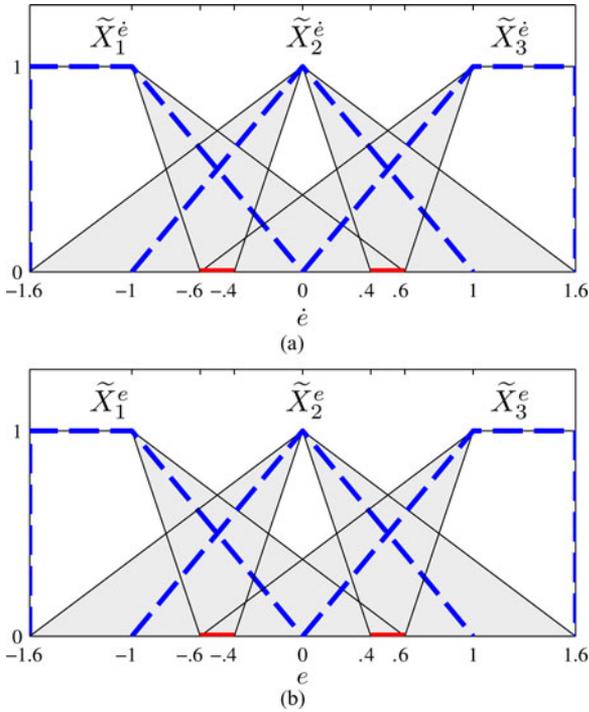


Fig. 11. Input IT2 FSs for \widetilde{FLC}_2 when $d_{\dot{e}} = d_e = 0.6$. (a) IT2 FSs for \dot{e} . (b) IT2 FSs for e . The dashed blue lines are the T1 FSs for FLC_2 .

V. VISUALIZE THE DIFFERENCES BETWEEN INTERVAL TYPE-2 AND TYPE-1 FUZZY LOGIC CONTROLLERS: P-MAP

The control surfaces in the previous section give us some intuitive understandings about the FLCs, e.g., complexity, continuity, monotonicity, etc; however, it is not easy to connect these properties with control performance. In this section, we use P-map [50] to compare the output of a baseline T1 PI FLC $\dot{u}_1(\dot{e}, e)$ and the output of an IT2 PI FLC $\dot{u}_2(\dot{e}, e)$, as well as to relate the difference to control performance.

The P-map represents the difference between the IT2 and T1 FLCs as a variable gain proportional controller

$$K'_P(\dot{e}, e) = \frac{\dot{u}_2(\dot{e}, e) - \dot{u}_1(\dot{e}, e)}{\dot{e}}.$$

By visualizing the magnitude of $K'_P(\dot{e}, e)$ for different \dot{e} and e , we can get some intuitive understanding on the control performance difference between the two FLCs, taking advantage of the simple and well-known properties of a proportional controller.

As an example, the left column in Fig. 12 shows the difference between the control surfaces of \widetilde{FLC}_1 and FLC_1 . Observe that the difference is nonlinear and nonmonotonic, but it is difficult to discover other useful information. However, observe the following from the P-maps in the right column in Fig. 12.

- 1) The difference is nonlinear, as the proportional gain in the P-maps is not a constant.
- 2) Most parts of the P-maps are negative, especially for the area around the steady state (the origins in the two P-maps); therefore, generally, \widetilde{FLC}_1 has a smaller proportional gain than FLC_1 , which may result in less oscillations, as confirmed by previous experimental results [21],

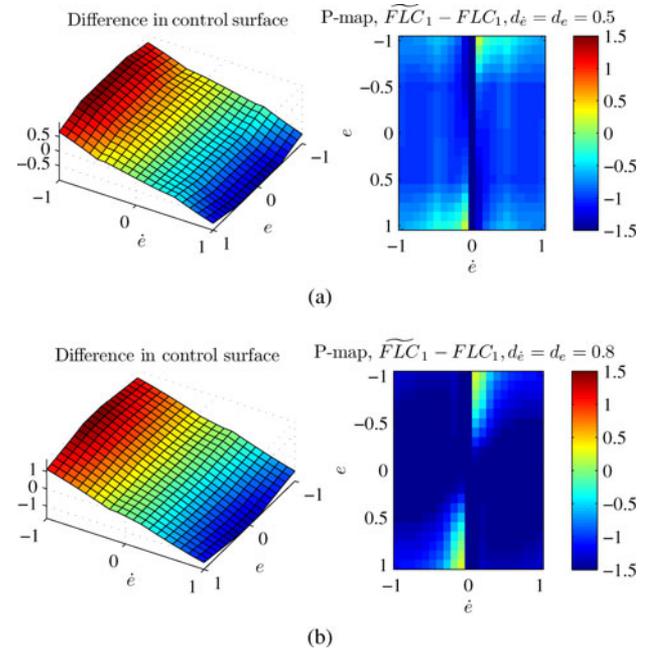


Fig. 12. (a) Difference between the two control surfaces in Fig. 9(b) and (a) and the corresponding P-map. (b) Difference between the two control surfaces in Fig. 9(c) and (a) and the corresponding P-map.

[57], [60], [61]. An intuitive reason, which is offered by one of the reviewers, is that the “blurring” of the antecedent MFs in an IT2 FLC introduces imprecision into the rulebase in determining the proper control response. Therefore, a more “cautious” control response is indicated as a result of the propagation of this imprecision to the output of the FLC.

In Section VII, we also derive the equivalent PI gains for \widetilde{FLC}_1 ; however, because of the iterative nature of the KM algorithms, it is very difficult, if not impossible, to extend those results to IT2 FLCs with arbitrary FOU. The P-map serves a tool to, conveniently, visualize the difference between the proportional gains of an IT2 FLC and a baseline controller, and to qualitatively understand the effect of this difference in control performance.

VI. VISUALIZE THE DIFFERENCE BETWEEN INTERVAL TYPE-2 AND TYPE-1 FUZZY LOGIC CONTROLLERS: EQUIVALENT GENERALIZED TYPE-1 FUZZY SETS

We have visualized the differences between the outputs of IT2 and T1 FLCs using control surface and P-map. In this section,⁹ we further visualize the differences between IT2 and T1 FLCs by comparing their MFs. First, two important definitions are introduced.

Definition 3: A generalized T1 FS is similar to an ordinary T1 FS, except that its *generalized membership grade* is not necessarily constrained in $[0, 1]$, i.e., the generalized membership grade can be negative or larger than 1.

⁹Preliminary results in this section have been presented in [59].

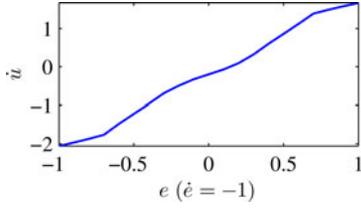


Fig. 13. Slice ($\dot{e} = -1$) of the control surface in Fig. 10(b).

Definition 4: A *generalized T1 FLC* is an FLC using generalized T1 FSs.

The key idea that is used in this section is that an IT2 FLC may be viewed as being equivalent to a group of generalized T1 FLCs, referred to as *equivalent generalized T1 FLCs* (EGT1FLCs), as long as both controllers have identical control surfaces. To identify the EGT1FLCs that can be used in place of an IT2 FLC, the FOU is first reduced into a group of EGT1FSs. By analyzing the characteristics of the EGT1FSs and EGT1FLCs, conclusions about the effects of FOU can be drawn, and the differences between IT2 and T1 FLCs can be made clear.

