Deriving and Analyzing Analytical Structures of a Class of Typical Interval Type-2 TS Fuzzy Controllers

Haibo Zhou, Member, IEEE, and Hao Ying, Fellow, IEEE

Abstract—A conventional controller’s explicit input–output mathematical relationship, also known as its analytical structure, is always available for analysis and design of a control system. In contrast, virtually all type-2 (T2) fuzzy controllers are treated as black-box controllers in the literature in that their analytical structures are unknown, which inhibits precise and comprehensive understanding and analysis. In this regard, a long-standing fundamental issue remains unresolved: how a T2 fuzzy set’s footprint of uncertainty, a key element differentiating a T2 controller from a type-1 (T1) controller, affects a controller’s analytical structure. In this paper, we describe an innovative technique for deriving analytical structures of a class of typical interval T2 (IT2) TS fuzzy controllers. This technique makes it possible to analyze the analytical structures of the controllers to reveal the role of footprints of uncertainty in shaping the structures. Specifically, we have mathematically proven that under certain conditions, the larger the footprints, the more the IT2 controllers resemble linear or piecewise linear controllers. That is to say the smaller the footprints, the more nonlinear the controllers. The most nonlinear IT2 controllers are attained at zero footprints, at which point they become T1 controllers. This finding implies that sometimes if strong nonlinearity is most important and desired, one should consider using a smaller footprint or even just a T1 fuzzy controller. This paper exemplifies the importance and value of the analytical structure approach for comprehensive analysis of T2 fuzzy controllers.

Index Terms—Analytical structure, footprint of uncertainty, TS fuzzy rules, type-2 (T2) fuzzy control.

I. INTRODUCTION

DEVELOPED from type-2 (T2) fuzzy logic and systems [1], [12], [13], [17], [23], T2 fuzzy control theory is an emerging field of research [5]–[7], [11], [15], [22], [28], [39]. Analytical structure of a T2 controller is related to a footprint of uncertainty of a T2 fuzzy set, which is an underlying factor differentiating a T2 fuzzy controller from a type-2 (T1) fuzzy controller. By analytical structure, we mean an explicit mathematical representation that precisely describes input–output relationship of a controller [34]. Finding an adequate footprint of uncertainty is a key design step. Unfortunately, there lacks a method that is supported by convincing evidence. At present, a footprint is largely determined through the trial-and-error effort involving computer simulation or experiment. This problem was well recognized a long time ago and effort has been made to reduce the experimentation. The results so far are some recommended rules of thumb, one of which, for instance, is the notion linking the magnitude of a footprint to the extent of uncertainty associated with a T2 controller’s input variable. Basically, the notion states that the more the uncertainty, the larger the footprint should be in order to achieve a good control performance. Notions like this were generated based on researchers’ observations in their computer simulation and experimental studies as well as from their personal understanding of how a footprint should be related to fuzzy control. Validity of this and other related notions for any T2 fuzzy controller other than those in the studies remains an open question. A related fundamental question that no one has raised before is this: is it possible for a T2 controller to be a linear or piecewise controller even though its footprint of uncertainty is nonzero? If this can be the case, we would then want to know condition under which this will occur so that in practice we can avoid the inadvertent mistake of producing a linear or piecewise linear controller while our real intention is, as always, to construct a nonlinear controller.

In our opinion, these issues cannot be adequately addressed by empirical approaches, such as computer simulation and lab experiments, due to the approaches’ inherent limitations. These approaches can study only a finite number of cases or scenarios and hence are not rigorous enough to generalize and produce conclusive findings. We argue that studying analytical structure of a T2 controller is one of the few effective ways to produce rigorous results. With exposed analytical structure, one will then be able to have a precise and comprehensive understanding of a T2 fuzzy controller. Moreover, analyzing and designing a T2 fuzzy control system can be carried out in the framework of the well-developed conventional nonlinear
control theory as a T2 fuzzy control problem actually becomes a nonlinear control problem. Many time-tested methods can be taken advantage of for T2 controllers, resulting in better analysis and design outcomes (e.g., less conservative stability criteria). The fruitful outcomes of the analytical structure studies on various T1 fuzzy controllers over the past 25 years (see [2], [4], [8], [10], [14], [16], [18]–[21], [29], [30], [32]) hint a bright future for this line of research for T2 controllers. Because a T2 fuzzy controller’s configuration is much more complex than a T1 controller’s, deriving its analytical structure is significantly more challenging. To date, only a handful of analytical structures of T2 fuzzy controllers have been uncovered, all of which are for the Mamdani IT2 fuzzy controllers [9], [24], [36], [37] except our preliminary results on the IT2 TS fuzzy controllers [36], [38].

In this paper, we focus on: 1) deriving analytical structures of a class of typical IT2 TS fuzzy controllers and 2) investigating the role of a footprint of uncertainty in shaping the analytical structures of two subsets of the controllers to reveal exactly how the footprint affects nonlinearity of the controllers. More concretely, we first present, step by step in Section III, an innovative method capable of deriving analytical structures of a class of typical IT2 TS fuzzy controllers whose configuration is given in Section II. The configuration is quite general and typical and no one, including ourselves, has studied them before. The configuration covers any type of IT2 input fuzzy sets, arbitrary TS fuzzy rules, Zadeh AND operator, the KM center-of-sets type reducer, and the centroid defuzzifier. We start Section IV with comparing the analytical structures of the IT2 TS fuzzy controllers derived using the new method with those of the comparable IT2 Mamdani fuzzy controllers that we attained before. We then mathematically prove that one subset of the IT2 TS controllers in Section II becomes linear controller and another subset piecewise linear controller when footprints of uncertainty of all these fuzzy controllers are at their maximum. To our knowledge, these are the first results that systematically link a footprint to an analytical structure and expose how the analytical structure is impacted by different levels of a footprint. We discuss how the new findings may be used to guide the process of choosing value of a footprint so that this design step becomes less challenging. In Section V, the conclusions are drawn.

II. Configuration of Class of Typical IT2 TS Fuzzy Controllers

IT2 TS fuzzy controllers in this paper have two input variables, \( x_1(n) \) and \( x_2(n) \), and one output variable, \( \Delta u(n) \), which is change of controller output [i.e., \( \Delta u(n) = u(n) - u(n-1) \)], where \( n \) represents a sampling instance. For simplicity, \( x_1 \) and \( x_2 \) will be used instead of \( x_1(n) \) and \( x_2(n) \). We suppose that \( x_1 \) is in \([R_1, Q_1]\) that is divided into \( N_1 - 1 \) subintervals: \([S_1, S_2], \ldots, [S_{i-1}, S_i], \ldots, [S_{N_1-1}, S_{N_1}]\), \( 1 \leq i \leq N_1 \). There are \( N_1 \) IT2 fuzzy sets, \( A_1, A_2, \ldots, A_{N_1} \). \( A_i \) is defined over \([S_{i-1}, S_i]\) and its membership value is zero outside the interval. The upper and lower primary membership functions of \( A_i \) are designated as \( \tilde{\mu}_{A_i}(x_1) \) and \( \underline{\mu}_{A_i}(x_1) \), respectively.

