2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

A Method for Deriving the Analytical Structure of the TS Fuzzy Controllers with Two Linear Interval Type-2 Fuzzy Sets for Each Input Variable

Haibo Zhou, Hao Ying

***Abstract*—Type-2 fuzzy controllers have been mostly viewed and treated as black boxes in that their input-output mathematical mappings (i.e., analytical structures) are unknown. In contrast, this is never the case for any conventional controller. In this paper, we show an innovative analytical structure derivation technique for the interval Type-2 TS fuzzy controllers whose configurations are as follows: two input variables, two linear input fuzzy sets for each input variable, linear TS fuzzy rules, Zadeh AND operator, the Karnik-Mendel center-of-sets type reducer, and the centroid defuzzifier. Revealing the analytical structure of any Type-2 fuzzy controller, this one included, is important as it can lead to better understanding of the controller and more productive analysis and design of the Type-2 fuzzy control system.**

I. Introduction

IN the past several years, a growing number of research results on type-2 (T2) fuzzy control have appeared to emerge in the literature [1-7]. Some authors claim that T2 fuzzy control outperforms its counterpart type-1 (T1) fuzzy control (e.g., traditional fuzzy control) by showing better control performance data. Some important questions are in order. Why is this the case? Do the T2 fuzzy controllers work better because their analytical structures are more advantageous? What are their analytical structures then? By "analytical structure," we mean a mathematical expression that precisely describes the input-output relationship of the controller [8, 9]. After the analytical structure becomes available, one can have deeper understanding of the T2 fuzzy controller and how it functions. One can also analyze and design the T2 fuzzy control system better and in the framework of the well-developed conventional (nonlinear) control theory as the fuzzy control problems will be transformed to nonlinear control problems. Many time-tested methods can be taken advantage of for the fuzzy control purpose, resulting in better analysis and design outcomes (e.g., a less conservative stability condition).

In the past two decades, researchers have investigated the analytical structure of various T1 fuzzy controllers [11, 12]. Because a T2 fuzzy controller is always more complicated than its T1 counterpart, deriving the analytical structure of a T2 fuzzy controller is obviously more difficult. So far, a very limited number of papers address the analytical structure of a T2 fuzzy controller [8-10, 13-15], and when doing so, it is about Mamdani T2 fuzzy controller. No paper has focused on the analytical structure of a TS fuzzy controller except [8].

In this paper, we show an innovative analytical structure derivation technique for the interval T2 (IT2) TS fuzzy controllers whose configurations are as follows: two input variables, two linear input fuzzy sets for each input variable, linear TS fuzzy rules, Zadeh AND operator, the Karnik-Mendel center-of-sets type reducer, and the centroid defuzzifier.

II. Configuration of the IT2 TS Fuzzy Controllers

The IT2 TS fuzzy controllers in this study had two input variables, $x_1(n)$ and $x_2(n)$, that were computed from one physical input variable (e.g., error and change of error of the physical variable), where *n* represented the *n*-th sampling instance. They had one output variable, $\triangle u(n)$, which was the change of controller output (i.e., $\Delta u(n) = u(n) - u(n-1)$). For simplicity, x_1, x_2 and $\triangle u$ will be used instead of $x_1(n)$, $x_2(n)$ and $\triangle u(n)$. Suppose that x_1 was defined on $[L_1, R_1]$, and two IT2 fuzzy sets, A_i $(i = 1,2)$, were used to fuzzify x_1. Their upper and lower primary membership functions of A_i are designated as $\overline{\mu}_{A_i}(x_1)$ and $\underline{\mu}_{A_i}(x_1)$, respectively, and their membership values were 0 outside the interval. Inside the interval, it was assumed that (1) $\overline{\mu}_{A_i}(x_1) \geq \underline{\mu}_{A_i}(x_1)$, and (2) $\overline{\mu}_{A_i}(x_1)$ and $\underline{\mu}_{A_i}(x_1)$ were linear functions. Fig. 1 provides two examples. Likewise, x_2 was defined in $[L_2, R_2]$ and was fuzzified by two IT2 fuzzy sets, B_j $(j = 1,2)$, that met the same assumptions above. Their upper and lower primary membership functions are denoted by $\overline{\mu}_{B_j}(x_2)$ and $\underline{\mu}_{B_j}(x_2)$, respectively.

Haibo Zhou is with the State Key Laboratory of High Performance Complex Manufacturing, Central South University, Changsha, Hunan 410083 China (corresponding author; phone: 86-731-88830291; e-mail: csuhaibo@ hotmail.com).

Hao Ying is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, Michigan 48202, USA (e-mail: hao.ying@wayne.edu).

This work was supported in part by the National Natural Science Foundation of China under Grant 51005253, the National Science and Technology Major Project of China under Grant 2012ZX02702006-003, and the Key Laboratory for Precision & Non-traditional Machining of Ministry of Education (Dalian University of Technology) under Grant JMTZ201101.

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

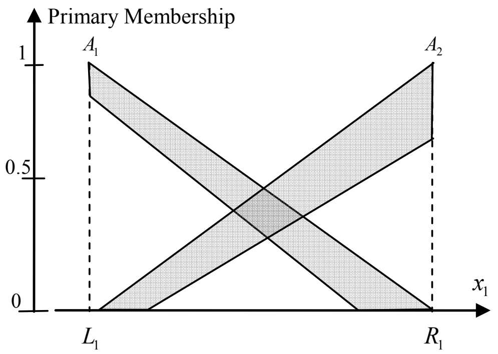

Fig. 1. Two example IT2 fuzzy sets for x_1.