A. Theory of Equivalent Generalized Type-1 Fuzzy Sets and Equivalent Generalized Type-1 Fuzzy Logic Controllers

Although the control surface of an IT2 FLC is generally more nonlinear and more complex than that of a baseline T1 FLC, the output corresponding to a particular input is still fixed once the system parameters are selected. As a T1 FLC has the same property, the implication is that the interval firing levels of the IT2 FSs corresponding to a particular input–output pair can, effectively, be replaced by crisp values without affecting the system output. The task of finding the EGT1FSs can be achieved by first making a vertical cut to obtain a slice of the control surface where all points have the same \dot{e} (or e) value. That is, for a particular input \dot{e} (or e), a curve representing the relationship between the output \dot{u} and the input e (or \dot{e}) can be plotted [e.g., see Fig. 13 for a slice ($\dot{e} = -1$) of the control surface in Fig. 10(b)]. Each slice is, then, replicated by replacing the IT2 FLC with a generalized T1 FLC, called EGT1FLC. There are countless generalized T1 FLCs that can duplicate that slice of the control surface. For best understanding and comparison, we would like the EGT1FLC to be meaningful and the EGT1FLC corresponding to different slices to share some common structure.

Assume the IT2 FLC has N IT2 FSs in its rulebase. We preassign an embedded T1 FS (chosen arbitrarily or based on intuitions) to $N - 1$ of them and then identify the EGT1FS to replace the N th IT2 FS so that the slice of the control surface is duplicated. The technique of designating the embedded T1 FSs for all but one IT2 FS is akin to amassing in the N th IT2 FS the degrees of freedom provided by all the IT2 FSs. Since the control surface of the IT2 FLC is more complex, the shape of the slice may change as \dot{e} (or e) is varied; therefore, different EGT1FSs may be needed to reproduce the curve corresponding to different \dot{e} (or e). By considering all \dot{e} (or e) within the

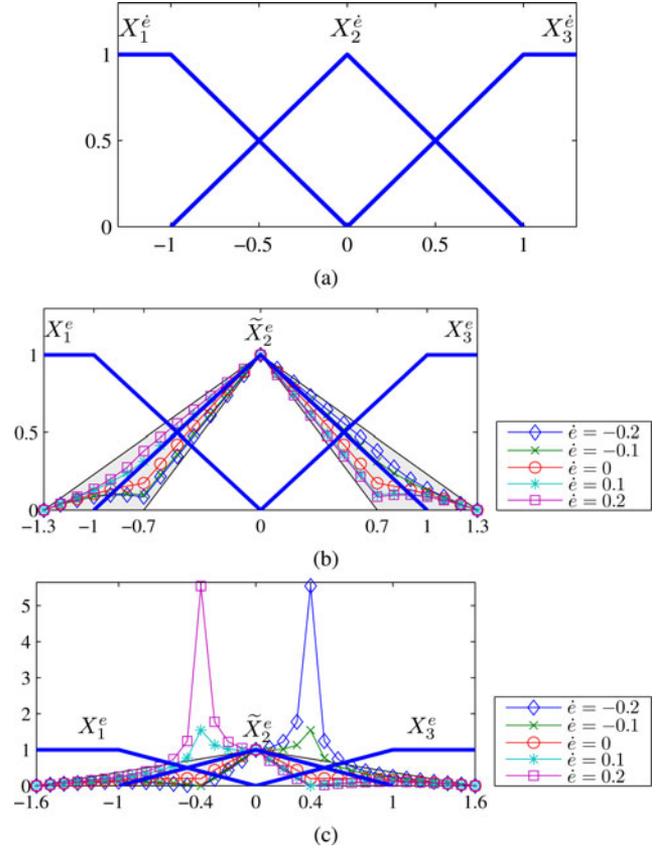


Fig. 14. EGT1FSs of \tilde{X}_2^e when the size of the FOU of \tilde{X}_2^e changes. (a) Input MFs for \widetilde{FLC}_2 in the e domain. (b) Input MFs and the EGT1FSs for \widetilde{FLC}_2 in the e domain when $d_e = 0.3$ in \tilde{X}_2^e . (c) Input MFs and the EGT1FSs for \widetilde{FLC}_2 in the e domain when $d_e = 0.6$ in \tilde{X}_2^e .

universe of discourse, the collection of EGT1FSs that duplicates all the slices and, therefore, the control surface can be found.

The formal definitions of EGT1FLCs and EGT1FSs are given in the following.

Definition 5: EGT1FLCs are the group of generalized T1 FLCs that, together, has the same control surface as an IT2 FLC. For an IT2 FLC that has N IT2 FSs, its EGT1FLCs comprise $N - 1$ embedded (traditional) T1 FS and a group of EGT1FSs.

Definition 6: EGT1FSs are the collection of generalized T1 FSs that can be used in place of the FOUs in an IT2 FLC.

The detailed procedure to identify the EGT1FSs and EGT1FLCs is given in Appendix B.

B. Results and Discussions

For ease of understanding, we introduce symmetrical FOU to only X_2^e to obtain \tilde{X}_2^e and keep all other FSs in \widetilde{FLC}_2 the same as those in FLC_2 and then use the EGT1FSs to visualize the effect of the FOU. When different FOU sizes are used, the resulting EGT1FSs are shown in Fig. 14. Note that, for clarity, we only plot the EGT1FSs corresponding to $\dot{e} = \{-0.2, -0.1, 0, 0.1, 0.2\}$. Observe the following.

- 1) As the FOUs of an IT2 FLC grow, its EGT1FSs become more diverse. This characteristics is very intuitive. Recall

that an EGT1FS at \dot{e}' is a curve that can be used to replace the FOUs when $\dot{e} = \dot{e}'$ without changing the output, which is a slice of the control surface of \widetilde{FLC}_2 at $\dot{e} = \dot{e}'$. Since the control surface of \widetilde{FLC}_2 is more complex when the FOU is larger, the EGT1FSs become more diverse.

- 2) *The EGT1FSs may not lie in the FOU of the corresponding IT2 FS. Moreover, the EGT1FSs are different from traditional T1 FSs in that their generalized membership grades can be larger than 1.* The reason is explained by an example in [59].

In summary, an IT2 FLC can be viewed as a combination of many different EGT1FLCs. A different EGT1FLC is utilized when the input is changed, thereby providing an IT2 FLC with more adaptiveness. Additionally, the EGT1FSs are different from traditional T1 FSs because of the novelty. These properties enable an IT2 FLC to implement more complex control surface than its T1 counterpart.

VII. ANALYZE THE DIFFERENCE BETWEEN INTEGRAL TYPE-2 AND TYPE-1 FUZZY LOGIC CONTROLLERS: EQUIVALENT PROPORTIONAL-INTEGRAL GAINS

We have visualized the difference between IT2 and T1 FLCs using three methods. Several researchers tried to explore the underlying reason mathematically. However, it is very difficult since the KM algorithms do not have a closed-form solution. Du and Ying [12] partitioned the input domain into many small regions and, then, derived and analyzed the analytical solutions in each region. They found that it is very difficult to perform the derivation and analysis using the popular center-of-sets type reducer; therefore, they proposed an average defuzzifier. However, as explained in [49] and [55] as well as the next section of this paper, the average defuzzifier is quite different from the center-of-sets type reducer, which is consistent with the extension principle and is theoretically well grounded. Nie and Tan [38] also partitioned the input domain into many small regions and, then, derived and analyzed the analytical solutions in each region. However, because there are too many small regions and the analytical expression in each region is very complex, it is difficult to gain useful insights. Furthermore, both previous works used the Zadeh AND operator (*min*), whereas the *product* AND operator is more popular in fuzzy modeling and control.