![Fig. 1. Example IT2 fuzzy sets for \( x_1 \).](image)

We assume that: 1) \( 0 \leq \underline{\mu}_{A_i}(x_1) \leq \tilde{\mu}_{A_i}(x_1) \leq 1 \) and 2) \( \tilde{\mu}_{A_i}(x_1) \) and \( \underline{\mu}_{A_i}(x_1) \) is a function that increases from a small value (e.g., 0), reaches its maximum (can be a range of the same maximum), and then decreases to another small value or 0. To illustrate these assumptions, Fig. 1 provides some example fuzzy sets. Likewise, we suppose \( x_2 \) is in \([R_2, Q_2]\) that is divided into \( N_2 - 1 \) subintervals, namely \([M_1, M_2], \ldots, [M_{j-1}, M_j], \ldots, [M_{N_2-1}, M_{N_2}]\), over which \( N_2 \) IT2 fuzzy sets meeting the same two assumptions above are defined. Each of them is denoted \( B_j(1 \leq j \leq N_2) \) whose upper and lower primary membership functions are \( \tilde{\mu}_{B_j}(x_2) \) and \( \underline{\mu}_{B_j}(x_2) \), respectively.

A total of \( N_1 \times N_2 \) TS fuzzy rules are used to cover all possible combinations of the input fuzzy sets. Because of the way the fuzzy sets are defined, at any sampling instance, only two adjacent \( A_i \) and two adjacent \( B_j \) will be involved in the fuzzification of \( x_1 \) and \( x_2 \), respectively. Without losing generality, assume \( A_i, A_{i+1}, B_j \), and \( B_{j+1} \) to be the ones that are involved. That means the following four rules will be executed:

\[
\text{IF } x_1 \text{ is } A_i \text{ AND } x_2 \text{ is } B_j \\
\text{THEN } \Delta u(n) = d_{i,j} x_1 + e_{i,j} x_2 + c_{i,j} \quad \text{(Rule 1)}
\]

\[
\text{IF } x_1 \text{ is } A_i \text{ AND } x_2 \text{ is } B_{j+1} \\
\text{THEN } \Delta u(n) = d_{i,j+1} x_1 + e_{i,j+1} x_2 + c_{i,j+1} \quad \text{(Rule 2)}
\]

\[
\text{IF } x_1 \text{ is } A_{i+1} \text{ AND } x_2 \text{ is } B_{j+1} \\
\text{THEN } \Delta u(n) = d_{i+1,j} x_1 + e_{i+1,j} x_2 + c_{i+1,j} \quad \text{(Rule 3)}
\]

\[
\text{IF } x_1 \text{ is } A_{i+1} \text{ AND } x_2 \text{ is } B_j \\
\text{THEN } \Delta u(n) = d_{i+1,j+1} x_1 + e_{i+1,j+1} x_2 + c_{i+1,j+1} \quad \text{(Rule 4)}
\]

where Zadeh fuzzy AND operator [i.e., \( \min() \)] is used and all the coefficients in the rule consequents, \( d_{i,j}, e_{i,j} \), and \( c_{i,j} \), are constants that are indexed in a way that mathematically links them to the input fuzzy sets in the rule premises.

Firing interval for rule \( k (k = 1, 2, 3, 4) \), denoted \([L_k(x_1, x_2), U_k(x_1, x_2)]\), is calculated as follows [23]:

\[
f_1(x_1, x_2) = \left[ L_1(x_1, x_2), U_1(x_1, x_2) \right] = \left[ \min(\tilde{\mu}_{A_i}(x_1), \tilde{\mu}_{B_j}(x_2)), \min(\tilde{\mu}_{A_i}(x_1), \tilde{\mu}_{B_j}(x_2)) \right]
\]

\[
f_2(x_1, x_2) = \left[ L_2(x_1, x_2), U_2(x_1, x_2) \right] = \left[ \min(\tilde{\mu}_{A_i}(x_1), \tilde{\mu}_{B_{j+1}}(x_2)), \min(\tilde{\mu}_{A_i}(x_1), \tilde{\mu}_{B_{j+1}}(x_2)) \right]
\]
\[ f_3(x_1, x_2) = \left[ f_3^u(x_1, x_2), f_3^l(x_1, x_2) \right] = \left[ \min(\mu_{A_{i+1}}(x_1), \mu_{B_j}(x_2)), \min(\mu_{A_{i+1}}(x_1), \mu_{B_j}(x_2)) \right] \]  

(3)

\[ f_4(x_1, x_2) = \left[ f_4^u(x_1, x_2), f_4^l(x_1, x_2) \right] = \left[ \min(\mu_{A_{i+1}}(x_1), \mu_{B_j}(x_2)), \min(\mu_{A_{i+1}}(x_1), \mu_{B_j}(x_2)) \right] \]  

(4)

The iterative KM center-of-sets type reducer [23] is employed to link the firing intervals to the rule consequents to produce \( \Delta u(n) = [\Delta u_L(n), \Delta u_R(n)] \), which is an interval set (i.e., a special kind of T1 fuzzy set). The KM type reducer requires arranging the consequent values in an ascending order. Denoting \( \Delta u(n) \) generated by rule \( k \) \( \Delta u_k(n) \). Without loss of generality, assume that the arrangement result is \( \Delta u_1^L(n) \leq \Delta u_2^L(n) \leq \Delta u_3^L(n) \leq \Delta u_4^L(n) \) [note that \( \Delta u_k^L(n) \) does not necessarily correspond to \( \Delta u_k(n) \)]. One then needs to arrange \( f_k^u(x_1, x_2) \) according to \( \Delta u_k^L(n) \leq \Delta u_k^U(n) \leq \Delta u_k^U(n) \), resulting in \( f_1^u(x_1, x_2) \), \( f_2^u(x_1, x_2) \), \( f_3^u(x_1, x_2) \), \( f_4^u(x_1, x_2) \), and \( f_5^u(x_1, x_2) \), \( f_6^u(x_1, x_2) \), \( f_7^u(x_1, x_2) \), \( f_8^u(x_1, x_2) \), respectively.

