ANALYSIS AND APPLICATIONS OF THE KM ALGORITHM IN TYPE-2 FUZZY LOGIC CONTROL AND DECISION MAKING

NIE MAOWEN

(B.Eng UESTC)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2011

Acknowledgments

I would like to express my thanks to all the tutors, colleagues, friends, and family for their support on my research and life. During the period of my PhD program, I benefited and learned much from them, especially when I met obstacles.

First of all, I want to thank my supervisor Assoc. Prof. Tan Woei Wan for her patient guidance and advice on my research, writing and presentation throughout the past four years. Her insights on the theory of fuzzy logic have greatly stimulated my research work, and her patient guidance on writing and presentation gives me much help.

I also wish to take this opportunity to thank Prof. Wang Qingguo, Prof. Ben. Chen, Assoc. Prof. Xiang Cheng and Prof. Xu Jianxin for their courses which build up my fundamentals on the theory of control. Besides, I am grateful to my colleagues for their constant support and encourage.

Finally, I would like to express my gratitude to my parents for their consistent support. Without their encouragement and love, I may not complete my research during the period at university.

Contents

Summary

The concept of fuzzy logic was introduced to handle the uncertainties and vagueness which widely exist due to inaccurate information, unmeasurable disturbance and noise in practical applications. Fuzzy logic, also called type-1 fuzzy logic, has been widely applied to a variety of fields such as control, pattern recognition, signal processing, decision making, etc. Results from a large amount of experiments have shown that type-1 fuzzy logic is able to better cope with uncertainties than other traditional methodologies. However, type-1 fuzzy logic has been shown to be limited in modelling and minimizing the effect of uncertainties, especially in the face of complex uncertainties. In order to improve the ability of fuzzy logic in handling complex uncertainties, type-2 fuzzy logic was introduced. While the concept of type-2 fuzzy set was introduced by Zadeh in 1975, interest in the field grew only after Mendel and his students developed a theoretical framework for type-2 fuzzy systems. This thesis focuses on studying and enhancing the Karnik-Mendel (KM) algorithm, an iterative technique widely used in type-2 fuzzy set operations.

As an important application of type-2 fuzzy logic, type-2 fuzzy logic control has been attracting increasing attention from the research community. An open research issue is that whether a type-2 fuzzy logic controller has the potential to outperform type-1 fuzzy logic controller. Although a large number of experiments show that type-2 fuzzy controller can produce more satisfactory performance, there is no rigorous theoretical analysis to explain the condition under which a type-2 fuzzy controller can outperform type-1 fuzzy controller. The main challenge that impedes the theoretical analysis is the lack of closed-form expressions for type-2 fuzzy controller, primarily because the widely adopted Karnik-Mendel (KM) type-reducer can be implemented through the KM iterative algorithm/ the enhanced KM (EKM) iterative algorithm only. To overcome this challenge, the input-output relationship of a class of symmetric type-2 fuzzy PD/PI controller was established. The significance is that these mathematical equations lay the foundation for the theoretical study of type-2 fuzzy logic controller. By comparing the derived expressions with its type-1 counterpart, four interesting properties of type-2 fuzzy logic controller were identified. These properties provide insights into why a type-2 fuzzy logic controller is better able to balance the amount of the compromise between faster response and smaller overshoot.

As an extension of these results, the input-output relationship of a class of non-symmetric type-2 fuzzy PD and PI controllers was established. By comparing the derived expressions with its type-1 counterpart, it was found that the properties of the symmetric type-2 fuzzy controller still hold true for the non-symmetric type-2 fuzzy PD and PI controller. More importantly, another two properties were identified to highlight the differences between the non-symmetric type-2 fuzzy controller and the symmetric type-2 fuzzy controller and to establish the unique characteristics of the non-symmetric type-2 fuzzy controller. The analysis demonstrated that the non-symmetric type-2 fuzzy controller is able to further alleviate the amount of the compromise between a fast response and smaller overshoot.

Another application of the KM iterative algorithm is the computation of fuzzy weighted average (FWA) and linguistic weighted average (LWA). FWA and LWA are important aggregation methods that have many engineering applications. However, even with the introduction of the KM iterative algorithm/the EKM iterative algorithm to assist with the necessary α-cut arithmetic, the computational efficacy of FWA and LWA remained poor because of the iterative nature of the KM/EKM algorithm. Three algorithms that further reduce the computational burden needed to calculate FWA and LWA were presented. In order to achieve lower computational overhead, the proposed algorithms optimize the choice of the initial switch point in three different manners and propose an alternative termination condition in the procedure for the KM iterative algorithm. Theoretical analysis showed that the number of the iterations may be significantly reduced by the proposed algorithms, especially when the required accuracy increases. Results from numerical studies were presented to demonstrate that all the three proposed algorithms take fewer iterations and less computational time to compute the FWA and LWA. Among the three proposed algorithms, the one which require the least computational overhead can achieve an approximately 60% reduction in the computational time of the KM iterative algorithm and an approximately 40% reduction of the EKM iterative algorithm.

In conclusion, the advances about the pivotal KM iterative algorithm presented in this thesis enhance the understanding of type-2 fuzzy logic and promote its practical application in various areas.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Fuzzy logic

Uncertainty exists in various aspects of life, resulting in an interesting world. In the real world, uncertainties may arise from a variety of sources: (1) measurement error due to unavoidable noise and resolution limits of measuring equipments; (2) incomplete information due to limited knowledge, corrupted data, or the loss of data; (3) vague natural language in communication among human beings. Since uncertainty arising from different sources always exists, there is a need to find ways to model uncertainty and to minimize its effect. Varieties of strategies have been developed to cope with different types of uncertainties. Among them, fuzzy logic introduced by Lotfi Zadeh has been shown to be an effective methodology for handling uncertainties.

In the framework of fuzzy logic, the concept of fuzzy set was introduced by allowing the membership grade to be any value within the interval $[0, 1]$, instead of the unity or zero membership grade of traditional sets, to represent the degree of the relevance. Fuzzy logic is essentially a reasoning process by performing logic operations such as union, intersection, etc, on fuzzy sets. Comparing with unity or zero-membership grade of traditional set concept, varying membership grade of a fuzzy set between 0 and 1 enables a fuzzy set to better model uncertainties and minimize the effect of the uncertainties.

A widely adopted reasoning method in the theory of fuzzy logic is fuzzy logic system which uses fuzzy sets and a rule base to describe the input-output relationship of a system. Fuzzy logic system has been widely applied to modelling, control, pattern recognition, signal processing, etc. A large number of literatures on the applications of fuzzy logic system have emerged to study how to utilize fuzzy logic system to cope with uncertainties. Another reasoning method is to perform aggregation operators on fuzzy sets to aggregate the information represented by fuzzy sets. Various aggregation operators have been developed to achieve satisfactory performance. Although the reasoning methods in the theory of fuzzy logic is not limited to these two, this thesis will focus on these two reasoning processes.

1.1.1 Fuzzy control

The application of fuzzy logic system in control, termed as fuzzy control, is primarily to design fuzzy controllers for the controlled plants. Literatures on fuzzy control emerged in the early 1970's. An early work [49] by Mamdani proposed to utilize a fuzzy algorithm to control plants and used the laboratory-built steam engine as a testbed to examine its performance. The algorithm was implemented by interpreting a collection of rules expressed in terms of fuzzy conditional statements. In 1975, the basic framework of Mamdani fuzzy control system was established based on Mamdani fuzzy logic system by Mamdani and Assilian [50], and applied to control a steam engine. Based on the framework of Mamdani fuzzy control system, a fuzzy controller [73] was designed to control temperature of a heat exchanger system by varying the steam pressure supplied to the heat exchanger. The controller was designed by translating the prior knowledge on how to maintain the temperature through varying the steam pressure to linguistic rules which supervise how the inputs of fuzzy controllers determined control signals. Converting heuristic experience or prior knowledge to linguistic rules was the main method of designing fuzzy controllers in early publications [18, 67]. Such design methods not only provide opportunities for interaction between human beings and computers by incorporating knowledge from human being into linguistic rules, but also avoid accurate modelling of the controlled plant required by traditional control methodologies. Furthermore, fuzzy controller designed in this way is amenable for engineers to understand.

However, fuzzy controller designed based on the designer's experience is not sufficient for complex systems. In order to design more efficient fuzzy controllers, efforts have been made to extend conventional control technologies to fuzzy controllers [3, 23, 51, 69, 70, 78, 81]. Among conventional control technologies, PID control has been widely applied in industry. To extend PID control technology to fuzzy control system, the knowledge inherent in conventional PID control laws is converted to linguistic rules supervising how the two inputs, i.e. the system error and the rate of change, determine control signals. Results from experimental research have shown that fuzzy PID controllers can also produce better performance than conventional PID controllers. More importantly, fuzzy PID control system incorporates the advantages of conventional PID techniques in rejecting disturbance and maintaining stability so that it can produce satisfactory performance, even in face of a large amount of disturbances or modelling errors. The study of fuzzy PID control system has been broaden to structural design, disturbance rejection, parameters tuning, etc [17, 75, 77, 80, 82, 83, 107, 109].

1.1.2 Fuzzy aggregation

Fuzzy aggregation is an important reasoning method in the theory of fuzzy logic. This reasoning method has been widely applied in decision making, signal processing, etc [5, 16, 24, 36, 38, 40, 48, 53, 68, 72, 93, 95]. It is primarily used to aggregate the information inherent in a certain number of fuzzy sets to produce an overall result. For example, fuzzy aggregation may be performed to aggregate opinions from different people in multi-persons decision making or different attributes in multi-attribute decision making to produce a representative result as a criteria for decision making. The result of fuzzy aggregation highly depends on the choice of aggregation operator, and thus fuzzy aggregation operator is an important research topic in the theory of fuzzy aggregation. Hence, it is necessary to investigate fuzzy aggregation operator.

Early aggregation operations are max and min operation. Using max or min as aggregation operator means that only the information contained in the largest or smallest fuzzy numbers representing attributes or opinions are kept to represent the overall result. Since the aggregation operators max and min are extreme cases, they are not sufficient to model complex aggregation process. Another widely used aggregation method is fuzzy weighted average. Fuzzy weighted average is similar to average arithmetic, except that the former is performed on fuzzy numbers, while the latter is performed on crisp values. Fuzzy weighted average has been widely studied. An important advancement in the theory of fuzzy aggregation operator is the introduction of ordered weighted average by Yager [94]. Ordered weighted average is an aggregation operation lying between max and min operation. Unlike fuzzy weighted average in which the weights are assigned depending on the importance of each opinion or attribute, ordered weighted average operator allows the weights for opinions or attributes to be assigned according to the relative values of the fuzzy numbers representing the opinions or attributes. In the implementation, the first step of performing ordered weighted average is to assign the predefined weights based on the relative values of the fuzzy numbers representing opinions or attributes, and then it becomes a fuzzy weighted average problem. The choice of the aggregation operator, fuzzy weighted average or ordered weighted average, depends on the need of practical applications.

1.2 Extensional fuzzy logic theory

1.2.1 Type-2 fuzzy logic

Type-1 fuzzy logic has been shown to be a useful tool in handling uncertainties in a variety of areas such as control, pattern recognition, signal processing, decision making; however, type-1 fuzzy logic is not sufficient for coping with complex uncertainties arising from different sources. A primary reason is that the membership grade of a type-1 fuzzy set is a crisp value so that the membership function is limited in modelling the position and shape of a fuzzy set. The introduction of type-2 fuzzy logic overcomes this limitation, since for any value of the variable, the membership grade of type-2 fuzzy set is a fuzzy set, instead of a crisp value. This architecture of type-2 fuzzy set allows more design freedoms for modelling and coping with uncertainties.

The concept of type-2 fuzzy set was first introduced as an extension of type-1 fuzzy set by Zadeh in 1975 [106]. Set operations of type-2 fuzzy sets including union, intersection, algebraic product, algebraic sum, etc, were widely studied [28, 33]. The composition of type-2 relations was discussed as an extension of super-star composition of type-1 fuzzy logic [28, 29]. Based on these results, the complete theory of type-2 fuzzy logic system was established by Karnik and Mendel in 1999 [32]. Fig. 1.1 depicts the structural diagram of type-2 fuzzy logic system consisting of the components: fuzzifier, inference engine, type-reducer and defuzzifier. Type-2 fuzzy logic has been gaining increasing attention from the research community [8, 30, 31, 34, 55, 56, 57, 59, 64, 92]. Interval type-2 fuzzy set is a special type of type-2 fuzzy set, and has been widely studied. Type-2 fuzzy logic using interval type-2 fuzzy set is a hot research topic and also the focus of this thesis.

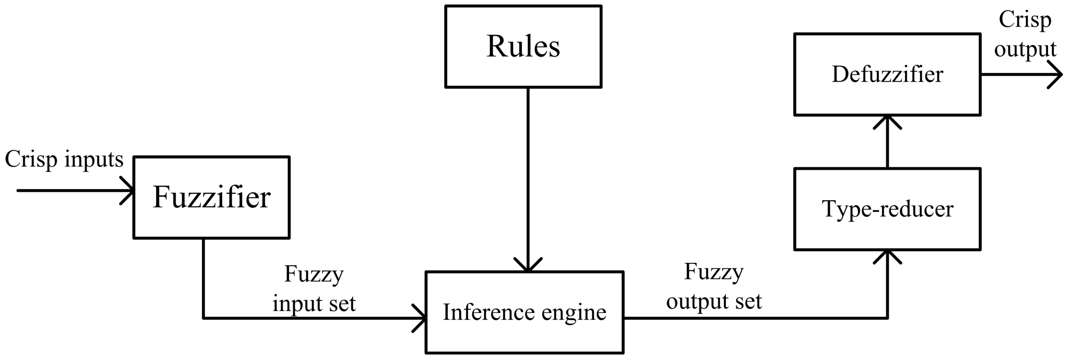


Figure 1.1: The structure of type-2 fuzzy logic system

1.2.2 Review of interval type-2 fuzzy control

With increasingly more researchers working on interval type-2 fuzzy control, the number of publications studying interval type-2 fuzzy controller rapidly increased [1, 6, 7, 37, 41, 76, 108]. Till now, there have been a great number of literatures on different types of interval type-2 fuzzy controllers. Tan and Lai investigated the robustness of an interval type-2 fuzzy proportional controller in the experiments of controlling the liquid-level process with biased parameters or delays, etc [74]. Hagras developed a hierarchical interval type-2 fuzzy controller for the navigation of mobile robots operating in varying indoor and outdoor environments [19]. Emmanuel, Martin and et al applied an interval type-2 fuzzy controller for video streaming across IP Networks by adjusting the bit rate to avoid both fluctuations and packet loss which may affect the end-users perception of the delivered video[25]. Liu, Zhang, and Wang proposed an interval type-2 fuzzy switching controller for the control of a biped robot with challenging dynamic characteristics such as its high-dimensional dynamics, the instability of two-legged motion, and multiple operating phases of the walking cycle [47]. Bartolomeo and Mose proposed an adaptive interval type-2 fuzzy controller for the control of the aerobic growth in a biomedical process [4].

Besides these applications of the interval type-2 fuzzy controller, there exist a certain number of literatures theoretically studying interval type-2 fuzzy controller. Wu and Tan investigated the robustness of the interval type-2 fuzzy proportional and derivative controller around the origin through studying the characteristics of its proportional and derivative gains [91]. Du and Ying proposed a method to approximate the output of interval type-2 fuzzy logic controller, derived the input-output relationship of a class of interval type-2 fuzzy proportional and derivative controller, and identified the characteristics of the interval type-2 fuzzy controllers [15]. To guarantee that the designed interval type-2 fuzzy controller produces a continuous control surface, Wu and Mendel [87] studied the continuity of an interval type-2 fuzzy system by identifying sufficient and necessary conditions for a continuous interval type-2 fuzzy system. Biglarbegian, Melek and Mendel studied the stability of interval type-2 fuzzy controller by proposing sufficient conditions for two classes of interval type-2 fuzzy controllers [6]. Although these results have shed some lights on interval type-2 fuzzy controller, few of them investigated the characteristics of interval type-2 fuzzy controllers using the widely adopted Karnik-Mendel type-reducer. Therefore, there is a need to study interval type-2 fuzzy controllers using the widely adopted Karnik-Mendel type-reducer theoretically.

1.2.3 Review of fuzzy aggregation using interval type-2 fuzzy set

"Words mean different things to different people"[54]. Hence, interval type-2 fuzzy set can be used to model a word so that opinions from different people can be incorporated into one fuzzy set. Hence, it is necessary to study fuzzy aggregation using interval type-2 fuzzy set. To perform fuzzy aggregation using interval type-2 fuzzy set, techniques for modelling words using interval type-2 fuzzy set need to be developed first. At this stage of development, there has been a number of publications studying how to use interval type-2 set to model words [45, 58, 63, 65, 66]. Mendel reported two approaches for mapping a word from a group of subjects into an interval type-2 fuzzy set for that word: the person-membership function approach and the interval endpoints approach [58, 65, 66]. In the person-membership function approach, a word for each subjective is represented using an interval type-2 fuzzy set while in the interval endpoints approach, a word for each subjective is represented using an interval on a scale of 0 -10, respectively. However, these two approaches require people to be knowledgable about fuzzy sets. To make it easier and practical, Liu and Mendel proposed an interval approach for encoding words into interval type-2 fuzzy sets by first mapping interval endpoints data for any subjective into a pre-specified type-1 membership function and then aggregating them into an interval type-2 fuzzy set for a word from these type-1 fuzzy sets [45].

The aggregation operator on interval type-2 fuzzy sets has gained much attention from the research community. Wu and Mendel extended the concept of fuzzy weighted average for type-1 fuzzy sets to interval type-2 fuzzy sets and called it linguistic weighted average [84]. Similarly, ordered linguistic weighted average for aggregating interval type-2 fuzzy sets was proposed as an extension of ordered fuzzy weighted average for type-1 fuzzy sets. These aggregation operators have been applied in many decision making processes. For example, Wu and Mendel applied the linguistics weighted average to a hierarchical decision making for evaluating a weapon system [86]. In this hierarchical decision making system, the performance of competing alternatives are evaluated by comparing the aggregation of hierarchical criteria and sub-criteria of alternatives. The linguistic weighted average was also applied for evaluating locations of international logistic centers by Han and Mendel [21]. Although the linguistic weighted average and ordered weighted average are efficient aggregation operators, the computational requirement of performing the linguistic weighted average is prohibitively high so that they may not be suitable for practical applications. Hence, developing efficient algorithms for implementing fuzzy aggregation is one focus of this thesis.

1.3 Aims and Scope of the Work

Although there have been an increasing number of publications on interval type-2 fuzzy logic, there are still many open issues that need further investigation. Most of these issues arise from the following common computation step needed in different type-2 fuzzy sets operations:

Given variables x_i and w_i satisfying

$$x_i \in [\underline{x}_i, \overline{x}_i], \quad \overline{x}_i > \underline{x}_i \tag{1.1}$$

$$w_i \in [\underline{w}_i, \overline{w}_i], \quad \overline{w}_i > \underline{w}_i \tag{1.2}$$

where $[\underline{x}_i, \overline{x}_i]$ and $[\underline{w}_i, \overline{w}_i]$ are interval sets bounded by crisp values $\underline{x}_i$ and $\overline{x}_i$, $\underline{w}_i$ and $\overline{w}_i$, respectively,

Compute

$$y_l = \min_{\substack{x_i \in [\underline{x}_i, \overline{x}_i] \\ w_i \in [\underline{w}_i, \overline{w}_i]}} \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \tag{1.3}$$

$$y_r = \max_{\substack{x_i \in [\underline{x}_i, \overline{x}_i] \\ w_i \in [\underline{w}_i, \overline{w}_i]}} \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \tag{1.4}$$

The Karnik-Mendel iterative algorithm is an efficient method for searching for y_l and y_r. A situation where the common computation step defined in (1.1)-(1.4) needs to be performed is to implement the Karnik-Mendel type-reducer which is an indispensable step in the implementation of an interval type-2 fuzzy logic controller. A result of applying the Karnik-Mendel type-reducer is the lack of closed from expression because the Karnik-Mendel type-reducer can be computed using the Karnik-Mendel iterative algorithm only. Due to the lack of closed form expressions, it is challenging to perform theoretical study of interval type-2 fuzzy logic controller. Hence, the research reported in the thesis starts by deriving the closed-form equations relating the output of the interval type-2 fuzzy system with the inputs, providing a platform for further theoretical study of interval type-2 fuzzy controller.

Once the mathematical expressions are established, the theoretical study of interval type-2 fuzzy logic controller can be performed. Despite several attempts made to compare interval type-2 fuzzy controller and type-1 fuzzy controller, it remains unclear whether an interval type-2 fuzzy controller can improve the performance of a type-1 fuzzy controller. Hence, there is a need to study the potential advantage of interval type-2 controller over its type-1 counterpart.

Another situation that requires the common computation step defined in (1.1)-(1.4) is the computation of the fuzzy weighted average and linguistic weighted average. Although the Karnik-Mendel iterative algorithm was introduced to reduce the computational overhead, they remain computationally intensive due to the iterative nature of the Karnik-Mendel iterative algorithm. Hence, it is necessary to develop efficient algorithms which reduce the computational overhead required by performing fuzzy weighted average and linguistic weighted average.

In summary, this thesis seeks a further understanding of the Karnik-Mendel algorithm that plays a pivotal role in the theory of interval type-2 fuzzy logic operations including the type-reduction step of an interval type-2 fuzzy logic controller and linguistic weighted average operator. Based on the above discussion, the detailed objectives are as follows:

1. To establish the mathematical equations relating the output and the inputs of interval type-2 fuzzy controller using the Karnik-Mendel type-reducer, as a tool for theoretical study of interval type-2 fuzzy logic controller.

2. To establish the potential advantage of interval type-2 fuzzy logic controller over type-1 fuzzy controller using the derived mathematical expressions, especially how the additional parameters introduced by antecedent interval type-2 sets affect the input-output relationship of interval type-2 fuzzy logic controller. To verify the established potential advantages through numerical experiments.

3. To propose efficient algorithms for the implementation of fuzzy weighted average and linguistic weighted average to reduce their computational overhead. To evaluate the computational performance of the proposed algorithms by comparing with the Karnik-Mendel iterative algorithm.

1.4 Organization of the Thesis

In order to facilitate an understanding of the theory of type-2 fuzzy logic, Chapter 2 provides a brief description of the fundamental theory of type-2 fuzzy logic including the basics of type-2 fuzzy set and type-2 fuzzy logic system. Chapter 3 derives the input-output relationship of a class of symmetric interval type-2 fuzzy

PD/PI controller using the Karnik-Mendel type-reducer by following the proposed algorithm which overcomes the limitation of no closed form equations for such interval type-2 fuzzy controller. Using the derived mathematical equations, this chapter investigates the potential advantage of interval type-2 fuzzy controller over its type-1 counterpart by identifying four interesting properties unique to interval type-2 fuzzy controller. Chapter 4 develops the input-output relationship for a more general class of interval type-2 fuzzy controller. Chapter 4 shows that those properties identified for the symmetric interval type-2 fuzzy logic controller still hold true and establishes unique properties of the general class of interval type-2 fuzzy logic controller. Chapter 5 presents efficient algorithms for fuzzy weighted average and linguistic weighted average to reduce their computational overhead and studies their computational performance through intensive experiments.

Chapter 2

Review of Type-2 Fuzzy Logic

This chapter aims to provide a brief introduction of the basics of type-2 fuzzy logic to assist readers to understand the results reported in this thesis. The basics about type-2 fuzzy set including the concept of type-2 fuzzy set and operations among type-2 fuzzy sets will be introduced first in Section I of this chapter. Section II reviews the centroid of a type-2 fuzzy set. Section III focuses on the theory of type-2 fuzzy logic system including the concept of type-2 fuzzy logic system and its implementation.

2.1 Type-2 Fuzzy Set

A type-1 fuzzy set is characterized by its membership grade which can be any value within the interval $[0, 1]$. A type-1 fuzzy set $\hat{A}$ can be characterized by

$$\hat{A} = \int_{x \in X} \mu_{\hat{A}}/x \tag{2.1}$$

where the variable x is defined in the domain X; $\mu_{\hat{A}}$ is the membership grade. Fig. 2.1 gives an example of type-1 membership function. Since the membership grade

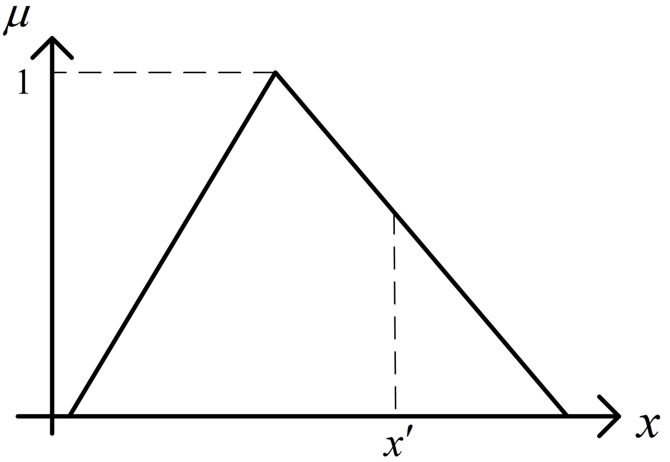


Figure 2.1: Type-1 membership function

is a crisp value, type-1 fuzzy set is limited in handling uncertainties in the shape and position of the fuzzy set. As an extension of type-1 fuzzy set, the membership grade of a type-2 fuzzy set is a type-1 fuzzy set instead of a crisp value, providing more freedoms in the membership grade for coping with complex uncertainties. The concept of type-2 fuzzy set plays an important role in the theory of type-2 fuzzy logic. This section will briefly review the concept of type-2 fuzzy set.

2.1.1 The Concept of Type-2 Fuzzy Set

Definition 2.1 *A type-2 fuzzy set $\tilde{A}$ is characterized as*

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \quad J_x \subset [0,1] \tag{2.2}$$

where the primary variable x is defined on the domain X; the secondary variable $u \in U$ has domain J_x for each $x \in X$. J_x, the primary membership of x is defined as

$$J_x = \{(x,u) : u \in [\underline{u}_{\tilde{A}}(x), \overline{u}_{\tilde{A}}(x)]\} \tag{2.3}$$

$\tilde{A}$ can also be expressed as

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) | \forall x \in X, \forall u \in J_x \subset [0,1]\} \tag{2.4}$$

The domain of a secondary membership function is called the primary membership of x, i.e. J_x is the primary membership of x, where $J_x \subset [0,1]$ for $\forall x \in X$. The amplitude of a secondary membership function is called a secondary grade, i.e. $\mu_{\tilde{A}}(x,u)$, is a secondary grade.

Fig. 2.2 shows an example of the membership function of a type-2 fuzzy set. The primary variable x and the secondary variable u are discrete. The primary membership J_x is restricted in the interval $[0,1]$, i.e. $0 \leq u \leq 1$. The secondary membership grade is also in the interval $[0,1]$, i. e. $0 \leq \mu_{\tilde{A}}(x) \leq 1$.