Only four fuzzy rules were needed and they were of the TS type as shown below:

IF x_1 is A_1 AND x_2 is B_1 THEN $\Delta u = d_{1,1}x_1 + e_{1,1}x_2 + c_{1,1}$ (Rule 1)

IF x_1 is A_1 AND x_2 is B_2 THEN $\Delta u = d_{1,2}x_1 + e_{1,2}x_2 + c_{1,2}$ (Rule 2)

IF x_1 is A_2 AND x_2 is B_1 THEN $\Delta u = d_{2,1}x_1 + e_{2,1}x_2 + c_{2,1}$ (Rule 3)

IF x_1 is A_2 AND x_2 is B_2 THEN $\Delta u = d_{2,2}x_1 + e_{2,2}x_2 + c_{2,2}$ (Rule 4)

where Zadeh fuzzy AND operator (i.e., min()) were used and $c_{i,j}$, $d_{i,j}$, and $e_{i,j}$ were constants whose values to be determined by the controller designer.

The firing interval for the IT2 fuzzy sets in Rule k ($k = 1,2,3,4$), denoted by $[\underline{f}_k(x_1,x_2), \overline{f}_k(x_1,x_2)]$, was calculated as follows [16]:

$$f_1(x_1,x_2) = [\underline{f}_1(x_1,x_2), \overline{f}_1(x_1,x_2)] = [\min(\underline{\mu}_{A_1}(x_1), \underline{\mu}_{B_1}(x_2)), \min(\overline{\mu}_{A_1}(x_1), \overline{\mu}_{B_1}(x_2))] \quad (1)$$

$$f_2(x_1,x_2) = [\underline{f}_2(x_1,x_2), \overline{f}_2(x_1,x_2)] = [\min(\underline{\mu}_{A_1}(x_1), \underline{\mu}_{B_2}(x_2)), \min(\overline{\mu}_{A_1}(x_1), \overline{\mu}_{B_2}(x_2))] \quad (2)$$

$$f_3(x_1,x_2) = [\underline{f}_3(x_1,x_2), \overline{f}_3(x_1,x_2)] = [\min(\underline{\mu}_{A_2}(x_1), \underline{\mu}_{B_1}(x_2)), \min(\overline{\mu}_{A_2}(x_1), \overline{\mu}_{B_1}(x_2))] \quad (3)$$

$$f_4(x_1,x_2) = [\underline{f}_4(x_1,x_2), \overline{f}_4(x_1,x_2)] = [\min(\underline{\mu}_{A_2}(x_1), \underline{\mu}_{B_2}(x_2)), \min(\overline{\mu}_{A_2}(x_1), \overline{\mu}_{B_2}(x_2))] \quad (4)$$

Then, the iterative Karnik-Mendel (KM) center-of-sets type reducer [17, 6] was employed to link the firing intervals to the rule consequents to create $\Delta u = [\Delta u_L, \Delta u_R]$, an interval set (a special kind of T1 fuzzy set). Denote the consequent of Rule k Δu_k. The type reducer arranged the four Δu_k in the ascending orders. Without loss of generality, assume that the result was $\Delta u_1^* \le \Delta u_2^* \le \Delta u_3^* \le \Delta u_4^*$ (note that Δu_k^* did not necessarily correspond to Δu_k). One then arranged $\overline{f}_k(x_1,x_2)$ and $\underline{f}_k(x_1,x_2)$ to correspond to $\Delta u_1^* \le \Delta u_2^* \le \Delta u_3^* \le \Delta u_4^*$. Suppose the results were $\overline{f}_1^*(x_1,x_2)$, $\overline{f}_2^*(x_1,x_2)$, $\overline{f}_3^*(x_1,x_2)$, $\overline{f}_4^*(x_1,x_2)$ and $\underline{f}_1^*(x_1,x_2)$, $\underline{f}_2^*(x_1,x_2)$, $\underline{f}_3^*(x_1,x_2)$, $\underline{f}_4^*(x_1,x_2)$, respectively. The terminal points Δu_L and Δu_R were computed [4]:

$$\Delta u_R = \frac{\sum_{k=1}^{P_R} \underline{f}_k^*(x_1,x_2)\Delta u_k^* + \sum_{k=P_R+1}^{4} \overline{f}_k^*(x_1,x_2)\Delta u_k^*}{\sum_{k=1}^{P_R} \underline{f}_k^*(x_1,x_2) + \sum_{k=P_R+1}^{4} \overline{f}_k^*(x_1,x_2)} \quad (5)$$

$$\Delta u_L = \frac{\sum_{k=1}^{P_L} \overline{f}_k^*(x_1,x_2)\Delta u_k^* + \sum_{k=P_L+1}^{4} \underline{f}_k^*(x_1,x_2)\Delta u_k^*}{\sum_{k=1}^{P_L} \overline{f}_k^*(x_1,x_2) + \sum_{k=P_L+1}^{4} \underline{f}_k^*(x_1,x_2)} \quad (6)$$

where integers P_R ($1 \le P_R \le 3$) and P_L ($1 \le P_L \le 3$) were switching points whose values depended on the input fuzzy sets, rules consequents and the values of x_1 and x_2 , and hence would vary with n . Finally, the centroid defuzzifier was used to reduce the interval set to a number [3]:

$$\Delta u = \frac{1}{2}\left(\Delta u_L + \Delta u_R\right) \quad (7)$$

III. A Technique for Deriving the Analytical Structure of the IT2 TS Fuzzy Controllers

To derive the explicit mathematical expression for Δu in equation (7), one must first determine the resulting membership value for each of the four rules involving the min() operation in equations (1)-(4). To achieve this goal the input space spanned by x_1 and x_2 was divided into a number of regions, each of which was called IC (input combination) so that in each IC, the inequality relationship between x_1 and x_2 was exclusive (i.e., one variable's membership value was always smaller than the other) [18]. Furthermore, this space division process had to be applied simultaneously to three different components, namely the input fuzzy sets, the rule consequents, and the type reducer. This was achieved by considering one component a time first and then superimposing the resulting space divisions from each component to form an overall input space division.