In this section, we study how the control law is changed when symmetrical FOUs are introduced to the baseline T1 FLC \widetilde{FLC}_1 . We consider a very special case (\widetilde{FLC}_1) because the iterative nature of the KM algorithms makes it very challenging to derive nice closed-form solutions for IT2 FLCs with arbitrary FOUs. However, we believe that our analysis can still shed light on more general cases. Moreover, for general cases in which accurate quantitative analysis is challenging or even impossible, we can always use the P-map introduced earlier to qualitatively compare the equivalent PI gains of an IT2 FLC with a baseline PI controller.

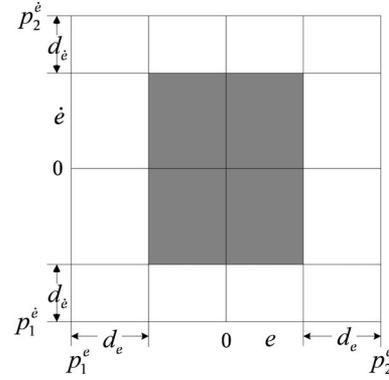


Fig. 15. Region of the input domain determined by (12) and (13). Observe that it is around the origin.

A. Equivalent Proportional-Integral Gains

We use the *product* AND operator. To simplify the computation, we only consider the region around the steady state bounded by the following inequalities:

$$p_1^{\dot{e}} + d_{\dot{e}} \leq \dot{e} \leq p_2^{\dot{e}} - d_{\dot{e}} \quad (12)$$

$$p_1^e + d_e \leq e \leq p_2^e - d_e \quad (13)$$

which are motivated by the observations that the robustness improvement occurs mainly when the system output is near the setpoint [60], [61]. A graphical illustration of the region is shown in Fig. 15. Note that $p_1^{\dot{e}} = -p_2^{\dot{e}}$ and $p_1^e = -p_2^e$ are used in this section.

According to the derivations given in Appendix C, when an input (\dot{e}, e) satisfies (12), (13), and the following constraint:

$$\begin{aligned} & |(K_P p_2^{\dot{e}} p_2^e - K_P p_2^{\dot{e}} d_e + K_I p_2^e d_e) \dot{e} + K_I p_2^{\dot{e}} p_2^e e| \\ & \leq K_P p_2^{\dot{e}} (p_2^{\dot{e}} - d_{\dot{e}}) (p_2^e - d_e) - K_I p_2^e (p_2^{\dot{e}} p_2^e + d_{\dot{e}} d_e) \end{aligned} \quad (14)$$

the output of \widetilde{FLC}_1 is

$$\dot{u} = \alpha K_P \dot{e} + \beta K_I e \quad (15)$$

where

$$\alpha = \frac{p_2^{\dot{e}2} (p_2^e2 - d_e2)}{(p_2^{\dot{e}} p_2^e + d_{\dot{e}} d_e)2 - d_{\dot{e}}2 \dot{e}2} \quad (16)$$

$$\beta = \frac{p_2^{\dot{e}} p_2^e (p_2^{\dot{e}} p_2^e + d_{\dot{e}} d_e)}{(p_2^{\dot{e}} p_2^e + d_{\dot{e}} d_e)2 - d_{\dot{e}}2 \dot{e}2} \quad (17)$$

αK_P is the *equivalent proportional gain* of \widetilde{FLC}_1 , and βK_I is the *equivalent integral gain*. Observe the following.

- 1) Both α and β are functions of \dot{e} , i.e., the equivalent PI gains of \widetilde{FLC}_1 change as the input \dot{e} changes.
- 2) α is always smaller than 1; when $|\dot{e}|$ is small, e.g., $|\dot{e}| \leq d_{\dot{e}}$, β is also smaller than 1. Therefore, for small inputs (disturbances) around the steady state, the equivalent PI gains of \widetilde{FLC}_1 are smaller than the PI gains of \widetilde{FLC}_1 . Consequently, the same amount of disturbance will cause a smaller control signal change in \widetilde{FLC}_1 and, hence, reduces the risk of oscillation.

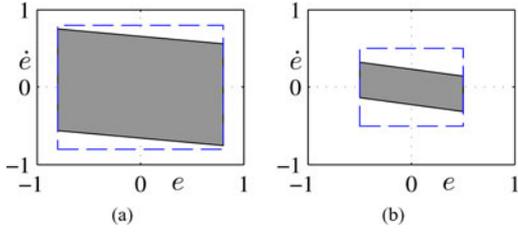


Fig. 16. Input regions where (15) is applicable when (a) $d_e = d_e = 0.2$ and (b) $d_e = d_e = 0.5$ in \widetilde{FLC}_1 . Observe that they are around the origin. The dashed squares are the input regions when the constraint (14) is not imposed.

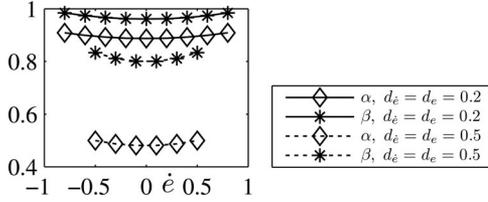


Fig. 17. Relationship between α , β , and \dot{e} .

- 3) Because $\frac{\partial \alpha}{\partial d_e} < 0$, $\frac{\partial \beta}{\partial d_e} < 0$, $\frac{\partial \alpha}{\partial d_{\dot{e}}} < 0$, and $\frac{\partial \beta}{\partial d_{\dot{e}}} < 0$ for small \dot{e} , generally, an increase in d_e and/or $d_{\dot{e}}$ will reduce both α and β , i.e., larger FOU will result in smaller equivalent PI gains around the steady state, and, hence, \widetilde{FLC}_1 is potentially more robust.
- 4) Dividing (16) by (17) yields

$$\frac{\alpha}{\beta} = \frac{p_2^{\dot{e}} p_2^{\dot{e}^2} - p_2^{\dot{e}} d_e^2}{p_2^{\dot{e}} p_2^{\dot{e}^2} + p_2^{\dot{e}} d_e d_{\dot{e}}} < 1$$

i.e., the equivalent proportional gain decreases, relatively, faster than the equivalent integral gain. Observe also that when the FOUs increase, i.e., d_e and/or $d_{\dot{e}}$ increase, α/β decreases. Consequently, a larger FOU will increase the damping of \widetilde{FLC}_1 , and, hence, reduces overshoots and oscillations.

B. Examples

Recent experimental results on IT2 fuzzy PI controller [21], [57], [60], [61] indicate that IT2 PI FLCs are more robust and are better able to eliminate oscillations. This section uses equivalent PI gains to explain the underlying reason.

Example 3: Consider FLC_1 and \widetilde{FLC}_1 introduced in Section II-C, where $K_P = 2.086$, $K_I = 0.2063$, and $P_2^{\dot{e}} = p_2^{\dot{e}} = 1$.

As $K_P \cdot p_2^{\dot{e}} > K_I \cdot p_2^{\dot{e}}$, the equivalent PI gains of \widetilde{FLC}_1 are determined by (15). The closed-form solutions of the equivalent PI gains are derived using the assumption in (21). Using (12)–(14), the input regions in which the equivalent PI gains of \widetilde{FLC}_1 are valid when $d_e = d_{\dot{e}} = \{0.2, 0.5\}$ are plotted in Fig. 16(a) and (b), respectively. The diagrams indicate that the constraint (14) further restricts the region where the equivalent PI gains of \widetilde{FLC}_1 are applicable.