The next step is to apply the following iterative procedure to obtain \( [\Delta u_L(n), \Delta u_R(n)] \) [23].

1) Compute \( \Delta u_R(n) = \left( \sum_{k=1}^{M} f_k^u(x_1, x_2) \Delta u_k^R(n) \right) / \left( \sum_{k=1}^{M} f_k^u(x_1, x_2) \right) \) by letting \( f_k^u(x_1, x_2) = (f_k^u(x_1, x_2) + f_k^u(x_1, x_2)/2) \) for \( k = 1, \ldots, M \) (\( M = 4 \) in this paper because four rules are fired a time). Afterward, let intermediate variable \( \Delta u_k^R(n) \) equal to \( \Delta u_R(n) \).

2) Find integer \( P_R \), where \( 1 \leq P_R \leq M - 1 \), such that \( \Delta u_{P_R}^R(n) \leq \Delta u_R(n) \leq \Delta u_{P_R+1}^R(n) \).

3) Compute \( \Delta u_R(n) = \left( \sum_{k=1}^{P_R} f_k^u(x_1, x_2) \Delta u_k^R(n) \right) / \left( \sum_{k=1}^{P_R} f_k^u(x_1, x_2) \right) \) again with \( f_k^u(x_1, x_2) = f_k^u(x_1, x_2) \) for \( k \leq P_R \) and \( f_k^u(x_1, x_2) = f_k^u(x_1, x_2) \) for \( k > P_R \). Then, let \( \Delta u_k^R(n) = \Delta u_R(n) \) where \( \Delta u_k^R(n) \) is another intermediate variable.

4) If \( \Delta u_R(n) \neq \Delta u_R(n) \), let \( \Delta u_R(n) = \Delta u_R(n) \) and go back to step 3). Otherwise, let \( \Delta u_R(n) = \Delta u_R(n) \) and \( \Delta u_R(n) \) is the sought result.

The iterative procedure for computing \( \Delta u_L(n) \) is the same as that for \( \Delta u_R(n) \) above except 1) subscript \( R \) is replaced by \( L \) and 2) let \( f_k^l(x_1, x_2) = f_k^l(x_1, x_2) \) for \( k \leq P_L \) and \( f_k^l(x_1, x_2) = f_k^l(x_1, x_2) \) for \( k > P_L \) in step 3). In summary, \( \Delta u_L(n) \) and \( \Delta u_R(n) \) are calculated by

\[ \Delta u_R(n) = \sum_{k=1}^{P_R} f_k^u(x_1, x_2) \Delta u_k^R(n) + \sum_{k=P_R+1}^{M} f_k^u(x_1, x_2) \Delta u_k^R(n) \] 

\[ \sum_{k=1}^{P_R} f_k^u(x_1, x_2) + \sum_{k=P_R+1}^{M} f_k^u(x_1, x_2) \]  

(5)

\[ \Delta u_L(n) = \sum_{k=1}^{P_L} f_k^l(x_1, x_2) \Delta u_k^L(n) + \sum_{k=P_L+1}^{M} f_k^l(x_1, x_2) \Delta u_k^L(n) \] 

\[ \sum_{k=1}^{P_L} f_k^l(x_1, x_2) + \sum_{k=P_L+1}^{M} f_k^l(x_1, x_2) \]  

(6)

where integers \( P_R (1 \leq P_R \leq 3) \) and \( P_L (1 \leq P_L \leq 3) \) are switching points whose values depend on the input fuzzy sets, rules consequents, and values of \( x_1 \) and \( x_2 \), and hence vary with \( n \).

Finally, the centroid defuzzifier is used to reduce the interval to a number [23]

\[ \Delta u(n) = \frac{1}{2} (\Delta u_L(n) + \Delta u_R(n)). \]  

(7)

III. Novel Technique for Deriving the Analytical Structures of the IT2 TS Fuzzy Controllers

To derive the explicit mathematical expression for \( \Delta u(n) \) in (7), one must first determine the membership value for each of the four rules involved in the min() operation in (1)–(4). Importantly, the membership value must be a mathematical expression rather than a numerical number because we are seeking the input–output relation in mathematical expression as opposed to a series of paired input–output values [34]. The only way to achieve this goal is to divide the 2-D input space spanned by \( x_1 \) and \( x_2 \) into a number of regions, each of which is called the input combination (IC) so that in each region, the membership value of \( x_1 \) is always larger (or smaller) than that of \( x_2 \) [33, 34]. Therefore, how to divide the input space into ICs is a key step. The dividing process must consider three different factors simultaneously: 1) the input fuzzy sets; 2) the rule consequents; and 3) the type reducer. This can be achieved by first considering one factor at a time and then superimposing the resulting space divisions due to each factor to form an overall and final space division. Below, we will use a relatively simple IT2 TS fuzzy controller to illustrate how our new technique achieves all of these divisions.

A. Divide the Input Space by Considering the Input Fuzzy Sets Only

Without loss of generality, we suppose that the universe of discourse of \( x_1 \) of the example controller is \([-9, 9]\) and four triangular IT2 fuzzy sets, \( A_1, A_2, A_3, \) and \( A_4 \) are used [Fig. 2(a)]. The symmetrical triangular type in Fig. 2 makes presentation clearer and more concise. But our method works for any type of fuzzy sets (Fig. 13 shows another example). Likewise, \( x_2 \) is in \([-6, 6]\) and four triangular IT2 fuzzy sets, \( B_1, B_2, B_3, \) and \( B_4 \), are assumed [Fig. 2(b)]. For illustration, we will only derive the analytical structure for the input space covered by \(-3 \leq x_1 \leq 3 \) and \(-2 \leq x_2 \leq 2 \). The analytical structures for the rest of the input space areas can be obtained in a similar fashion. Note that only \( A_2, A_3, B_2, \) and \( B_3 \) (hence \( i = 2, 3 \) and \( j = 2, 3 \)) are involved in the execution of the four TS fuzzy rules. The mathematical definitions for parts of \( A_2, A_3, B_2, \) and \( B_3 \) needed in the derivation are listed in Table I.

The next step is to find the input space division with regard to the determination of the eight interval terminal points in (1)–(4). There are eight different divisions, one for each of the eight terminal points. Fig. 3 provides one division as an example. It shows the ICs for \( f_4^u(x_1, x_2) = \min(\mu_{A_{i+1}}(x_1), \mu_{B_j}(x_2)) \) from rule 1. There are two regions and they are labeled as \( f_1-IC1 \) and \( f_1-IC2 \). In \( f_1-IC1 \),
These eight divisions are for the input fuzzy sets in each individual fuzzy rule. Because all the four rules are simultaneously used in the calculation of $\Delta u(n)$, the divisions must be simultaneously considered. This amounts to superimposing the eight space divisions to form an overall input space division (Fig. 4), leading to four ICs in this particular case.