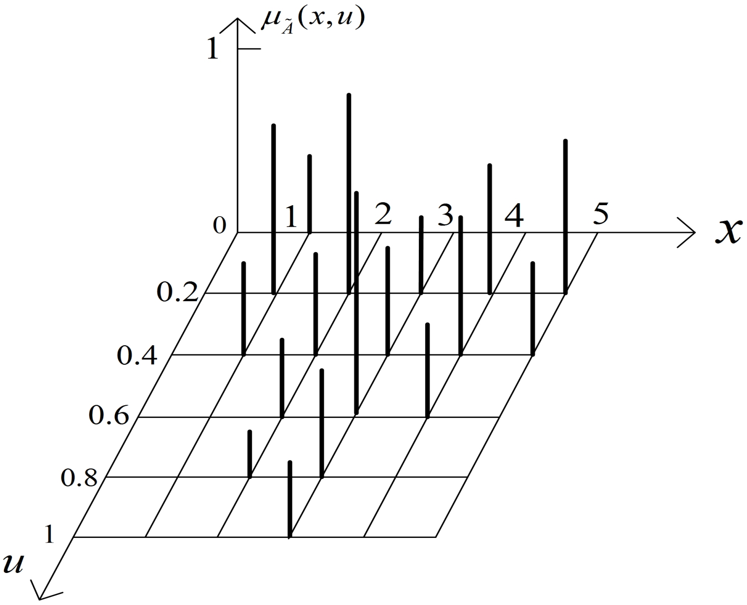


Figure 2.2: An example of type-2 membership function

For each value of x, for example $x = x'$, the 2D plane whose axes are u and $\mu_{\tilde{A}}(x',u)$ is called a vertical slice of $\mu_{\tilde{A}}(x,u)$. A secondary membership function

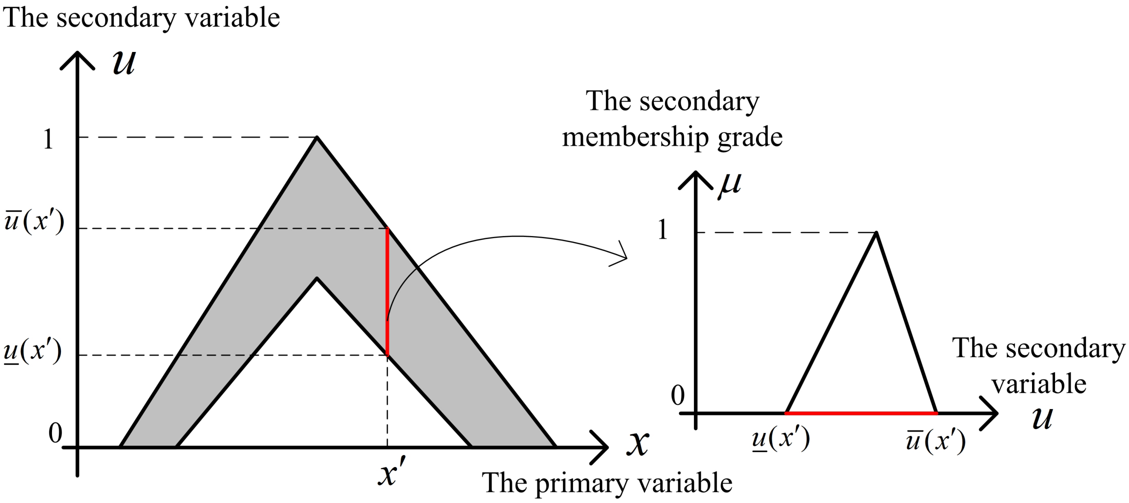


Figure 2.3: Vertical-slice of a type-2 fuzzy set

is a vertical slice of $\mu_{\tilde{A}}(x,u)$, i. e.

$$\mu_{\tilde{A}}(x=x',u)=\mu_{\tilde{A}}(x')=\int_{u\in J_{x'}}\mu_{\tilde{A}}(u)/u \tag{2.5}$$

A type-2 fuzzy set can be considered as the union of all its vertical slices, which is the vertical-slice representation for a type-2 fuzzy set. Fig. 2.3 shows the membership function of a type-2 fuzzy set where the 2D plane whose axes are u and $\mu_{\tilde{A}}(x',u)$ at each x' is a vertical-slice. Each vertical slice is a type-1 fuzzy set and such type-1 fuzzy set is called as a secondary set. Based on the concept of a secondary set, a type-2 fuzzy set can be interpreted as the union of all secondary sets (vertical slices), i. e.

$$\tilde{A}=\int_{x\in X}\mu_{\tilde{A}}(x)/x=\int_{x\in X}[\int_{u\in J_x}\mu_{\tilde{A}}(u)/u]/x \tag{2.6}$$

Interval type-2 fuzzy set is a special case of type-2 fuzzy set when the secondary grades equal to unity.

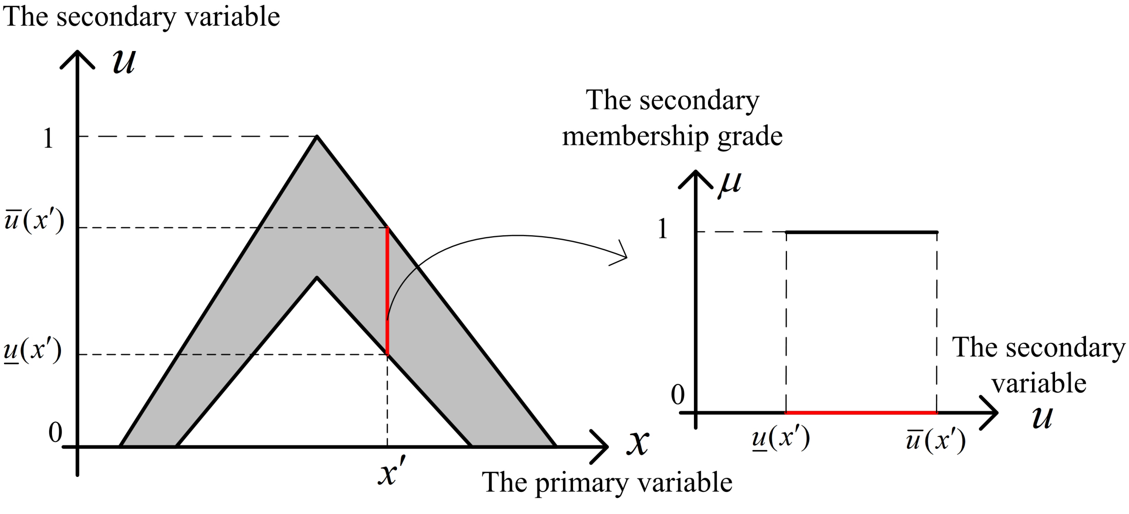


Figure 2.4: Vertical-slice of an interval type-2 fuzzy set

Definition 2.2 *An interval type-2 fuzzy set is characterized by*

$$A = \int_{x\in X}\int_{u\in J_x} 1/(x,u) = \int_{x\in X}[\int_{u\in J_x\subset[0,1]} 1/u]/x \tag{2.7}$$

where x is the primary variable and u is the secondary variable.

Fig. 2.4 shows the membership function of an interval type-2 fuzzy set whose primary variable is x, the secondary variable is u and the secondary membership grade are all unity. For each x', the 2D plane whose axis is u and $\mu_{\tilde{A}}(x',u)$ is a vertical slice. Unlike a type-2 fuzzy set where each vertical slice at x' is a type-1 fuzzy set, each vertical slice of an interval type-2 fuzzy set at x' is an interval set.

Since all the secondary grades of an interval type-2 fuzzy set A are unity, all the uncertainty modelled by an interval type-2 fuzzy set A can be completely described by the union of all the primary memberships, which is called the footprint of uncertainty (FOU) of A, i. e.

$$FOU(A) = \bigcup_{\forall x\in X} J_x = \{(x,u) : u \in J_x \subset [0,1]\} \tag{2.8}$$

The FOU of an interval type-2 fuzzy set is bounded by two type-1 membership functions, called the upper membership function (UMF) and the lower membership function (LMF). The UMF and LMF are associated with the upper bound and the lower bound of $FOU(A)$, denoted by $\underline{u}(x)$ and $\overline{u}(x)$, respectively, i. e.

$$\underline{u}(x) = \underline{FOU(A)} \quad \forall x \in X \tag{2.9}$$

$$\overline{u}(x) = \overline{FOU(A)} \quad \forall x \in X \tag{2.10}$$

Fig. 2.5 shows the FOU of an interval type-2 fuzzy set, in which the shaded area is the FOU bounded by the UMF and LMF. Since the FOU of an interval fuzzy set can be completely described by the UMF and the LMF, an interval type-2 fuzzy set can be completely determined by its UMF and LMF. The concept of UMF and LMF of an interval type-2 fuzzy set are useful in the theory of interval type-2 fuzzy logic.

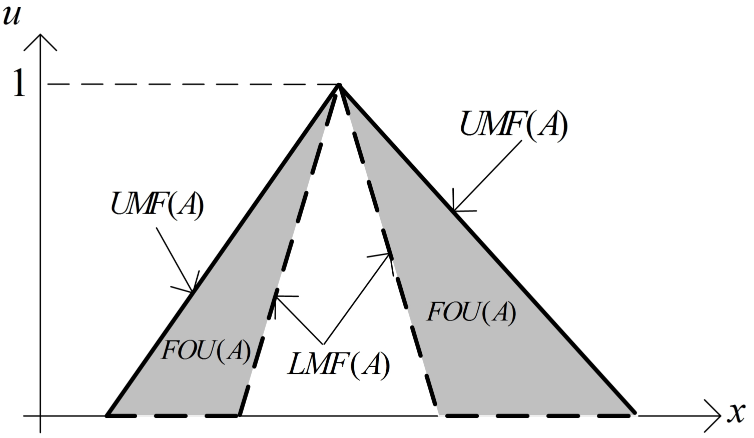


Figure 2.5: Interval type-2 membership function: UMF, LMF and FOU

2.1.2 Representation of Type-2 Fuzzy Set

A type-2 fuzzy set $\tilde{A}$ can be interpreted as a collection of type-2 fuzzy sets $\tilde{A}_e$, which we call embedded type-2 fuzzy sets in $\tilde{A}$.

Definition 2.3 *For a type-2 fuzzy set defined in continuous of discourse X and U, an embedded type-2 fuzzy set $\tilde{A}_e$ is*

$$\tilde{A}_e = \int_{x \in X} [\mu_{\tilde{A}}(x,\theta)/\theta]/x \quad \theta \in J_x \subset U = [0,1] \tag{2.11}$$

An embedded type-2 fuzzy set can be constructed by choosing a primary membership θ from the primary membership grade J_x for each value of the primary variable x, and the associated secondary membership grade $\mu_{\tilde{A}}(x,\theta)$. Since there are an infinite number of the possibilities in choosing θ from a continuous interval J_x, there are a countless number of embedded type-2 fuzzy sets. A type-2 fuzzy set can be represented as a union of an infinitely embedded type-2 fuzzy set, which is called the wavy-slice representation for a type-2 fuzzy set.

Another important concept is embedded type-1 set A_e.

Definition 2.4 *For a type-2 fuzzy set defined in continuous universe of discourse X and U, an embedded type-1 fuzzy set A_e is:*

$$A_e = \int_{x \in X} \theta/x, \quad ,\theta \in J_x \subset U = [0,1] \tag{2.12}$$

An embedded type-1 fuzzy set A_e is the union of all the primary memberships of set $\tilde{A}_e$ defined in (2.11), and thus there are an infinite number of A_e.

Fig. 2.6 shows an example of an embedded type-1 fuzzy set of a type-2 fuzzy set. The concept of embedded type-1 fuzzy set is very useful in the theory of interval type-2 fuzzy set. An application of embedded type-1 fuzzy set is the wavy-slice representation for an interval type-2 fuzzy set, which may be formally stated as follows:

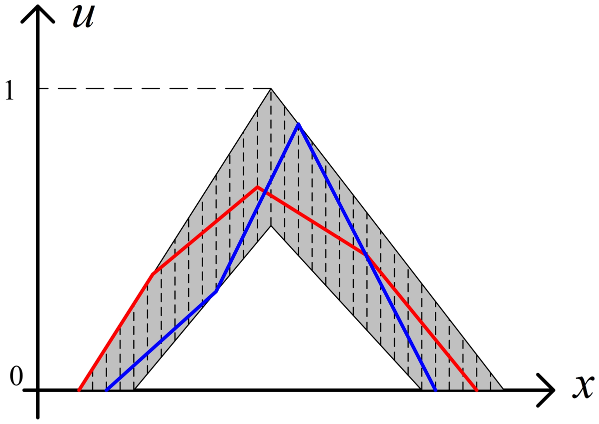


Figure 2.6: Embedded type-1 set (Red or green thick solid lines)

Theorem 2.1 *(Wavy-slice representation) [54]: For an interval type-2 fuzzy set, the domain of A is equal to the union of all of its embedded type-1 fuzzy set, i.e.*

$$A = 1/FOU(A) = 1/\bigcup_{i=1}^{n} A_e^j \tag{2.13}$$

This theorem indicates that the domain of an interval type-2 fuzzy set can be considered as a union of all its embedded type-1 fuzzy sets. This representation theorem provides an efficient method for operations on interval type-2 fuzzy sets by representing an interval type-2 fuzzy set as a collection of all its embedded type-1 fuzzy sets.

2.1.3 Operations among Type-2 Fuzzy Sets

Operations among type-2 fuzzy sets are complex to implement, and these complex operations impede the study of type-2 fuzzy logic. Comparing with operations among type-2 fuzzy sets, the operations on interval type-2 fuzzy sets are easy to implement, primarily because an interval type-2 fuzzy set can be completely described by its UMF and LMF. Since this thesis centers on interval type-2 fuzzy logic, this subsection will review the operations among interval type-2 fuzzy sets.

The widely used operations on interval type-2 fuzzy sets include union, intersection and complement. Suppose interval fuzzy sets A and B are characterized by

$$A = 1/FOU(A) = 1/\bigcup_{\forall x \subset X} [\underline{\mu}_A(x), \overline{\mu}_A(x)] \tag{2.14}$$

$$B = 1/FOU(B) = 1/\bigcup_{\forall x \subset X} [\underline{\mu}_B(x), \overline{\mu}_B(x)] \tag{2.15}$$

then the operations are as follows

1. The union of A and B is an interval type-2 fuzzy set, i. e.

$$A \cup B = 1/\bigcup_{\forall x \in X} [\underline{\mu}_A(x) \vee \underline{\mu}_B(x), \overline{\mu}_A(x) \vee \overline{\mu}_B(x)] \tag{2.16}$$

2. The intersection of A and B is an interval type-2 fuzzy set, i. e.

$$A \cap B = 1/\bigcap_{\forall x \in X} [\underline{\mu}_A(x) \star \underline{\mu}_B(x), \overline{\mu}_A(x) \star \overline{\mu}_B(x)] \tag{2.17}$$

3. The implementation of A is also an interval type-2 fuzzy set, i. e.

$$\overline{A} = 1/\bigcup_{\forall x \in X} [1 - \underline{\mu}_A(x), 1 - \overline{\mu}_A(x)] \tag{2.18}$$

Form the above operations, it may be observed that the results of the union, intersection and complement operation are interval type-2 fuzzy sets, and the implementation of these operations is equivalent to computing the UMF and LMF of the resulted interval type-2 fuzzy set. The UMF and LMF of the resulted interval type-2 fuzzy sets are the results of the corresponding operations between the UMF and LMF of interval type-2 fuzzy sets A and B, respectively.

2.2 Centroid of a Type-2 Fuzzy Set

Defuzzification is an indispensable step of converting the output of the inference engine to a crisp value in a type-1 fuzzy logic system, i.e. computing the centroid of a type-1 fuzzy set $\hat{A}$,

$$C_{\hat{A}} = \frac{\sum_{i=1}^{N} \mu_i x_i}{\sum_{i=1}^{N} \mu_i} \tag{2.19}$$

where $x_i, i = 1, 2, \cdots, N$ are the discretized points of the variable; μ_i are the corresponding membership grade. The important step of an interval type-2 fuzzy set that corresponds to the defuzzification procedure is type-reduction, i.e. calculating the centroid of an interval type-2 fuzzy set.

2.2.1 Centroid of a Type-2 Fuzzy Set

Given a type-2 fuzzy set $\tilde{A} = \{(x, \mu_A(x))|x \in X\}$, discretize the x-domain into N points, $x_1, x_2, \cdots, x_N$, and then the type-2 fuzzy set A can be expressed as

$$\tilde{A} = \sum_{i=1}^{N} [\int_{u \in J_x} f_{x_i}(u)/u]/x_i \tag{2.20}$$

By applying the extension principle to the centroid of a type-1 fuzzy set defined in (2.19), the centroid of a type-2 fuzzy set can be defined as

$$C_{\tilde{A}} = \int_{w_1 \in J_{x_1}} \cdots \int_{w_N \in J_{x_N}} [f_{x_1}(w_1) \star \cdots \star f_{x_N}(w_N)] / \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{2.21}$$

Every possible combination of the the primary membership grade $w_1, \cdots, w_N$ and the corresponding secondary membership grade $f_{x_1}(w_1), \cdots, f_{x_N}(w_N)$ forms an embedded type-2 fuzzy set, $\tilde{A}_e$. The centroid of a type-2 fuzzy set is a type-1 fuzzy set whose primary variable is the result of

$$\frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{2.22}$$

and the corresponding membership grade is the result of $f_{x_1}(w_1) \star \cdots \star f_{x_N}(w_N)$. A method that may be used to compute the centroid of a type-2 fuzzy set is to exhaustively calculate all possible embedded type-2 fuzzy set, i.e. all the possible combination of w_i and x_i and the corresponding $f_{x_1}(w_1) \star \cdots \star f_{x_N}(w_N)$.

2.2.2 Centroid of an Interval Type-2 Fuzzy Set

For an interval type-2 fuzzy set, all $f_{x_i}(u)$ for any x_i and u equals 1. Then the centroid can be defined as

$$GC = \int_{x_1 \in X_1} \cdots \int_{x_N \in X_N} \int_{w_1 \in W_N} \cdots \int_{w_N \in W_N} 1 / \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} = 1/[y_l, y_r] \quad (2.23)$$

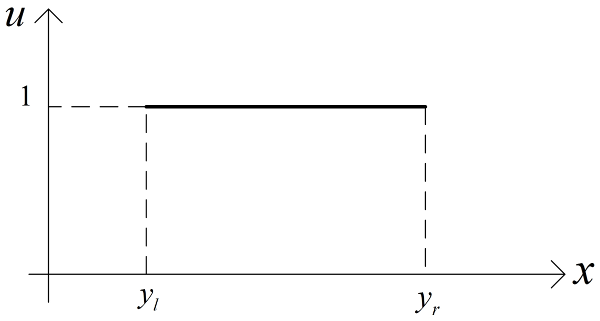


Figure 2.7: Centroid of an interval type-2 fuzzy set

Fig. 2.7 shows the centroid of an interval type-2 fuzzy set is an interval set bounded by y_l and y_r. It has been proven that y_l and y_r are the maximum and minimum value of all possible combinations of $\frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i}$, i.e.

$$y_l = \min_{w_i \in [\underline{w}_i, \overline{w}_i]} \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \quad (2.24)$$

$$y_r = \max_{w_i \in [\underline{w}_i, \overline{w}_i]} \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \quad (2.25)$$

Suppose

$$y(w_1, \cdots, w_N) = \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{2.26}$$

Differentiating $y(w_1, \cdots, w_N)$ with respect to w_k leads to

$$\frac{\partial}{\partial w_k} = \frac{\partial}{\partial w_k}[\frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i}] = \frac{x_k - y(w_1, \cdots, w_N)}{\sum_{i=1}^{N} w_i} \tag{2.27}$$

From (2.27), the following can be observed that

1. When $y(w_1, \cdots, w_N) = x_k$,

$$y(w_1, \cdots, w_N) = x_k \to \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} = x_k \to \frac{\sum_{i \neq k}^{N} x_i w_i}{\sum_{i \neq k}^{N} w_i} = x_k \tag{2.28}$$

   w_k does not appear in the final expression, indicating that the value of w_k does not affect the result of $y(w_1, \cdots, w_N)$ when $y(w_1, \cdots, w_N) = x_k$.

2. The equation indicates the direction in which w_k should be changed to increase or decrease the value of $y(w_1, \cdots, w_N)$:

   - If $x_k > y(w_1, \cdots, w_N)$, $y(w_1, \cdots, w_N)$ increases as w_k increases;
   - If $x_k < y(w_1, \cdots, w_N)$, $y(w_1, \cdots, w_N)$ decreases as w_k increases.

Based on the above results, y_l and y_r can be expressed as

$$y_l = \frac{\sum_{i=1}^{L} x_i \overline{w}_i + \sum_{i=L+1}^{N} x_i \underline{w}_i}{\sum_{i=1}^{L} \overline{w}_i + \sum_{i=L+1}^{N} \underline{w}_i} \tag{2.29}$$

$$y_r = \frac{\sum_{i=1}^{R} x_i \underline{w}_i + \sum_{i=R+1}^{N} x_i \overline{w}_i}{\sum_{i=1}^{R} \underline{w}_i + \sum_{i=R+1}^{N} \overline{w}_i} \tag{2.30}$$

It should be noted that the switch point L and R satisfying

$$x_L \leq y_l \leq x_{L+1} \tag{2.31}$$

$$x_R \leq y_r \leq x_{R+1} \tag{2.32}$$

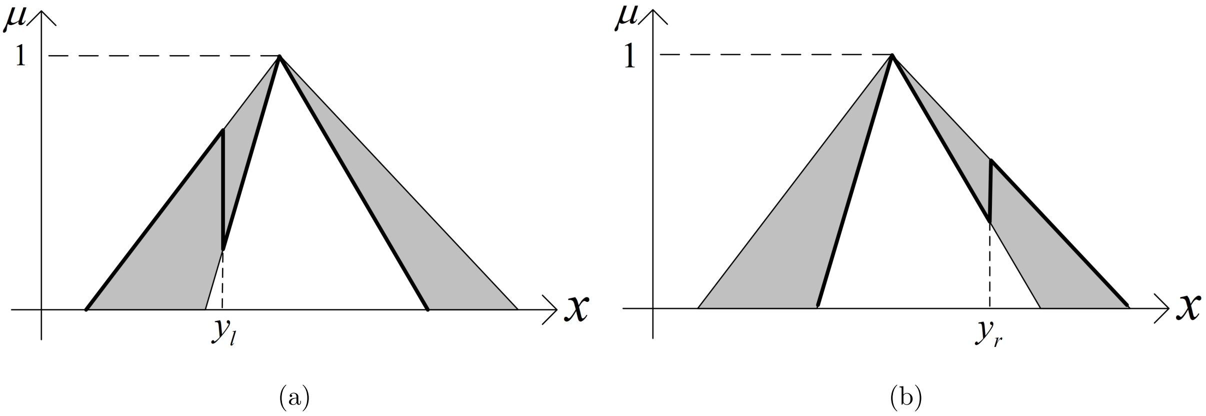


Figure 2.8: The left and right endpoints y_l and y_r with switch point L and R

For an interval type-2 fuzzy set defined in continuous domain, y_l and y_r can be expressed as

$$y_l = \frac{\int_{-\infty}^{l} x\overline{w}_i(x)dx + \int_{l}^{\infty} x\underline{w}_i(x)dx}{\int_{-\infty}^{l} \overline{w}_i(x)dx + \int_{l}^{\infty} \underline{w}_i(x)dx} \tag{2.33}$$

$$y_r = \frac{\int_{-\infty}^{r} x\underline{w}_i(x)dx + \int_{r}^{\infty} x\overline{w}_i(x)dx}{\int_{-\infty}^{r} \underline{w}_i(x)dx + \int_{r}^{\infty} \overline{w}_i(x)dx} \tag{2.34}$$

where the switch point l and r satisfy $l = y_l$ and $r = y_r$. Fig. 2.8 illustrates that y_l and y_r can be achieved involving a switch point L and R.

2.2.3 The Karnik-Mendel Iterative Algorithm and The Enhanced Karnik-Mendel Iterative Algorithm

Although the left and right bounds of the type-reduced set y_l and y_r can be expressed mathematically as (2.29) and (2.30), these expressions can not be directly used to calculate y_l and y_r because the switch points L and R are unknown. The Karnik-Mendel iterative algorithm is an efficient algorithm that searches for the switch points L and R. The detailed procedure of the Karnik-Mendel (KM) iterative algorithm may be stated as

- Step 1: Sort x_i in an increasing order and label them as $x_1 < x_2 < \cdots < x_n$. Let $[\underline{w}_i, \overline{w}_i]$ be the corresponding weight of x_i.

- Step 2: Set w_i as

$$w_i = \frac{\underline{w}_i + \overline{w}_i}{2} \tag{2.35}$$

and then compute

$$y = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{2.36}$$

- Step 3: Find the switch point k such as

$$x_k \leq y \leq x_{k+1} \tag{2.37}$$

- Step 4: Set w_i as:

  - y_l:

$$w_i = \begin{cases} \overline{w}_i & \text{for } i \leq k \\ \\ \underline{w}_i & \text{for } i > k \end{cases} \tag{2.38}$$

  - y_r:

$$w_i = \begin{cases} \underline{w}_i & \text{for } i \leq k \\ \\ \overline{w}_i & \text{for } i > k \end{cases} \tag{2.39}$$

and compute

$$y' = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{2.40}$$

- Step 5: If $y' = y$, stop. k is the actual switch point L (R) and $y_l = y$ ($y_r = y$). Otherwise, set $y = y'$ and go to Step 3.

The Enhanced Karnik-Mendel (EKM) iterative algorithm is an enhanced version of the KM iterative algorithm through optimizing the initial switch point, the terminal condition and the computational process [85]. The strategies used to optimize the initialization, termination and computation are as follows:

1. Initialization: The initialization of the KM iterative algorithm is inefficient and thus may cause a large number of iterations. To reduce the number of iterations required in the search process, the switch point is initialized as $k = [n/2.4]$ for y_l or $k = [n/1.7]$ for y_r ($[\cdot]$ denotes the nearest integer.), where n is the number of x_i. The heuristic number 2.4 and 1.7 are obtained from intensive simulation studies.

2. Computation: In the KM iterative algorithm, to compute y in (5.48), $\sum_{i=1}^{m} w_i$ and $\sum_{i=1}^{n} x_i w_i$ need to be computed in each iteration. To reduce computational cost, the new computation technique is introduced by utilizing results from the last iteration. Suppose that in the jth iteration, the switch point k, $\sum_{i=1}^{m} w_i$ and $\sum_{i=1}^{n} x_i w_i$ are denoted by k_j, $(\sum_{i=1}^{m} w_i)_j$ and $(\sum_{i=1}^{n} x_i w_i)_j$, respectively. $(\sum_{i=1}^{m} w_i)_{j+1}$ can be computed by adding the difference between $(\sum_{i=1}^{m} w_i)_j$ and $(\sum_{i=1}^{m} w_i)_{j+1}$ to $(\sum_{i=1}^{m} w_i)_j$. Similarly, $(\sum_{i=1}^{m} w_i x_i)_{j+1}$ can be computed by adding the difference between $(\sum_{i=1}^{m} w_i x_i)_j$ and $(\sum_{i=1}^{m} w_i x_i)_{j+1}$.

3. Termination: In the KM iterative algorithm, the termination is identified by comparing the output of the current iteration with the last iteration, indicating that another iteration is needed although the actual switch point is found. To avoid the computation in the unnecessary iteration, the termination of iterations is proposed to be identified by comparing the switch point

of the current iteration with that of the last iteration.

In summary, the procedures of the EKM iterative algorithm may be described as follows:

- Step 1: Sort $x_i, i = 1, 2, \cdots, n$ in an increasing order and label them as $x_1 < x_2 < \cdots < x_n$.