A. Input Space Division Concerning Input Fuzzy Sets Only

Without loss of generality, we suppose that the universe of inputs x_1 and x_2 are the same, which are $[-2,2]$. A_1 , A_2 , B_1 and B_2 are shown in Fig. 2. These fuzzy sets are symmetrically triangular. The mathematical definitions for A_1 , A_2 , B_1 and B_2 are listed in Table I. Fig. 3 shows the input space division results when the eight terminal points in equations (1) to (4) are individually evaluated.

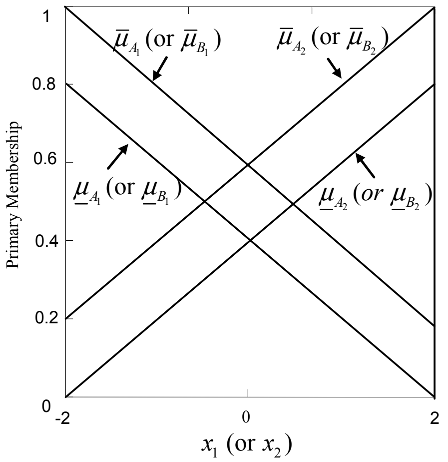

Fig. 2. Example interval T2 fuzzy sets for x_1 and x_2.

TABLE I
THE MATHEMATICAL DEFINITIONS FOR A_1, A_2, B_1 AND B_2.

Definition	Interval
$\bar{\mu}_{A_1}(x_1) = -0.2x_1 + 0.6$	$x_1 \in [-2,2]$
$\underline{\mu}_{A_1}(x_1) = -0.2x_1 + 0.4$	$x_1 \in [-2,2]$
$\bar{\mu}_{A_2}(x_1) = 0.2x_1 + 0.6$	$x_1 \in [-2,2]$
$\underline{\mu}_{A_2}(x_1) = 0.2x_1 + 0.4$	$x_1 \in [-2,2]$
$\bar{\mu}_{B_1}(x_2) = -0.2x_2 + 0.6$	$x_2 \in [-2,2]$
$\underline{\mu}_{B_1}(x_2) = -0.2x_2 + 0.4$	$x_2 \in [-2,2]$
$\bar{\mu}_{B_2}(x_2) = 0.2x_2 + 0.6$	$x_2 \in [-2,2]$
$\underline{\mu}_{B_2}(x_2) = 0.2x_2 + 0.4$	$x_2 \in [-2,2]$

In Fig. 3, two of the eight divisions of the input space $[-2, 2]\times[-2, 2]$ for determining the terminal points in equations (1)-(4) are shown. For instance, Fig. 3(a) shows the ICs for $\underline{f}_1(x_1, x_2) = \min(\underline{\mu}_{A_1}(x_1), \underline{\mu}_{B_1}(x_2))$ from Rule 1, which are $\underline{f}_1$-IC1 and $\underline{f}_1$-IC2. In $\underline{f}_1$-IC2, $\underline{\mu}_{A_1}(x_1) \le \underline{\mu}_{B_1}(x_2)$, thus $\underline{f}_1(x_1, x_2) = \underline{\mu}_{A_1}(x_1)$. Notice the notations - each IC label is followed by its resulting membership value in parentheses (e.g., $\underline{f}_1$-IC2 $\left(\underline{\mu}_{B_1}(x_2)\right)$). There should be eight different divisions because of the eight terminal points. Due to the space limitation, just one more division result (i.e., $\bar{f}_3(x_1, x_2)$) is given (Fig. 3(b)). In Fig. 3(b), boundary A is described by the solution of solving $\underline{\mu}_{A_2}(x_1) = \underline{\mu}_{B_1}(x_2)$: $0.2x_1 + 0.4 = -0.2x_2 + 0.4$, yielding the boundary $x_1 = -x_2$. Once the boundary is determined, just compute the values of $\underline{\mu}_{A_2}(x_1)$ and $\underline{\mu}_{B_1}(x_2)$ using values of any pair of x_1 and x_2 in either IC. The membership function producing the smaller value is the resulting function of min( ) for that IC.

These divisions are for the input fuzzy sets in each individual fuzzy rule only. Because all the four rules are involved in calculating Δu, they must be simultaneously considered. This simultaneous consideration is achieved if we superimpose the eight individual input space divisions to form an overall input space division (Fig. 4). It turns out that there are a total of four ICs, labeled from input-IC1 to input-IC4. To differentiate other ICs generated by considering the other two factors below, we call each of the ICs an input-fuzzy-sets-related IC, or input-IC for short.

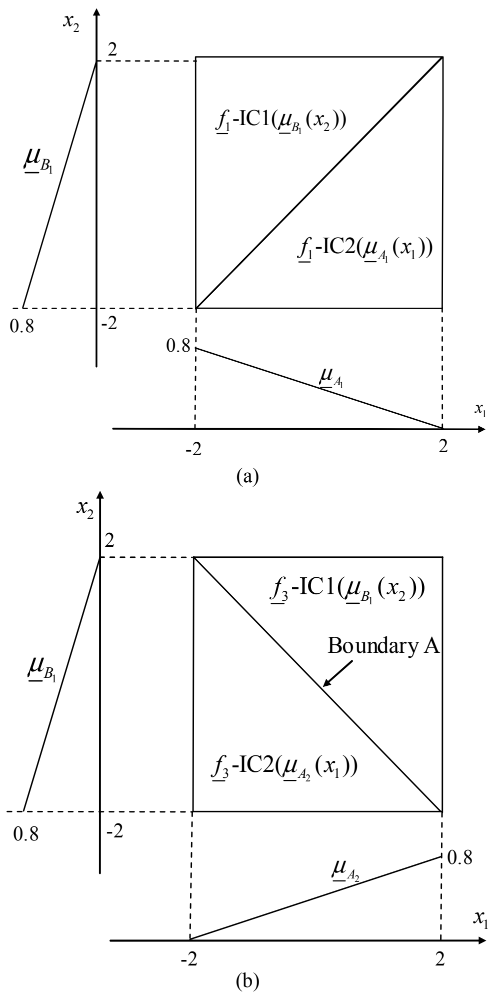

Fig. 3. Two of the eight divisions of the input space $[-2, 2]\times[-2, 2]$ for determining the terminal points in equations (1)-(4): (a) for $\underline{f}_1(x_1, x_2)$ in equation (1), and (b) for $\underline{f}_3(x_1, x_2)$ in equation (3).