To differentiate these ICs from the ICs generated later in this section when considering the rule consequents and the type reducer, we call each of the ICs here the input-fuzzy-sets-related IC, or input-IC for short. Because the primary membership functions of all the input fuzzy sets are the symmetrical triangles, the number of ICs is relatively small.

For each IC in Fig. 4, we can decide $f_k(x_1, x_2)$ (i.e., $[f_1(x_1, x_2), \bar{f}_1(x_1, x_2)]$). For example, for input-IC4, $f_4(x_1, x_2) = \mu_{A_2}(x_1), \bar{f}_4(x_1, x_2) = \mu_{B_2}(x_1), \tilde{f}_4(x_1, x_2) = \mu_{B_3}(x_2), \bar{f}_4(x_1, x_2) = \bar{\mu}_{B_2}(x_1), \tilde{\mu}_4(x_1, x_2) = \mu_{B_3}(x_2),$ and $\tilde{\mu}_4(x_1, x_2) = \bar{\mu}_{B_3}(x_2).$ This process needs to be carried out for all the four input-ICs. The outcomes for rules 1 and 2 are listed in Table II as example.

TABLE I
MATHEMATICAL DEFINITIONS FOR PARTS OF $A_2, A_3, B_2,$ AND $B_3$ THAT ARE USED IN THE ANALYTICAL STRUCTURE DERIVATION

<table>
<thead>
<tr>
<th>Definition</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{\mu}_{A_2}(x_1) = -0.133x_1 + 0.6$</td>
<td>$x_1 \in [-3, 3]$</td>
</tr>
<tr>
<td>$\mu_{A_2}(x_1) = -0.133x_1 + 0.4$</td>
<td>$x_1 \in [-3, 3]$</td>
</tr>
<tr>
<td>$\bar{\mu}_{B_2}(x_1) = 0.133x_1 + 0.6$</td>
<td>$x_1 \in [-3, 3]$</td>
</tr>
<tr>
<td>$\mu_{B_2}(x_1) = 0.133x_1 + 0.4$</td>
<td>$x_1 \in [-3, 3]$</td>
</tr>
<tr>
<td>$\bar{\mu}_{B_2}(x_2) = -0.2x_2 + 0.6$</td>
<td>$x_2 \in [-2, 2]$</td>
</tr>
<tr>
<td>$\mu_{B_2}(x_2) = 0.2x_2 + 0.4$</td>
<td>$x_2 \in [-2, 2]$</td>
</tr>
<tr>
<td>$\bar{\mu}_{B_3}(x_2) = 0.2x_2 + 0.4$</td>
<td>$x_2 \in [-2, 2]$</td>
</tr>
</tbody>
</table>

$\mu_{A_2}(x_1) \geq \mu_{B_3}(x_3),$ thus $f_1(x_1, x_2) = \mu_{B_3}(x_2)$ and in $f_1-IC2$ it is the other way around. Notice the notations—each IC label is followed by the resulting membership function for the IC [e.g., $f_1-IC2(\mu_{A_2}(x_1))]$. The boundary between the two ICs in Fig. 3 is the solution of the equation $\mu_{A_2}(x_1) = \mu_{B_2}(x_2)$, which is $x_1 = (3/2)x_2$.

TABLE II
UPPER AND LOWER LIMITS OF THE FIRING INTERVALS FOR THE FOUR INPUT-ICS IN FIG. 4

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>$f_1(x_1, x_2)$</th>
<th>$\bar{f}_1(x_1, x_2)$</th>
<th>$\mu_{A_2}(x_1)$</th>
<th>$\bar{\mu}_{A_2}(x_1)$</th>
<th>$\mu_{B_2}(x_1)$</th>
<th>$\bar{\mu}_{B_2}(x_1)$</th>
<th>$\mu_{B_3}(x_1)$</th>
<th>$\bar{\mu}_{B_3}(x_1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 2</td>
<td>$f_2(x_1, x_2)$</td>
<td>$\bar{f}_2(x_1, x_2)$</td>
<td>$\mu_{A_2}(x_1)$</td>
<td>$\bar{\mu}_{A_2}(x_1)$</td>
<td>$\mu_{B_2}(x_1)$</td>
<td>$\bar{\mu}_{B_2}(x_1)$</td>
<td>$\mu_{B_3}(x_1)$</td>
<td>$\bar{\mu}_{B_3}(x_1)$</td>
</tr>
</tbody>
</table>
B. Divide the Input Space by Considering the Input Fuzzy Sets and Rule Consequents at the Same Time

For every input-IC, the value of \( \Delta u(n) \) for the four rule consequents varies with \( x_1 \) and \( x_2 \). This means whether or not the ascending order \( \Delta u_1^*(n) \leq \Delta u_2^*(n) \leq \Delta u_3^*(n) \leq \Delta u_4^*(n) \) holds depends on the variable values. As a result, \( \Delta u_L(n) \) and \( \Delta u_R(n) \) cannot be determined if the variable values are not specified. This problem is unique to a TS rule and is indeed irrelevant to a Mamdani rule. It has not been studied by other researchers in the literature before. An innovative solution is required so that the input space could be divided into regions in such a manner that in each region the ascending order would be maintained despite variations of \( x_1 \) or \( x_2 \). We call such a region the rule-consequents-related IC or rule-IC for short.

We show our approach to resolving this issue. Like before, it suffices to explain it using the example input space \([-3, 3] \times [-2, 2]\). Suppose that following four rule consequents are used: \( \Delta u_1(n) = 5 + 3x_1 + 4x_2 \), \( \Delta u_2(n) = 1 + 2x_1 - 3x_2 \), \( \Delta u_3(n) = 1 - 2x_1 + 3x_2 \), and \( \Delta u_4(n) = 5 - 3x_1 - 4x_2 \). The boundaries of the rule-ICs are obtained by first letting two of the four rule consequents equal and then solving the resulting equation. There are a total of six such equations. For example, solving the equation involving rules 1 and 2

\[
5 + 3x_1 + 4x_2 = 1 + 2x_1 - 3x_2
\]

gives the boundary

\[
x_1 + 7x_2 + 4 = 0
\]

which divides the space into two rule-ICs. In one IC, \( \Delta u_1(n) \geq \Delta u_2(n) \) and in the other IC, the opposite is true. On the boundary, they are equal. By the same token, the other five boundaries can be attained. We name the boundary involving rules \( p \) and \( q \) \( L_{pq} \). The six boundaries are shown in Fig. 5, which happen to be lines dividing the input space into 16 rule-ICs. The sorted orders of the rule consequents for select ICs are provided in Table III. Take rule-IC1 as an example, the order is \( \Delta u_2(n) \leq \Delta u_1(n) \leq \Delta u_4(n) \leq \Delta u_3(n) \), which means \( \Delta u_1^*(n) = \Delta u_2(n), \Delta u_2^*(n) = \Delta u_1(n), \Delta u_3^*(n) = \Delta u_4(n), \) and \( \Delta u_4^*(n) = \Delta u_3(n) \).