- Step 2:

  - y_l: Set $k = [n/2.4]$ (the nearest integer to $n/2.4$), and compute

$$a = \sum_{i=1}^{k} x_i \overline{w}_i + \sum_{i=k+1}^{n} x_i \underline{w}_i \tag{2.41}$$

$$b = \sum_{i=1}^{k} \overline{w}_i + \sum_{i=k+1}^{n} \underline{w}_i \tag{2.42}$$

and

$$y = \frac{a}{b} \tag{2.43}$$

  - y_r: Set $k = [n/1.7]$ (the nearest integer to $n/1.7$), and compute

$$a = \sum_{i=1}^{k} x_i \underline{w}_i + \sum_{i=k+1}^{n} x_i \overline{w}_i \tag{2.44}$$

$$b = \sum_{i=1}^{k} \underline{w}_i + \sum_{i=k+1}^{n} \overline{w}_i \tag{2.45}$$

and

$$y = \frac{a}{b} \tag{2.46}$$

- Step 3: Find $k' \in [1, n-1]$ such that

$$x_{k'} \leq y \leq x_{k'+1} \tag{2.47}$$

- Step 4: Check if $k' = k$. If yes, stop, set $y_l = y$ ($y_r = y$), and $L = k$ ($R = k$). If no, continue.

- Step 5: Compute $s = \text{sign}(k' - k)$, and

$$a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} x_i(\overline{w}_i - \underline{w}_i) \tag{2.48}$$

$$b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} (\overline{w}_i - \underline{w}_i) \tag{2.49}$$

$$y' = \frac{a'}{b'} \tag{2.50}$$

- Step 6: Set $y = y'$, $a = a'$, $b = b'$, and $k = k'$. Go to Step 3.

2.3 Type-2 Fuzzy Logic System

This section will provide a brief introduction of the theory of type-2 fuzzy logic system. A description of a type-2 fuzzy logic system is provided in the following subsection before its implementation will be described in the next subsection.

2.3.1 Components of a Type-2 Fuzzy Logic System

A type-1 fuzzy logic system comprises of four components: fuzzifier, rule base, inference engine and defuzzifier. Fig. 2.9 depicts the diagram structure of type-1 fuzzy logic system. The fuzzifier maps crisp inputs into type-1 input fuzzy sets. Rules are expressed into IF-THEN statement relating antecedent sets and consequent sets. In the inference engine, the operations between type-1 input fuzzy sets and antecedent sets output the firing degree of each rule, while the operations between the firing degree and the consequent sets produce a type-1 output fuzzy set. The defuzzifier is to convert this output fuzzy set to a crisp value as the output of a type-1 fuzzy logic system.

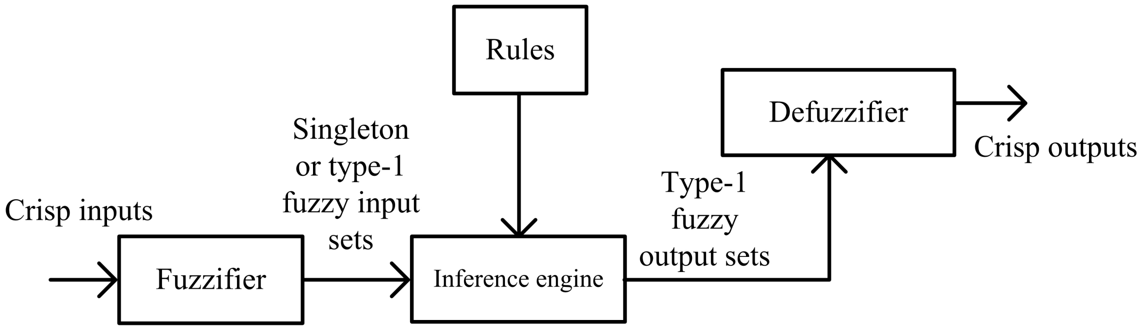


Figure 2.9: The structure of type-1 fuzzy logic system

In type-1 fuzzy logic system, the antecedent and consequent sets are type-1 fuzzy sets; while in a type-2 fuzzy logic system, antecedent and consequent sets are type-2 fuzzy sets. Fig. 2.10 shows the structural diagram of type-2 fuzzy logic system comprising of five components—fuzzifier, rule base, inference engine, type-reducer and defuzzifier. In the implementation of a type-2 fuzzy logic system, crisp inputs are mapped via the fuzzifier into fuzzy sets called fuzzy input sets which can be singleton, type-1 or type-2 fuzzy sets, depending on the requirement of practical application. Then the operation between fuzzy input sets and antecedent sets produce the firing set for each rule, which is a type-1 set. The combination of the firing sets and the corresponding consequent sets will output a type-2 fuzzy set called fuzzy output set. The type-reducer reduces this output set into a type-1 fuzzy set, and the defuzzifier maps this interval type-1 fuzzy set into a crisp output value.

An interval type-2 fuzzy logic system is similar to a type-2 fuzzy logic system in components, except that in the former the antecedent and consequent sets are interval type-2 fuzzy sets, while in the latter, the antecedent and consequent sets are type-2 fuzzy sets. Unlike a type-2 fuzzy logic system where the fuzzifier can map crisp inputs into singleton, type-1 or type-2 fuzzy sets, the output of the fuzzifier in an interval type-2 fuzzy logic system can be singleton, type-1 or interval type-2 fuzzy sets. As a result, the firing set for each rule is an interval set instead of a type-1 set and the output of the inference engine is an interval type-2 fuzzy set instead of a type-2 fuzzy set. The type-reducer converts the output of the inference engine into an interval set.

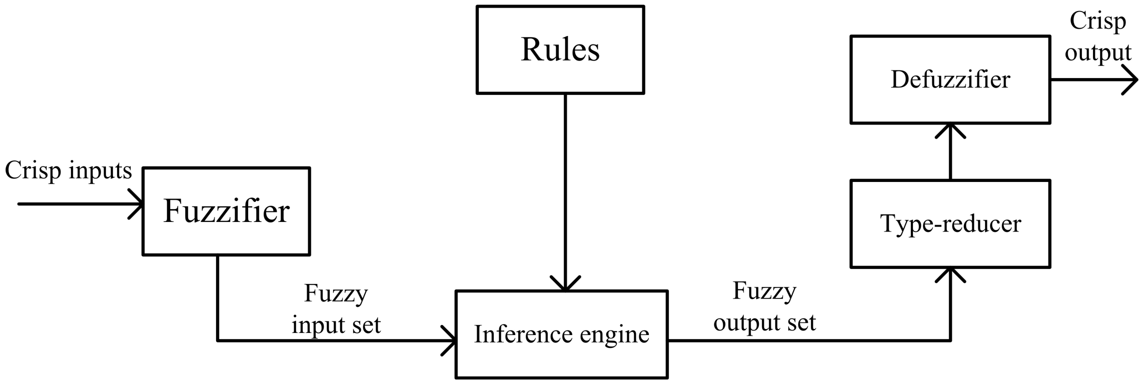


Figure 2.10: The structure of type-2 fuzzy logic system

2.3.2 The Sup-star Composition Inference System

Since the focus of this thesis is on interval type-2 fuzzy logic, this subsection reviews the implementation of an interval type-2 fuzzy logic system.

Consider a multi-inputs single-output interval type-2 FLS with p inputs $x_1 \in X_1, \cdots, x_p \in X_p$ and the output $y \in Y$. The rule base comprises the following

IF-THEN statement:

R^i : IF x_1 is $\tilde{F}_1^i$ AND$\cdots$ AND x_p is $\tilde{F}_p^i$, THEN y is $\tilde{G}^i$ $i = 1, \cdots, M$

where all $\tilde{F}_j^i, i = 1, \cdots, M, j = 1, \cdots, p$ are interval type-2 fuzzy sets. These rules provide a mapping from the input space $X_1 \times \cdots X_p$ to the output space Y.

The sup-star composition is the widely adopted computation for a type-1 fuzzy system. This computation can be extended to an interval type-2 fuzzy logic system by replacing type-1 fuzzy sets in the operation with interval type-2 fuzzy sets and the *sup*, t-norm operation with the join and the meet operation. Then each rule can be interpreted as

$$R^i : \quad \tilde{F}_1^i \times \cdots \times \tilde{F}_p^i \quad \rightarrow \tilde{G}^i = \tilde{A}^i \rightarrow \tilde{G}^i \quad i = 1, \cdots, M \tag{2.51}$$

Let the membership function $\mu_{R^i}(x, y) = \mu_{R^i}(x_1, \cdots, x_p, y)$ be R^i, i.e.

$$\mu_{R^i}(x, y) = \mu_{\tilde{A}^i \rightarrow \tilde{G}^i}(x, y) \tag{2.52}$$

Then

$$\begin{aligned} \mu_{R^i}(x, y) = \mu_{\tilde{A}^i \rightarrow \tilde{G}^l}(x, y) = \mu_{\tilde{F}_1^i}(x_1) \sqcap \cdots \sqcap \mu_{\tilde{F}_p^i}(x_p) \sqcap \mu_{\tilde{G}^i}(y) \\ = [\sqcap_{j=1}^p \mu_{\tilde{F}_j^i}(x_j)] \sqcap \mu_{\tilde{G}^i}(y) \end{aligned} \tag{2.53}$$

The p-dimensional input to R^i is given by the type-2 fuzzy set $\tilde{A}_x$, whose membership function is

$$\mu_{\tilde{A}_x} = \mu_{\tilde{X}_1}(x_1) \sqcap \cdots \sqcap \mu_{\tilde{X}_p}(x_p) = \sqcap_{j=1}^p \mu_{\tilde{X}_j}(x_i) \tag{2.54}$$

where $\tilde{X}_j (j = 1, \cdots, p)$ are the labels of the fuzzy sets describing the inputs. Each rule R^i determines a type-2 fuzzy set $\tilde{B}^i = \tilde{A}_x \cdot R^i$ such that

$$\mu_{\tilde{B}^i}(y) = \mu_{\tilde{A}_x \cdot R^i}(y) = \sqcup_{x \in X}[\mu_{\tilde{A}_x} \sqcap \mu_{R^i}(x, y)] \quad y \in Y, i = 1, \cdots, M \tag{2.55}$$

By substituting (2.53) and (2.54) into (2.55), it follows

$$\begin{aligned}
\mu_{\tilde{B}^i}(y) &= \sqcup_{x\in X}[\mu_{\tilde{A}_x} \sqcap \mu_{R^i}(x,y)] \\
&= \sqcup_{x\in X}[\sqcap_{j=1}^{p}\mu_{\tilde{X}_j}(x_j)] \sqcap [\sqcap_{j=1}^{p}\mu_{\tilde{F}_j^i}(x_j)] \sqcap \mu_{\tilde{G}^i}(y) \\
&= \sqcup_{x\in X}[\sqcap_{j=1}^{p}\mu_{\tilde{X}_j}(x_j)] \sqcap \mu_{\tilde{F}_j^i}(x_j)] \sqcap \mu_{\tilde{G}^i}(y) \\
&= \mu_{\tilde{G}^i}(y) \sqcap \{[\sqcup_{x_1\in X_1}\mu_{\tilde{X}_1}(x_1) \sqcap \mu_{\tilde{F}_1^i}(x_1)] \sqcap \\
&\quad \cdots \sqcap [\sqcup_{x_p\in X_p}\mu_{\tilde{X}_p}(x_p) \sqcap \mu_{\tilde{F}_p^i}(x_p)]\} \qquad (2.56)
\end{aligned}$$

Equation (2.56) follows in part from the commutativity of the meet operator using the minimum or product function and the fact that $\mu_{\tilde{X}_j}(x_j) \sqcap \mu_{\tilde{F}_j^i}(x_j)$ is only a function of x_j. Hence, the result of each join operation is just a scalar variable. Summarily, the implementation of an interval type-2 fuzzy logic system can be formally stated as:

Theorem 2.2 *[54] For an interval singleton type-2 fuzzy logic system using product or minimum t-norm to implement the meet operator:*

1. *the result of the input and antecedent operations, which are contained in the firing set $F_i(x')$ is an interval type-1 set, i.e.*

$$F_i(x') = [\underline{f}_i(x'), \overline{f}_i(x')] = [\underline{f}_i, \overline{f}_i] \qquad (2.57)$$

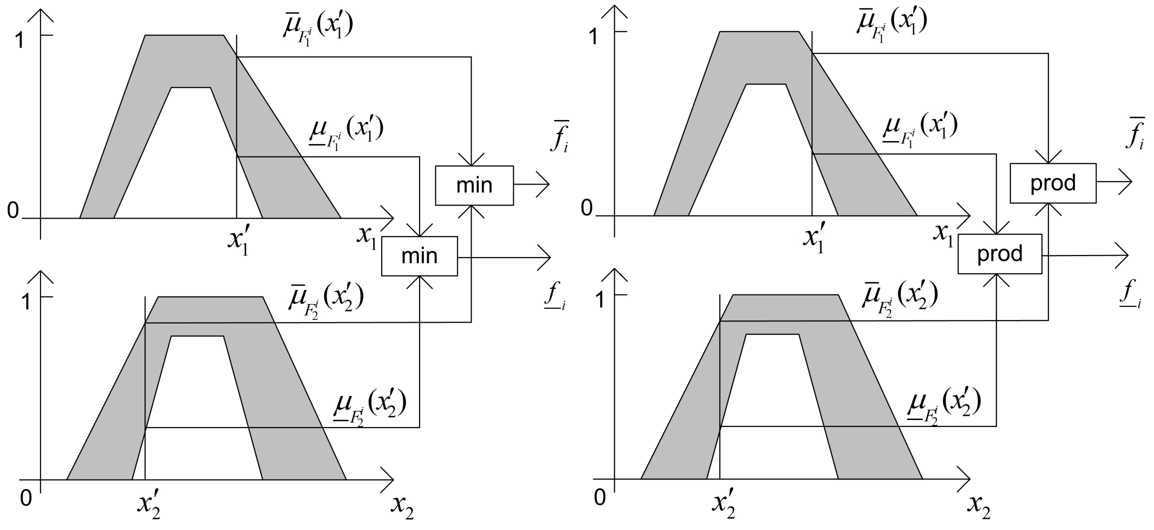
where

$$\underline{f}_i(x') = \underline{\mu}_{\tilde{F}_1^i}(x_1') \star \cdots \star \mu_{\tilde{F}_p^i}(x_p') \qquad (2.58)$$

$$\overline{f}_i(x') = \mu_{\tilde{F}_1^i}(x_1') \star \cdots \star \overline{\mu}_{\tilde{F}_p^i}(x_p) \qquad (2.59)$$

2. *the rule R^i fired output consequent, $\mu_{\tilde{B}_i}(y)$ is the type-1 fuzzy set:*

$$\mu_{\tilde{B}_i} = \int_{b^i\in[\underline{f}_i\star\underline{\mu}_{\tilde{G}_i}(y),\overline{f}_i\star\overline{\mu}_{\tilde{G}_i}(y)]} 1/b_i \quad y\in Y \qquad (2.60)$$


1
0
$\overline{\mu}_{F_1^i}(x_1')$
$\underline{\mu}_{F_1^i}(x_1')$
x_1'
x_1
min
min
$\overline{f}_i$
$\underline{f}_i$
$\overline{\mu}_{F_2^i}(x_2')$
$\underline{\mu}_{F_2^i}(x_2')$
x_2'
x_2
prod
prod

(a) (b)

Figure 2.11: Pictorial description of input and antecedent operation for an interval singleton type-2 fuzzy logic system. (a) minimum t-norm, and (b) product t-norm

where $\underline{\mu}_{\tilde{G}_i}(y)$ *and* $\overline{\mu}_{\tilde{G}_i}(y)$ *are the lower and upper membership grades of* $\mu_{\tilde{G}_i}(y)$.

3. *suppose that* N *of the* M *rules in the FLS are fired, where* $N \leq M$*, and the combined output type-1 fuzzy set is obtained by combining the fired output consequent set, i.e.* $\mu_{\tilde{B}}(y) = \sqcap_{i=1}^{N} \mu_{\tilde{B}_i}(y), \quad y \in Y$*, then*

$$\mu_{\tilde{B}}(y) = \int_{b \in [[\underline{f}_1 \star \underline{\mu}_{\tilde{G}_1}(y)] \vee \cdots \vee [\underline{f}_1 \star \underline{\mu}_{\tilde{G}_1}(y)], [\overline{f}_1 \star \overline{\mu}_{\tilde{G}_1}(y)] \vee \cdots \vee [\overline{f}_1 \star \overline{\mu}_{\tilde{G}_1}(y)]]} 1/b, \quad y \in Y (2.61)$$

Several defuzzification methods such as center-of-sets, centroid, height have been developed. Depending on the adopted defuzzification method, Theorem 2.2 may be interpreted in different manners. Since this thesis involves the interval type-2 fuzzy logic system using widely adopted center-of-sets defuzzification only, the procedures of implementing a singleton interval type-2 fuzzy logic system using the Karnik-Mendel type-reducer and the center-of-sets defuzzificantion are provided based on Theorem 2.2 as follows:

- Step 1: Compute the firing set $F_i(x') = [\underline{f}_i(x'), \overline{f}_i(x')]$ for each rule R^i by intersecting the antecedent sets and the firing strength according to

$$F_i(x') = [\underline{f}_i(x'), \overline{f}_i(x')] = [\underline{f}_i, \overline{f}_i] \tag{2.62}$$

where

$$\underline{f}_i(x') = \underline{\mu}_{\tilde{F}_1^i}(x'_1) \star \cdots \star \underline{\mu}_{\tilde{F}_p^i}(x'_p) \tag{2.63}$$

$$\overline{f}_i(x') = \overline{\mu}_{\tilde{F}_1^i}(x'_1) \star \cdots \star \overline{\mu}_{\tilde{F}_p^i}(x_p) \tag{2.64}$$

Fig. 2.11 gives a pictorial description of calculating $\underline{f}_i$ and $\overline{f}_i$ according to (2.63) and (2.64).

- Step 2: Combine the firing sets $[\underline{f}_i, \overline{f}_i]$ and the corresponding singleton consequent sets $y_i, i = 1, 2, \cdots, M$, resulting in an interval type-2 fuzzy set in which the primary variable consists of the points $y_i, i = 1, 2, \cdots, M$ and the primary membership grade is $[\underline{f}_i, \overline{f}_i]$. Fig. 2.12 shows the resulted interval type-2 fuzzy set.

- Step 3: Compute the type-reduced set $[y_l, y_r]$ using the Karnik-Mendel iterative algorithm:

$$y_l = \frac{\sum_{i=1}^{L} \overline{f}_i y_i + \sum_{i=L+1}^{M} \underline{f}_i y_i}{\sum_{i=1}^{L} \overline{f}_i + \sum_{i=L+1}^{M} \underline{f}_i} \tag{2.65}$$

$$y_r = \frac{\sum_{i=1}^{R} \underline{f}_i y_i + \sum_{i=R+1}^{M} \overline{f}_i y_i}{\sum_{i=1}^{R} \underline{f}_i + \sum_{i=R+1}^{M} \overline{f}_i} \tag{2.66}$$

where L and R are the switch point satisfying

$$y_L \leq y_l < y_{L+1} \tag{2.67}$$

$$y_R \leq y_r < y_{R+1} \tag{2.68}$$

Fig. 2.13 shows a pictorial description of y_l and y_r.

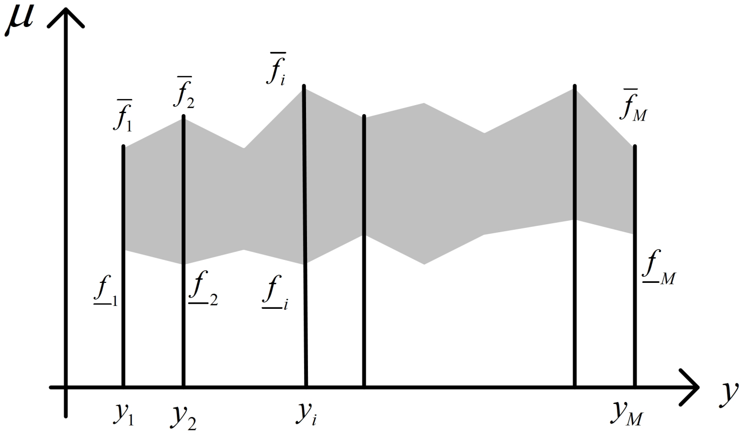
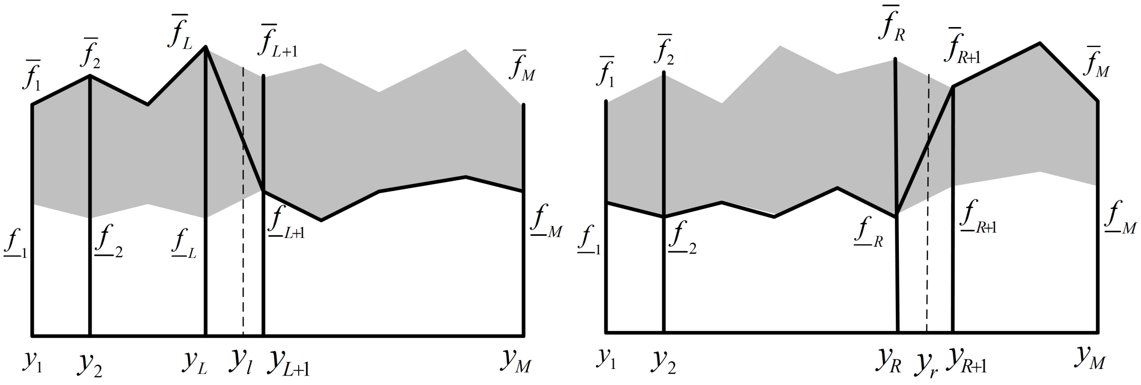


Figure 2.12: The output of the inference engine of an IT2 FLS ($y_i, i = 1, 2, \cdots, M$ represent the points where singleton consequent set have unity membership grade; $\underline{\mathrm{f}}^i$ and $\bar{f}_i$ are the lower and upper bound of the firing set for the ith rule; M is the number of fired rules.)



(a) (b)

Figure 2.13: Pictorial description of type-reduction. (a) y_l (b) y_r.

- Step 4: The result of the type-reduction is an interval set called type-reduced set, i.e. $[y_l, y_r]$. The output of an interval type-2 fuzzy system is the average of the type-reduced set, i.e.

$$y = \frac{y_l + y_r}{2} \tag{2.69}$$

Chapter 3

Analytical Structure and Characteristics of Symmetrical Karnik-Mendel Type-Reduced Interval Type-2 Fuzzy PI and PD Controllers

3.1 Introduction

Fuzzy logic system theory (Type-1 fuzzy logic system, T1 FLS), first proposed by Zadeh in 1965, has been widely applied for control, signal processing and pattern recognition. T1 FLS was introduced for modelling and handling the uncertainties in real systems; however, it has been demonstrated to be limited in modelling the uncertainties in the shape and position of the fuzzy membership set. To overcome this limitation, the interval type-2 (IT2) fuzzy logic system (FLS), in which IT2 fuzzy sets (FSs) act as antecedent sets or consequent sets, was introduced. A reason why IT2 FLS may be able to outperform its T1 counterpart is the additional design freedoms provided by the footprint of uncertainty (FOU) associated with the membership function of an IT2 FS. Numerical study has shown that IT2 FLS can outperform T1 counterpart in some control problems [10, 12, 14, 20, 22, 42, 71, 88, 89, 90]. However, the simulation analysis cannot be generalized and applied into other systems due to the limitation of case studies. While theoretical study of IT2 FLS may provide more general results, it is a challenging task.

For T1 fuzzy logic controllers (FLCs), the analytical structure approach has been established as a useful tool to identify the relationship between the system inputs and output [2, 9, 97, 98, 99, 100, 101, 103, 104, 105]. Most notably in [105], a class of Mamdani fuzzy PD and PI controllers with the Zadeh AND and linear FSs (triangular or trapezoidal) for the input variables were proved to be equivalent to nonlinear PD and PI controllers with variable gains. The mathematical study is performed by first dividing the input space into several subregions, and then determining the relationship between the firing strength and the inputs i.e. whether the firing strength is 0, 1 or the membership of some antecedent set. Next, the FLC was expressed as nonlinear PD or PI controller by replacing all the firing strength with the specific expressions in every subregion. As pointed out [105], it is the variable gains in different subregions that made it possible for a T1 FLC to outperform conventional PD/PI controllers. Generally speaking, this analytical structure theory of T1 fuzzy PD and PI controllers provides a general framework to perform theoretical study of fuzzy systems in these aspects: (1) insights into the internal mechanism of T1 FLC — structurally PD and PI controller with variables gains; (2) the advantages over conventional PD and PI controller; (3) the characteristics of fuzzy logic system with different configurations; (4) design methods including the tuning of the parameters [98].

An effort [15] has been made to derive the analytical structure of a class of IT2 FLCs. Instead of using common type-reduction methods, the IT2 FLC analyzed in [15] approximates the type-reduced set by averaging 16 embedded IT2 FLSs. Unlike [15], this chapter considers the IT2 FLCs that use the widely adopted Karnik-Mendel (KM) type-reduction method. In the KM type-reduction method, the domain of an IT2 FS is interpreted as a collection of embedded T1 FSs. As the centroids of these embedded T1 FSs would form an interval, the type-reduced T1 FS is bounded by the minimum and maximum centroid of all embedded T1 FSs. It has been proved that for an IT2 FS the bounds y_l and y_r of the type-reduced set are the centroids of the two embedded T1 sets that involves one switch between the lower and upper membership function of the IT2 FS respectively[54], i.e.

$$y_l = \frac{\sum_{i=1}^{L} \overline{\mu}_i y_i + \sum_{i=L+1}^{m} \underline{\mu}_i y_i}{\sum_{i=1}^{L} \overline{\mu}_i + \sum_{i=L+1}^{m} \underline{\mu}_i} \tag{3.1}$$

$$y_r = \frac{\sum_{i=1}^{R} \underline{\mu}_i y_i + \sum_{i=R+1}^{m} \overline{\mu}_i y_i}{\sum_{i=1}^{R} \underline{\mu}_i + \sum_{i=R+1}^{m} \overline{\mu}_i} \tag{3.2}$$

where the primary variable $y_i, i = 1, 2, \cdots, m$ satisfies $y_i < y_{i+1}$; $\overline{\mu}_i$ and $\underline{\mu}_i$ are respectively the upper and lower bound of its primary membership. From the derivation of the KM type-reduction method, the following property can be observed [54] as presented in (2.31) and (2.32):

$$y_L \le y_l < y_{L+1} \tag{3.3}$$

$$y_R \le y_r < y_{R+1} \tag{3.4}$$

This type-reduction process is generally implemented by using the Karnik-Mendel (KM) algorithm to determine the bounds iteratively. Due to the iterative nature of the KM type-reduction process, the relationship between the output of an IT2 FLS and the input signals cannot be expressed in closed form, thereby impeding the theoretical study of IT2 FLSs. Hence, a key result of this chapter is a methodology that enables the input space to be partitioned such that the output for an IT2 fuzzy logic system may be explicitly related to the firing strength of all rules. Comparing with their T1 counterpart, there are more parameters in antecedent sets or consequent sets of an IT2 FLS. These additional parameters may create more partitions in the input space and modify the relationship between the input and output signals. This chapter also seeks to identify the advantages that the additional FOU parameters may provide for a class of simple fuzzy PD and PI controllers with symmetrical rule base and symmetrical consequent sets.