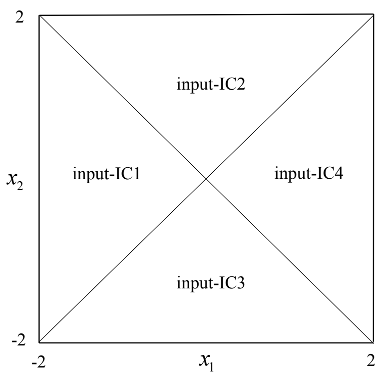

Fig. 4. Superimposing the ICs in the eight individual input space divisions to create an overall input space division, which results in four input-ICs.

Now, for each IC in Fig. 4, we can decide $f_k(x_1,x_2)$ (i.e., $[\underline{f}_k(x_1,x_2),\overline{f}_k(x_1,x_2)]$). For example, for input-IC4, $\underline{f}_1(x_1,x_2)=\underline{\mu}_{A_1}(x_1)$, $\underline{f}_2(x_1,x_2)=\underline{\mu}_{A_1}(x_1)$, $\underline{f}_3(x_1,x_2)=\underline{\mu}_{B_1}(x_2)$, $\underline{f}_4(x_1,x_2)=\underline{\mu}_{B_2}(x_2)$, $\overline{f}_1(x_1,x_2)=\overline{\mu}_{A_1}(x_1)$, $\overline{f}_2(x_1,x_2)=\overline{\mu}_{A_1}(x_1)$, $\overline{f}_3(x_1,x_2)=\overline{\mu}_{B_1}(x_2)$ and $\overline{f}_4(x_1,x_2)=\overline{\mu}_{B_2}(x_2)$. This process is carried out for all the four input-ICs. The results are listed in Table II.

B. Input Spaces Division Concerning Input Fuzzy Sets and Rule Consequents at the Same Time

For every input-IC, values of Δu for the four rule consequents vary with x_1 and x_2. This means whether or not the ascending order $\Delta u_1^* \le \Delta u_2^* \le \Delta u_3^* \le \Delta u_4^*$ holds depends on the values of x_1 and x_2. As a result, Δu_L and Δu_R cannot be determined if the variable values are not specified. This is a new problem that has never been addressed 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 change 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. Suppose that following four linear and symmetrical rule consequents are used: $\Delta u_1 = 5+3x_1+4x_2$, $\Delta u_2 = 1+2x_1-3x_2$, $\Delta u_3 = 1-2x_1+3x_2$ and $\Delta u_4 = 5-3x_1-4x_2$. 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 six of such equations). For example, solving the equation involving Rule 1 and Rule 2:

$$5+3x_1+4x_2 = 1+2x_1-3x_2$$

gives the boundary

$$x_1+7x_2+4=0$$

that divides the space into two rule-ICs. In one IC, Δu of Rule 1 is always larger than that of Rule 2; in the other IC, the opposite is true. On the boundary, they are equal. By the same token, the other five boundaries can be determined. For distinction, the boundary formed by Rule p and Rule q is denoted by L_{pq} . For this specific fuzzy controller, there are six L_{pq} as shown in Fig. 5. All the boundaries are lines owing to the symmetric linear rule consequents. The entire input space is segmented into 16 rule-ICs (i.e., rule-IC1 to rule-IC16), and the sorted order $\Delta u_1^* \le \Delta u_2^* \le \Delta u_3^* \le \Delta u_4^*$ for each IC is different. Table III lists some of the 16 sorted orders of the rule consequents. Take rule-IC1 in Fig. 5 as an example, the sorted order is $\Delta u_2 \le \Delta u_1 \le \Delta u_4 \le \Delta u_3$, which means that $\Delta u_1^* = \Delta u_2$, $\Delta u_2^* = \Delta u_1$, $\Delta u_3^* = \Delta u_4$ and $\Delta u_4^* = \Delta u_3$. Importantly, the number and line-bounded shapes of the ICs are dependent of the rule consequent coefficients.

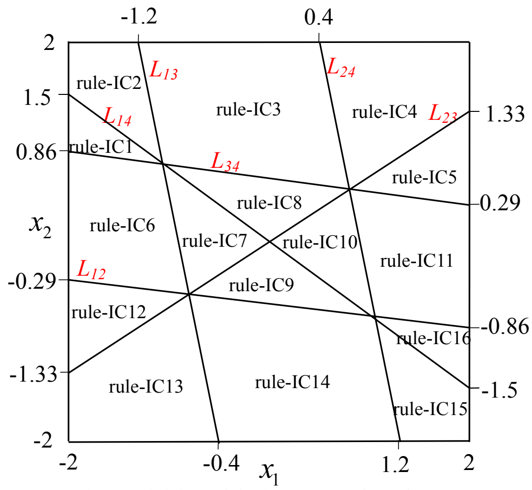

Fig. 5. Division of the input space by rule-ICs.

TABLE III
SORTED ORDERS OF THE RULE CONSEQUENTS FOR SOME OF THE 16 RULE-ICS.