Both the input-ICs and the rule-ICs must be taken into account simultaneously, which means the superimposition of these two different types of ICs. Superimposing Figs. 4 and 5 will result in 22 ICs of a new type (Fig. 6), which is termed the input-rule-IC. In an input-rule-IC, the criteria on the input fuzzy sets and the rule consequents are met simultaneously, thus \( f_k^*(x_1, x_2), f_l^*(x_1, x_2) \) and \( \Delta u_k^*(n) \) can all be determined. For each input-rule-IC, putting the eight membership functions resulted from the \( \min() \) operations in the four rules (Table II) and the related rules consequents \( \Delta u_k^*(n) \) into the type reducer in (5) and (6), one will obtain \( \Delta u_R(n) \) and \( \Delta u_L(n) \) if \( P_L \) and \( P_R \) are known.

C. Divide the Input Space by Considering the Type Reducer Only

The exact values of integers \( P_L \) and \( P_R \) in the type reducer can be computed only if \( x_1 \) and \( x_2 \), all the parameter values of the input fuzzy sets, and all the rule consequent coefficients are numerically available. For the example fuzzy controller, the values of \( P_L \) and \( P_R \) vary with \( x_1 \) and \( x_2 \) and range from 1 to 3. Hence there are a total of \( 3 \times 3 = 9 \) possible combinations of \( P_L \) and \( P_R \) values. We call each of them the case and they are numbered (Table IV). A point in the input space belongs to a case and the connective points of the same case form a new type of IC, called the case-IC. We now show how to determine them.

In the type reducer, when \( \Delta u_R(n) \leq \Delta u_k^*(n) \), \( P_R = 1 \); when \( \Delta u_2^*(n) \leq \Delta u_R(n) \leq \Delta u_3^*(n) \), \( P_R = 2 \); when \( \Delta u_4^*(n) \geq \Delta u_R(n) \), \( P_R = 3 \). Therefore, solving \( \Delta u_R(n) = \Delta u_k^*(n) \) and \( \Delta u_R(n) = \Delta u_k^*(n) \) will generate two curves related to \( P_R \). For convenience, let us name them \( B_1 \) and \( B_2 \), respectively. Similarly, curves \( B_3 \) and \( B_4 \) are for \( P_L \). They are obtained.

![Fig. 5. Dividing the input space into the rule-ICs.](image)

![Fig. 6. Input-rule-ICs produced after Figs. 4 and 5 are superimposed, which are numbered from 1 to 22.](image)

<table>
<thead>
<tr>
<th>Sorted order of the rule consequent</th>
<th>Sorted order of the rule consequent</th>
</tr>
</thead>
<tbody>
<tr>
<td>rule-IC1</td>
<td>( \Delta u_k(n) \leq \Delta u_L(n) \leq \Delta u_R(n) \leq \Delta u_4(n) )</td>
</tr>
<tr>
<td>rule-IC3</td>
<td>( \Delta u_k(n) \leq \Delta u_L(n) \leq \Delta u_R(n) \leq \Delta u_4(n) )</td>
</tr>
<tr>
<td>rule-IC13</td>
<td>( \Delta u_k(n) \leq \Delta u_L(n) \leq \Delta u_R(n) \leq \Delta u_4(n) )</td>
</tr>
</tbody>
</table>
respectively, by solving the equations $\Delta u_L(n) = \Delta u_2^d(n)$ and $\Delta u_U(n) = \Delta u_3^d(n)$. Let us use input-rule-IC4 in Fig. 6 (the gray region labeled 4) as an example to explain. This input-rule-IC is resulted from superimposing input-IC2 in Fig. 4 and rule-IC3 in Fig. 5. From Table III, we know that $\Delta u_1^d(n) = \Delta u_2^d(n)$, $\Delta u_3^d(n) = \Delta u_4^d(n)$, and $\Delta u_5^d(n) = \Delta u_6^d(n)$ for input-rule-IC4. The type reducer requires that $\mu_{A_L}(x_1)$ and $\mu_{A_R}(x_2)$ be arranged to correspond to $\Delta u_2^d(n) \leq \Delta u_3^d(n) \leq \Delta u_6^d(n)$. Subsequently, $f_{\mu_1}(x_1, x_2)$ and $f_{\mu_2}(x_1, x_2)$ in the type reducer can be obtained as follows: $f_{\mu_1}(x_1, x_2) = f_{\mu_1}(x_1, x_2) = \mu_{A_L}(x_1)$, $f_{\mu_2}(x_1, x_2) = f_{\mu_2}(x_1, x_2) = \mu_{A_R}(x_2)$, $f_{\mu_1}(x_1, x_2) = f_{\mu_1}(x_1, x_2) = \mu_{B_L}(x_1)$, $f_{\mu_2}(x_1, x_2) = f_{\mu_2}(x_1, x_2) = \mu_{B_R}(x_2)$. Putting these terminal points and their associated rule conse- quents into (5) and (6) will produce $\Delta u_R(n)$ and $\Delta u_U(n)$. Then use them to form the following equations: $\Delta u_R(n) = \Delta u_2^d(n)$, $\Delta u_R(n) = \Delta u_3^d(n)$, $\Delta u_4^d(n) = \Delta u_5^d(n)$, and $\Delta u_6^d(n) = \Delta u_7^d(n)$. Their solutions are the four curves for input-rule-IC4

\begin{align*}
B_1 : & \quad 10x_1^2 + 45x_2^2 + 23x_1x_2 - 101x_1 - 153x_2 + 60 = 0 \\
B_2 : & \quad 10x_1^2 + 3x_2^2 + 171x_2 - 71x_1 + 81x_2 + 60 = 0 \\
B_3 : & \quad 10x_1^2 + 45x_1^2 + 23x_1x_2 - 95x_1 - 111x_2 + 60 = 0 \\
B_4 : & \quad 10x_1^2 + 3x_2^2 + 171x_2 - 65x_1 - 123x_2 + 60 = 0
\end{align*}

which are shown in Fig. 7. Curves $B_1$ and $B_2$ divide input-rule-IC4 into three subdivisions, each of which has a different $P_R$[Fig. 7(a)]. Curves $B_3$ and $B_4$ play a similar role in creating the three subdivisions for $P_L$[Fig. 7(b)]. Superimposing Fig. 7(a) and (b) leads to the case-ICs. There are six of them [Fig. 7(c)]. It is obvious that the boundaries of the case-ICs are defined by $B_1$ to $B_4$.