The rest of the chapter is organized as follows : Section II describes the configurations of IT2 fuzzy PD and PI controllers; Section III presents the algorithm to derive their analytical structure followed by detailed derivation of the analytical structure of IT2 FLCs in Section IV. Readers who are interested in understanding the potential performance improvements due to the inclusion of the footprint of uncertainty may jump directly to Section V, where four interesting characteristics are presented to explain why an IT2 fuzzy controller may better achieve the conflicting aims of fast rise time and small overshoot. Finally, Section VI presents results from a comparative study between IT2 fuzzy PD controller and its T1 counterpart to highlight the advantages provided by the extra design parameters.

3.2 Configuration of Interval T2 Fuzzy PD and PI Controller

As shown in in Fig. 3.1, a two-inputs single-output fuzzy system commonly used for feedback control may be defined as

$$\triangle U(n) = f(E(n), R(n)) \tag{3.5}$$

where $E(n) = K_e e(n) = K_e(SP(n) - y(n))$, $R(n) = K_r r(n) = K_r(e(n) - e(n-1))$, $y(n)$ is the output of the closed-loop system, $SP(n)$ is the reference signal, K_e and K_r are the scaling constants. The output $\triangle U(n)$ may be interpreted directly as the control signal or as the rate of change in the actuation signal. Depending how the output of the fuzzy controller is defined, it may be interpreted as a fuzzy PD or a fuzzy PI controller.

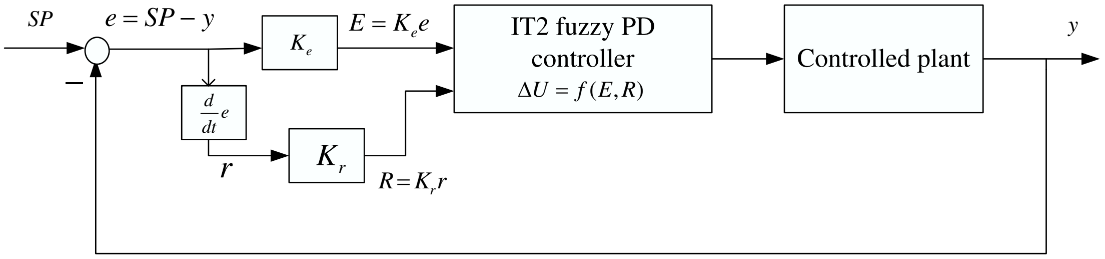


Figure 3.1: The structure of a fuzzy PD control system

Fig. 3.2 contains the schematic diagram of an IT2 FLS. For this study, two IT2 FSs are used to partition the space of each input: EN and EP for $E(n)$, RN and RP for $R(n)$. By shifting the membership of two symmetrical T1 FSs horizontally by the amount of θ_1, the upper and lower membership of EN and EP, $\overline{\mathbf{EN}}$, $\underline{\mathbf{EN}}$, $\overline{\mathbf{EP}}$ and $\underline{\mathbf{EP}}$ in Fig. 3.3(a), can be obtained [15]. In the same

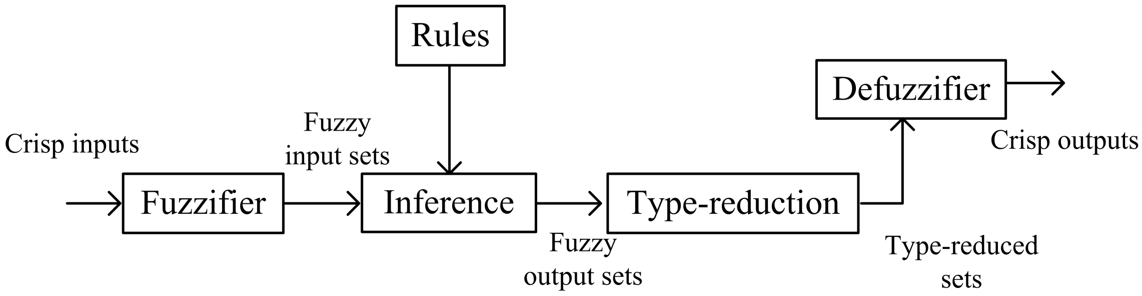


Figure 3.2: The structure of IT2 FLS

way, θ_2 represents the amount by which the upper and lower membership of RN and RP are shifted to obtain $\overline{\mathbf{RN}}$, $\underline{\mathbf{RN}}$, $\overline{\mathbf{RP}}$ and $\underline{\mathbf{RP}}$ in Fig. 3.3(b). Based on whether its value depends on the inputs, the lower bound or upper bound of any antecedent set can be decomposed into dependent part or independent segment (0 or 1). For example,

$$\overline{\mathbf{EN}} = \begin{cases} \overline{EN} = -\frac{1}{2L_1}E(n) + 0.5 + \theta_1 & \text{for } -L_1 + P_1 \leq E(n) \leq L_1 + P_1 \\ 0 & \text{for } E(n) \geq L_1 + P_1 \\ 1 & \text{for } E(n) \leq -L_1 + P_1 \end{cases} \tag{3.6}$$

where $\overline{EN}$ is the segment of the membership function of $\overline{\mathbf{EN}}$ that is linearly related with the input, while the membership grade of the other two segments of $\overline{\mathbf{EN}}$ is fixed at 0 or 1. (3.6) reveals the relationship between the membership grade and the input.

For fuzzy systems that partition the input space using IT2 FSs shown in Fig. 3.3(a) and 3.3(b), a commonly used rule base comprises of the following IF-THEN statements:

- Rule 1: IF $E(n)$ is Positive AND $R(n)$ is Positive THEN $\triangle U(n)$ is H_1

- Rule 2: IF $E(n)$ is Positive AND $R(n)$ is Negative THEN $\triangle U(n)$ is H_2
- Rule 3: IF $E(n)$ is Negative AND $R(n)$ is Positive THEN $\triangle U(n)$ is H_3
- Rule 4: IF $E(n)$ is Negative AND $R(n)$ is Negative THEN $\triangle U(n)$ is H_4

where H_1, H_2, H_3 and H_4 are the four singleton consequent FSs.

The result of the input and antecedent operations is an IT1 set called the firing set. Using Zadeh fuzzy AND operator as the t-norm operator, the firing set for each rule are as follows:

$$R_1 = [\underline{R}_1, \overline{R}_1] = [\min(\underline{\mathbf{EP}}, \underline{\mathbf{RP}}), \min(\overline{\mathbf{EP}}, \overline{\mathbf{RP}})] \quad \text{for} \triangle U(n) = H_1 \tag{3.7}$$

$$R_2 = [\underline{R}_2, \overline{R}_2] = [\min(\underline{\mathbf{EP}}, \underline{\mathbf{RN}}), \min(\overline{\mathbf{EP}}, \overline{\mathbf{RN}})] \quad \text{for} \triangle U(n) = H_2 \tag{3.8}$$

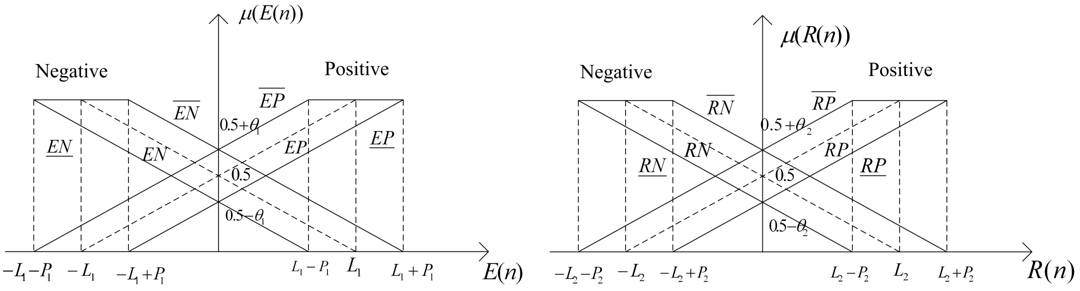
$$R_3 = [\underline{R}_3, \overline{R}_3] = [\min(\underline{\mathbf{EN}}, \underline{\mathbf{RP}}), \min(\overline{\mathbf{EN}}, \overline{\mathbf{RP}})] \quad \text{for} \triangle U(n) = H_3 \tag{3.9}$$

$$R_4 = [\underline{R}_4, \overline{R}_4] = [\min(\underline{\mathbf{EN}}, \underline{\mathbf{RN}}), \min(\overline{\mathbf{EN}}, \overline{\mathbf{RN}})] \quad \text{for} \triangle U(n) = H_4 \tag{3.10}$$

The IT2 FLS using the Karnik-Mendel type-reduction and center-of-sets defuzzification method is the research objective of this chapter. Using the Mendel-John Representation theorem [60] (Theorem 1), the IT2 FS formed by the fuzzy inference engine may be viewed as the collection of all of its embedded IT1 FSs. Hence, the output of the inference engine may be type-reduced into an IT1 set comprising of the centroids of all embedded T1 FSs.

$$\triangle U_j = \frac{R_1^* * H_1 + R_2^* * H_2 + R_3^* * H_3 + R_4^* * H_4}{R_1^* + R_2^* + R_3^* + R_4^*} \tag{3.11}$$

where R_i^* is a value within the lower and upper bound of the firing set for the ith rule, R_i. In summary, the type-reduced set $\triangle U_{TR}(n)$ may be expressed mathematically as $[\triangle U_j^{min}, \triangle U_j^{max}]$, $\triangle U_j^{min}$ and $\triangle U_j^{max}$ are respectively the smallest



(a) (b)

Figure 3.3: IT2 antecedent FSs: (a) IT2 FSs EN and EP for the input $E(n)$ $(P_1 = 2L_1\theta_1)$. (b) IT2 FSs RN and RP for the input $R(n)$ $(P_2 = 2L_2\theta_2)$.

and largest centroid of all the possible embedded T1 FS. Lastly, using the height defuzzification, the crisp output of the IT2 FLC is

$$\triangle U(n) = \frac{\triangle U_j^{min} + \triangle U_j^{max}}{2} \tag{3.12}$$

It has been proved that the upper and lower bound of the type-reduced set, $\triangle U_j^{min}$ and $\triangle U_j^{max}$, are respectively the centroids of two unique embedded type-1 sets that involves only one switch between the lower and upper MF of the IT2 fuzzy set produced by the fuzzy inference engine [54]. The position of the switch point depends on the values of the singleton consequent sets. Hence, unlike T1 fuzzy controller where the partitions of the input space are independent of the consequent sets, there is a need for the following assumptions in order to simplify the derivation of the analytical structure of IT2 fuzzy PD controller :

1. The rule base is symmetrical. In other words, $H_2 = H_3$.

2. H_1, $H_2 = H_3$ and H_4 are equally spaced in Fig. 3.4. The assumption $H_4 < H_3 = H_2 < H_1$ is made based on the observation that the output of a linear PD controller increases as any input increases.

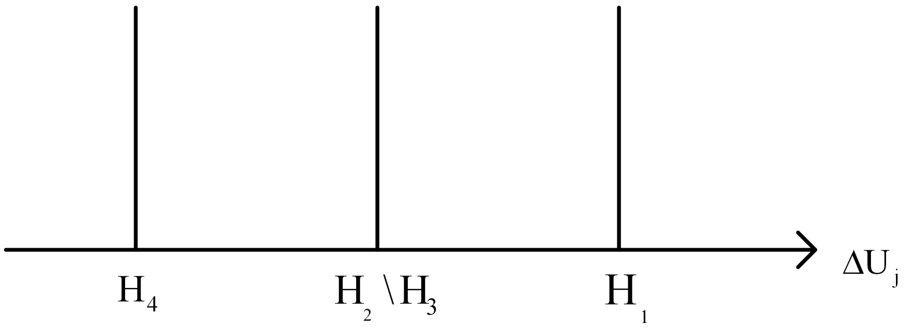


Figure 3.4: Singleton consequent FSs of IT2 fuzzy PD controller

3.3 Analysis of the Karnik-Mendel Type-Reduced IT2 Fuzzy PD Controller

The analytical structure of a FLS may be established by deriving the mathematical relationship between the inputs and the output. Similar to the case of a T1 FLC, the main concept used to determine the input-output relationship in an IT2 fuzzy PD controller is to specify the firing strength by dividing the input space into regions and to replace each firing strength with its corresponding mathematical expression. For an IT2 FLC, the key step is to identify equations for each firing strength that should be used to calculate $\triangle U_j^{min}$ and $\triangle U_j^{max}$.

(3.1) and (3.2) show that $\triangle U_j^{min}$ and $\triangle U_j^{max}$ may be expressed as the average of all singleton consequents weighted by the lower or upper bound of the firing strength and an embedded T1 FS. Once the switch points L in (3.1) and R in (3.2) are known, then the problem of analyzing the structure of an IT2 fuzzy PD controller reduces to a T1 system that can be studied using the following well-established techniques [102]:

1. Partition the input space into regions by applying the Zadeh AND operation (minimum operator) to the antecedent membership functions;

2. Determine the specific expression for each firing strength respectively in the corresponding subregion of each embedded T1 FLS.

The foundation stone of the the proposed method for deriving the analytical structure of an IT2 fuzzy PD controller is the identification of the embedded T1 FS. For the IT2 FLC described in Section 3.2, the type-reduced set may be constructed from (3.1) and (3.2) by making the following substitution as

$$\overline{\mu}_i = \overline{R}_j, \quad \underline{\mu}_i = \underline{R}_j, \quad y_i = H_j \tag{3.13}$$

where $\overline{R}_j$ and $\underline{R}_j$ are the upper and lower bound of the firing set associated with the jth rule in (3.7)-(3.10). Furthermore, since it is assumed that $H_4 < H_3 = H_2 < H_1$, (3.1) and (3.2) may be re-expressed as

$$\triangle U_j^{min} = \frac{\sum_{i=1}^{L-1} \underline{R}_i H_i + \sum_{i=L}^{4} \overline{R}_i H_i}{\sum_{i=1}^{L-1} \underline{R}_i + \sum_{i=L}^{4} \overline{R}_i} \tag{3.14}$$

$$\triangle U_j^{max} = \frac{\sum_{i=1}^{R-1} \overline{R}_i H_i + \sum_{i=R}^{4} \underline{R}_i H_i}{\sum_{i=1}^{R-1} \overline{R}_i + \sum_{i=R}^{4} \underline{R}_i} \tag{3.15}$$

where the switch point L and R coincide with the location of one of the 3 singleton consequent sets (H_1, $H_2 = H_3$, H_4). In the case of the IT2 fuzzy PD controller, (3.3) and (3.4) which are the conditions for finding the switch points L and R are

$$H_L \leq \triangle U_j^{min} < H_{L-1} \tag{3.16}$$

$$H_R \leq \triangle U_j^{max} < H_{R-1} \tag{3.17}$$

Due to the constraint that the switch point L and R must be positioned at one of the three singleton consequent sets and the inequalities in (3.16) and

(3.17), it may be concluded that L and R can assume one of only two values i.e. $H_L = \{H_4, H_2 = H_3\}$ and $H_R = \{H_4, H_2 = H_3\}$. Hence, the only two possible expressions for $\triangle U_j^{min}$ may be written as

1. Mode 1: When $H_4 \leq \triangle U_j^{min} \leq H_2 = H_3 \Leftrightarrow$ The left switch point L coincides with H_4,

$$\triangle U_j^{min} = \triangle U_{j1}^{min} = \frac{\overline{R}_4 * H_4 + \underline{R}_3 * H_3 + \underline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \underline{R}_3 + \underline{R}_2 + \underline{R}_1} \tag{3.18}$$

2. Mode 2: When $H_2 = H_3 \leq \triangle U_j^{min} \leq H_1 \Leftrightarrow$ The left switch point L coincides with $H_2 = H_3$

$$\triangle U_j^{min} = \triangle U_{j2}^{min} = \frac{\overline{R}_4 * H_4 + \overline{R}_3 * H_3 + \overline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \overline{R}_3 + \overline{R}_2 + \underline{R}_1} \tag{3.19}$$

By comparing the expressions of Mode 1 and 2 in (3.18) and (3.19), their properties can be generalized as

1. The weight associated with H_1 is always the lower bound of the firing set for Rule 1. Similarly, H_4 is weighted by the upper bound of the firing set for Rule 4. Consequently, the weight on H_1 and H_4 is independent of the switch points L and R.

2. In Mode 1, $H_2 = H_3$ are weighted by the lower bounds $\underline{R}_2$ and $\underline{R}_3$, while the corresponding upper bounds $\overline{R}_2$ and $\overline{R}_3$ are used to weight $H_2 = H_3$ in Mode 2. The condition when the switch point changes from the position of $H_2 = H_3$ to H_1 and vice versa can be established as:

$$\triangle U_j^{min} = \triangle U_{j1}^{min} = \triangle U_{j2}^{min} = H_2 = H_3 \tag{3.20}$$

By replacing $\triangle U_{j1}^{min}$ and $\triangle U_{j2}^{min}$ with their corresponding expressions in (3.18) and (3.19), the above equation may be written as :

$$\begin{aligned}
&\frac{\overline{R}_4 * H_4 + \underline{R}_3 * H_3 + \underline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \underline{R}_3 + \underline{R}_2 + \underline{R}_1} \\
&= \frac{\overline{R}_4 * H_4 + \overline{R}_3 * H_3 + \overline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \overline{R}_3 + \overline{R}_2 + \underline{R}_1} = H_2 = H_3 \\
&\Leftrightarrow \overline{R}_4(H_4 - H_2) = \underline{R}_1(H_2 - H_1) \;/\; \overline{R}_4(H_4 - H_3) = \underline{R}_1(H_3 - H_1)
\end{aligned}$$

Due to the assumption that the three consequent sets are equally spaced and $H_4 < H_3 = H_2 < H_1$, the condition derived above may be simplified to

$$\overline{R}_4 = \underline{R}_1 \tag{3.21}$$

Further, the subregions where (3.16) is used to calculate the left endpoint of the type-reduced set (Mode 1) should satisfy the condition

$$\triangle U_{j1}^{min} \leq H_2 = H_3 \Leftrightarrow \overline{R}_4 \geq \underline{R}_1 \tag{3.22}$$

and the areas where the IT2 fuzzy PD controller operate in Mode 2, i.e. the output is defined by (3.17), can be found using the following equality:

$$\triangle U_{j2}^{min} \geq H_2 = H_3 \Leftrightarrow \overline{R}_4 \leq \underline{R}_1 \tag{3.23}$$

The first property shows that the firing strength of Rule 1 and Rule 4 used to calculate $\triangle U_j^{min}$ is independent of the switch point L, and the firing strength of Rule 1 and Rule 4 is always governed by the lower and upper bound of the firing set i.e. $\underline{R}_1$ and $\overline{R}_4$. Furthermore, the boundary defined by the conditions in (3.22)-(3.23) also depends on $\underline{R}_1$ and $\overline{R}_4$. Based on this observation, the first step in partitioning the input space in order to derive closed form firing level is performed by considering the outcomes of Zadeh AND operations (minimum operator) for Rule 1 and Rule 4. Next, the relative firing strengths of Rule 1 and Rule 4 in each sub-space is compared in order to determine whether the IT2 fuzzy PD controller will operate in Mode 1 and Mode 2. In the subregions under Mode 1 and 2, the embedded type-1 set is completely defined by (3.18) and (3.19) so the partitions can be found using existing technique i.e. ascertaining the minimum of the lower or upper bound of the firing sets. In summary, an algorithm to derive the partitions of the input space by $\triangle U_j^{min}$ can be generalized as:

- *Step 1* : The firing strength of Rule 1 ($\underline{R}_1$) and Rule 4 ($\overline{R}_4$) can be specified by dividing the input space via the outcomes of Zadeh AND operations (minimum operator) for Rule 1 and Rule 4 i.e.

$$\underline{R}_1 = \min\{\underline{\mathbf{EP}}, \underline{\mathbf{RP}}\} \tag{3.24}$$

$$\overline{R}_4 = \min\{\overline{\mathbf{EN}}, \overline{\mathbf{RN}}\} \tag{3.25}$$

- *Step 2* : The partitions obtained using $\underline{R}_1$ and $\overline{R}_4$ in Step 1 is further subdivided into the following two groups that correspond to one of the two possible operating modes :

$$\text{Mode 1 :} \quad \overline{R}_4 > \underline{R}_1 \tag{3.26}$$

$$\text{Mode 2 :} \quad \overline{R}_4 < \underline{R}_1 \tag{3.27}$$

- *Step 3* : To specify the firing strength of Rule 2 and Rule 3 by dividing the corresponding regions for Mode 1 and Mode 2 using (3.18) and (3.19) respectively. Under Mode 1, the firing strength of Rule 2 and Rule 3 is given by the lower MF. Hence, partitioning can be achieved from the outcomes of the following Zadeh AND operations :

$$\underline{R}_2 = \min\{\underline{\mathbf{EP}}, \underline{\mathbf{RN}}\} \tag{3.28}$$

$$\underline{R}_3 = \min\{\underline{\mathbf{EN}}, \underline{\mathbf{RP}}\} \tag{3.29}$$

For the regions where the IT2 fuzzy PD controller is operating in Mode 2, the following Zadeh AND operations should be used to divide the input space :

$$\overline{R}_2 = \min\{\overline{\mathbf{EP}}, \overline{\mathbf{RN}}\} \tag{3.30}$$

$$\overline{R}_3 = \min\{\overline{\mathbf{EN}}, \overline{\mathbf{RP}}\} \tag{3.31}$$

- *Step 4* : Superimpose all the partitions obtained by considering the mode switch and Zadeh AND operations.

Similarly, the firing strength in the equation for deriving the right endpoint $\triangle U_j^{max}$ can be specified. The procedures to derive the analytical structure of IT2 fuzzy PD controller can be generalized in Fig. 3.5.

3.4 Derivation of the Analytical Structure of IT2 Fuzzy PD Controller

In this section, the analytical structure of the IT2 fuzzy PD controller will be derived by following the proposed algorithms in last section.

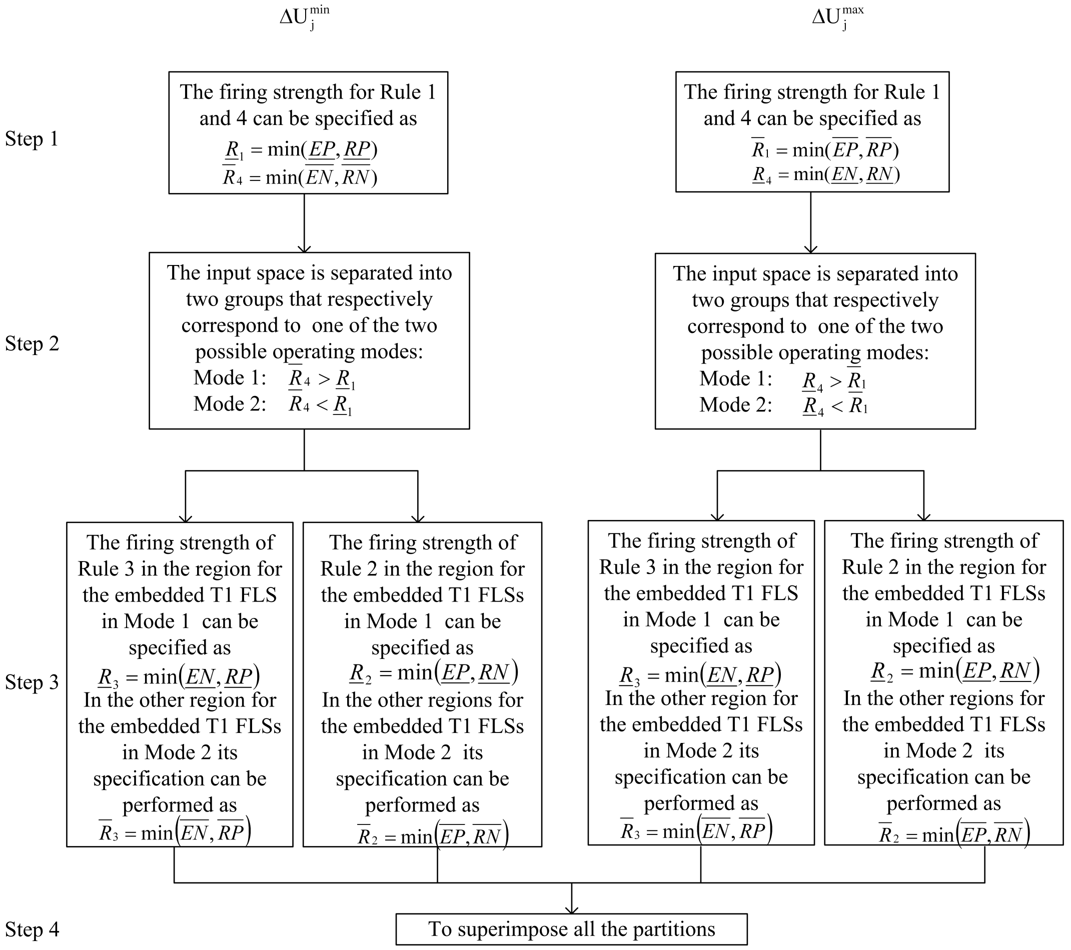

ΔU_j^{min}
ΔU_j^{max}
Step 1
The firing strength for Rule 1 and 4 can be specified as
$\underline{R}_1 = \min(\underline{EP}, \underline{RP})$
$\overline{R}_4 = \min(\overline{EN}, \overline{RN})$
The firing strength for Rule 1 and 4 can be specified as
$\overline{R}_1 = \min(\overline{EP}, \overline{RP})$
$\underline{R}_4 = \min(\underline{EN}, \underline{RN})$
Step 2
The input space is separated into two groups that respectively correspond to one of the two possible operating modes:
Mode 1: $\overline{R}_4 > \underline{R}_1$
Mode 2: $\overline{R}_4 < \underline{R}_1$
The input space is separated into two groups that respectively correspond to one of the two possible operating modes:
Mode 1: $\underline{R}_4 > \overline{R}_1$
Mode 2: $\underline{R}_4 < \overline{R}_1$
Step 3
The firing strength of Rule 3 in the region for the embedded T1 FLS in Mode 1 can be specified as
$\underline{R}_3 = \min(\underline{EN}, \underline{RP})$
In the other region for the embedded T1 FLSs in Mode 2 its specification can be performed as
$\overline{R}_3 = \min(\overline{EN}, \overline{RP})$
The firing strength of Rule 2 in the region for the embedded T1 FLSs in Mode 1 can be specified as
$\underline{R}_2 = \min(\underline{EP}, \underline{RN})$
In the other regions for the embedded T1 FLSs in Mode 2 its specification can be performed as
$\overline{R}_2 = \min(\overline{EP}, \overline{RN})$
The firing strength of Rule 3 in the region for the embedded T1 FLS in Mode 1 can be specified as
$\underline{R}_3 = \min(\underline{EN}, \underline{RP})$
In the other region for the embedded T1 FLSs in Mode 2 its specification can be performed as
$\overline{R}_3 = \min(\overline{EN}, \overline{RP})$
The firing strength of Rule 2 in the region for the embedded T1 FLSs in Mode 1 can be specified as
$\underline{R}_2 = \min(\underline{EP}, \underline{RN})$
In the other regions for the embedded T1 FLSs in Mode 2 its specification can be performed as
$\overline{R}_2 = \min(\overline{EP}, \overline{RN})$
Step 4
To superimpose all the partitions

Figure 3.5: Flowchart of the algorithm to specify the firing strength of IT2 fuzzy PD controller

3.4.1 Input Conditions for Left Endpoint, $\triangle U_j^{min}$

1. Step 1 seeks to partition the input space into regions where the firing strength of Rule 1 ($\underline{R}_1$) and Rule 4 ($\overline{R}_4$) can be expressed mathematically.