	Sorted orders of the rule consequents
rule-IC1	$\Delta u_2 \le \Delta u_1 \le \Delta u_4 \le \Delta u_3$
rule-IC2	$\Delta u_2 \le \Delta u_4 \le \Delta u_1 \le \Delta u_3$
rule-IC3	$\Delta u_2 \le \Delta u_4 \le \Delta u_3 \le \Delta u_1$
rule-IC4	$\Delta u_4 \le \Delta u_2 \le \Delta u_3 \le \Delta u_1$
rule-IC5	$\Delta u_4 \le \Delta u_3 \le \Delta u_2 \le \Delta u_1$
rule-IC6	$\Delta u_2 \le \Delta u_1 \le \Delta u_3 \le \Delta u_4$
rule-IC7	$\Delta u_2 \le \Delta u_3 \le \Delta u_1 \le \Delta u_4$
rule-IC8	$\Delta u_2 \le \Delta u_3 \le \Delta u_4 \le \Delta u_1$
rule-IC9	$\Delta u_3 \le \Delta u_2 \le \Delta u_1 \le \Delta u_4$
rule-IC10	$\Delta u_3 \le \Delta u_2 \le \Delta u_4 \le \Delta u_1$

TABLE II
THE UPPER AND LOWER LIMITS OF THE FIRING INTERVALS FOR ALL THE INPUT-ICS.

	Rule 1		Rule 2		Rule 3		Rule 4	
	$\underline{f}_1(x_1,x_2)$	$\overline{f}_1(x_1,x_2)$	$\underline{f}_2(x_1,x_2)$	$\overline{f}_2(x_1,x_2)$	$\underline{f}_3(x_1,x_2)$	$\overline{f}_3(x_1,x_2)$	$\underline{f}_4(x_1,x_2)$	$\overline{f}_4(x_1,x_2)$
input-IC1	$\underline{\mu}_{B_1}(x_2)$	$\overline{\mu}_{B_1}(x_2)$	$\underline{\mu}_{B_2}(x_2)$	$\overline{\mu}_{B_2}(x_2)$	$\underline{\mu}_{A_2}(x_1)$	$\overline{\mu}_{A_2}(x_1)$	$\underline{\mu}_{A_2}(x_1)$	$\overline{\mu}_{A_2}(x_1)$
input-IC2	$\underline{\mu}_{B_1}(x_2)$	$\overline{\mu}_{B_1}(x_2)$	$\underline{\mu}_{A_1}(x_1)$	$\overline{\mu}_{A_1}(x_1)$	$\underline{\mu}_{B_1}(x_2)$	$\overline{\mu}_{B_1}(x_2)$	$\underline{\mu}_{A_2}(x_1)$	$\overline{\mu}_{A_2}(x_1)$
input-IC3	$\underline{\mu}_{A_1}(x_1)$	$\overline{\mu}_{A_1}(x_1)$	$\underline{\mu}_{B_2}(x_2)$	$\overline{\mu}_{B_2}(x_2)$	$\underline{\mu}_{A_2}(x_1)$	$\overline{\mu}_{A_2}(x_1)$	$\underline{\mu}_{B_2}(x_2)$	$\overline{\mu}_{B_2}(x_2)$
input-IC4	$\underline{\mu}_{A_1}(x_1)$	$\overline{\mu}_{A_1}(x_1)$	$\underline{\mu}_{A_1}(x_1)$	$\overline{\mu}_{A_1}(x_1)$	$\underline{\mu}_{B_1}(x_2)$	$\overline{\mu}_{B_1}(x_2)$	$\underline{\mu}_{B_2}(x_2)$	$\overline{\mu}_{B_2}(x_2)$

In order to derive the analytical structure of the fuzzy controller, 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 Fig. 4 to Fig. 5 will result in 28 new ICs as shown in Fig. 6, each of which is termed input-rule-IC. In each input-rule-IC, the criteria regarding both the input fuzzy sets and the rule consequents are met simultaneously, thus $\overline{f}_k^*(x_1,x_2)$, $\underline{f}_k^*(x_1,x_2)$ and Δu_k^* can all be determined. Now for each input-rule-IC, putting the eight membership functions resulted from the min() operations in the four rules (Table II) and the related rule consequents Δu_k^* into the type reducer in equations (5) and (6), one will obtain Δu_R and Δu_L if P_L and P_R in equations (5) and (6) are known. We now show how to determine the IC boundaries for different values of P_L and P_R.

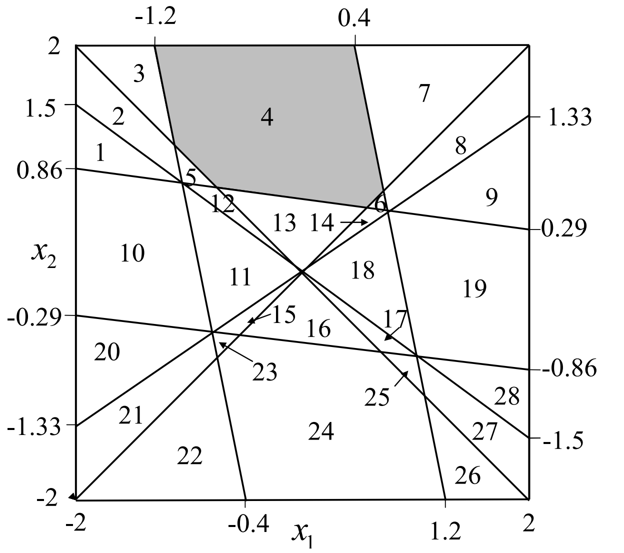

Fig. 6. Twenty-eight input-rule-ICs are produced after Fig. 4 and Fig. 5 are superimposed (only the sequence numbers of the ICs are shown, e.g., 18 means input-rule-IC18).

C. *Input Space Division Concerning P_L and P_R Only*

The exact values of P_L and P_R vary with x_1 and x_2 and they range from 1 to 3. Consequently there are a total of $3\times3=9$ possible situations to consider. We call each of them a case. Table IV defines these nine cases. Any point in the input space is associated with one and only one case. If the boundary of every case is drawn, a new type of IC, called a case-IC, will be created.