Fig. 17 shows how α and β vary with \dot{e} in the range where the equivalent PI gains are valid. Observe that the extra degrees of freedom provided by the FOU result in varying equivalent PI

gains. Unlike FLC_1 whose input–output relationship is linear, \widetilde{FLC}_1 realizes a nonlinear PI control law around the steady state, as we have seen in previous sections from the control surface and the P-map. Because both α and β are smaller than unity, the equivalent PI gains of \widetilde{FLC}_1 are smaller than the PI gains of FLC_1 . The deviation of \widetilde{FLC}_1 from FLC_1 becomes larger as $d_{\dot{e}}$ increases, i.e., as the FOU increases.

Simulation results on comparing the control performances of FLC_1 and \widetilde{FLC}_1 can be found in [62], and they are consistent with the theoretical analysis in this section.

VIII. ALTERNATIVE TYPE REDUCERS: DO THEY CAPTURE THE FUNDAMENTALS?

In the literature, there are many alternative type reducers, which are used to approximate the KM algorithms, or to simplify computation [51], [52]. With the two fundamental differences between IT2 and T1 FLCs, and Theorems 1–3, in mind, it is interesting to examine their reasonableness. Five¹⁰ alternative type reducers are studied in this paper.

A. Five Alternative Type Reducers

Five popular alternative type reducers [5], [12], [37], [58], [63] are the following.

- 1) *Uncertainty Bound Method:* The UB type reducer, which was proposed by Wu and Mendel [63], computes the output of the IT2 FLC by (5), but

$$y_l = \frac{\underline{y}_l + \bar{y}_l}{2}$$

$$y_r = \frac{\underline{y}_r + \bar{y}_r}{2}$$

where

$$\bar{y}_l = \min\{\underline{y}^{(0)}, \underline{y}^{(N)}\}$$

$$\underline{y}_r = \max\{\bar{y}^{(0)}, \bar{y}^{(N)}\}$$

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

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

¹⁰There are several other methods [16], [17], [32], [39] that bypass type reduction; however, they require the rule consequents to be IT2 FSs so that the union of the fired rule output sets can be computed and used, whereas our IT2 FLC structure, which was defined in Section II, only uses the centroids of the consequent IT2 FSs. Because our structure is simpler and much more widely used in fuzzy logic modeling and control, those methods are not considered in this paper. Additionally, these are some new algorithms for computing the centroid of IT2 FSs [10], [18], [19], which may also be used for type reduction and defuzzification of IT2 fuzzy logic systems; however, since they have not been used in fuzzy logic control, they are not considered in this paper.

in which

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

$$\bar{y}^{(0)} = \frac{\sum_{n=1}^N \bar{f}^n \bar{y}^n}{\sum_{n=1}^N \bar{f}^n}, \quad \bar{y}^{(N)} = \frac{\sum_{n=1}^N \bar{f}^n \bar{y}^n}{\sum_{n=1}^N \bar{f}^n}.$$

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

- 2) *Wu–Tan Method*: Wu and Tan [58] proposed a closed-form type reduction and defuzzification method by making use of the equivalent T1 membership grades. The basic idea is to first find an equivalent T1 membership grade $\mu_{X_i^n}(x_i)$ to replace each firing interval $[\mu_{\underline{X}_i^n}(x_i), \mu_{\bar{X}_i^n}(x_i)]$, i.e.,

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

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

$$y = \frac{\sum_{n=1}^N \mu_{X_i^n}(x_i) y^n}{\sum_{n=1}^N \mu_{X_i^n}(x_i)}.$$

- 3) *Nie–Tan Method*: Nie and Tan [37] proposed another closed-form type reduction and defuzzification method, where the output of an IT2 FLC is computed as

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

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

- 4) *Du–Ying Method*: Du and Ying [12] proposed an average defuzzifier. It first computes 2^N crisp outputs obtained by all possible combinations of the lower and upper firing levels, i.e.,

$$y_m = \frac{\sum_{n=1}^N \underline{f}^{n*} y^n}{\sum_{n=1}^N \underline{f}^{n*}}, \quad m = 1, 2, \dots, 2^N$$

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

$$y = \frac{1}{2^N} \sum_{m=1}^{2^N} y_m.$$

- 5) *Begian–Melek–Mendel Method*: Begian *et al.* [5] proposed another closed-form type reduction and defuzzification method for TSK IT2 FLCs, i.e.,

$$y = \alpha \frac{\sum_{n=1}^N \underline{f}^n y^n}{\sum_{n=1}^N \underline{f}^n} + \beta \frac{\sum_{n=1}^N \bar{f}^n y^n}{\sum_{n=1}^N \bar{f}^n}.$$

where α and β are adjustable coefficients.

B. Do the Alternative Type Reducers Capture the Fundamentals?

We examine the reasonableness of the five alternative type reducers using adaptiveness, novelty, and Theorem 3. The conclusions are the following.

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

In summary, none of the five alternative type reducers captures both adaptiveness and novelty, which are demonstrated in the KM type-reducer-based IT2 FLCs. We emphasize a KM type-reducer-based IT2 FLC because the KM type reducer is consistent with the extension principle and is, theoretically, well grounded. However, this does not mean that the five alternative type reducers are not good. Actually, some of them are very close approximations to the KM type reducer. They have less computational cost and have demonstrated good performance in some applications [30], [37], [43].

IX. CONCLUSION

IT2 FLCs have been widely used and demonstrated for their better ability to handle uncertainties than their T1 counterparts. A challenging question is what the fundamental differences between IT2 and T1 FLCs are. Once the fundamental differences are clear, we can better understand the advantages of IT2 FLCs and, hence, better make use of them. In this paper, we have pointed out two fundamental differences between IT2 and T1 FLCs: 1) *Adaptiveness*, meaning that the embedded T1 FSs used to compute the bounds of the type-reduced interval change as input changes, and 2) *Novelty*, meaning that the UMF and LMF of the same IT2 FS may be used simultaneously in computing each bound of the type-reduced interval. As a result, an IT2 FLC can implement a complex control surface that cannot be achieved by a T1 FLC using the same rulebase.

We have also introduced several methods to visualize and analyze the effects of these two differences, including

- 1) *the control surface*, in which we have shown that the control surface of an IT2 FLC can have discontinuities that cannot be achieved by a T1 FLC;
- 2) *the P-map*, in which we have shown that the difference between the control surfaces of an IT2 FLC and a T1 FLC may be equivalent to a variable gain proportional

controller, whose proportional gain is usually negative around the steady state;

- 3) *the EGTIFSs*, in which we have shown that the FOU's may be replaced by a group of EGTIFSs, whose shape changes as the input changes, and whose generalized membership grade can be larger than 1.
- 4) *the equivalent PI gains*, in which we have shown that a special IT2 PI FLC is equivalent to a variable gain PI controller, whose PI gains change with the input and are smaller than the PI gains of a baseline T1 PI FLC.

All the aforementioned results help understand why an IT2 FLC can be better at eliminating oscillations than a T1 FLC. We have also examined five alternative type reducers for IT2 FLCs and explained why they do not capture the fundamentals of IT2 FLCs.

Finally, we need to point out that although the results in this paper explain the fundamental differences between IT2 and T1 FLCs, they do not directly answer another fundamental question on how to ensure that an IT2 FLC outperforms a T1 FLC. That will be one of our future research directions.

APPENDIX A

PROOF OF THEOREM 1

In this proof, we make use of the following two facts.

- 1) *Fact 1*: The rule firing levels that are used in the KM algorithms are the bounds of the firing intervals. For an upper bound, all involved embedded T1 FSs must be UMFs, and for a lower bound, all involved embedded T1 FSs must be LMFs. There is no mixture of UMFs and LMFs in computing the firing level of any rule.
- 2) *Fact 2*: \bar{f}^1 and \underline{f}^N are, always, used for computing y_l in (3), although we are not sure about whether the upper or lower firing levels should be used for the rest of the rules. For \bar{f}^1 , all involved embedded T1 FSs must be UMFs. For \underline{f}^N , all involved embedded T1 FSs must be LMFs.