   - The firing strength of Rule 1 ($\underline{R}_1$) can be calculated via (3.24)

   $$\underline{R}_1 = \min\{\underline{\mathbf{EP}}, \underline{\mathbf{RP}}\}.$$

   Hence, the input space is divided into three regions: $\underline{R}_1$(IC1, IC2 and IC3) in Fig. 3.6a. In these three regions, the firing strength $\underline{R}_1$ is 0, $\underline{EP}$ and $\underline{RP}$ respectively.

- As the firing strength of Rule 4 is defined as (3.25)

$$\overline{R}_4 = \min\{\overline{\mathbf{EN}}, \overline{\mathbf{RN}}\},$$

the input space is divided into three regions: $\overline{R}_4$(IC1, IC2 and IC3) in Fig. 3.6b. In these three regions, the firing strength $\overline{R}_4$ is $\overline{RN}$, 1 and $\overline{EN}$ respectively.

2. Step 2 separates the input space into regions corresponding to Mode 1 (M_{L1}) and Mode 2 (M_{L2}) operation respectively.

By superimposing the partitions derived in Fig. 3.6 (a-b), five regions are generated: S_{L1}, S_{L2}, S_{L3}, S_{L4} and S_{L5} (boundaries denoted by dashed lines) in Fig. 3.7. Whether the IT2 fuzzy PD controller operates in Mode 1 or Mode 2 in each of the 5 regions depends on the relative values of $\underline{R}_1$ and $\overline{R}_4$.

- Region S_{L1}: $\overline{R}_4 = \overline{RN}$ and $\underline{R}_1 = 0$. Hence, $\overline{R}_4 > \underline{R}_1$ and according to (3.22), an input pair in Region S_{L1} triggers Mode 1 (M_{L1}) operation.
- Region S_{L2}: $\overline{R}_4 = 1$, $\underline{R}_1 = 0$. As a result, $\overline{R}_4 > \underline{R}_1$ and thus the region S_{L2} belongs to M_{L1}.
- Region S_{L3}: $\overline{R}_4 = \overline{EN}$ and $\underline{R}_1 = 0$. Region S_{L3} belongs to M_{L1} due to $\overline{R}_4 > \underline{R}_1$.
- Region S_{L4}: $\overline{R}_4 = \overline{EN}$ and $\underline{R}_1 = \underline{RP}$. The red line segment in Fig. 3.7 denotes the points in S_{L4} where $\overline{EN} = \underline{RP}$. Hence, the red line divides S_{L4} into two regions (unshaded and shaded) corresponding to each of the operating modes. In the unshaded space, $\overline{R}_4 > \underline{R}_1$ so the region belongs to M_{L1}. Conversely, the shaded part corresponds to M_{L2}.

- Region S_{L5}: $\overline{R}_4 = \overline{RN}$ and $\underline{R}_1 = \underline{EP}$. Again, the boundary where $\overline{R}_4 = \underline{R}_1$ is shown as a red line segment in Fig. 3.7. The unshaded and shaded area in S_{L5} corresponds to M_{L1} and M_{L2} respectively.

3. Step 3 partitions the input space according to the firing strength of Rule 2 and Rule 3.

   - The partitions due to Rule 2 and Rule 3 when the controller is working in M_{L1} are specified by (3.28) and (3.29). The result of considering the Zadeh AND operations defined by (3.28) $\underline{R}_2 = \min\{\underline{\mathbf{EP}}, \underline{\mathbf{RN}}\}$ and (3.29) $\underline{R}_3 = \min\{\underline{\mathbf{EN}}, \underline{\mathbf{RP}}\}$ is shown as the green and blue lines in Fig. 3.8 respectively.

   - Region M_{L2} is divided according to the firing strength of Rule 2 and Rule 3 given by (3.30) and (3.31) i.e.

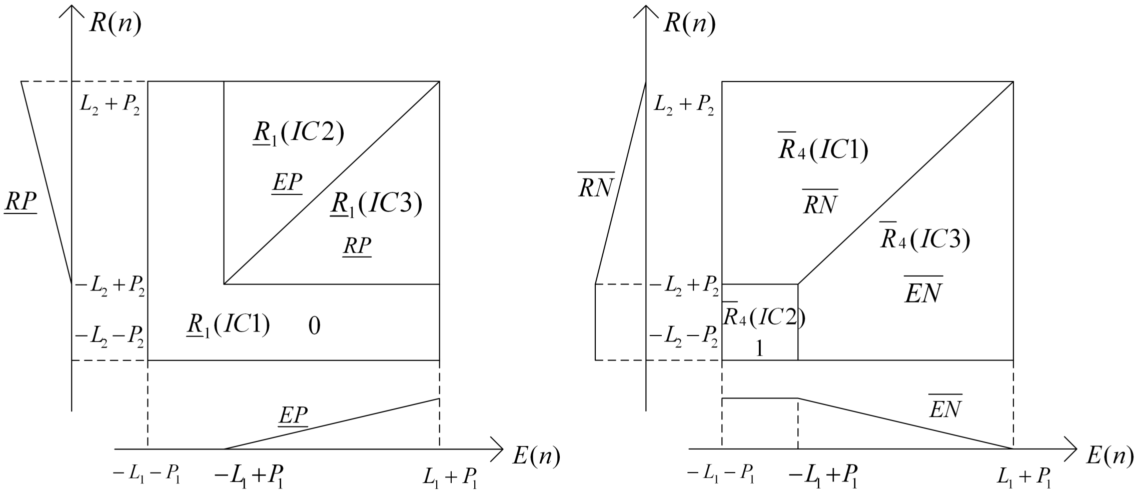
$$\overline{R}_2 = \min\{\overline{\mathbf{EP}}, \overline{\mathbf{RN}}\}$$

$$\overline{R}_3 = \min\{\overline{\mathbf{EN}}, \overline{\mathbf{RP}}\}$$

   As shown in Fig. 3.8, these two Zadeh AND operations does not lead to any further partitions of the region M_{L2}.

4. Finally, the input space partitions for $\triangle U_j^{min}$ shown in Fig. 3.9 is obtained by superimposing all the results obtained in Steps 1–3.

By following the proposed algorithm in Fig. 3.5, the partition of the input space for $\triangle U_j^{max}$ shown in Fig. 3.10 can be derived. Since the output of IT2 FLS is the average of the two endpoints, the partition of the input space by IT2 fuzzy PD controller can be derived by superimposing the partition by $\triangle U_j^{min}$ and



(a) (b)

Figure 3.6: The partitions by rules and switch mode in $\triangle U_j^{min}$ (a) The partition by $\underline{R}_1$. (b) The partition by $\bar{R}_4$.

$\triangle U_j^{max}$ in Fig. 3.11. The corresponding firing strengths in 35 local subregions for $\triangle U_j^{min}$ and $\triangle U_j^{max}$ are listed in Table 3.3.

3.4.2 The Expressions for IT2 Fuzzy PD Controller

By replacing each firing strengths in (3.14) and (3.15) with the expressions in Table 3.3 in every subregion, mathematical expressions for the two endpoints $\triangle U_j^{min}$ and $\triangle U_j^{max}$ can be derived. As an example, consider the region defined by the input conditions IC 1. The first row in Table 3.3 shows that the firing strengths in (3.14) for $\triangle U_j^{min}$ should be $R_4 = R_3 = \overline{EN} = -\frac{1}{2L_1}E(n) + 0.5 + \theta_1, R_2 = \underline{RN} = -\frac{1}{2L_2}R(n) + 0.5 - \theta_2, R_1 = \underline{RP} = \frac{1}{2L_1}E(n) + 0.5 - \theta_1$. After mathematical manipulations, the expression for $\triangle U_j^{min}$ is shown as (3.32). A similar expression can be obtained for $\triangle U_j^{max}$. As both equations are expressed in terms of $E(n)$

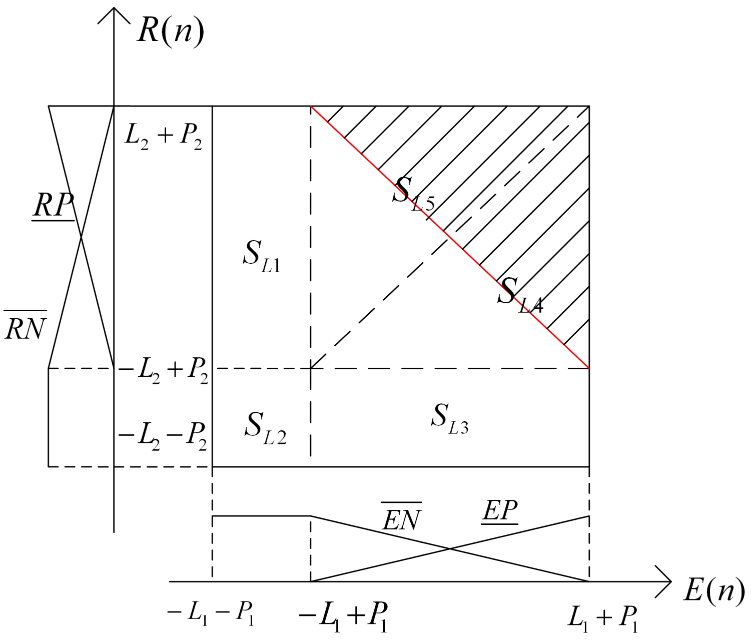


Figure 3.7: The boundary (red line) that divides the input space into the two operating mode in $\triangle U_j^{min}$

and $R(n)$, each subregion is equivalent to a non-linear PI/PD controller.

$$\begin{aligned}
\triangle U_{jIC1}^{min} &= \frac{R_4^* * H_4 + R_3^* * H_3 + R_2^* * H_2 + R_1^* * H_1}{R_4^* + R_3^* + R_2^* + R_1^*} \\
&= \frac{\overline{EN} * H_4 + \overline{EN} * H_3 + \overline{RN} * H_2 + \underline{RP} * H_1}{\overline{EN} + \overline{EN} + \overline{RN} + \underline{RP}} \\
&= \frac{[-\frac{1}{2L_1}E(n) + 0.5 + \theta_1](H_3 + H_4) + [-\frac{1}{2L_2}R(n) + 0.5 + \theta_2]H_2 + [\frac{1}{2L_2}R(n) + 0.5 - \theta_2]H_1}{2[-\frac{1}{2L_1}E(n) + 0.5 + \theta_1] + [-\frac{1}{2L_2}R(n) + 0.5 + \theta_2] + [\frac{1}{2L_2}R(n) + 0.5 - \theta_2]} \\
&= \frac{-L_2(H_3 + H_4)E(n) + L_1(H_1 - H_2)R(n)}{4L_1L_2(1 + \theta_1) - 2L_2E(n)} \\
&\qquad + \frac{L_1L_2[(0.5 + \theta_1)(H_3 + H_4) + (0.5 - \theta_2)H_1 + (0.5 + \theta_2)H_2]}{2L_1L_2(1 + \theta_1) - L_2E(n)} \\
&= K_p^* E(n) + K_d^* R(n) + \delta^* \qquad (3.32)
\end{aligned}$$

Similarly, the relationship between the output of IT2 fuzzy PD controller and its inputs in other subregions can be expressed into the following form:

$$\triangle^s U_{jICq} = {}^sK_p^q E(n) + {}^sK_d^q R(n) + {}^s\delta^q \qquad (3.33)$$

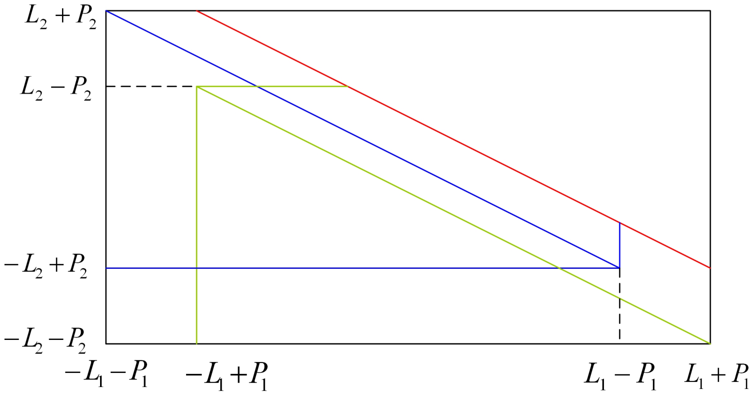


Figure 3.8: Partition of the input space by Rule 2 (green line), Rule 3 (blue line) and the boundary between the two operating modes (red line) in $\triangle U_j^{min}$ when $\theta_1 < \theta_2$

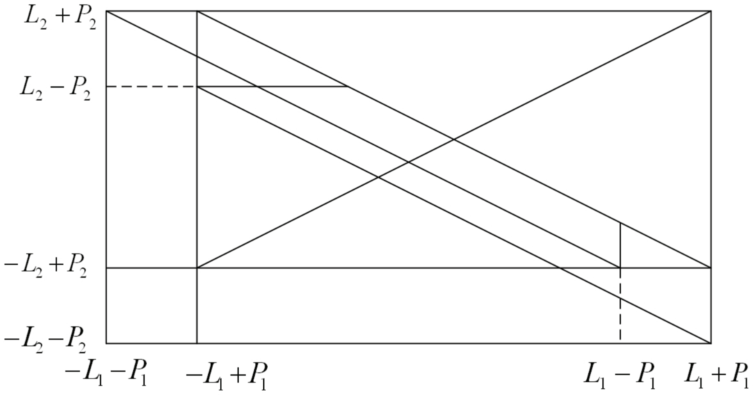


Figure 3.9: Partition of the input space by the left endpoint $\triangle U_j^{min}$ when $\theta_1 < \theta_2$

where $\triangle {}^sU_{jICq}$ is the output of IC q, ${}^sK_p^q$ is the corresponding proportional gain, ${}^sK_d^q$ is the derivative gain and ${}^s\delta^q$ is the offset.

3.5 Characteristics of IT2 Fuzzy PD Controller

This section aims at using the analytical structure and the equivalent proportional and derivative gains of the IT2 fuzzy PD controller, derived in previous sections, to compare the controller with those of its T1 counterpart and to highlight in-

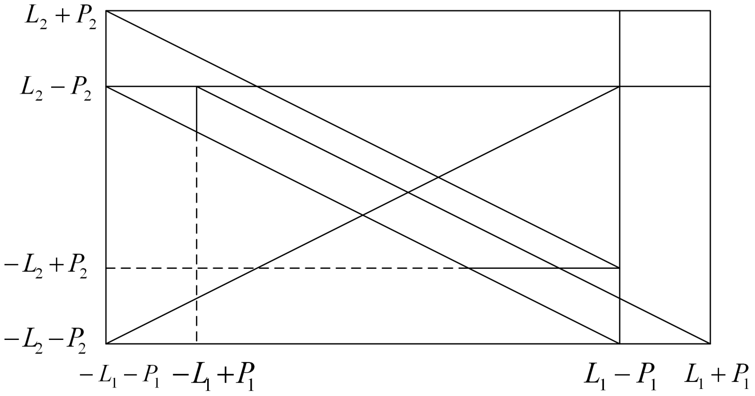


Figure 3.10: Partition of the input space by the right endpoint $\triangle U_j^{max}$ when $\theta_1 < \theta_2$

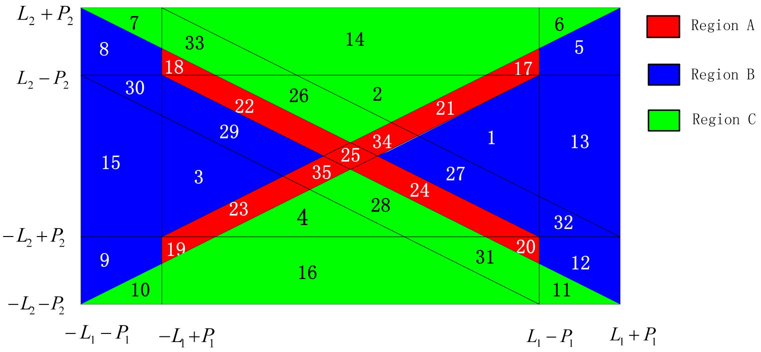


Figure 3.11: Partition of the input space by the IT2 FLS when $\theta_1 < \theta_2$

teresting characteristics of the IT2 fuzzy PD controller. To simplify the following investigation, the consequent sets of the IT2 and T1 fuzzy PD controllers in this study are assumed to be $H_2 = H_3 = 0$. Theorem 3.1 proves that shifting all consequents sets of an IT2 FLS or a T1 FLS horizontally by δ will yield an output that is altered by δ. (A proof of Theorem 3.1 is provided in Appendix A.) Consequently, the assumption that $H_2 = H_3 = 0$ does not affect the generality of the study when the separation between the consequent sets are maintained.

Theorem 3.1 *For multi-input single-output T1 FLS using the center-of-sets defuzzification and IT2 FLS using the KM type-reducer and the center-of-sets de-*

fuzzification, shifting all consequent sets by δ will cause the output to be shifted by δ.

Fig. 3.12(a) and 3.12(b) shows the antecedent sets of the T1 FLC for inputs $E(n)$ and $R(n)$ used in this study. They are constructed by replacing every IT2 FS with a T1 FS such that both the IT2 and T1 FLC have the same input space. Fig. 3.13 shows the four separate partitions for analyzing the gain characteristics of a T1 FLC [102]. The output value of the T1 FLC may be written as:

$$\triangle u_{jICh} = k_p^h E(n) + k_d^h R(n) \tag{3.34}$$

The proportional and derivative gains expressions (k_p^q, k_d^q) for the T1 fuzzy PD controller are listed in Table 3.1.

As shown in Fig. 3.11, the input space of the IT2 fuzzy PD controller needs to be decomposed into the 35 subregions before equivalence to nonlinear PD controllers with variable gains and the offset can be established. The proportional, derivative gains and the offset are tabulated in Table 3.4 and 3.5. By comparison, Fig. 3.13 shows that the input space of a T1 fuzzy PD controller is only divided into 4 subregions. The equivalent gains for the T1 FLC are tabulated in Table 3.1. Hence, it may be concluded that more subregions with special characteristics are provided by the IT2 fuzzy PD controller at the cost of two additional independent parameters θ_1 and θ_2 (Parameters P_1 and P_2 are related to θ_1 and θ_2 via the equations : $P_1 = 2L_1\theta_1, P_2 = 2L_2\theta_2$). Next, four interesting properties of the IT2 FLC will be highlighted.

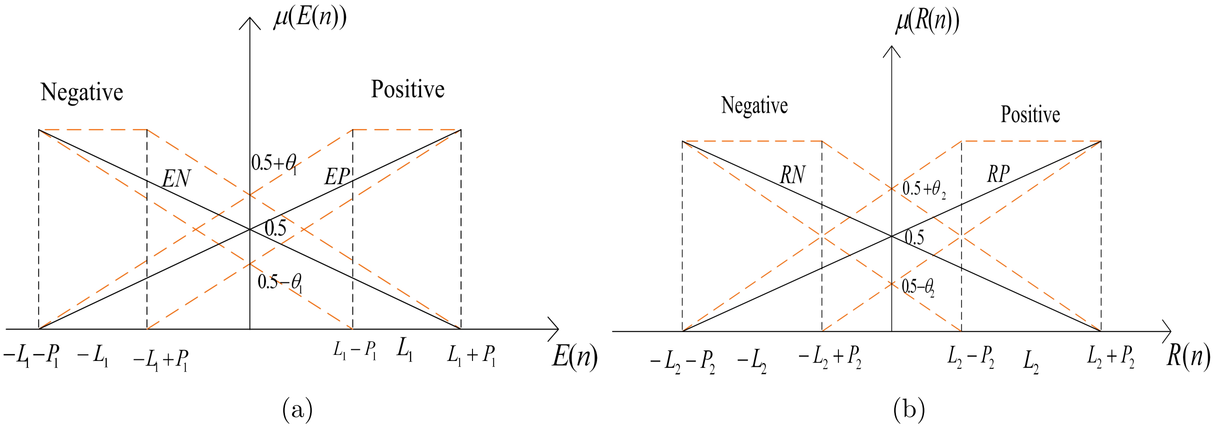


Figure 3.12: (a) T1 FSs EN and EP (solid lines) as antecedent sets for the input $E(n)$. (b)T1 FSs RN and RP (solid lines) as antecedent sets for the input $R(n)$.)

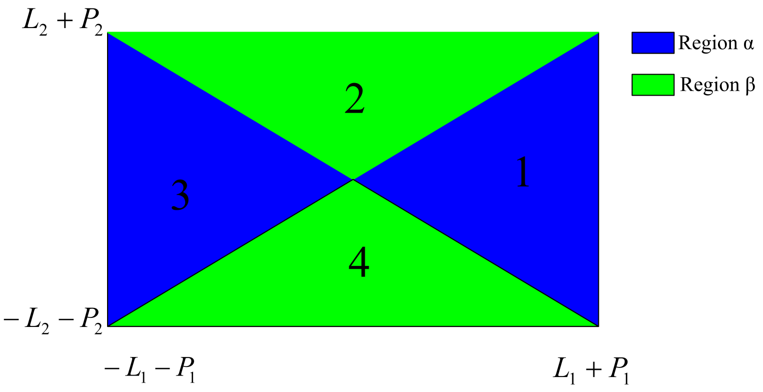


Figure 3.13: The partitions of the input space by T1 FLS

3.5.1 Characteristics of the Regions that Exist Only When $\theta_1 \neq \theta_2$

Fig. 3.11 and 3.14 contain the partition of the input space when $\theta_1 \neq \theta_2$ and when $\theta_1 = \theta_2$. By comparing the two figures, it may be concluded that $\theta_1 \neq \theta_2$ introduces Region A. An inspection of Table 3.4 and 3.5 shows that the equivalent proportional and derivative gains of the IT2 FLC for Region B and Region C in Fig. 3.11 are functions of $E(n)$ and $R(n)$ respectively while the gains for Region A except IC 17-20 are functions of both $E(n)$ and $R(n)$. In contrast, Fig. 3.13

Table 3.1: The output of the T1 fuzzy PD controller $\triangle u_j(n)$ for IC 1 to IC 4

IC No.	$\triangle u_j(n)$
1	$\frac{L_1(1+2\theta_1)R(n)+L_2(1+2\theta_2)E(n)}{-2L_2(1+2\theta_2)E(n)+4L_1L_2(1+2\theta_1)(1+2\theta_2)}H_1$
2	$\frac{L_1(1+2\theta_1)R(n)+L_2(1+2\theta_2)E(n)}{-2L_1(1+2\theta_1)R(n)+4L_1L_2(1+2\theta_1)(1+2\theta_2)}H_1$
3	$\frac{L_1(1+2\theta_2)R(n)+L_2(1+2\theta_2)E(n)}{2L_2(1+2\theta_2)E(n)+4L_1L_2(1+2\theta_1)(1+2\theta_2)}H_1$
4	$\frac{L_1(1+2\theta_1)R(n)+L_2(1+2\theta_2)E(n)}{2L_1(1+2\theta_1)R(n)+4L_1L_2(1+2\theta_1)(1+2\theta_2)}H_1$

Table 3.2: The geometrical relationship of the input space between IT2 fuzzy PD controller and its T1 counterpart

IC. q	IC. h
$1, 27, 13$	1
$2, 26, 14$	2
$3, 29, 15$	3
$4, 28, 14$	4

and Table 3.1 indicates that the equivalent gains (k_p^h, k_d^h) for a T1 FLC in Region α are functions of one input $E(n)$ while the equivalent gains (k_p^h, k_d^h) in Region β are functions of the other input $R(n)$. Region A exists near the boundaries of the partitions lines for the T1 FLC, which may be expressed mathematically as $L_2(1+2\theta_2)E(n)+L_1(1+2\theta_1)R(n)=0$ and $L_2(1+2\theta_2)E(n)-L_1(1+2\theta_1)R(n)=0$. This characteristic will first be stated formally before its significance is elucidated: *Property 1* In the IT2 fuzzy PD controller, the gains ${}^sK_p^q$, ${}^sK_d^q$ and ${}^s\delta^q$ of these regions IC 21-25, 34 and 35 (Region A except IC 17-20 in Fig. 3.11) are functions of both $E(n)$ and $R(n)$, while these gains for the other regions are functions of only $E(n)$ or $R(n)$. The size of these areas increases as $|\theta_1-\theta_2|$ increases.

Comment : As Region A connects Region B and Region C in Fig. 3.11, Region A whose gains are functions of both the inputs $E(n)$ and $R(n)$ may provide

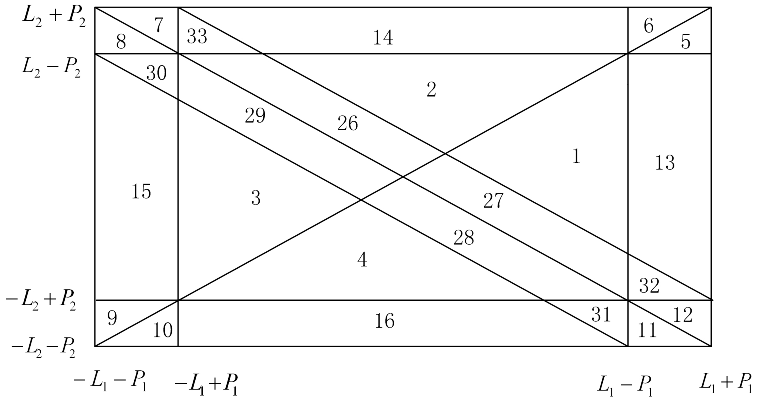


Figure 3.14: Partition of the input space by the IT2 FLS when $\theta_1 = \theta_2$

smoother transitions between Region B and Region C whose gains are functions of $E(n)$ and $R(n)$ respectively. In other words, these regions make the surface of IT2 fuzzy PD controller smoother around the zero feedback error point. Consequently, these regions may help to eliminate the overshoot despite greater control efforts provided by the IT2 fuzzy PD controller.

3.5.2 Gains Relationship between Internal Regions and External Regions

In this subsection, Region B and Region C in Fig. 3.11 are the study objective. A unique property of IT2 FSs are regions where the upper or lower membership grade remains constant at zero or unity. This characteristics enables Region B and Region C in Fig. 3.11 to be classified into two categories:

1. internal subregions comprising IC 1 - IC4, IC 26- IC 29.
2. external subregions consisting of the other regions i.e. IC 5-IC 16 and IC 30- IC 33

Property 2 In the IT2 fuzzy PD controller, the proportional gain of the internal subregions IC 1, 27 (IC 3, 29) is bigger than that of their adjacent external subregion IC 13 (IC 15) while the internal subregion IC 2, 26 (IC 4,28) exhibit bigger derivative gain than their adjacent external subregion IC 14 (IC 16).

Proof : The symmetry of the antecedent sets and consequent sets allows IC 1 and 27 to be taken as an example to illustrate this property. The observation are

$$ {}^{s}K_p^{1} = {}^{s}K_p^{13} + \frac{H_1}{8L_1(1-\theta_1)-4E(n)} = \frac{H_1}{8L_1(1+\theta_1)-4E(n)} + \frac{H_1}{8L_1(1-\theta_1)-4E(n)} $$

$$ \Rightarrow {}^{s}K_p^{1} > {}^{s}K_p^{13} > 0 $$

$$ {}^{s}K_p^{27} > {}^{s}K_p^{1} > {}^{s}K_p^{13} $$

This shows that in the direction of $E(n)$, IC 1 and 27 exhibits bigger slope than IC 13.