We now show how to determine the boundaries of the case-ICs, each of which is called a case boundary. Two case boundaries are due to P_R and another two are caused by P_L. In the iterative KM type reducer, when $\Delta u_R \le \Delta u_2^*$, P_R=1; when $\Delta u_2^* \le \Delta u_R \le \Delta u_3^*$, P_R=2; when $\Delta u_R \ge \Delta u_3^*$, P_R=3. Therefore, solving $\Delta u_R = \Delta u_2^*$ and $\Delta u_R = \Delta u_3^*$ will generate two case boundaries related to P_R. For convenience, let us name them

TABLE IV
DEFINING THE NINE CASES RELATED TO P_L AND P_R.

Case No.	Switching Points
1	P_L=1, P_R=1
2	P_L=1, P_R=2
3	P_L=1, P_R=3
4	P_L=2, P_R=1
5	P_L =2, P_R=2
6	P_L=2, P_R=3
7	P_L=3, P_R=1
8	P_L=3, P_R=2
9	P_L=3, P_R=3

B_1 and B_2. Similarly, there are case boundaries B_3 and B_4 for the three integers of P_L. They are obtained by solving equations $\Delta u_L = \Delta u_2^*$ and $\Delta u_L = \Delta u_3^*$. Note that because Δu_R (or Δu_L) and Δu_k^* are different for different input-rule-ICs, each input-rule-IC has four curves to form its case boundary. The curves divide the input-rule-IC, not the entire input space. For the 28 input-rule-ICs of the input space $[-2,2]\times[-2,2]$ in Fig. 6, there are at most 28×4 = 112 case boundaries. The shape and number of the case-ICs depend on the nature of the equation sets (four equations form a set; two due to P_R and two involve P_L) that generate the case boundaries, which in turn depend on the input fuzzy sets and coefficients of the rule consequents.

Let us use input-rule-IC4 in Fig. 6 (the grey region labeled 4) to explain. Keep in mind that input-rule-IC4 is resulted from superimposing input-IC2 in Fig. 4 and rule-IC3 in Fig. 5. From Table III, we know that $\Delta u_1^* = \Delta u_2$, $\Delta u_2^* = \Delta u_4$, $\Delta u_3^* = \Delta u_3$ and $\Delta u_4^* = \Delta u_1$ for input-rule-IC4. According to the iterative KM type-reducer, one must arranges $\underline{f}_k(x_1,x_2)$ and $\overline{f}_k(x_1,x_2)$ to correspond to $\Delta u_1^* \le \Delta u_2^* \le \Delta u_3^* \le \Delta u_4^*$. Therefore, $\underline{f}_k^*(x_1,x_2)$ and $\overline{f}_k^*(x_1,x_2)$ in the type-reducer can be accordingly obtained: $\underline{f}_1^*(x_1,x_2) = \underline{f}_2(x_1,x_2) = \underline{\mu}_{A_1}(x_1)$, $\underline{f}_2^*(x_1,x_2) = \underline{f}_4(x_1,x_2) = \underline{\mu}_{A_2}(x_1)$, $\underline{f}_3^*(x_1,x_2) = \underline{f}_3(x_1,x_2) = \underline{\mu}_{B_1}(x_2)$, $\underline{f}_4^*(x_1,x_2) = \underline{f}_1(x_1,x_2) = \underline{\mu}_{B_1}(x_2)$, $\overline{f}_1^*(x_1,x_2) = \overline{f}_2(x_1,x_2) = \overline{\mu}_{A_1}(x_1)$, $\overline{f}_2^*(x_1,x_2) = \overline{f}_4(x_1,x_2) = \overline{\mu}_{A_2}(x_1)$, $\overline{f}_3^*(x_1,x_2) = \overline{f}_3(x_1,x_2) = \overline{\mu}_{B_1}(x_2)$ and $\overline{f}_4^*(x_1,x_2) = \overline{f}_1(x_1,x_2) = \overline{\mu}_{B_1}(x_2)$. Putting these firing intervals and the associated rule consequents into equations (5) and (6) and solving $\Delta u_R = \Delta u_2^*$, $\Delta u_R = \Delta u_3^*$, $\Delta u_L = \Delta u_2^*$ and $\Delta u_L = \Delta u_3^*$, the case boundaries for input-rule-IC4 in Fig. 6 can be found as follows:

B_1: $5x_1^2 + 15x_2^2 + 8x_1x_2 - 33x_1 - 37x_2 + 20 = 0$

B_2: $5x_1^2 + x_2^2 + 6x_1x_2 - 28x_1 + 40x_2 - 24 = 0$

B_3: $5x_1^2 + 15x_2^2 + 8x_1x_2 - 35x_1 - 51x_2 + 20 = 0$

B_4: $5x_1^2 + x_2^2 + 6x_1x_2 - 25x_1 + 27x_2 - 20 = 0$

which are shown in Fig. 7. Curves B_1 and B_2 divide input-rule-IC4 into three subdivisions and each subdivision has different P_R. B_3 and B_4 play a similar role by creating three subdivisions for P_L. Superimposing the six subdivisions leads to the case boundaries of six case-ICs for input-rule-IC4. Each case-IC is associated with a case number and in this example, there are case 1, case 2, case 3, case 5, case 6, and case 9, as shown in Fig. 7.