We consider the following two cases separately.

- 1) *Rules \tilde{R}^1 and \tilde{R}^N share at least one IT2 FS \tilde{X}_i* . In this case, according to Fact 2, for Rule \tilde{R}^1 , \bar{X}_i must be used, whereas for Rule \tilde{R}^N , \underline{X}_i must be used. This novelty cannot be implemented by a T1 FLC using the same rulebase.
- 2) *Rules \tilde{R}^1 and \tilde{R}^N do not have any IT2 FS in common*, (e.g., for y_l in (10), \tilde{R}^1 involves \tilde{X}_1^e and \tilde{X}_1^e , whereas \tilde{R}^4 involves \tilde{X}_2^e and \tilde{X}_2^e). This case is more complicated than the previous one. We prove it by contradiction. Assume y_l in (3) can be implemented by a T1 FLC, where the same T1 MFs are used in computing all firing levels, e.g., if the UMF of \tilde{X}_1^e is used in computing the firing level of Rule \tilde{R}^1 , it must also be used in computing the firing levels of all other rules involving \tilde{X}_1^e .

In this second case, it is always possible to find a Rule \tilde{R}^k such that Rules \tilde{R}^1 and \tilde{R}^k share at least one common IT2 FS \tilde{X}_i , and Rules \tilde{R}^k and \tilde{R}^N share at least one common IT2 FS \tilde{X}_j (e.g., for y_l in (10), Rules \tilde{R}^1 and \tilde{R}^2 share \tilde{X}_1^e , and Rules \tilde{R}^2 and \tilde{R}^4 share \tilde{X}_2^e). According to Fact 2, \bar{X}_i must be used in Rule \tilde{R}^1 to compute \bar{f}^1 . If y_l

TABLE VII
FIRING LEVELS OF THE NINE RULES OF THE EGT1FLC

Rule No.:	Firing Level	→	Rule Consequent
R^1 :	$\mu_{X_1^e}(e')\mu_{X_1^e}(e')$	→	y_{11}
R^2 :	$\mu_{X_1^e}(e')\mu_{X_2^e}(e')$	→	y_{12}
R^3 :	$\mu_{X_1^e}(e')\mu_{X_3^e}(e')$	→	y_{13}
R^4 :	$\mu_{X_2^e}(e')\mu_{X_1^e}(e')$	→	y_{21}
R^5 :	$\mu_{X_2^e}(e')\mu_{X_2^e}(e')$	→	y_{22}
R^6 :	$\mu_{X_2^e}(e')\mu_{X_3^e}(e')$	→	y_{23}
R^7 :	$\mu_{X_3^e}(e')\mu_{X_1^e}(e')$	→	y_{31}
R^8 :	$\mu_{X_3^e}(e')\mu_{X_2^e}(e')$	→	y_{32}
R^9 :	$\mu_{X_3^e}(e')\mu_{X_3^e}(e')$	→	y_{33}

can be implemented by a T1 FLC using the same rulebase, then \bar{X}_i must also be used in Rule \tilde{R}^k . According to Fact 1, \bar{X}_j must also be used for Rule \tilde{R}^k . For a T1 FLC, this means \bar{X}_j must also be used in Rule \tilde{R}^N , which is a contradiction with Fact 2. Therefore, again, y_l in (3) cannot be implemented by a T1 FLC using the same rulebase.

Theorem 1 is, hence, proved.

APPENDIX B

METHOD FOR IDENTIFYING EQUIVALENT GENERALIZED TYPE-1 FUZZY SETS AND EQUIVALENT GENERALIZED TYPE-1 FUZZY LOGIC CONTROLLERS

In this Appendix, we use FLC_2 and \widetilde{FLC}_2 as an example to illustrate the procedure to identify EGTIFSs and EGT1FLCs; however, the method can be generalized to any IT2 FLC with any FOU shape.

We want to compare \widetilde{FLC}_2 and FLC_2 ; therefore, we replace \tilde{X}_1^e , \tilde{X}_2^e , \tilde{X}_3^e , \tilde{X}_1^e , and \tilde{X}_3^e in \widetilde{FLC}_2 by X_1^e , X_2^e , X_3^e , X_1^e , and X_3^e in FLC_2 , respectively. We, then, find EGTIFSs to replace \tilde{X}_2^e in \widetilde{FLC}_2 so that its control surface can be duplicated.

Let \mathcal{X}_2^e be an EGTIFS of \widetilde{FLC}_2 . Then, the rulebase of the corresponding EGT1FLC is

- $$\begin{aligned}
 R^1 &: \text{ IF } \dot{e} \text{ is } X_1^e \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y_{11} \\
 R^2 &: \text{ IF } \dot{e} \text{ is } X_1^e \text{ and } e \text{ is } \mathcal{X}_2^e, \text{ THEN } \dot{u} \text{ is } y_{12} \\
 R^3 &: \text{ IF } \dot{e} \text{ is } X_1^e \text{ and } e \text{ is } X_3^e, \text{ THEN } \dot{u} \text{ is } y_{13} \\
 R^4 &: \text{ IF } \dot{e} \text{ is } X_2^e \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y_{21} \\
 R^5 &: \text{ IF } \dot{e} \text{ is } X_2^e \text{ and } e \text{ is } \mathcal{X}_2^e, \text{ THEN } \dot{u} \text{ is } y_{22} \\
 R^6 &: \text{ IF } \dot{e} \text{ is } X_2^e \text{ and } e \text{ is } X_3^e, \text{ THEN } \dot{u} \text{ is } y_{23} \\
 R^7 &: \text{ IF } \dot{e} \text{ is } X_3^e \text{ and } e \text{ is } X_1^e, \text{ THEN } \dot{u} \text{ is } y_{31} \\
 R^8 &: \text{ IF } \dot{e} \text{ is } X_3^e \text{ and } e \text{ is } \mathcal{X}_2^e, \text{ THEN } \dot{u} \text{ is } y_{32} \\
 R^9 &: \text{ IF } \dot{e} \text{ is } X_3^e \text{ and } e \text{ is } X_3^e, \text{ THEN } \dot{u} \text{ is } y_{33}.
 \end{aligned}$$

Consider an input pair (\dot{e}', e') . The firing levels of the nine rules of the EGT1FLC are shown in Table VII. The output of the EGT1FLC is

$$\dot{u}_1 = [\mu_{X_1^e}(\dot{e}')\mu_{X_1^e}(e')y_{11} + \mu_{X_1^e}(\dot{e}')\mu_{\mathcal{X}_2^e}(e')y_{12}$$