3.5.3 Comparative Output Values of IT2 Fuzzy PD Controller and its T1 Counterpart

For any given input pair $(E(n), R(n))$ in the internal subregions, the relationship in the output between the IT2 FLC and its T1 counterpart is the focus of this subsection. By comparing Fig. 3.11 with Fig. 3.13, it may be observed that the subregions ICq in the input space of IT2 fuzzy PD controller is a subset of the subregion ICh for its T1 counterpart. Table. 3.2 list the values of ICq for the IT2 controller that make up one partition ICh for the T1 controller. By contrasting the characteristics of subregions q in the IT2 fuzzy PD controller with the corresponding subregion h in its T1 counterpart, conclusions about the relative sizes of the output values of the IT2 and T1 controller can be established. This relationship is interesting because a bigger control effort may lead to smaller rise time (faster transient response). Since the outputs of the IT2 and T1 controllers for a given input pair depends the equivalent gains, the study is performed by analyzing the equivalent proportional and derivative gains of the two fuzzy controllers tabulated in Tables 3.5, 3.4 and 3.1.

Property 3 Given any input pair in IC 1 or IC 27 (IC 3 or IC 29 due to symmetry in the input space), the IT2 fuzzy PD controller has bigger proportional gain than that of its T1 counterpart, while the IT2 fuzzy PD controller exhibit bigger derivative gain than its T1 counterpart for any input pair in its subregions IC 2 or IC 26 (IC 4 or IC 28). The differences between these gains of the IT2 FLC and the T1 FLC increases as θ_1 and θ_2 increase.

Proof : It is sufficient to consider IC 1 and IC 27 (subset of IC 1 for the T1 controller) due to the symmetry of the two fuzzy PD controllers. The relationships between the proportional gains of the IT2 fuzzy PD controller and the T1 FLC are

$$\begin{aligned}
&{}^{s}K_p^1 = \frac{H_1}{8L_1(1+\theta_1)-4E(n)} + \frac{H_1}{8L_1(1-\theta_1)-4E(n)}\\
&> (\frac{1}{8L_1-4E(n)} + \frac{1}{8L_1-4E(n)})H_1 = \frac{1}{4L_1-2E(n)}H_1\\
&\geq \frac{1}{4L_1(1+2\theta_1)-2E(n)}H_1 = k_p^1\\
&{}^{s}K_p^{27} > {}^{s}K_p^1 > k_p^1
\end{aligned}$$

The above inequalities show the proportional gain of IT2 fuzzy PD controller in IC 1 and IC 27 is bigger than its T1 counterpart.

For the special case when $\theta_1 = \theta_2$, a more rigorous result governing the output relationships between the IT2 fuzzy PD controller and its T1 controller can be established. Fig. 3.14 shows the manner in which the input space is partitioned when $\theta_1 = \theta_2$, while Property 2 formally states the relationship.

Property 4 Any input pair $(E(n), R(n))$ in IC 1–IC 4 and IC 26–IC 29 of the IT2 fuzzy PD controller will produce an output that is bigger magnitude than that of its T1 counterpart when $\theta_1 = \theta_2$; and the difference increases as θ_1 and θ_2 increase.

Proof: Due to the symmetry in the IT2 fuzzy PD controller and its T1 counterpart, the output value property for IC 1–IC4 and IC 26–IC 29 will be illustrated by comparing the equivalent gains expressions for IC 1 and 27 (subset of IC 1 of the T1 counterpart). From Table 3.4, 3.5 and 3.1, the relationship between the outputs of the two fuzzy PD controllers can be established as

$$\begin{aligned}
&\triangle {}^{s}U_{jIC1} = (\frac{H_1}{8L_1(1+\theta_1) - 4E(n)} + \frac{H_1}{8L_1(1-\theta_1) - 4E(n)})E(n) \\
&+(\frac{L_1H_1}{8L_1L_2(1+\theta_1) - 4L_2E(n)} + \frac{L_1H_1}{8L_1L_2(1-\theta_1) - 4L_2E(n)})R(n) + \delta_1 \\
&= (\frac{1}{8L_1L_2(1+\theta_1) - 4L_2E(n)} + \frac{1}{8L_1L_2(1-\theta_1) - 4L_2E(n)})(L_1R(n) + L_2E(n))H_1 + {}^{s}\delta_1
\end{aligned}$$

$$\begin{aligned}
&\triangle u_{jIC1} = \frac{L_1(1+2\theta_1)R(n) + L_2(1+2\theta_2)E(n)}{4L_1L_2(1+2\theta_1)(1+2\theta_2) - 2L_2(1+2\theta_2)E(n)}H_1 \\
&= \frac{1}{4L_1L_2(1+2\theta_1) - 2L_2E(n)}(L_1R(n) + L_2E(n))H_1 \\
&= (\frac{1}{8L_1L_2(1+2\theta_1) - 4L_2E(n)} + \frac{1}{8L_1L_2(1+2\theta_1) - 4L_2E(n)})(L_1R(n) + L_2E(n))H_1
\end{aligned}$$

Because

$$\frac{1}{8L_1L_2(1-\theta_1) - 4L_2E(n)} > \frac{1}{8L_1L_2(1+2\theta_1) - 4L_2E(n)}$$
$$\frac{1}{8L_1L_2(1+\theta_1) - 4L_2E(n)} > \frac{1}{8L_1L_2(1+2\theta_1) - 4L_2E(n)}$$

and $\delta_1 > 0$, $L_1R(n) + L_2E(n) > 0$ in IC 1, then $\triangle u_{jIC1} > \triangle u_{jIC1} > 0$ and their difference increases as θ_1 increases because θ_1 appears in the denominator.

Another observation is

$$
\begin{aligned}
&\triangle^{s}U_{jIC27} = (\frac{H_1}{8L_1(1-\theta_2)-4E(n)} + \frac{H_1}{8L_1(1-\theta_1)-4E(n)})E(n)\\
&+(\frac{L_1H_1}{8L_1L_2(1-\theta_1)-4L_2E(n)} + \frac{L_1H_1}{8L_1L_2(1-\theta_2)-4L_2E(n)})R(n)\\
&= (\frac{1}{8L_1L_2(1-\theta_1)-4L_2E(n)} + \frac{1}{8L_1L_2(1-\theta_2)-4L_2E(n)})(L_1R(n)+L_2E(n))H_1
\end{aligned}
$$

Since $L_1R(n) + L_2E(n) > 0$ in IC 27 and

$$
\begin{aligned}
\frac{1}{8L_1L_2(1-\theta_1)-4L_2E(n)} &> \frac{1}{8L_1L_2(1+2\theta_1)-4L_2E(n)}\\
\frac{1}{8L_1L_2(1-\theta_2)-4L_2E(n)} &> \frac{1}{8L_1L_2(1+2\theta_1)-4L_2E(n)}
\end{aligned}
$$

then the magnitude of the control effort provided by IT2 fuzzy PD controller for the same input pair is bigger. Furthermore, the difference increases as θ_1 and θ_2 increase.

3.5.4 Discussion

Based on the results in Property 1-4, the characteristics of an IT2 FLC that may enable it to outperform a T1 FLC by providing fast rise time and small overshoot may be summarized as:

1. As stated in Property 1, Region A whose gains are functions of both the inputs $E(n)$ and $R(n)$ connects Region B and Region C whose gains are functions of $E(n)$ and $R(n)$ respectively; thus such architecture enables the IT2 FLC to achieve smoother surface. More importantly, the area of Region A depends on the value of $|\theta_1 - \theta_2|$.

2. The greater proportional and derivative gains (Property 3) may enable the IT2 FLC to provide better disturbance rejection ability. It should be noted

that the difference between the gains of the IT2 FLC and the T1 FLC increases as θ_1 and θ_2 increase.

3. Property 4 indicates that the IT2 FLC can produce control efforts that are larger in magnitude for certain input pairs. Greater control effort from the IT2 FLC may decrease the rise time; the amount by which the control effort can be enlarged depends on the value of θ_1 and θ_2.

Nevertheless, the trade-off between fast rise time and small overshoot still exists. This is because θ_1 and θ_2 need to be large for the IT2 FLC to produce bigger control efforts that result in fast rise time. However, a small overshoot requires the size of Region A to be enlarged by setting a large absolute value $|\theta_1 - \theta_2|$. The two conditions cannot be achieved simultaneously.

3.6 Numerical Studies

Results from a numerical study is presented here to further illustrate the above properties of an IT2 fuzzy PD controller gained by the analysis in the previous sections. The test bed is a coupled tank whose behavior is defined by the following differential equations [90]:

$$A_1 \frac{dH_1}{dt} = Q_1 - \alpha_1 \sqrt{H_1} - \alpha_3 \sqrt{H_1 - H_2} \tag{3.35}$$

$$A_2 \frac{dH_2}{dt} = -\alpha_2 \sqrt{H_2} - \alpha_3 \sqrt{H_1 - H_2} \tag{3.36}$$

where H_1 and H_2 are the water level of the tank 1 and 2; Q_1 and Q_2 are the rate at which water is pumped into tank 1 and tank 2. The two tanks in the system are connected by a baffle that may be raised or lowered to vary the amount of water that flows between them. Outlets, at the base of the two tanks, enable water to flow out from each tank. The control task is to vary the amount of water (Q_1) that enters tank 1 in order to regulate the water level (H_2) in tank 2. In the simulation, it is assumed that the nominal plant has the following parameters : $A_1 = A_2 = 36.52$, $\alpha_1 = \alpha_2 = 5.6186$, $\alpha_3 = 10$.

The configurations of IT2 fuzzy PI controller and T1 fuzzy PI controller are identical to those in Section 3.2 where $K_r = K_e = 1, K_{\triangle U} = 75$. The predefined antecedent sets of T1 fuzzy PI controller for $Error(E(n))$ and $Rate(R(n))$ are shown in Fig. 3.15(a) and 3.15(b). The singleton consequent sets are predefined as $H_2 = H_3 = 0, H_4 = -H_1$. The two parameters θ_{Error} and θ_{Rate} are defined as the distance between the upper bound and the lower bound for every antecedent sets when α-cut is 1.

Analysis in the last subsection shows that IT2 fuzzy PD controller can outperform T1 counterpart in rise time, overshoot and disturbance rejection. To substantiate the theoretical study, the following three cases are simulated:

1. Case 1: The parameters of IT2 fuzzy PD controller are optimized as $\theta_{Error} = 5$, $\theta_{Rate} = 2$ and $H_1 = 8$ through genetic algorithm with ITAE as the fitness function. As shown in Fig. 3.16(a), The response obtained using the IT2 FLC has comparative rise time with the T1 case, but exhibits smaller overshoot and is less oscillatory. Fig. 3.16(b) shows the error versus rate trajectory and the trajectory for IT2 FLC is much more smooth when it is near $Error = 0$.

2. Case 2: In terms of rise time, these two FLCs are compared by choosing

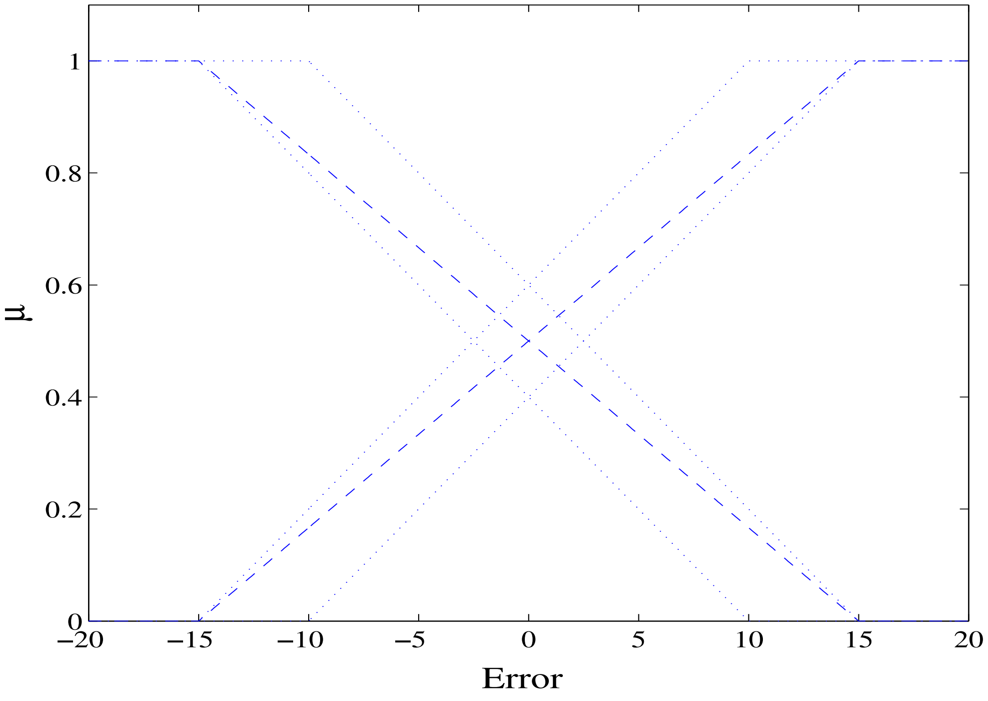
$H_1 = 0.08$. With a small H_1, their difference is more obvious in rise time. Fig. 3.17(a) shows the step responses for Case 2, while the error versus rate trajectory is shown in Fig. 3.17(b). IT2 FLC achieves bigger convergence rate and less rise time as it can provide bigger control effort than T1 case.

3. Case 3: By keeping $H_1 = 8$, the results for the plant with random disturbances are compared with those in Case 1 to show their disturbance rejection ability. As shown in Fig .3.18(a) and 3.18(b), more oscillation is caused by disturbance in T1 FLC while IT2 FLC exhibits similar performance with the one in Case 1.

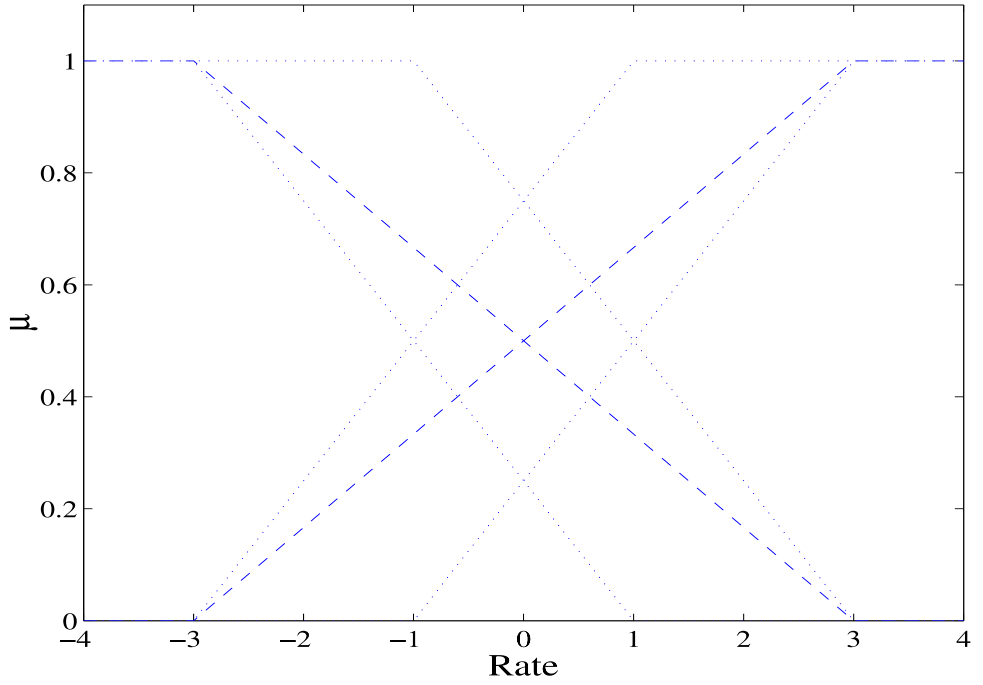
In terms of overshoot, rise time and disturbance rejection, IT2 FLC can outperform T1 counterpart, which is identical to the theoretical analysis in the last subsection. To find out how much IT2 FLC can outperform T1 case, further study is done by gradually increasing the value of H_1 and comparing the value of their ITAE. Fig. 3.19 shows that as H_1 increases, the rate of ITAE reduced by IT2 FLC increases. The increase of this rate becomes slow when H_1 is increased to some big value. Fig. 3.20 and 3.21 show the control surface of the T1 FLC and the IT2 FLC respectively. From their difference in Fig. 3.22 it can be observed that for most of the input pairs the IT2 FLC generates control efforts that are larger in magnitude, which is consistent with Property 4.

3.7 Conclusion

The analytical structure of a special class of IT2 fuzzy PD and PI controllers that uses the KM iterative algorithm for type-reduction is presented. Due to the iterative nature of the widely adopted KM algorithm, the well-established techniques for deriving the analytical structure of T1 FLCs cannot be applied directly. A methodology for identifying the input space boundaries where the KM algorithm uses a new switch point to compute the bounds of the type-reduced set is first established. This result resolves one of the main issues hindering theoretical analysis of IT2 FLCs that employs the KM type reducer. Based on the finding, the analytical structure of an IT2 FLC is derived by treating each input region as a T1 FLS problem using the well-established techniques for T1 FLSs. A comparative study of the derived analytical structure and its T1 counterpart identified four characteristics unique to an IT2 FLC. The property that has the most practical relevance is that an IT2 FLC can provide greater control effort and achieve smoother control surface simultaneously. This result provides initial theoretical basis for explaining the ability of an IT2 FLC to alleviate the trade-off between fast rise time and small overshoot, which was derived from experimental studies. Having developed the analytical structure for a simple IT2 FLC, a future research direction is to relax the design constraints and study the properties of more complex IT2 FLCs.

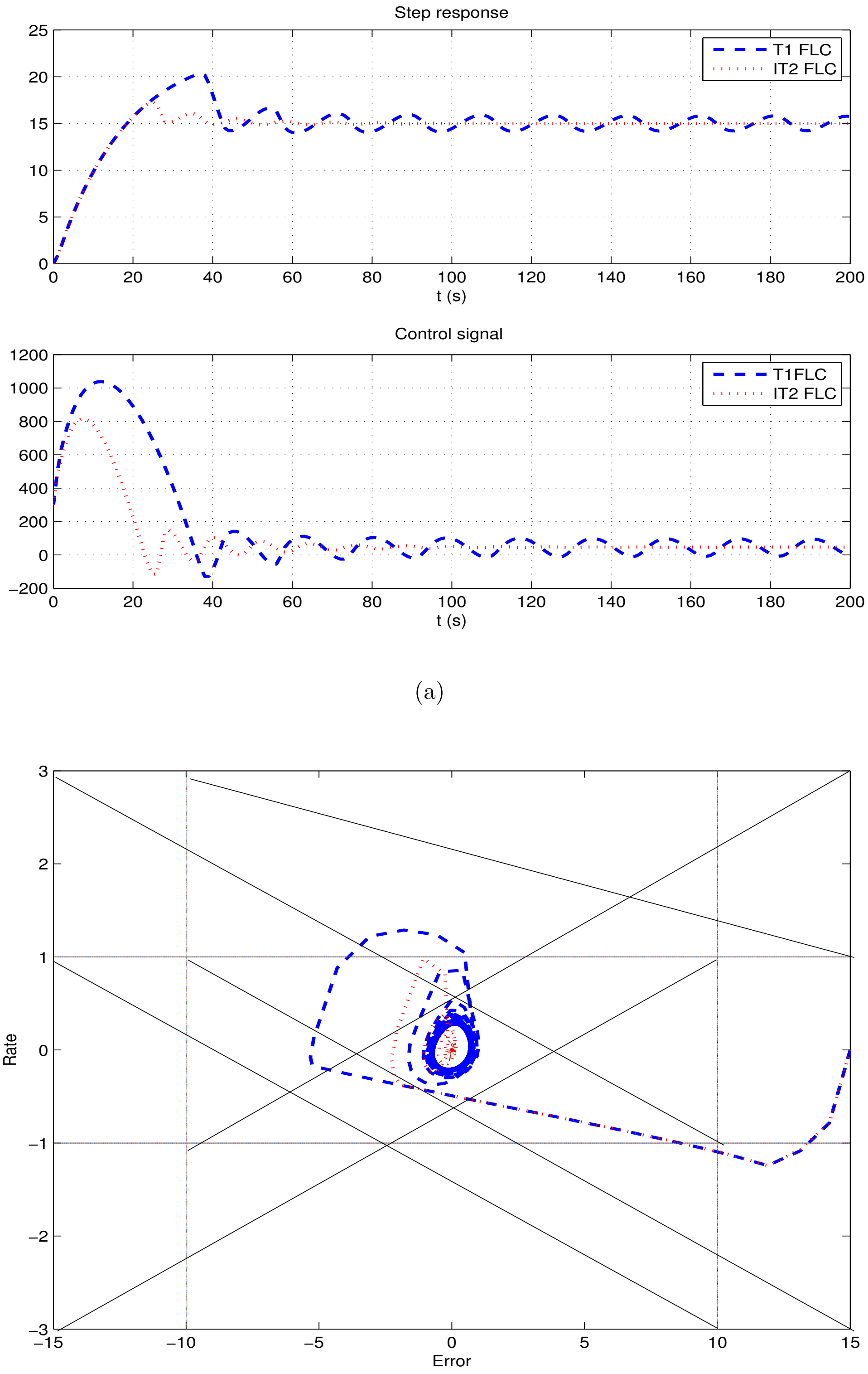


(a)



(b)

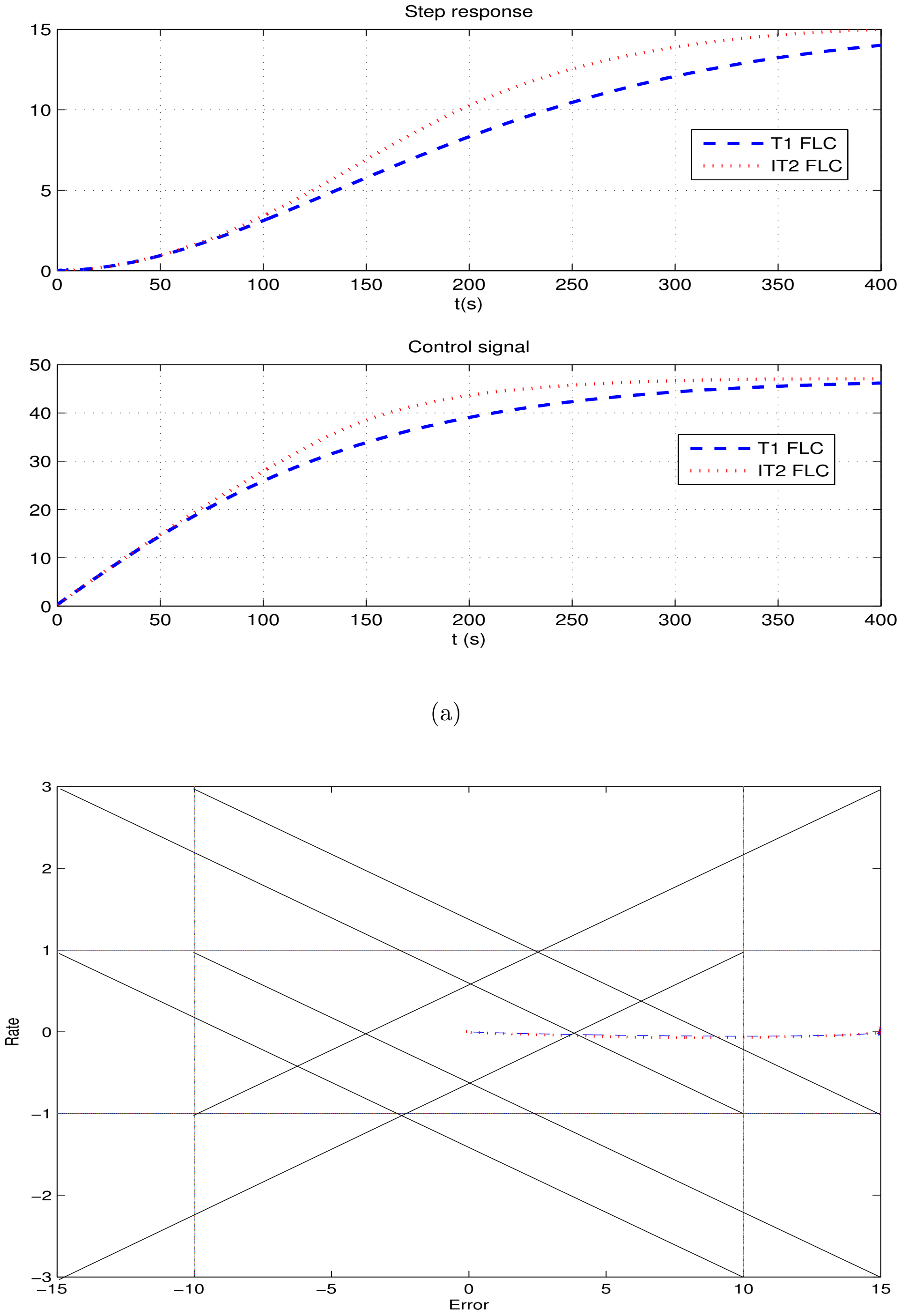
Figure 3.15: IT2 antecedent FSs: (a) Antecedent sets of Error. (b) Antecedent sets of Rate. (The dashed line for T1 FLS, the dotted line for IT2 FLS)



(a)

(b)

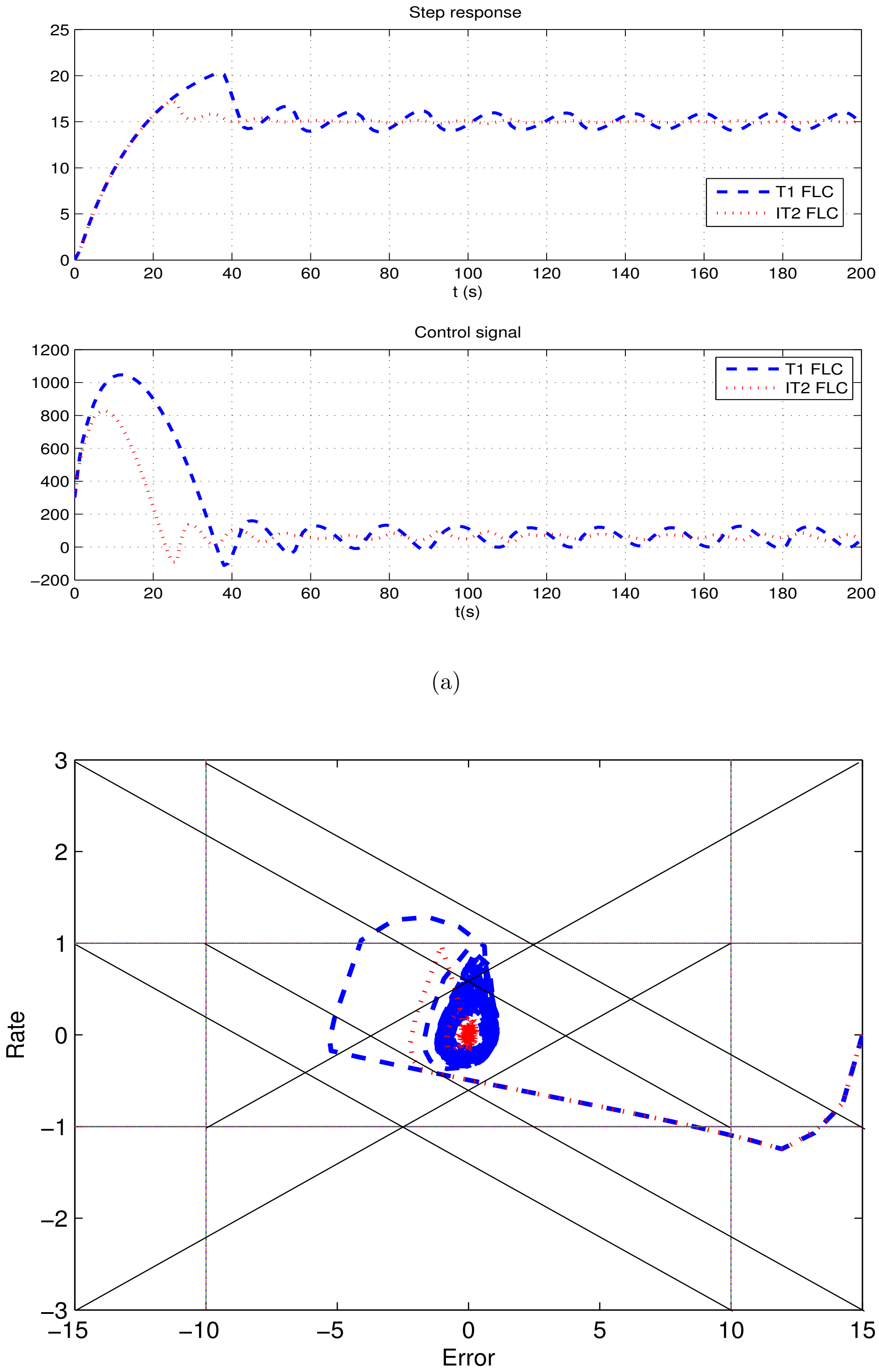
Figure 3.16: Case 1 (a) The output of the system using T1 FLC and IT2 FLC (The dashed line for T1 FLC, the dotted line for IT2 FLC). (b)The trajectory of Error and Rate(Red line for IT2 FLC, Blue line for T1 FLC).)