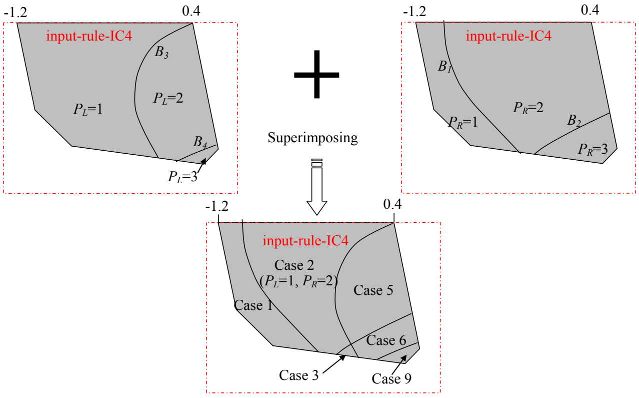

Fig. 7. Boundaries of case-ICs and their associated case numbers (Table IV) for input-rule-IC4 in Fig. 6.

Extending this example to cover all the input-rule-ICs in Fig. 6 will reveal the case boundaries and case numbers for the entire input space. Because there can be up to 112 case boundaries, obtaining them manually is tedious. A computer program can be made to automate this task. We outline the computation procedure for Fig. 6 as follows: (1) choose a number of points in [-2, 2] for x_1 and x_2 (e.g., -2, -1.99,…,1.99, 2 for x_1 and -2, -1.99,…,1.99, 2 for x_2 - total $401\times401=160{,}801$ combinations of x_1 and x_2 values; (2) Each of the combinations is associated with one of the nine pairs of P_L and P_R (i.e., the nine cases defined in Table IV). The case numbers of the 160,801 combinations can be determined by executing the KM iterative type-reducer; (3) the program will then form case boundaries by finding the points that are connected one another and have the same case number. Fig. 8 shows the result of this procedure for Fig. 6. The entire input space is divided into 33 case-ICs. Note that the more points chosen for x_1 and/or x_2, the more accurate the boundaries will be. Superimposing Fig. 6 to Fig. 8 produces Fig. 9, which is the final division of the entire input space. There are 102 regions in Fig. 9 and we call each of such region a final-IC.

D. *Deriving the Analytical Structure of the Fuzzy Controllers*

We continue to use the above specific T2 TS controller as an example. There will be 102 different analytical structures, one for each final-IC. Let's use final-IC53 in Fig. 9 as an example. Final-IC53 is a result of superimposing input-IC3 in Fig. 4 and rule-IC9 in Fig. 5, hence $\underline{f}_1(x_1,x_2)=\underline{\mu}_{A_1}(x_1)$, $\overline{f}_1(x_1,x_2)=\overline{\mu}_{A_1}(x_1)$, $\underline{f}_2(x_1,x_2)=\underline{\mu}_{B_2}(x_2)$, $\overline{f}_2(x_1,x_2)=\overline{\mu}_{B_2}(x_2)$,

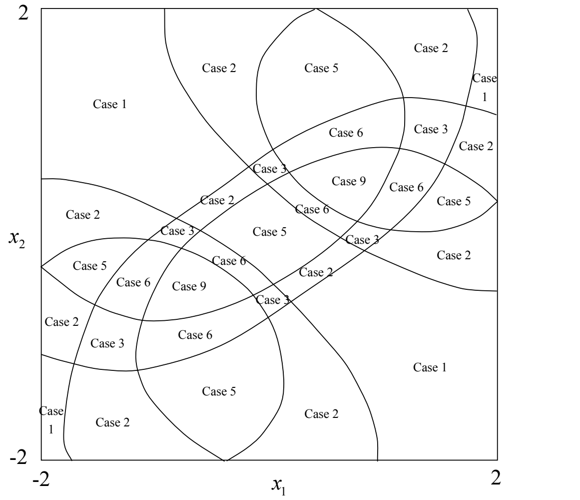

Fig. 8. Thirty-three case-ICs obtained by our MATLAB program for Fig. 6.

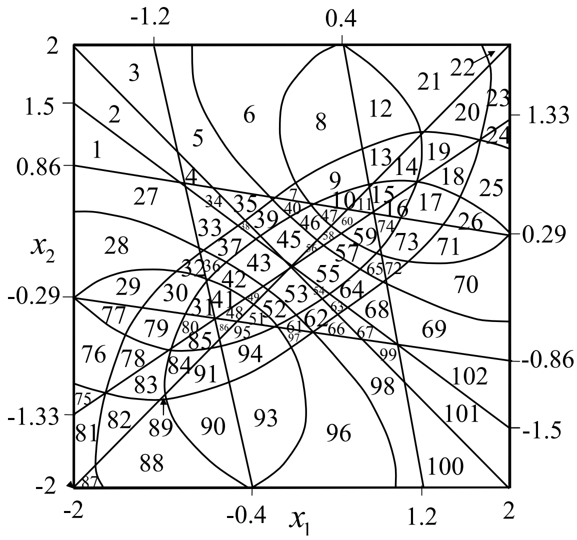

Fig. 9. Result of superimposing all the input-ICs, rule-ICs, and case-ICs, resulting in 102 final-ICs (due to the space limitation, only the sequence numbers of the ICs are shown).

$\underline{f}_3(x_1,x_2)=\underline{\mu}_{A_2}(x_1)$, $\overline{f}_3(x_1,x_2)=\overline{\mu}_{A_2}(x_1)$, $\underline{f}_4(x_1,x_2)=\underline{\mu}_{B_1}(x_2)$, $\overline{f}_4(x_1,x_2)=\overline{\mu}_{B_1}(x_2)$, $\Delta u_1^*=\Delta u_3$, $\Delta u_2^*=\Delta u_2$, $\Delta u_3^*=\Delta u_1$ and $\Delta u_4^*=\Delta u_4$. From Figs. 8 and 9, it can be found that the case number for final-IC53 is 5 (i.e., P_L=2, P_R=2). Using all this information along with Table I and equations (5), (6) and (7), one obtains the analytical structure for final-IC53 after algebraic simplifications:

$$\Delta u=\frac{-5x_1^2-7x_2^2-2x_1x_2-4x_1+6x_2+26}{4x_2+20}+\frac{-5x_1^2-7x_2^2-2x_1x_2-4x_1+6x_2+34}{4x_2+20}$$

This derivation procedure can be automated by using such software as MATLAB Symbolic Toolbox, Mathematica or Maple (we used Mathematica). Table V shows the analytical structures for some select final-ICs in Fig. 9.