TABLE VIII
FIRING INTERVALS OF THE FOUR RULES OF \widetilde{FLC}_1

Rule No.:	Firing Interval	→	Rule Consequent
\widetilde{R}^1 :	$[\underline{f}^1, \overline{f}^1] = \left[\frac{(p_2^e - d_e - \dot{e})(p_2^e - d_e - \dot{e})}{4p_2^e p_2^e}, \frac{(p_2^e + d_e - \dot{e})(p_2^e + d_e - \dot{e})}{4p_2^e p_2^e} \right]$	→	$y^1 = K_P \cdot p_1^e + K_I \cdot p_1^e = -K_P \cdot p_2^e - K_I \cdot p_2^e$
\widetilde{R}^2 :	$[\underline{f}^2, \overline{f}^2] = \left[\frac{(p_2^e - d_e - \dot{e})(e + p_2^e - d_e)}{4p_2^e p_2^e}, \frac{(p_2^e + d_e - \dot{e})(e + p_2^e + d_e)}{4p_2^e p_2^e} \right]$	→	$y^2 = K_P \cdot p_1^e + K_I \cdot p_2^e = -K_P \cdot p_2^e + K_I \cdot p_2^e$
\widetilde{R}^3 :	$[\underline{f}^3, \overline{f}^3] = \left[\frac{(e + p_2^e - d_e)(p_2^e - d_e - \dot{e})}{4p_2^e p_2^e}, \frac{(e + p_2^e + d_e)(p_2^e + d_e - \dot{e})}{4p_2^e p_2^e} \right]$	→	$y^3 = K_P \cdot p_2^e + K_I \cdot p_1^e = K_P \cdot p_2^e - K_I \cdot p_2^e$
\widetilde{R}^4 :	$[\underline{f}^4, \overline{f}^4] = \left[\frac{(e + p_2^e - d_e)(e + p_2^e - d_e)}{4p_2^e p_2^e}, \frac{(e + p_2^e + d_e)(e + p_2^e + d_e)}{4p_2^e p_2^e} \right]$	→	$y^4 = K_P \cdot p_2^e + K_I \cdot p_2^e = K_P \cdot p_2^e + K_I \cdot p_2^e$

$$\begin{aligned}
& + \mu_{X_1^e}(\dot{e})\mu_{X_3^e}(e')y_{13} + \mu_{X_2^e}(\dot{e})\mu_{X_1^e}(e')y_{21} \\
& + \mu_{X_2^e}(\dot{e})\mu_{X_2^e}(e')y_{22} + \mu_{X_2^e}(\dot{e})\mu_{X_3^e}(e')y_{23} \\
& + \mu_{X_3^e}(\dot{e})\mu_{X_1^e}(e')y_{31} + \mu_{X_3^e}(\dot{e})\mu_{X_2^e}(e')y_{32} \\
& + \mu_{X_3^e}(\dot{e})\mu_{X_3^e}(e')y_{33} / [\mu_{X_1^e}(\dot{e})\mu_{X_1^e}(e') \\
& + \mu_{X_1^e}(\dot{e})\mu_{X_2^e}(e') + \mu_{X_1^e}(\dot{e})\mu_{X_3^e}(e') + \mu_{X_2^e}(\dot{e})\mu_{X_1^e}(e') \\
& + \mu_{X_2^e}(\dot{e})\mu_{X_2^e}(e') + \mu_{X_2^e}(\dot{e})\mu_{X_3^e}(e') + \mu_{X_3^e}(\dot{e})\mu_{X_1^e}(e') \\
& + \mu_{X_3^e}(\dot{e})\mu_{X_2^e}(e') + \mu_{X_3^e}(\dot{e})\mu_{X_3^e}(e')]. \quad (18)
\end{aligned}$$

Denote the output of \widetilde{FLC}_2 for input pair (\dot{e}, e') as \dot{u}_2 . Then, to duplicate a slice of the control surface, we must have

$$\dot{u}_1 = \dot{u}_2. \quad (19)$$

From (18) and (19), it is easy to solve for $\mu_{X_2^e}(e')$, as

$$\begin{aligned}
& \mu_{X_2^e}(e') \\
& = \{ \dot{u}_2 [\mu_{X_1^e}(e') + \mu_{X_3^e}(e')] [\mu_{X_1^e}(\dot{e}') + \mu_{X_2^e}(\dot{e}') + \mu_{X_3^e}(\dot{e}')] \\
& \quad - \mu_{X_1^e}(\dot{e}') [\mu_{X_1^e}(\dot{e}')y_{11} + \mu_{X_2^e}(\dot{e}')y_{12} + \mu_{X_3^e}(\dot{e}')y_{13}] \\
& \quad - \mu_{X_3^e}(\dot{e}') [\mu_{X_1^e}(\dot{e}')y_{31} + \mu_{X_2^e}(\dot{e}')y_{32} + \mu_{X_3^e}(\dot{e}')y_{33}] \} \\
& \quad / \{ \mu_{X_1^e}(\dot{e}')y_{21} + \mu_{X_2^e}(\dot{e}')y_{22} + \mu_{X_3^e}(\dot{e}')y_{23} \\
& \quad - \dot{u}_2 [\mu_{X_1^e}(\dot{e}') + \mu_{X_2^e}(\dot{e}') + \mu_{X_3^e}(\dot{e}')] \}. \quad (20)
\end{aligned}$$

When e' changes from p_1^e to p_3^e , its $\mu_{X_2^e}(e')$ form the EGT1FS corresponding to $\dot{e} = \dot{e}'$. The complete group of EGT1FSs can, then, be identified by discretizing the \dot{e} domain and applying (20) repeatedly.

In summary, the procedure for finding EGT1FSs for \widetilde{FLC}_2 is as follows.

- 1) Replace $\widetilde{X}_1^e, \widetilde{X}_2^e, \widetilde{X}_3^e, \widetilde{X}_1^e,$ and \widetilde{X}_3^e in \widetilde{FLC}_2 by $X_1^e, X_2^e, X_3^e, X_1^e,$ and X_3^e in FLC_2 , respectively.
- 2) Discretize the \dot{e} domain into m points $\{\dot{e}_1, \dot{e}_2, \dots, \dot{e}_m\}$.
- 3) Discretize the e domain into k points $\{e_1, e_2, \dots, e_k\}$.
- 4) Set $\dot{e} = \dot{e}_1$. Use (20) to calculate the k -generalized membership grades $\mu_{X_2^e}(e_i)$ ($i = 1, \dots, k$) to duplicate the slice of the control surface corresponding to $\dot{e} = \dot{e}_1$. By joining the k -generalized membership grades, the EGT1FS corresponding to $\dot{e} = \dot{e}_1$ is obtained.
- 5) Repeat Step (3) for the remaining $(m - 1)$ elements in $\{\dot{e}_2, \dots, \dot{e}_m\}$.

APPENDIX C

DERIVATION OF THE EQUIVALENT PROPORTIONAL-INTEGRAL GAINS

When an input (\dot{e}, e) falls into the shaded region in Fig. 15, the firing intervals of the four IT2 FSs of \widetilde{FLC}_1 (see Fig. 4) are

$$\begin{aligned}
[\mu_{\underline{X}_1^e}(\dot{e}), \mu_{\overline{X}_1^e}(\dot{e})] &= \left[\frac{p_2^e - d_e - \dot{e}}{2p_2^e}, \frac{p_2^e + d_e - \dot{e}}{2p_2^e} \right] \\
[\mu_{\underline{X}_2^e}(\dot{e}), \mu_{\overline{X}_2^e}(\dot{e})] &= \left[\frac{\dot{e} + p_2^e - d_e}{2p_2^e}, \frac{\dot{e} + p_2^e + d_e}{2p_2^e} \right] \\
[\mu_{\underline{X}_1^e}(e), \mu_{\overline{X}_1^e}(e)] &= \left[\frac{p_2^e - d_e - e}{2p_2^e}, \frac{p_2^e + d_e - e}{2p_2^e} \right] \\
[\mu_{\underline{X}_2^e}(e), \mu_{\overline{X}_2^e}(e)] &= \left[\frac{e + p_2^e - d_e}{2p_2^e}, \frac{e + p_2^e + d_e}{2p_2^e} \right].
\end{aligned}$$

The firing intervals of the four rules of \widetilde{FLC}_1 are listed in Table VIII. To save space, we only consider the case $K_P \cdot p_2^e > K_I \cdot p_2^e$. The case $K_P \cdot p_2^e < K_I \cdot p_2^e$ can be found in [62].