(a)

(b)

Figure 3.17: Case 2 (a) The output of the system using T1 FLC and IT2 FLC (The dashed line for T1 FLC, the dotted line for IT2 FLC). (b)The trajectory of Error and Rate(Red line for IT2 FLC, Blue line for T1 FLC).)



(a)

(b)

Figure 3.18: Case 3 (a) The output of the system using T1 FLC and IT2 FLC (The dashed line for T1 FLC, the dotted line for IT2 FLC). (b)The trajectory of Error and Rate(Red line for IT2 FLC, Blue line for T1 FLC).)

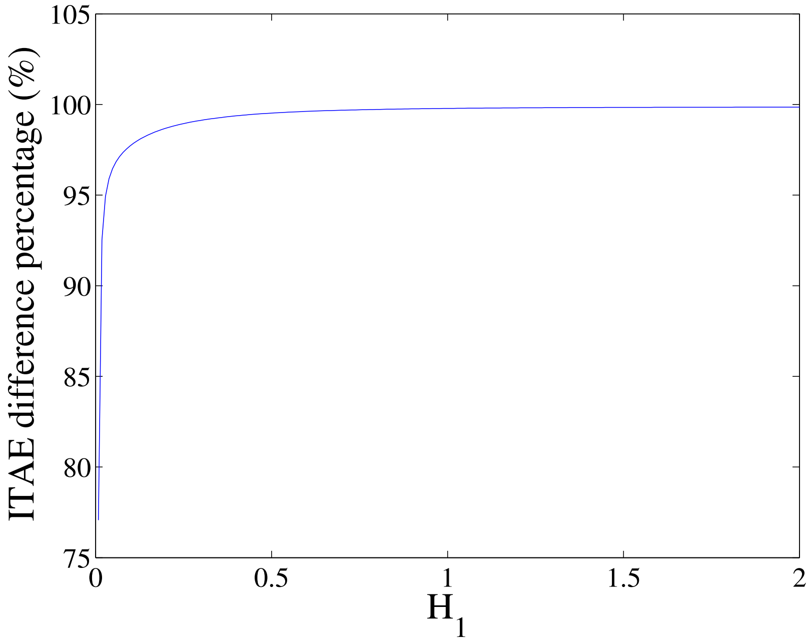


Figure 3.19: The ITAE difference percentage: $\frac{\text{ITAE for T1 FLC} - \text{ITAE for IT2 FLC}}{\text{ITAE for T1 FLC}} \times 100\%$

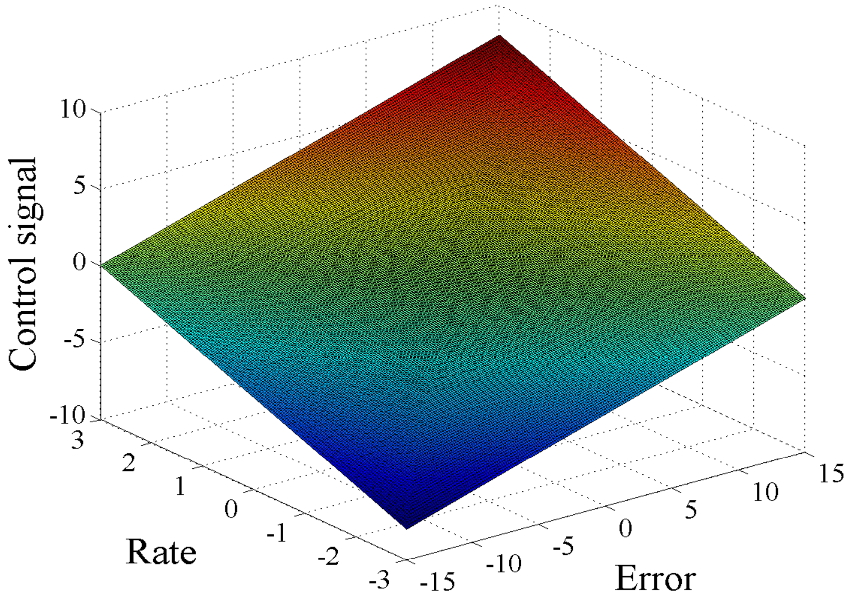


Figure 3.20: The control surface produced by the T1 FLC ($H_1 = 8$)

Table 3.3: The Firing strengths of four rules in $\triangle U_j^{min}$ and $\triangle U_j^{max}$

IC No.	$\triangle U_j^{min}$				$\triangle U_j^{max}$			
	Rule 4	Rule 3	Rule 2	Rule 1	Rule 4	Rule 3	Rule 2	Rule 1
1	$\overline{EN}$	$\overline{EN}$	$\overline{RN}$	$\underline{RP}$	$\underline{EN}$	$\underline{EN}$	$\underline{RN}$	$\overline{RP}$
2	$\overline{RN}$	$\overline{EN}$	$\overline{RN}$	$\underline{EP}$	$\underline{RN}$	$\underline{\underline{EN}}$	$\underline{\underline{RN}}$	$\overline{EP}$
3	$\overline{RN}$	$\underline{RP}$	$\underline{EP}$	$\underline{EP}$	$\underline{RN}$	$\overline{\overline{RP}}$	$\overline{\overline{EP}}$	$\overline{EP}$
4	$\overline{EN}$	$\underline{\underline{RP}}$	$\underline{\underline{EP}}$	$\underline{RP}$	$\underline{EN}$	$\overline{RP}$	$\overline{EP}$	$\overline{RP}$
5	$\overline{EN}$	$\overline{\overline{EN}}$	$\overline{\overline{RN}}$	$\underline{RP}$	0	0	0	1
6	$\overline{RN}$	$\overline{EN}$	$\overline{RN}$	$\underline{EP}$	0	0	0	1
7	$\overline{RN}$	$\underline{EN}$	0	0	0	$\underline{EN}$	0	$\overline{EP}$
8	$\overline{RN}$	$\underline{RP}$	0	0	0	$\underline{\underline{RP}}$	0	$\overline{EP}$
9	1	0	0	0	$\underline{RN}$	$\overline{\overline{RP}}$	$\overline{EP}$	$\overline{EP}$
10	1	0	0	0	$\underline{EN}$	$\overline{RP}$	$\overline{EP}$	$\overline{RP}$
11	$\overline{EN}$	0	$\underline{EP}$	0	0	0	$\underline{EP}$	$\overline{RP}$
12	$\overline{EN}$	0	$\underline{\underline{RN}}$	0	0	0	$\underline{RN}$	$\overline{RP}$
13	$\overline{EN}$	$\overline{EN}$	$\overline{\overline{RN}}$	$\underline{RP}$	0	0	RN_L	$\overline{RP}$
14	$\overline{RN}$	$\overline{EN}$	$\overline{RN}$	$\underline{EP}$	0	$\underline{\underline{EN}}$	0	$\overline{EP}$
15	$\overline{RN}$	$\underline{RP}$	0	0	$\underline{RN}$	$\overline{\overline{RP}}$	$\overline{EP}$	$\overline{EP}$
16	$\overline{EN}$	0	$\underline{\underline{EP}}$	0	$\underline{EN}$	$\overline{RP}$	$\overline{EP}$	$\overline{RP}$
17	$\overline{EN}$	$\overline{EN}$	$\overline{\overline{RN}}$	$\underline{RP}$	0	$\underline{EN}$	0	$\overline{EP}$
18	$\overline{RN}$	$\underline{RP}$	0	$\underline{EP}$	0	$\underline{\underline{RP}}$	0	$\overline{EP}$
19	$\overline{EN}$	0	$\underline{EP}$	0	$\underline{RN}$	$\overline{\overline{RP}}$	$\overline{EP}$	$\overline{EP}$
20	$\overline{EN}$	0	$\underline{\underline{RN}}$	0	$\underline{EN}$	0	$\underline{RN}$	$\overline{RP}$
21	$\overline{EN}$	$\overline{EN}$	$\overline{\overline{RN}}$	$\underline{RP}$	$\underline{RN}$	$\underline{EN}$	$\underline{RN}$	$\overline{EP}$
22	$\overline{RN}$	$\underline{RP}$	$\underline{RN}$	$\underline{EP}$	$\underline{RN}$	$\underline{\underline{RP}}$	$\underline{\underline{RN}}$	$\overline{EP}$
23	$\overline{EN}$	$\underline{RP}$	$\underline{EP}$	$\underline{RP}$	$\underline{RN}$	$\overline{\overline{RP}}$	$\overline{\overline{EP}}$	$\overline{EP}$
24	$\overline{EN}$	$\underline{RP}$	$\underline{RN}$	$\underline{RP}$	$\underline{EN}$	$\underline{RP}$	$\underline{RN}$	$\overline{RP}$
25	$\overline{EN}$	$\underline{RP}$	$\underline{RN}$	$\underline{RP}$	$\underline{RN}$	$\underline{RP}$	$\underline{RN}$	$\overline{EP}$
26	$\overline{RN}$	$\underline{EN}$	$\underline{RN}$	$\underline{EP}$	$\underline{RN}$	$\underline{EN}$	$\underline{RN}$	$\overline{EP}$
27	$\overline{EN}$	$\underline{EN}$	$\underline{RN}$	$\underline{RP}$	$\underline{EN}$	$\underline{EN}$	$\underline{RN}$	$\overline{RP}$
28	$\overline{EN}$	$\underline{RP}$	$\underline{EP}$	$\underline{RP}$	$\underline{EN}$	$\underline{RP}$	$\underline{EP}$	$\overline{RP}$
29	$\overline{RN}$	$\underline{RP}$	$\underline{EP}$	$\underline{EP}$	$\underline{RN}$	$\underline{RP}$	$\underline{EP}$	$\overline{EP}$
30	$\overline{RN}$	$\underline{RP}$	0	0	$\underline{RN}$	$\underline{RP}$	0	$\overline{EP}$
31	$\overline{EN}$	0	$\underline{EP}$	0	$\underline{EN}$	0	$\underline{EP}$	$\overline{RP}$
32	$\overline{EN}$	0	$\underline{RN}$	$\underline{RP}$	0	0	$\underline{RN}$	$\overline{RP}$
33	$\overline{RN}$	$\underline{EN}$	0	$\underline{EP}$	0	$\underline{EN}$	0	$\overline{EP}$
34	$\overline{EN}$	$\underline{EN}$	$\underline{RN}$	$\underline{RP}$	$\underline{RN}$	$\underline{EN}$	$\underline{RN}$	$\overline{EP}$
35	$\overline{EN}$	$\underline{RP}$	$\underline{EP}$	$\underline{RP}$	$\underline{RN}$	$\underline{RP}$	$\underline{EP}$	$\overline{EP}$

Table 3.4: The gains for the external subregions

IC No.	${}^{s}K_p$	${}^{s}K_d$	${}^{s}\delta$
5	$\frac{H_1}{8L_1(1+\theta_1)-4E(n)}$	$\frac{L_1H_1}{8L_1L_2(1+\theta_1)-4L_2E(n)}$	$-\frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1+\theta_1)-2E(n)}+0.5H_1$
6	$\frac{L_2H_1}{8L_1L_2(1+\theta_2)-4L_1R(n)}$	$\frac{H_1}{8L_2(1+\theta_2)-4R(n)}$	$-\frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1+\theta_2)-2R(n)}+0.5H_1$
7	$\frac{H_1}{4L_1}$	$\frac{L_1H_1}{4L_1L_2(1-\theta_1+\theta_2)-2L_1R(n)-2L_2E(n)}$	$\frac{L_1L_2(0.5+\theta_2)H_4}{2L_1L_2(1-\theta_1+\theta_2)-L_1R(n)-2L_2E(n)}$ $+0.5(0.5+\theta_1)H_1$
8	$\frac{L_2H_1}{4L_1L_2(1-\theta_2+\theta_1)+2L_1R(n)+2L_2E(n)}$	$\frac{H_1}{4L_2}$	$\frac{L_1L_2(0.5+\theta_1)H_1}{2L_1L_2(1-\theta_2+\theta_1)+L_1R(n)+L_2E(n)}$ $+0.5(0.5+\theta_2)H_4$
9	$\frac{H_1}{8L_1(1+\theta_1)+4E(n)}$	$\frac{L_1H_1}{8L_1L_2(1+\theta_1)+4L_2E(n)}$	$\frac{L_1H_1(\theta_1+\theta_2)}{4L_1(1+\theta_1)+2E(n)}+0.5H_4$
10	$\frac{H_1L_2}{8L_1L_2(1+\theta_2)+4L_1R(n)}$	$\frac{H_1}{8L_2(1+\theta_2)+4R(n)}$	$\frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1+\theta_2)+2R(n)}+0.5H_4$
11	$\frac{H_1}{4L_1}$	$\frac{L_1H_1}{4L_1L_2(1-\theta_1+\theta_2)+2L_2E(n)+2L_1R(n)}$	$\frac{L_1L_2(0.5+\theta_2)H_1}{2L_1L_2(1-\theta_1+\theta_2)+L_2E(n)+L_1R(n)}$ $+0.5(0.5+\theta_1)H_4$
12	$\frac{L_2H_1}{4L_1L_2(1+\theta_1-\theta_2)-2L_2E(n)-2L_1R(n)}$	$\frac{H_1}{4L_2}$	$\frac{L_1L_2(0.5+\theta_1)H_4}{2L_1L_2(1+\theta_1-\theta_2)-L_2E(n)-L_1R(n)}$ $+0.5(0.5+\theta_2)H_1$
13	$\frac{H_1}{8L_1(1+\theta_1)-4E(n)}$	$\frac{L_1H_1}{8L_1L_2(1+\theta_1)-4L_2E(n)}+\frac{H_1}{4L_2}$	$-\frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1+\theta_1)-2E(n)}$ $+0.5(0.5+\theta_2)H_1$
14	$\frac{L_2H_1}{8L_1L_2(1+\theta_2)-4L_1R(n)}+\frac{H_1}{4L_1}$	$\frac{H_1}{8L_2(1+\theta_2)-4R(n)}$	$-\frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1+\theta_2)-2R(n)}$ $+0.5(0.5+\theta_1)H_1$
15	$\frac{H_1}{8L_1(1+\theta_1)+4E(n)}$	$\frac{L_1H_1}{8L_1L_2(1+\theta_1)+4L_2E(n)}+\frac{H_1}{4L_2}$	$\frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1+\theta_1)+2E(n)}$ $+0.5(0.5+\theta_2)H_4$
16	$\frac{L_2H_1}{8L_1L_2(1+\theta_2)+4L_1R(n)}+\frac{H_1}{4L_1}$	$\frac{H_1}{8L_2(1+\theta_2)+4R(n)}$	$\frac{L_2\theta_1+\theta_2)H_1}{4L_2(1+\theta_2)+2R(n)}$ $+0.5(0.5+\theta_1)H_4$
17	$\frac{H_1}{4L_1}+\frac{H_1}{8L_1(1+\theta_1)-4E(n)}$	$\frac{L_1H_1}{8L_1L_2(1+\theta_1)-4L_2E(n)}$	$-\frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1+\theta_1)-2E(n)}$ $+0.5(0.5+\theta_1)H_1$
18	$\frac{H_1}{4L_1(1.5-\theta_1)+2E(n)}+$ $\frac{L_2H_1}{4L_1L_2(1-\theta_2+\theta_1)+2L_1R(n)+2L_2E(n)}$	$\frac{L_1H_1}{4L_1L_2(1.5-\theta_1)+2L_2E(n)}$	$\frac{L_1L_2(0.5+\theta_1)H_1}{2L_1L_2(1-\theta_2+\theta_1)+L_1R(n)+L_2E(n)}$ $+\frac{L_1H_4(\theta_1+\theta_2)}{2L_1(1.5-\theta_1)+E(n)}$
19	$\frac{H_1}{8L_1(1+\theta_1)+4E(n)}+\frac{H_1}{4L_1}$	$\frac{L_1H_1}{8L_1L_2(1+\theta_1)+4L_2E(n)}$	$\frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1+\theta_1)+2E(n)}$ $+0.5(0.5+\theta_1)H_4$
20	$\frac{H_1}{4L_1(1.5-\theta_1)-2E(n)}+$ $\frac{L_2H_1}{4L_1L_2(1+\theta_1-\theta_2)-2L_2E(n)-2L_1R(n)}$	$\frac{L_1H_1}{4L_1L_2(1.5-\theta_1)-2L_2E(n)}$	$\frac{L_1L_2(0.5+\theta_1)H_4}{2L_1L_2(1+\theta_1-\theta_2)-L_2E(n)-L_1R(n)}$ $+\frac{L_1(\theta_1+\theta_2)H_1}{2L_1(1.5-\theta_1)-E(n)}$
30	$\frac{H_1}{4L_1(1.5+\theta_1-2\theta_2)+2E(n)}$	$\frac{L_1H_1}{4L_1L_2(1.5+\theta_1-2\theta_2)+2L_2E(n)}+\frac{H_1}{4L_2}$	$\frac{L_1(\theta_1+\theta_2)H_1}{2L_1(1.5+\theta_1-2\theta_2)+E(n)}$ $+0.5(0.5+\theta_2)H_4$
31	$\frac{H_1}{4L_1}+\frac{L_2H_1}{4L_1L_2(2.5-2\theta_1+\theta_2)+2L_1R(n)}$	$\frac{H_1}{4L_2(2.5-2\theta_1+\theta_2)+R(n)}$	$\frac{L_2(\theta_1+\theta_2)H_1}{2L_2(2.5-2\theta_1+\theta_2)+R(n)}$ $+0.5(0.5+\theta_1)H_4$
32	$\frac{H_1}{4L_1(1.5+\theta_1-2\theta_2)-2E(n)}$	$\frac{L_1H_1}{4L_1L_2(1.5+\theta_1-2\theta_2)-2L_2E(n)}+\frac{H_1}{4L_2}$	$\frac{L_1H_4(\theta_1+\theta_2)}{2L_1(1.5+\theta_1-2\theta_2)-E(n)}$ $+0.5(0.5+\theta_2)H_1$
33	$\frac{L_2H_1}{4L_1L_2(1.5-2\theta_1+\theta_2)-2L_1R(n)}+\frac{H_1}{4L_1}$	$\frac{L_2H_1}{4L_1L_2(1.5-2\theta_1+\theta_2)-2L_1R(n)}$	$\frac{L_2H_4(\theta_1+\theta_2)}{2L_2(1.5-2\theta_1+\theta_2)-R(n)}$ $+0.5(0.5+\theta_1)H_1$

Table 3.5: The gains for the internal subregions

IC No.	${}^{s}K_p$	${}^{s}K_d$	${}^{s}\delta$
1	$\frac{H_1}{8L_1(1+\theta_1)-4E(n)} + \frac{H_1}{8L_1(1-\theta_1)-4E(n)}$	$\frac{L_1H_1}{8L_1L_2(1+\theta_1)-4L_2E(n)} + \frac{L_1H_1}{8L_1L_2(1-\theta_1)-4L_2E(n)}$	$\frac{L_1(\theta_1+\theta_2)H_4}{4L_1(1+\theta_1)-2E(n)} + \frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1-\theta_1)-2E(n)}$
2	$\frac{L_2H_1}{8L_1L_2(1+\theta_2)-4L_1R(n)} + \frac{L_2H_1}{8L_1L_2(1-\theta_2)-4L_1R(n)}$	$\frac{H_1}{8L_2(1+\theta_2)-4R(n)} + \frac{H_1}{8L_1L_2(1-\theta_2)-4R(n)}$	$\frac{L_2(\theta_1+\theta_2)H_4}{4L_2(1+\theta_2)-2R(n)} + \frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1-\theta_2)-2R(n)}$
3	$\frac{H_1}{8L_1(1-\theta_1)+4E(n)} + \frac{H_1}{8L_1(1+\theta_1)+4E(n)}$	$\frac{L_1H_1}{8L_1L_2(1-\theta_1)+2L_2E(n)} + \frac{L_1H_1}{8L_1L_2(1+\theta_1)+4l_2E(n)}$	$\frac{L_1(\theta_1+\theta_2)H_4}{4L_1(1-\theta_1)+2E(n)} + \frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1+\theta_1)+2E(n)}$
4	$\frac{L_2H_1}{8L_1L_2(1-\theta_2)+4L_1R(n)} + \frac{L_2H_1}{8L_1L_2(1+\theta_2)+4L_1R(n)}$	$\frac{H_1}{8L_2(1-\theta_2)+4R(n)} + \frac{H_1}{8L_2(1+\theta_2)+4R(n)}$	$\frac{L_2(\theta_1+\theta_2)H_4}{4L_2(1-\theta_2)+2R(n)} + \frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1+\theta_2)+2R(n)}$
21	$\frac{H_1}{8L_1(1+\theta_1)-4E(n)} + \frac{L_2H_1}{8L_1L_2(1-\theta_2)-4L_1R(n)}$	$\frac{L_1H_1}{8L_1L_2(1+\theta_1)-4L_2E(n)} + \frac{H_1}{8L_1L_2(1-\theta_2)-4R(n)}$	$\frac{L_1(\theta_1+\theta_2)H_4}{4L_1(1+\theta_1)-2E(n)} + \frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1-\theta_2)-2R(n)}$
22	$\frac{L_2H_1}{4L_1L_2(2-\theta_1-\theta_2)+2L_2E(n)-2L_1R(n)} + \frac{L_2H_1}{4L_1L_2(2-3\theta_2+\theta_1)+2L_2E(n)-2L_1R(n)}$	$\frac{L_1H_1}{4L_1L_2(2-\theta_1-\theta_2)+2L_2E(n)-2L_1R(n)} + \frac{L_1H_1}{4L_1L_2(2-3\theta_2+\theta_1)+2L_2E(n)-2L_1R(n)}$	$\frac{L_1L_2(\theta_1+\theta_2)H_4}{2L_1L_2(2-\theta_1-\theta_2)+L_2E(n)-L_1R(n)} + \frac{L_1L_2(\theta_1+\theta_2)H_1}{2L_1L_2(2-3\theta_2+\theta_1)+L_2E(n)-L_1R(n)}$
23	$\frac{L_2H_1}{8L_1L_2(1-\theta_2)+4L_1R(n)} + \frac{H_1}{8L_1(1+\theta_1)+4E(n)}$	$\frac{H_1}{8L_2(1-\theta_2)+4R(n)} + \frac{L_1H_1}{8L_1L_2(1+\theta_1)+4l_2E(n)}$	$\frac{L_2(\theta_1+\theta_2)H_4}{4L_2(1-\theta_2)+2R(n)} + \frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1+\theta_1)+2E(n)}$
24	$\frac{L_2H_1}{4L_1L_2(2+\theta_1-3\theta_2)+2L_1R(n)-2L_2E(n)} + \frac{L_2H_1}{4L_1L_2(2-\theta_1-\theta_2)+2L_1R(n)-2L_2E(n)}$	$\frac{L_1H_1}{4L_1L_2(2+\theta_1-3\theta_2)+2L_1R(n)-2L_2E(n)} + \frac{L_1H_1}{4L_1L_2(2-\theta_1-\theta_2)+2L_1R(n)-2L_2E(n)}$	$\frac{L_1L_2(\theta_1+\theta_2)H_4}{2L_1L_2(2+\theta_1-3\theta_2)+L_1R(n)-L_2E(n)} + \frac{L_1L_2(\theta_1+\theta_2)H_1}{2L_1L_2(2-\theta_1-\theta_2)+L_1R(n)-L_2E(n)}$
25	$\frac{L_2H_1}{4L_1L_2(2+\theta_1-3\theta_2)+2L_1R(n)-2L_2E(n)} + \frac{L_2H_1}{4L_1L_2(2-3\theta_2+\theta_1)+2L_2E(n)-2L_1R(n)}$	$\frac{L_1H_1}{4L_1L_2(2+\theta_1-3\theta_2)+2L_1R(n)-2L_2E(n)} + \frac{L_1H_1}{4L_1L_2(2-3\theta_2+\theta_1)+2L_2E(n)-2L_1R(n)}$	$\frac{L_1L_2(\theta_1+\theta_2)H_4}{2L_1L_2(2+\theta_1-3\theta_2)+L_1R(n)-L_2E(n)} + \frac{L_1L_2(\theta_1+\theta_2)H_1}{2L_1L_2(2-3\theta_2+\theta_1)+L_2E(n)-L_1R(n)}$
26	$\frac{L_2H_1}{8L_1L_2(1-\theta_1)-4L_1R(n)} + \frac{L_2H_1}{8L_1L_2(1-\theta_2)-4L_1R(n)}$	$\frac{L_1H_1}{8L_1L_2(1-\theta_1)-4L_1R(n)} + \frac{H_1}{8L_1L_2(1-\theta_2)-4R(n)}$	$\frac{L_2(\theta_1+\theta_2)H_4}{4L_2(1-\theta_1)-2R(n)} + \frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1-\theta_2)-2R(n)}$
27	$\frac{H_1}{8L_1(1-\theta_2)-4E(n)} + \frac{H_1}{8L_1(1-\theta_1)-4E(n)}$	$\frac{L_1H_1}{8L_1L_2(1-\theta_2)-4L_2E(n)} + \frac{L_1H_1}{8L_1L_2(1-\theta_1)-4L_2E(n)}$	$\frac{L_1(\theta_1+\theta_2)H_4}{4L_1(1-\theta_2)-2E(n)} + \frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1-\theta_1)-2E(n)}$
28	$\frac{L_2H_1}{8L_1L_2(1-\theta_2)+4L_1R(n)} + \frac{L_2H_1}{8L_1L_2(1-\theta_1)+4L_1R(n)}$	$\frac{H_1}{8L_2(1-\theta_2)+4R(n)} + \frac{H_1}{8L_2(1-\theta_1)+4R(n)}$	$\frac{L_2(\theta_1+\theta_2)H_4}{4L_2(1-\theta_2)+2R(n)} + \frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1-\theta_1)+2R(n)}$
29	$\frac{H_1}{8L_1(1-\theta_1)+4E(n)} + \frac{L_1H_1}{8L_1L_2(1-\theta_2)+4L_2E(n)}$	$\frac{L_1H_1}{8L_1L_2(1-\theta_1)+2L_2E(n)} + \frac{H_1}{8L_1(1-\theta_2)+4E(n)}$	$\frac{L_1(\theta_1+\theta_2)H_4}{4L_1(1-\theta_1)+2E(n)} + \frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1-\theta_2)+2E(n)}$
34	$\frac{H_1}{8L_1(1-\theta_2)-4E(n)} + \frac{L_2H_1}{8L_1L_2(1-\theta_2)-4L_1R(n)}$	$\frac{L_1H_1}{8L_1L_2(1-\theta_2)-4L_2E(n)} + \frac{H_1}{8L_1L_2(1-\theta_2)-4R(n)}$	$\frac{L_1(\theta_1+\theta_2)H_4}{4L_1(1-\theta_2)-2E(n)} + \frac{L_2(\theta_1+\theta_2)H_1}{4L_2(1-\theta_2)-2R(n)}$
35	$\frac{L_2H_1}{8L_1L_2(1-\theta_2)+4L_1R(n)} + \frac{L_1H_1}{8L_1L_2(1-\theta_2)+4L_2E(n)}$	$\frac{H_1}{8L_2(1-\theta_2)+4R(n)} + \frac{H_1}{8L_1(1-\theta_2)+4E(n)}$	$\frac{L_2(\theta_1+\theta_2)H_4}{4L_2(1-\theta_2)+2R(n)} + \frac{L_1(\theta_1+\theta_2)H_1}{4L_1(1-\theta_2)+2E(n)}$