Obtaining the case-ICs manually can be laborious and time consuming. Alternatively, one may write a computer program to generate them numerically with the caveat that their boundary- mathematical expressions will not be known. We now use Fig. 6 as an example to outline the procedure. First, choose a sufficient number of points in $[-3, 3]$ for $x_1$ and in $[-2, 2]$ for $x_2$ to form pairs (e.g., $-3, -2.99, \ldots, 2.99, 3$ for $x_1$ and $-2, -1.99, \ldots, 1.99, 2$ for $x_2$—total $601 \times 401 = 241,001$ pairs of $x_1$ and $x_2$). Each pair will be associated with one case (Table IV). The case numbers for all the pairs can be determined by executing the type reducer. Then, the program will form and plot the case-ICs by finding the points that are connected and have the same case number. Fig. 8 shows the result for Fig. 6 generated by our simple MATLAB program. The entire input space is divided into 33 case-ICs. Obviously, the more points chosen for $x_1$ and $x_2$, the more accurate the case-IC boundaries will be. We point out that because the case depends on the controller parameters and the values of the input variables, not all nine cases will necessarily show up in the case-ICs for a particular controller.

### D. Input Space Division When All the Three Factors Are Considered at the Same Time

Superimposing Figs. 6–8 produces Fig. 9, which shows the division of the input space when the input fuzzy sets, the rule consequents, and the type reducer are simultaneously considered. There are 93 regions in Fig. 9 and we call each of them the final-IC. We have just finished dividing the input space into final-ICs for the purpose of analytically evaluating the min() operations in the fuzzy rules. We are ready to derive the analytical structures of the IT2 fuzzy controllers.
Fuzzy Controllers

E. Deriving the Analytical Structures of the IT2 Fuzzy Controllers

We will continue to use the above example IT2 controller as an example. There will be 93 different analytical structures, one for each of the final-ICs in Fig. 9. Take final-IC82. It is the result of superimposing input-IC in Fig. 4 and rule-IC13 in Fig. 5. According to Tables III and IV, \( \bar{\mu}_A_1(x_1), \bar{\mu}_A_2(x_1), \bar{\mu}_B_1(x_1), \bar{\mu}_B_2(x_1), \mu_3(x_1), \mu_4(x_1), \mu_5(x_1), \mu_6(x_1), \mu_7(x_1), \mu_8(x_1), \mu_9(x_1), \mu_10(x_1) \) are superimposed. The final-ICs are numbered from 1 to 93.

Fig. 9. Generation of the final-ICs after the input-ICs, rule-ICs, and case-ICs are superimposed. The final-ICs are numbered from 1 to 93.

ANALYTICAL STRUCTURES OF SELECT FINAL-ICs SHOWN IN FIG. 9

<table>
<thead>
<tr>
<th>Final-IC</th>
<th>( \Delta u(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>( \frac{10x^2 + 21x^2 + 5x^2 + 14x + 27x - 75 + 10x^2 + 21x^2 + 5x^2 + 11x - 3x - 90}{12x - 54} )</td>
</tr>
<tr>
<td>17</td>
<td>( \frac{10x^2 + 21x^2 + 5x^2 + 20x - 36x - 72}{12x - 54} )</td>
</tr>
<tr>
<td>40</td>
<td>( \frac{10x^2 + 21x^2 + 5x^2 + 10x + 8x - 18x - 102}{-12x - 60} )</td>
</tr>
<tr>
<td>90</td>
<td>( \frac{10x^2 + 21x^2 + 5x^2 + 14x - 27x - 75 + 10x^2 + 21x^2 + 5x^2 + 2x - 9x - 105}{12x - 54} )</td>
</tr>
</tbody>
</table>

This derivation process can be automated by utilizing symbolic software such as MATLAB Symbolic Toolbox or Mathematica. We wrote a Mathematica program to derive the analytical structures for all the 93 final-ICs and found that the resulting structures can be summarized as

\[
\Delta u(n) = \frac{10x^2 + 21x^2 + 5x^2 + 17x - 6x^2 - 87}{12x - 54} + \frac{10x^2 + 21x^2 + 5x^2 + 5x^2 - 39x - 90}{12x - 60}.
\]

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.
of the fractions are the linear functions of \(x_1\) and \(x_2\) (the coefficients of the functions are different though). Nevertheless, there exists a significant difference—the numerators in (8) are the second-order polynomial functions of \(x_1\) and \(x_2\) whereas those in (9) are the linear functions of \(x_1\) and \(x_2\). Clearly, the analytical structure of the IT2 TS fuzzy controller is more complex than that of the IT2 Mamdani fuzzy controller.

If one of the input fuzzy sets is not of the piecewise linear type (i.e., not of the triangle or trapezoid type), the analytical structure involving that fuzzy set will be more complex than (8). For instance, if one of the fuzzy sets is of the Gaussian type for \(x_1\), the natural exponential function with an exponent involving \(x_1\) will appear in the analytical structure, making it more complicated than (8). In general, the piecewise linear IT2 fuzzy sets should be employed if one desires a simpler analytical structure.

\section{Maximal Footprint of Uncertainty Turns Some of the IT2 TS Fuzzy Controllers Into Linear or Piecewise Linear Controllers}

The new structure-deriving technique empowered us to comprehensively understand the role of the footprints of uncertainty in shaping the IT2 controllers’ characteristics. Through two subsets of the IT2 controllers in Section II, namely subsets A and B, we now show that under certain conditions, the larger the footprints, the closer these fuzzy controllers are to linear or piecewise linear controllers. Furthermore, when the footprints reach their maximum, these fuzzy controllers actually become linear or piecewise linear controllers.