When $K_P \cdot p_2^e > K_I \cdot p_2^e$, observe from Table VIII that

$$y^1 < y^2 < 0 < y^3 < y^4.$$

To derive closed-form solutions, we further impose the following constraint:

$$y^2 \leq y_l \leq y_r \leq y^3. \quad (21)$$

As will be shown in the next section, (21) reduces the shaded region shown in Fig. 15; however, it still ensures that the inputs under consideration are around the steady state.

According to the KM algorithms, (21) indicates that the switch points¹¹ $L = 2$ and $R = 2$. Therefore

$$\begin{aligned}
y_l &= \frac{\overline{f}^1 y^1 + \overline{f}^2 y^2 + \underline{f}^3 y^3 + \underline{f}^4 y^4}{\overline{f}^1 + \overline{f}^2 + \underline{f}^3 + \underline{f}^4} \\
&= \frac{p_2^e (K_P d_e p_2^e - K_P p_2^e \dot{e} + K_P p_2^e d_e + K_I p_2^e e)}{p_2^e p_2^e + d_e d_e - d_e \dot{e}} \quad (22)
\end{aligned}$$

$$\begin{aligned}
y_r &= \frac{\underline{f}^1 y^1 + \underline{f}^2 y^2 + \overline{f}^3 y^3 + \overline{f}^4 y^4}{\underline{f}^1 + \underline{f}^2 + \overline{f}^3 + \overline{f}^4} \\
&= \frac{p_2^e (K_P d_e p_2^e + K_P p_2^e d_e + K_P p_2^e \dot{e} + K_I p_2^e e)}{p_2^e p_2^e + d_e d_e - d_e \dot{e}}. \quad (23)
\end{aligned}$$

¹¹Note that $\{y^n\}$ has been sorted in ascending order.

Hence, the output of \widetilde{FLC}_1 is

$$\begin{aligned} \dot{u} &= \frac{y_l + y_r}{2} \\ &= \frac{p_2^{e^2}(p_2^{e^2} - d_e^2)K_P \dot{e} + p_2^e(p_2^e p_2^{e^2} + p_2^e d_e d_e)K_I e}{(p_2^e p_2^e + d_e d_e)^2 - d_e^2 \dot{e}^2} \\ &= \frac{p_2^{e^2}(p_2^{e^2} - d_e^2)}{(p_2^e p_2^e + d_e d_e)^2 - d_e^2 \dot{e}^2} K_P \dot{e} + \frac{p_2^e p_2^e (p_2^e p_2^e + d_e d_e)}{(p_2^e p_2^e + d_e d_e)^2 - d_e^2 \dot{e}^2} K_I e. \end{aligned} \quad (24)$$

Knowing y_l and y_r in (22) and (23), constraint (21) can be reexpressed as (14). Equation (14), together with (12) and (13), determines the complete input region in which (24) holds.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers and Prof. J. M. Mendel, whose critical comments have greatly improved the readability and technical quality of this paper.

REFERENCES

- [1] J. Aisbett, J. Rickard, and D. Morgenthaler, "Type-2 fuzzy sets as functions on spaces," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 4, pp. 841–844, Aug. 2010.
- [2] J. Aisbett, J. T. Rickard, and D. Morgenthaler, "Multivariate modeling and type-2 fuzzy sets," *Fuzzy Sets Syst.*, vol. 163, no. 1, pp. 78–95, 2011.
- [3] K. Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*. New York: Physica-Verlag, 1999.
- [4] K. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets Syst.*, vol. 20, pp. 87–97, 1986.
- [5] M. Begian, W. Melek, and J. Mendel, "Stability analysis of type-2 fuzzy systems," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Hong Kong, Jun. 2008, pp. 947–953.
- [6] J. Bezdek, "Fuzzy models—what are they, and why?" *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 1–5, 1993.
- [7] H. Bustince, "Indicator of inclusion grade for interval-valued fuzzy sets. Application to approximate reasoning based on interval-valued fuzzy sets," *Int. J. Approx. Reason.*, vol. 23, no. 3, pp. 137–209, 2000.
- [8] O. Castillo and P. Melin, *Type-2 Fuzzy Logic Theory and Applications*. Berlin, Germany: Springer-Verlag, 2008.
- [9] J. L. Castro, "Fuzzy logic controllers are universal approximators," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 4, pp. 629–635, Apr. 1995.
- [10] S. Coupland and R. I. John, "Geometric type-1 and type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 1, pp. 3–15, Feb. 2007.
- [11] G. Deschrijver and E. E. Kerre, "On the relationship between some extensions of fuzzy set theory," *Fuzzy Sets Syst.*, vol. 133, no. 2, pp. 227–235, 2003.
- [12] X. Du and H. Ying, "Derivation and analysis of the analytical structures of the interval type-2 fuzzy-PI and PD controllers," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 4, pp. 802–814, Aug. 2010.
- [13] J. M. Garibaldi and S. Guadarrama, "Constrained type-2 fuzzy sets," in *Proc. IEEE Symp. Adv. Type-2 Fuzzy Logic Syst.*, Paris, France, Apr. 2011, pp. 66–73.
- [14] J. Goguen, "L-fuzzy sets," *J. Math. Anal. Appl.*, vol. 18, pp. 145–174, 1967.
- [15] M. B. Gorzalczy, "A method of inference in approximate reasoning based on interval-valued fuzzy sets," *Fuzzy Sets Syst.*, vol. 21, pp. 1–17, 1987.
- [16] M. Gorzalczy, "Decision making in signal transmission problems with interval-valued fuzzy sets," *Fuzzy Sets Syst.*, vol. 23, pp. 191–203, 1987.
- [17] M. Gorzalczy, "Interval-valued fuzzy controller based on verbal model of object," *Fuzzy Sets Syst.*, vol. 28, pp. 45–53, 1988.
- [18] S. Greenfield, R. John, and S. Coupland, "A novel sampling method for type-2 defuzzification," in *Proc. U.K. Workshop Comput. Intell.*, London, U.K., Sep. 2005, pp. 120–127.
- [19] S. Greenfield, F. Chiclana, S. Coupland, and R. John, "The collapsing method of defuzzification for discretised interval type-2 fuzzy sets," *Inf. Sci.*, vol. 179, no. 13, pp. 2055–2069, 2008.
- [20] H. Hagnas, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 524–539, Aug. 2004.
- [21] H. Hagnas, "Type-2 FLCs: A new generation of fuzzy controllers," *IEEE Comput. Intell. Mag.*, vol. 2, no. 1, pp. 30–43, Feb. 2007.
- [22] K. Hirota and W. Pedrycz, "Fuzzy computing for data mining," *Proc. IEEE*, vol. 87, no. 9, pp. 1575–1600, Sep. 1999.
- [23] E. A. Jammeh, M. Fleury, C. Wagner, H. Hagnas, and M. Ghanbari, "Interval type-2 fuzzy logic congestion control for video streaming across IP networks," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 5, pp. 1123–1142, Oct. 2009.
- [24] J. Kacprzyk, A. Wilbik, and S. Zadrozny, "Linguistic summarization of time series using a fuzzy quantifier driven aggregation," *Fuzzy Sets Syst.*, vol. 159, pp. 1485–1499, 2008.
- [25] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Inf. Sci.*, vol. 132, pp. 195–220, 2001.
- [26] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 643–658, Dec. 1999.
- [27] N. K. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, Apr. 2002.
- [28] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. Comput.*, vol. 43, no. 11, pp. 1329–1333, Nov. 1994.
- [29] S. S. Liao, T. H. Tang, and W.-Y. Liu, "Finding relevant sequences in time series containing crisp, interval, and fuzzy interval data," *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 5, pp. 2071–2079, Oct. 2004.
- [30] C. Lynch, H. Hagnas, and V. Callaghan, "Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Vancouver, BC, Canada, Jul. 2006, pp. 7217–7224.
- [31] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice-Hall, 2001.
- [32] J. M. Mendel, "Advances in type-2 fuzzy sets and systems," *Inf. Sci.*, vol. 177, no. 1, pp. 84–110, 2007.
- [33] J. M. Mendel, "Computing with words: Zadeh, Turing, Popper and Occam," *IEEE Comput. Intell. Mag.*, vol. 2, no. 4, pp. 10–17, Nov. 2007.
- [34] J. M. Mendel and R. I. John, "Type-2 fuzzy sets made simple," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 117–127, Apr. 2002.
- [35] J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*. Hoboken, NJ: Wiley-IEEE Press, 2010.
- [36] M. Mizumoto, "Realization of PID controls by fuzzy control methods," *Fuzzy Sets Syst.*, vol. 70, pp. 171–182, 1995.
- [37] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Hong Kong, Jun. 2008, pp. 1425–1432.
- [38] M. Nie and W. W. Tan, "Derivation of the analytical structure of symmetrical IT2 fuzzy PD and PI controllers," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Barcelona, Spain, Jul. 2010, pp. 1–8.
- [39] A. Niewiadomski, J. Ochelska, and P. Szczepaniak, "Interval-valued linguistic summaries of databases," *Control Cybern.*, vol. 35, no. 2, pp. 415–443, 2006.
- [40] W. Pedrycz, "Fuzzy set technology in knowledge discovery," *Fuzzy Sets Syst.*, vol. 98, pp. 279–290, 1998.
- [41] M. R. Rajati, J. M. Mendel, and D. Wu, "Solving Zadeh's Magnus challenge problem on linguistic probabilities via linguistic weighted averages," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Taipei, Taiwan, Jun. 2011, pp. 2177–2184.
- [42] M. R. Rajati, D. Wu, and J. M. Mendel, "On solving Zadeh's Tall Swedes problem," presented at the World Conf. Soft Comput., San Francisco, CA, May 2011.
- [43] W. W. Tan and D. Wu, "Design of type-reduction strategies for type-2 fuzzy logic systems using genetic algorithms," in *Advances in Evolutionary Computing for System Design*, L. Jain, V. Palade, and D. Srinivasan, Eds. New York: Springer-Verlag, 2007, pp. 169–188.
- [44] I. B. Türkşen, "Interval-valued fuzzy sets based on normal forms," *Fuzzy Sets Syst.*, vol. 20, pp. 191–210, 1986.
- [45] M. Versaci and F. C. Morabito, "Fuzzy time series approach for disruption prediction in Tokamak reactors," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1503–1506, May 2003.
- [46] T. S. Wallsten and D. V. Budesu, "A review of human linguistic probability processing: General principles and empirical evidence," *Knowl. Eng. Rev.*, vol. 10, no. 1, pp. 43–62, 1995.