TABLE V
THE ANALYTICAL STRUCTURE OF SOME SELECT FINAL-ICS IN FIG. 9.

final-IC No.	Δu
6	$\frac{-5x_1^2-7x_2^2-2x_1x_2+6x_1-9x_2+25}{-4x_2+18}+\frac{-5x_1^2-7x_2^2-2x_1x_2+5x_1+x_2+30}{-4x_2+20}$
21	$\frac{-5x_1^2-7x_2^2-2x_1x_2+x_1-10x_2+29}{-4x_2+18}+\frac{-5x_1^2-7x_2^2-2x_1x_2+5x_1+x_2+30}{-4x_2+20}$
91	$\frac{-5x_1^2-7x_2^2-2x_1x_2-3x_1+13x_2+30}{4x_2+20}+\frac{-5x_1^2-7x_2^2-2x_1x_2-7x_1+2x_2+29}{4x_2+18}$
98	$\frac{-5x_1^2-7x_2^2-2x_1x_2-6x_1+9x_2+25}{4x_2+18}+\frac{-5x_1^2-7x_2^2-2x_1x_2-2x_1+3x_2+35}{4x_2+22}$

IV. CONCLUSION

A novel analytical structure-deriving technique for a class of interval T2 TS fuzzy controllers has been developed. The analytical structures derived will further our understanding on the fuzzy controllers as nonlinear controllers and can be useful in control system design.

REFERENCES

[1] H. Hagras, "Type-2 FLCs: A new generation of fuzzy controllers," *IEEE Computational Intelligence Magazine,* vol. 2, pp. 30-43, 2007.

[2] R. I. John and S. Coupland, "Type-2 fuzzy logic: A historical view," *IEEE Computational Intelligence Magazine,* vol. 2, pp. 57-62, 2007.

[3] N. N. Karnik, J. M. Mendel, and Q. L. Liang, "Type-2 fuzzy logic systems," *IEEE Transactions on Fuzzy Systems,* vol. 7, pp. 643-658, 1999.

[4] Q. L. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Transactions on Fuzzy Systems,* vol. 8, pp. 535-550, 2000.

[5] J. M. Mendel, R. I. John, and F. L. Liu, "Interval type-2 fuzzy logic systems made simple," *IEEE Transactions on Fuzzy Systems,* vol. 14, pp. 808-821, 2006.

[6] C.-S. Chen, "Supervisory Interval Type-2 TSK Neural Fuzzy Network Control for Linear Microstepping Motor Drives With Uncertainty Observer," *IEEE Transactions on Power Electronics,* vol. 26, pp. 2049-2064, Jul.

[7] H. B. Zhou, H. Ying, and J. A. Duan, "Adaptive Control Using Interval Type-2 Fuzzy Logic," *2009 IEEE International Conference on Fuzzy Systems, Vols 1-3,* pp. 836-841, 2009.

[8] H. Zhou and H. Ying, "A Technique for Deriving Analytical Structure of a General Class of Interval Type-2 TS Fuzzy Controllers," *31st Annual Meeting of North American Fuzzy Information Processing Socitey* vol. 2012.

[9] X. Du and H. Ying, "Derivation and Analysis of the Analytical Structures of the Interval Type-2 Fuzzy-PI and PD Controllers," *IEEE Transactions on Fuzzy Systems,* vol. 18, pp. 802-814, Aug 2010.

[10] H. B. Zhou and H. Ying, "A Method for Deriving the Analytical Structure of a Broad Class of Typical Interval Type-2 Mamdani Fuzzy Controllers," *IEEE Transactions on Fuzzy Systems,* vol. 21 pp. 447-458, 2013.

[11] G. K. I. Mann, B. G. Hu, and R. G. Gosine, "Analysis of direct action fuzzy PID controller structures," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics,* vol. 29, pp. 371-388, 1999.

[12] H. Ying, *Fuzzy Control and Modeling: Analytical Foundations and Applications*. Piscataway, NJ: IEEE Press, 2000.

[13] H. B. Zhou and H. Ying, "Deriving the Input-Output Mathematical Relationship for a Class of Interval Type-2 Mamdani Fuzzy Controllers," in *2011 IEEE International Conference on Fuzzy Systems*, Taipei, Taiwan, 2011, pp. 2589-2593.

[14] M. W. Nie and W. W. Tan, "The Derivation of the Analytical Structure of a class of Interval Type-2 Fuzzy PD and PI Controllers," in *2011 IEEE International Conference on Fuzzy Systems*, Taipei, Taiwan, 2011, pp. 635-642.

[15] M. W. Nie and W. W. Tan, "Derivation of the analytical structure of symmetrical IT2 fuzzy PD and PI controllers," in *2010 IEEE International Conference on Fuzzy Systems*, Barcelona, Spain, 2010, pp. 1-8.

[16] J. M. Mendel, "Type-2 fuzzy sets and systems: An overview," *IEEE Computational Intelligence Magazine,* vol. 2, pp. 20-29, 2007.

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

[18] H. Ying, "Deriving analytical input-output relationship for fuzzy controllers using arbitrary input fuzzy sets and Zadeh fuzzy AND operator," *IEEE Transactions on Fuzzy Systems,* vol. 14, pp. 654-662, 2006.

[19] H. Ying, "A general technique for deriving analytical structure of fuzzy controllers using arbitrary trapezoidal input fuzzy sets and Zadeh AND operator," *Automatica,* vol. 39, pp. 1171-1184, 2003.