\subsection{Configuration of Subset A}

The controllers in subset A all use only two IT2 fuzzy sets, \(A_1\) and \(A_2\), for \(x_1\) and two IT2 fuzzy sets, \(B_1\) and \(B_2\), for \(x_2\), all of which are defined in Fig. 10. The footprints of uncertainty of these fuzzy sets are solely represented and regulated by the parameter \(\theta\). Here, without loss of generality, it is assumed that \(L_1 > L_2\). The controllers use four rules:

\begin{enumerate}
  \item IF \(x_1\) is \(A_1\) AND \(x_2\) is \(B_1\) THEN \(\Delta u(n) = \beta_1(x_1 + x_2 + \delta)\)
  \item IF \(x_1\) is \(A_1\) AND \(x_2\) is \(B_2\) THEN \(\Delta u(n) = \beta_2(x_1 + x_2 + \delta)\)
  \item IF \(x_1\) is \(A_2\) AND \(x_2\) is \(B_1\) THEN \(\Delta u(n) = \beta_3(x_1 + x_2 + \delta)\)
  \item IF \(x_1\) is \(A_2\) AND \(x_2\) is \(B_2\) THEN \(\Delta u(n) = \beta_4(x_1 + x_2 + \delta)\).
\end{enumerate}

\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\(x_1\) & \(x_2\) & \(\Delta u(n)\) & \(\beta_1\) & \(\beta_2\) & \(\beta_3\) & \(\beta_4\) \\
\hline
\hline
\end{tabular}
\caption{Parameter values for Subset A.}
\end{table}

\section{Analytical Structure of Subset A}

The analytical structure of subset A for the final-ICs is then derived. It can be expressed as a nonlinear controller with variable gains in the following form:

\[ \Delta u(n) = K(x_1, x_2)x_1 + K(x_1, x_2)x_2 \]  \hspace{1cm} (10)

where variable gains represented by \(K(x_1, x_2)\) are described in Table VI. We point out that as a special case, (10) represents a nonlinear PI controller in incremental form with variable proportional-gain and integral-gain if \(x_2\) is the error of the system output and \(x_1\) is the change of error of the system output. Alternatively, it can represent a nonlinear PD controller in incremental form with variable proportional-gain and derivative-gain if \(x_2\) is the change of change of error of the
system output and $x_1$ is the change of error of the system output.

3) How the Footprints of Uncertainty Impact the Analytical Structure of Subset A: With the analytical structure available, the role of the footprints of uncertainty of the IT2 fuzzy sets, characterized by $\theta$, can now be clearly seen, making precise analysis possible. According to (10), the role of the footprints is to parameterize the variable gains of the nonlinear controller. At one extreme, when $\theta = 0$ (i.e., zero footprint), subset A degenerates to a T1 fuzzy controller whose analytical structure is a nonlinear controller with variable gains. At the other extreme, if $\theta = 1$ (i.e., the maximum footprint), the analytical structures for all the six different final-ICs become the same as follows:

$$\Delta u(n) = (\beta_1 + \beta_2)x_1 + (\beta_1 + \beta_2)x_2$$

(11)

which represents a linear controller with two equal constant gains of $\beta_1 + \beta_2$. In other words, the variable gains disappear, so does the nonlinearity of subset A as controllers! Any other value of $\theta$ in $[0, 1]$ will preserve the variable gain characteristics as well as the nonlinearity. According to Table VI, the denominator of $K(x_1, x_2)$ approaches 1 whereas the numerator approaches $\beta_1 + \beta_2$ for all the ICs as $\theta$ approaches 1. Therefore, (10) approaches (11) as $\theta$ approaches 1 and the two become identical when $\theta = 1$. This is to say that increasing the footprints of uncertainty makes subset A more resemble the linear controller (11) and ultimately become it when the footprints are at their maximum.

![Fig. 12. Control surface of the IT2 fuzzy controller described by (10) when (a) $\theta = 0$ and (b) $\theta = 0.9$. The other parameter values are $\beta_1 = \beta_4 = 1$, $\beta_2 = \beta_3 = 0.2$, $L_1 = 1$, and $L_2 = 0.8$.](image)

The analysis above serves as a proof. We prefer concise, clear presentation to formality.

We would like to add that when the footprints in Fig. 10 are at their maximum (i.e., $\theta = 1$), the four IT2 fuzzy sets will be in extreme, unusual shapes. More specifically, their lower linear membership functions will become lines that coincide with the respective horizontal axes while their upper linear membership functions will become horizontal lines coinciding, respectively, with the lines whose equations are: Primary Membership $= 1$. Obviously, no one would use the IT2 fuzzy sets in such a radical setting in practice. However, this setting is meaningful and indeed necessary for our theoretical investigation to exhibit what will ultimately happen to the
analytical structure of the IT2 controllers when the footprints reach their maximum.

4) Generalizing the Analysis to the IT2 TS Fuzzy Controllers in Subset B: Unlike subset A, subset B uses more than two IT2 fuzzy sets for the two input variables.

As stated in Section II, at any sampling moment, only two adjacent triangular $A_i$ and two adjacent triangular $B_j$ will be relevant to the fuzzification of $x_1$ and $x_2$, respectively. Let us suppose $A_i$ and $A_{i+1}$ are on $[S_i, S_{i+1}]$ (Fig. 13) and $B_j$ and $B_{j+1}$ are on $[M_j, M_{j+1}]$. The fuzzy sets can be various kinds of irregular triangles, which is another difference between subsets B and A. There are four simplified TS rules with constraints covering the region $[S_i, S_{i+1}] \times [M_j, M_{j+1}]$

IF $x_1 \in A_i$ AND $x_2 \in B_j$ THEN $\Delta u(n) = \beta_p(x_1 + x_2)$
IF $x_1 \in A_i$ AND $x_2 \in B_{j+1}$ THEN $\Delta u(n) = \beta_{p+1}(x_1 + x_2)$
IF $x_1 \in A_{i+1}$ AND $x_2 \in B_j$ THEN $\Delta u(n) = \beta_p(x_1 + x_2)$
IF $x_1 \in A_{i+1}$ AND $x_2 \in B_{j+1}$ THEN $\Delta u(n) = \beta_{p+1}(x_1 + x_2)$.

In $[S_i, S_{i+1}] \times [M_j, M_{j+1}]$, if a controller in subset B uses the same four IT2 fuzzy sets that are used by subset A, its configuration will be exactly the same as that of subset A. Of course, the mathematical notations will have to differ [e.g., $L_1 = (S_{i+1} - S_i)/2$, $L_2 = (M_{j+1} - M_j)/2$, $\beta_p = \beta_1$, and $\beta_{p+1} = \beta_2$]. Subsequently, the analytical structure and gain variation characteristics of that fuzzy controller will be identical to those given in (10) and Table VI. The fuzzy controller will become the linear controller (11) when the footprints of uncertainty equal to 1. On the other hand, if a controller in subset B employs different amounts of footprint for different fuzzy sets, its gain variation characteristics will be different from those in Table VI. We do not know exactly what the characteristics will look like. Luckily, it is unnecessary to explicitly know them in order to attain our goal. This is because when all the footprints are close to 1, they can be considered almost equal. In fact, we can treat them being equal for our purpose and replace them by $\theta$. As a result, the fuzzy controller’s analytical structure can be considered approximate to the nonlinear controller with variable gains in (10) with modified mathematical notations. Furthermore, when each of the four footprints equals to 1, the fuzzy controller will become the linear controller (11). Moreover, just like subset A, the analytical structure of subset B becomes closer and closer to a linear controller as the footprints become larger and larger. This analysis is for $[S_i, S_{i+1}] \times [M_j, M_{j+1}]$ only. We apply this analysis to the rest of such regions in the input space $[R_1, Q_1] \times [R_2, Q_2]$. The end result is obvious: the controller resembles more and more a piecewise linear controller as footprints increase and eventually becomes it when the footprints are at their maximum. Using this line of analysis as proof, we attain the following.