- [47] L.-X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ: Prentice-Hall, 1997.
- [48] D. Wu, "A constrained representation theorem for interval type-2 fuzzy sets using convex and normal embedded type-1 fuzzy sets, and its application to centroid computation," presented at the World Conf. Soft Comput., San Francisco, CA, May 2011.
- [49] D. Wu, "An interval type-2 fuzzy logic system cannot be implemented by traditional type-1 fuzzy logic systems," presented at the World Conf. Soft Comput., San Francisco, CA, May 2011.
- [50] D. Wu, "P-map: An intuitive plot to visualize, understand, and compare variable-gain PI controllers," presented at the Int. Conf. Auton. Intell. Syst., Burnaby, BC, Canada, Jun. 2011.
- [51] D. Wu, "Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: Overview and comparison," *IEEE Trans. Fuzzy Syst.*, 2012, to be published.
- [52] D. Wu, "An overview of alternative type-reduction approaches for reducing the computational cost of interval type-2 fuzzy logic controllers," in *Proc. IEEE World Congr. Comput. Intell.*, Brisbane, Australia, Jun. 2012.
- [53] D. Wu and J. M. Mendel, "Enhanced Karnik-Mendel algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 923-934, Aug. 2009.
- [54] D. Wu and J. M. Mendel, "Linguistic summarization using IF-THEN rules and interval type-2 fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 136-151, Feb. 2011.
- [55] D. Wu and J. M. Mendel, "On the continuity of type-1 and interval type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 179-192, Feb. 2011.
- [56] D. Wu and M. Nie, "Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Taipei, Taiwan, Jun. 2011, pp. 2131-2138.
- [57] D. Wu and W. W. Tan, "A type-2 fuzzy logic controller for the liquid-level process," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, vol. 2, Budapest, Hungary, Jul. 2004, pp. 953-958.
- [58] D. Wu and W. W. Tan, "Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Reno, NV, May 2005, pp. 353-358.
- [59] D. Wu and W. W. Tan, "Type-2 FLS modeling capability analysis," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Reno, NV, May 2005, pp. 242-247.
- [60] D. Wu and W. W. Tan, "Genetic learning and performance evaluation of type-2 fuzzy logic controllers," *Eng. Appl. Artif. Intell.*, vol. 19, no. 8, pp. 829-841, 2006.
- [61] D. Wu and W. W. Tan, "A simplified type-2 fuzzy controller for real-time control," *ISA Trans.*, vol. 15, no. 4, pp. 503-516, 2006.
- [62] D. Wu and W. W. Tan, "Interval type-2 fuzzy PI controllers: Why they are more robust," in *Proc. IEEE Int. Conf. Granular Comput.*, San Jose, CA, Aug. 2010, pp. 802-807.
- [63] H. Wu and J. M. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 622-639, Oct. 2002.
- [64] R. Yager, "A new approach to the summarization of data," *Inf. Sci.*, vol. 28, pp. 69-86, 1982.
- [65] R. Yager and D. Filev, *Essentials of Fuzzy Modeling and Control*. New York: Wiley, 1994.
- [66] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, pp. 338-353, 1965.
- [67] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-1," *Inf. Sci.*, vol. 8, pp. 199-249, 1975.
- [68] L. A. Zadeh, "Fuzzy logic = Computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 103-111, May 1996.
- [69] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 19, pp. 111-127, 1997.



Dongrui Wu (S'05-M'09) received the B.E. degree in automatic control from the University of Science and Technology of China, Hefei, China, in 2003, the M.Eng. degree in electrical engineering from the National University of Singapore, Singapore, in 2005, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 2009.

He is currently with the Machine Learning Laboratory, GE Global Research, Niskayuna, NY. He has more than 50 publications, including a book entitled

Perceptual Computing: Aiding People in Making Subjective Judgments (with J. M. Mendel, New York: Wiley-IEEE, 2010). His research interests include affective computing, computational intelligence, decision-support systems, intelligent control, machine learning, optimization, speech and physiological signal processing, and smart oilfield technologies.

Dr. Wu has been an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS since 2011. He is the Vice Chair of the IEEE Computational Intelligence Society Social Media Subcommittee (2011) and the Vice Chair of the Computing With Words Task Force of the Fuzzy Systems Technical Committee (2011). He is also an elected Executive Committee member of HUMAINE: the international affective computing research society. He received the 2005 IEEE International Conference on Fuzzy Systems Best Student Paper Award, the 2012 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award, and an Award of Excellence from GE Global Research in 2010 for outstanding performance.