**Theorem 2:** For the IT2 fuzzy controllers in Section II that use more than two triangular IT2 fuzzy sets for $x_1$ and $x_2$ and the simplified TS fuzzy rules with constraints (i.e., subset B), their analytical structure is a piecewise nonlinear controller with variable gains. The larger the footprints of uncertainty, the more the fuzzy controllers resemble a piecewise linear controller. The fuzzy controllers become the piecewise linear controller when maximal footprints of uncertainty are used.

It is noted that piecewise linear control is a type of nonlinear control, a representative of which is the gain-scheduling control. Because of its relative simplicity in comparison with other nonlinear controllers, piecewise controller began to be studied over 40 years ago [25], [27]. Other than the relative simplicity, piecewise linear controller is not known to be superior to other types of nonlinear controllers. There exists no control literature that suggests a piecewise linear controller to be capable of performing better than other controllers when its input variables are subject to significant uncertainty.

#### C. Discussion

It is known that under certain conditions, some T1 fuzzy controllers can become linear or piecewise controllers [3], [26]. That means the IT2 controllers whose configurations degenerate to the T1 controllers in [3] and [26] can become linear or piecewise controllers when their footprints of uncertainty are all 0. Theorems 1 and 2 show that this can happen to IT2 controllers even when the footprint of uncertainty is not zero, which might be a surprising finding for some people as it seems counter-intuitive.

This paper shows that the role of a footprint is dictated by controller configuration and hence may differ from controller to controller. This point can be seen concretely from the example controller used in Section III (Table V) and subset A controllers in Section IV-B (Table VI). Regarding the rules of thumb for determining a footprint value in the literature, one should be cautious when using them because they are established based on empirical studies (i.e., simulation and experiments) and thus may not be applicable outside the original test settings.

Absence of a general rule of thumb, determining a footprint currently relies on blind search through the trial-and-error method, which is widely adopted in the field. Blind searching of a multidimensional parameter space is not only time consuming but incomprehensive with subpar outcome. In light of our results, we propose a knowledge-guided process. As an example, the analytical structure knowledge of subsets A and B can be used to define an appropriate range for a footprint. Assume that a subsets A or B controller is desired because linear and piecewise controllers have attempted and failed to satisfactorily control a nonlinear system. (The reader is reminded that linear and piecewise linear controllers are
effective not only for linear systems, but also for mildly nonlinear systems and linear systems with a moderate time delay.) In this scenario, using a smaller footprint (e.g., \( \theta = 0.1 \) to 0.3) can be beneficial as the fuzzy controller will be less resemble to a linear or piecewise linear controller. The controller will be more nonlinear. After selecting a footprint value in this range, one may fine tune it, if needed, for control performance. As another example, if strong nonlinearity is most important and desirable, a very small footprint value (\( \theta < 0.1 \)) or even zero footprint value (i.e., the T1 controller) should be chosen. In either scenario, the analytical structure information leads to a substantial reduction in developer’s effort and time.

V. CONCLUSION

We present a novel technique for deriving the analytical structure of a class of typical TS fuzzy controllers that employ a wide range of nonlinear IT2 fuzzy sets, including, but not limited to, triangular and trapezoidal sets. The principle of this technique works for other IT2 fuzzy controllers whose configurations are similar to that defined in Section II but have \( n \) (\( n \geq 3 \)) input variables. The analytical structure derivation task will be more challenging, especially when \( n > 3 \), with the main obstacle being dividing the \( n \)-dimensional input space into \( n \)-dimensional final-ICs. Realistically speaking though, most IT2 fuzzy controllers employ two input variables.

We have proven that under certain conditions, two subsets of the IT2 TS fuzzy controllers resemble linear or piecewise linear controllers more and more as the footprints of uncertainty increase. Eventually, they become linear or piecewise controllers when the footprints are at their maximum. One implication of this finding is that a large footprint may not be very desirable. Another is that validity and utility of the footprint design rules of thumb in the literature may be rather limited outside the original settings in which the rules were generated. We discussed how the analytical structure information could help the developer narrow his/her selection of a proper footprint value.

This paper exemplifies the importance of investigating analytical structure of a T2 fuzzy controller because availability of such structure information can lead to comprehensive and precise analysis of a T2 fuzzy control system.

REFERENCES


**Haibo Zhou** (M’11) received the Ph.D. degree in mechanical and electrical engineering from Central South University, Changsha, China, in 2010. He has been an Associate Professor with the School of Mechanical and Electrical Engineering, Central South University, since 2012. From 2007 to 2008, he was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA. His current research interests include type-2 fuzzy systems, and modeling and control of electromechanical systems.

Dr. Zhou was a member of the Competition Task Force of the IEEE Computational Intelligence Society, which was responsible for the competitions at the IEEE International Conference on Fuzzy Systems held in 2009, 2011, and 2013.

**Hao Ying** (S’88–M’90–SM’97–F’12) received the B.S. and M.S. degrees in electrical engineering from Donghua University (formerly, China Textile University), Shanghai, China, in 1982 and 1984, respectively, and the Ph.D. degree in biomedical engineering from the University of Alabama at Birmingham, Birmingham, AL, USA, in 1990.

He is a Professor with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA. He has authored the book entitled *Fuzzy Control and Modeling: Analytical Foundations and Applications* (IEEE Press, 2000) and co-authored the book entitled *Type-2 Fuzzy Logic Control: Introduction to Theory and Applications* (John Wiley & Sons Inc., 2014). He has published over 100 journal papers and 150 conference papers with a Google H-index of 43.

Prof. Ying serves as an Associate Editor or an Editorial Board Member for eight international journals, including the *IEEE TRANSACTIONS ON FUZZY SYSTEMS*. He is on the IEEE Computational Intelligence Society’s Fuzzy Systems Technical Committee and also on the IEEE Systems, Man, and Cybernetics Society’s Fellow Evaluation Committee.