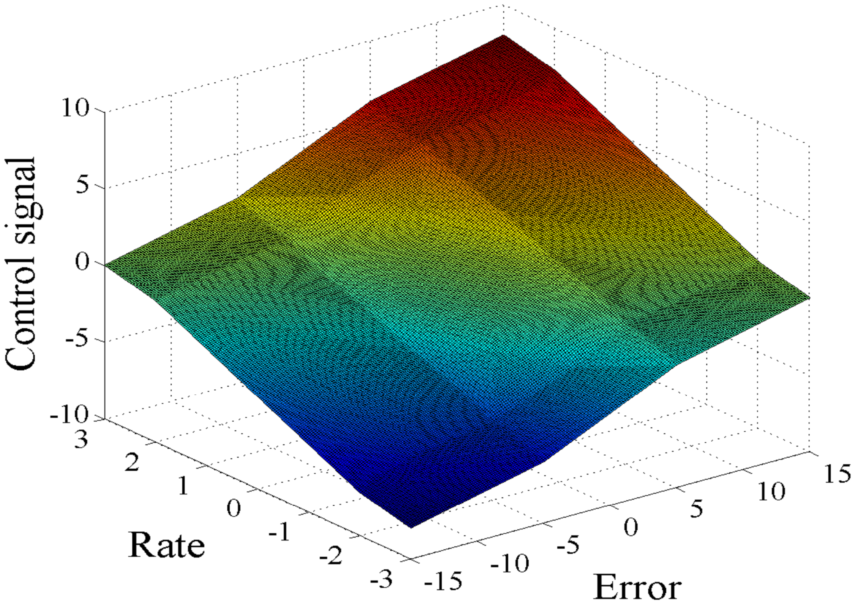


Figure 3.21: The control surface produced by the IT2 FLC ($H_1 = 8$)

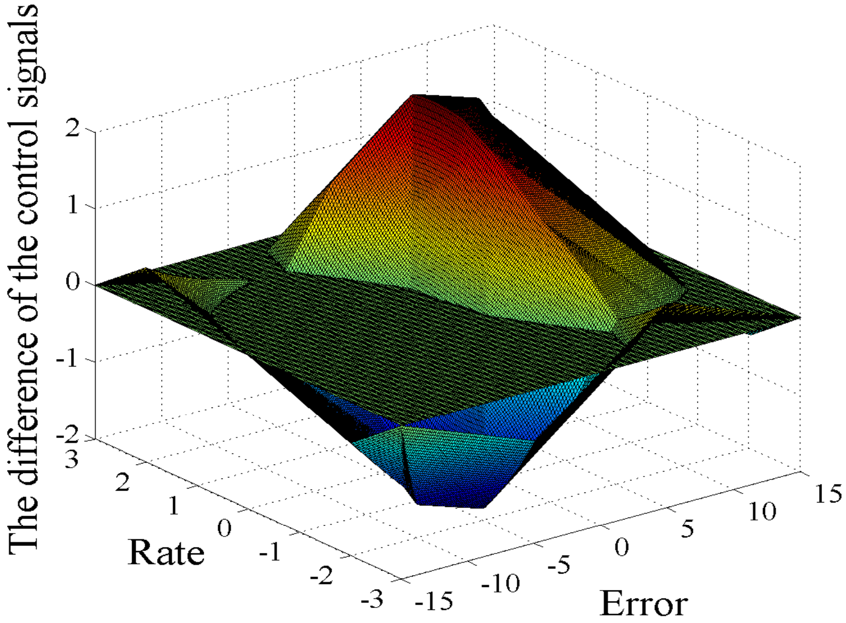


Figure 3.22: The surface difference between the IT2 FLC and the T1 FLC ($H_1 = 8$)

Chapter 4

Analytical Structure and Characteristics of Non-symmetric Karnik-Mendel Type-Reduced Interval Type-2 Fuzzy PI and PD Controllers

Analytical structure is an efficient method of establishing the input-output relationship of a fuzzy logic system. Chapter 3 established the analytical structure for a class of symmetric interval type-2 (IT2) fuzzy PD/ PI controller and addressed the potential advantages of the IT2 fuzzy logic controller (FLC) over the type-1 (T1) FLC by identifying four interesting properties using the derived analytical structure of the symmetric IT2 FLC. However, the results reported in Chapter 3 are limited to the class of symmetric IT2 FLC. To overcome this limitation, this chapter considers a class of non-symmetric IT2 fuzzy PD / PI controller. This chapter will establish the analytical structure of this class of non-symmetric IT2 fuzzy controller, examine whether the four properties of the symmetric IT2 FLC still hold true, and establish whether there is any unique characteristic of the non-symmetric IT2 FLC existing.

The following parts of this chapter are organized as: Section I describes the configurations of the non-symmetric IT2 fuzzy PD controller. Section II presents the algorithm to derive their analytical structure followed by the detailed derivation of the analytical structure of IT2 FLCs in Section III. Section IV presents results from a comparative study between non-symmetric IT2 fuzzy PD controller and its T1 counterpart to highlight its unique characteristics.

4.1 Configuration of Non-symmetric Interval T2 Fuzzy PD and PI Controller

The non-symmetric IT2 fuzzy PD controller considered in this chapter has similar structure as the symmetric IT2 fuzzy controller of Chapter 3. They share the same inputs, output, rule base and antecedent sets. The unique difference between the non-symmetric IT2 fuzzy controller and the symmetric one is that the non-symmetric IT2 fuzzy controller uses four different fuzzy sets in the consequent part of the fuzzy rules in Fig. 4.2, while the symmetric IT2 fuzzy controller has three.

As shown in Fig. 4.1, the notations of the non-symmetric IT2 fuzzy controller are the same with the symmetric IT2 FLC in Chapter 3 for ease of illustration.

For any given input pair $(E(n), R(n))$, the firing sets $R_i, i = 1, 2, 3, 4$ for Rule 1-4 are computed as follows:

$$R_1 = [\underline{R}_1, \overline{R}_1] = [\min(\underline{\mathbf{EP}}, \underline{\mathbf{RP}}), \min(\overline{\mathbf{EP}}, \overline{\mathbf{RP}})] \quad \text{for} \triangle U(n) = H_1 \tag{4.1}$$

$$R_2 = [\underline{R}_2, \overline{R}_2] = [\min(\underline{\mathbf{EP}}, \underline{\mathbf{RN}}), \min(\overline{\mathbf{EP}}, \overline{\mathbf{RN}})] \quad \text{for} \triangle U(n) = H_2 \tag{4.2}$$

$$R_3 = [\underline{R}_3, \overline{R}_3] = [\min(\underline{\mathbf{EN}}, \underline{\mathbf{RP}}), \min(\overline{\mathbf{EN}}, \overline{\mathbf{RP}})] \quad \text{for} \triangle U(n) = H_3 \tag{4.3}$$

$$R_4 = [\underline{R}_4, \overline{R}_4] = [\min(\underline{\mathbf{EN}}, \underline{\mathbf{RN}}), \min(\overline{\mathbf{EN}}, \overline{\mathbf{RN}})] \quad \text{for} \triangle U(n) = H_4 \tag{4.4}$$

Then the endpoint of the type-reduced set $\triangle U_j^{min}$ and $\triangle U_j^{max}$ can be expressed as [54]

$$\triangle U_j^{min} = \frac{\sum_{i=1}^{L-1} \underline{R}_i H_i + \sum_{i=L}^{4} \overline{R}_i H_i}{\sum_{i=1}^{L-1} \underline{R}_i + \sum_{i=L}^{4} \overline{R}_i} \tag{4.5}$$

$$\triangle U_j^{max} = \frac{\sum_{i=1}^{R-1} \overline{R}_i H_i + \sum_{i=R}^{4} \underline{R}_i H_i}{\sum_{i=1}^{R-1} \overline{R}_i + \sum_{i=R}^{4} \underline{R}_i} \tag{4.6}$$

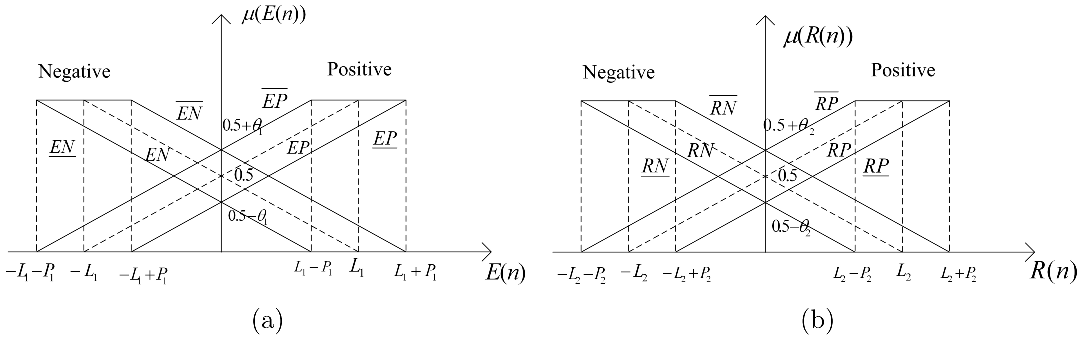
where L and R are the left and right switch point satisfying

$$H_L \leq \triangle U_j^{min} < H_{L-1} \tag{4.7}$$

$$H_R \leq \triangle U_j^{max} < H_{R-1} \tag{4.8}$$

The position of the switch point L and R depend on the values of the singleton consequent sets. Hence, unlike T1 fuzzy controller where the partitions of the input space is independent of the consequent sets, there is a need for the following assumptions in order to simplify the derivation of the analytical structure of IT2 fuzzy PD controller :

The four singleton consequent sets H_4, H_3, H_2, H_1 are equally spaced. Further, it is assumed that $H_4 < H_3 < H_2 < H_1$, which is shown in Fig. 4.2.



(a)

(b)

Figure 4.1: IT2 antecedent FSs: (a) IT2 FSs EN and EP for the input $E(n)$ $(P_1 = 2L_1\theta_1)$. (b) IT2 FSs RN and RP for the input $R(n)$ $(P_2 = 2L_2\theta_2)$.

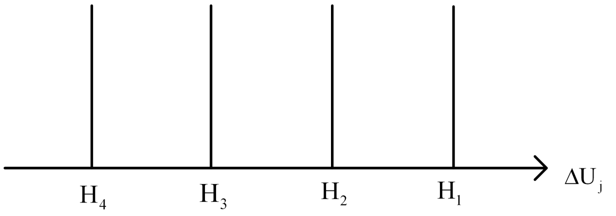


Figure 4.2: Singleton consequent fuzzy sets of non-symmetric IT2 fuzzy PD controller

4.2 Algorithms to Derive the Analytical Structure of non-symmetric IT2 Fuzzy PD Controllers

The analytical structure for a FLS is to establish the mathematical expressions relating the output and the inputs. Similar to the case of a T1 FLC, the main concept used to determine the input-output relationship in an IT2 fuzzy PD controller is to specify the firing strength by dividing the input space into regions and to replace each firing strength with its corresponding mathematical expression. For an IT2 FLC, the key step is to identify equations for each firing strength that should be used to calculate the endpoints $\triangle U_j^{min}$ and $\triangle U_j^{max}$ via (4.5) and (4.6).

4.2.1 General Idea for Deriving Mathematical Expressions of Each Firing Strength

Eq. (4.5) and (4.6) show that each firing strength R_i for $\triangle U_j^{min}$ and $\triangle U_j^{max}$ can be the lower bound or the upper bound of the corresponding firing set $[\underline{R}_i, \overline{R}_i]$ only. Whether the lower bound $\underline{R}_i$ or the upper bound $\overline{R}_i$ is used as the firing strength is determined by the position of the switch points L and R. Once the position of the switch point L and R are known, the endpoints $\triangle U_j^{min}$ and $\triangle U_j^{max}$ can be expressed into an embedded T1 FLS and each of their firing strength, R_i, can be specified by considering the Zadeh AND operation of $\underline{R}_i$ or $\overline{R}_i$. Hence, the key step of specifying each firing strength R_i is to identify each possible position of the switch point L and R by dividing the input space into parts, each of which corresponds to a position of switch point and an embedded T1 FLS.

Since deriving the mathematical expressions for $\triangle U_j^{min}$ is similar to that of $\triangle U_j^{max}$, $\triangle U_j^{min}$ is taken as an example for illustration. (4.5) indicates that the switch point L must be positioned at one of the three singleton consequent sets and thus L can assume one of the three values i.e. $L = \{4, 3, 2\}$. Each value of L corresponds to an unique embedded T1 FLS used to calculate $\triangle U_j^{min}$. Based on (4.5) and (4.8), the embedded T1 FLS and the output condition corresponding to each value of the switch point L may be established as follows:

$$\begin{aligned}
\text{Mode 1}: &\quad \text{When } H_4 \leq \triangle U_j^{min} \leq H_3 \Leftrightarrow L = 4, \\
\triangle U_j^{min} &= \triangle U_{j1}^{min} = \frac{\overline{R}_4 * H_4 + \underline{R}_3 * H_3 + \underline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \underline{R}_3 + \underline{R}_2 + \underline{R}_1} \qquad (4.9) \\
\text{Mode 2}: &\quad \text{When } H_3 \leq \triangle U_j^{min} \leq H_2 \Leftrightarrow L = 3 \\
\triangle U_j^{min} &= \triangle U_{j2}^{min} = \frac{\overline{R}_4 * H_4 + \overline{R}_3 * H_3 + \underline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \overline{R}_3 + \underline{R}_2 + \underline{R}_1} \qquad (4.10) \\
\text{Mode 3}: &\quad \text{When } H_2 \leq \triangle U_j^{min} \leq H_1 \Leftrightarrow L = 2 \\
\triangle U_j^{min} &= \triangle U_{j3}^{min} = \frac{\overline{R}_4 * H_4 + \overline{R}_3 * H_3 + \overline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \overline{R}_3 + \overline{R}_2 + \underline{R}_1} \qquad (4.11)
\end{aligned}$$

To identify when the three embedded T1 FLSs in (4.9)-(4.11) are respectively used as the left endpoint $\triangle U_j^{min}$ in the input space, the input conditions for each of them may be established as follows by analyzing the output conditions for each embedded T1 FLSs, i.e. $H_{i+1} \leq \triangle U_j^{min} \leq H_i, i = 1, 2, 3$:

1. The condition when the switch point changes from the position of $L = 4$ to $L = 3$ and vice versa can be established as

$$\triangle U_j^{min} = \triangle U_{j1}^{min} = \triangle U_{j2}^{min} = H_3 \qquad (4.12)$$

By replacing $\triangle U_{j1}^{min}$ and $\triangle U_{j2}^{min}$ with the expressions (4.9) and (4.10), the above equation may be written as

$$\begin{aligned}
&\frac{\overline{R}_4 * H_4 + \underline{R}_3 * H_3 + \underline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \underline{R}_3 + \underline{R}_2 + \underline{R}_1} \\
&= \frac{\overline{R}_4 * H_4 + \overline{R}_3 * H_3 + \underline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \overline{R}_3 + \underline{R}_2 + \underline{R}_1} = H_3 \\
&\Leftrightarrow \overline{R}_4(H_4 - H_3) = \underline{R}_2(H_3 - H_2) + 2\underline{R}_1(H_3 - H_1) \qquad (4.13)
\end{aligned}$$

Because of the assumption that the four consequent sets are equally spaced and $H_4 < H_2 < H_3 < H_1$, the condition above may be reduced to

$$\overline{R}_4 = \underline{R}_2 + 2\underline{R}_1 \qquad (4.14)$$

Further, the input subregion for Mode 1 satisfies the following inequalities:

$$\triangle U_j^{min} = \triangle U_{j1}^{min} \leq H_3 \Leftrightarrow \overline{R}_4 \geq \underline{R}_2 + 2\underline{R}_1 \tag{4.15}$$

and the subregion for Mode 2 satisfies the following inequalities:

$$\triangle U_j^{min} = \triangle U_{j2}^{min} \geq H_3 \Leftrightarrow \overline{R}_4 \leq \underline{R}_2 + 2\underline{R}_1 \tag{4.16}$$

2. The condition when the switch point changes from the position of $L = 3$ to $L = 2$ and vice versa can be established as

$$\triangle U_j^{min} = \triangle U_{j2}^{min} = \triangle U_{j3}^{min} = H_2 \tag{4.17}$$

By replacing $\triangle U_{j2}^{min}$ and $\triangle U_{j3}^{min}$ with the corresponding expression in (4.10)-(4.11), the above equation may be written as

$$\begin{aligned}&\frac{\overline{R}_4 * H_4 + \overline{R}_3 * H_3 + \underline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \overline{R}_3 + \underline{R}_2 + \underline{R}_1} \\ &= \frac{\overline{R}_4 * H_4 + \overline{R}_3 * H_3 + \overline{R}_2 * H_2 + \underline{R}_1 * H_1}{\overline{R}_4 + \overline{R}_3 + \overline{R}_2 + \underline{R}_1} = H_3 \\ &\Leftrightarrow \overline{R}_4(H_4 - H_2) + \overline{R}_3(H_3 - H_2) = \underline{R}_1(H_2 - H_1)\end{aligned} \tag{4.18}$$

The assumptions that the four singleton consequent sets are equally spaced and $H_4 < H_3 < H_2 < H_1$ reduce the condition established above to

$$2\overline{R}_4 + \overline{R}_3 = \underline{R}_1 \tag{4.19}$$

Further, the subregion in the input space for Mode 2 satisfies the following equality:

$$\triangle U_j^{min} = \triangle U_{j1}^{min} \leq H_2 \Leftrightarrow 2\overline{R}_4 + \overline{R}_3 \geq \underline{R}_1 \tag{4.20}$$

and the subregion for Mode 3 satisfy the following inequalities:

$$\triangle U_j^{min} = \triangle U_{j3}^{min} \geq H_2 \Leftrightarrow 2\overline{R}_4 + \overline{R}_3 \leq \underline{R}_1 \tag{4.21}$$

From the input conditions (4.15)-(4.16) and (4.20)-(4.21), the following can be found:

1. The regions in the input space where the left endpoint $\triangle U_j^{min}$ is operating at Mode 1 and Mode 2 satisfy the condition (4.15) and (4.16), respectively. The subregions for Mode 3 satisfy (4.21), i. e. $2\overline{R}_4 + \overline{R}_3 \leq \underline{R}_1$, and thus it can be concluded that in the subregions for Mode 3, the inequality (4.16), i. e. $\overline{R}_4 \leq \underline{R}_2 + 2\underline{R}_1$, must hold true because all $\underline{R}_i, \overline{R}_i, i = 1, 2, 3, 4$ are nonnegative. Since the subregions operating in Mode 1 satisfy (4.15), i.e. $\overline{R}_4 \geq \underline{R}_2 + 2\underline{R}_1$, while those for Mode 2 and 3 satisfy (4.16), i. e. $\overline{R}_4 \leq \underline{R}_2 + 2\underline{R}_1$, the subregion where the left endpoint $\triangle U_j^{min}$ operates in Mode 1 can be identified using (4.15) due to the contradiction between (4.15) and (4.16).

2. Similarity, the regions which correspond to Mode 2 and Mode 3 satisfy (4.20) and (4.21), respectively. The region for Mode 1 in the input space also satisfies the inequality (4.15), i.e. $\overline{R}_4 \geq \underline{R}_2 + 2\underline{R}_1$, and thus they must satisfy (4.20), i.e. $2\overline{R}_4 + \overline{R}_3 \geq \underline{R}_1$. Since the subregions that correspond to Mode 3 satisfy (4.21) while the subregions for Mode 1 and Mode 2 satisfy (4.20), the subregion for Mode 3 can be identified using the inequality (4.21).

In summary, the regions in the input space that correspond to Mode 1 and 3 can be identified according to (4.15) and (4.21), respectively. The other region in the input space corresponds to Mode 2.

4.2.2 The Algorithm for Deriving Mathematical Expressions of Each Firing Strength

Based on the general idea for specifying each firing strength in Section 4.2.1, this subsection will present an algorithm for deriving mathematical expressions of each firing strength.

From (4.9)-(4.11), the following properties of the firing strength for $\triangle U_j^{min}$ can be observed:

1. The weight associated with H_1 is always the lower bound of the firing set for Rule 1. Similarly, H_4 is weighted by the upper bound of the firing set for Rule 4. Consequently, the weight on H_1 and H_4 is independent of the switch points L.

2. In Mode 1, H_3 is weighted by the lower bound $\underline{R}_3$, while the corresponding upper bounds $\overline{R}_3$ is used to weight H_3 in Mode 2 and 3. On the other hand, the lower bound $\underline{R}_2$ weights H_2 in Mode 1 and 2, while in Mode 3 H_2 is weighted by the corresponding upper bound $\overline{R}_2$.

The first property shows that the firing strength of Rule 1 and Rule 4 used to calculate $\triangle U_j^{min}$ is independent of the switch point L, and the firing strength of Rule 1 and Rule 4 is always governed by the lower and upper bound of the firing set, i.e. $R_1 = \underline{R}_1$, $R_4 = \overline{R}_4$. Therefore, the first step in partitioning the input space in order to derive closed form firing level is performed by considering the outcomes of Zadeh AND operations (minimum operator) for Rule 1 and Rule 4. The weights associated with H_2 and H_3 can be the lower or upper bound of their corresponding firing set, depending on which embedded T1 FLS in (4.9)-(4.11) dominates the sub-region. Hence, the weights for H_2 and H_3 can not be specified before the subregions for each mode are identified to clarify which bound of the firing sets for Rule 2 and Rule 3 should be used to weight H_2 and H_3.

As discussed in Section 4.2.1, the regions for Mode 1 and Mode 3 can be identified using (4.15) and (4.21). (4.15) indicates that whether the left endpoint $\triangle U_j^{min}$ is operating at Mode 1 depends on the relative value of $\underline{R}_1$, $\underline{R}_2$ and $\overline{R}_4$. Similarly, (4.21) indicates that whether the left endpoint $\triangle U_j^{min}$ is operating at Mode 3 depends on the relative value of $\underline{R}_1$, $\overline{R}_3$ and $\overline{R}_4$. To identify the regions for Mode 1 and Mode 3 using (4.15) and (4.21), the following strategies will be used in the second step:

- *The region for Mode 1*: The mathematical expression of $\underline{R}_2$ is first obtained by considering the following Zadeh AND operation in the whole input space:

$$\underline{R}_2 = \min(\underline{\mathbf{EP}}, \underline{\mathbf{RN}}) \tag{4.22}$$

  Then the subregion that correspond to Mode 1 can be identified by examining the relative value of $\underline{R}_1$, $\underline{R}_2$ and $\overline{R}_4$ via (4.15).

- *The region for Mode 3*: The mathematical equations for $\overline{R}_3$ is first derived by considering the corresponding Zadeh operation:

$$\overline{R}_3 = \min(\overline{\mathbf{EN}}, \overline{\mathbf{RP}}) \tag{4.23}$$

  Then the subregions for Mode 3 may be identified by examining the relative value of $\underline{R}_1$, $\overline{R}_3$ and $\overline{R}_4$ according to (4.21).

Once the regions for Mode 1 and Mode 3 are respectively identified, the firing strength for Rule 2 and Rule 3 may also be mathematically expressed by considering the corresponding Zadeh AND operations in the subregions. In summary, the proposed algorithm to derive the mathematical expression of the firing strength for $\triangle U_j^{min}$ may be presented as:

- *Step 1* : The firing strength of Rule 1 ($\underline{R}_1$) and Rule 4 ($\overline{R}_4$) can be specified by dividing the input space via the outcomes of Zadeh AND operations (minimum operator) for Rule 1 and Rule 4 i.e.

$$R_1 = \underline{R}_1 = \min\{\underline{\mathbf{EP}}, \underline{\mathbf{RP}}\} \tag{4.24}$$

$$R_4 = \overline{R}_4 = \min\{\overline{\mathbf{EN}}, \overline{\mathbf{RN}}\} \tag{4.25}$$

- *Step 2* : This step is to identify the regions in the input space that correspond to Mode 1 and Mode 3 respectively. Then the other region must correspond to Mode 2.

  – *Step 2.1* : The mathematical expressions for $\underline{R}_2$ can be derived by dividing the input space via the corresponding Zadeh AND operation:

  $$\underline{R}_2 = \min(\underline{\mathbf{EP}}, \underline{\mathbf{RN}}) \tag{4.26}$$

  The partitions obtained using $\underline{R}_1$, $\overline{R}_4$ in Step 1 and $\underline{R}_2$ are further examined to find out the group for Mode 1 (M_{L1}) satisfying the following inequality (4.15):

  $$\text{Mode 1}: \quad \overline{R}_4 \geq \underline{R}_2 + 2\underline{R}_1$$

  – *Step 2.2* : The mathematical expression for $\overline{R}_3$ can be specified by dividing the input space via the corresponding Zadeh AND operation:

  $$\overline{R}_3 = \min(\overline{\mathbf{EN}}, \overline{\mathbf{RP}}) \tag{4.27}$$

The relative values of $\overline{R}_4$, $\underline{R}_1$ obtained in Step 1 and $\overline{R}_3$ are examined to find out the group for Mode 3 (M_{L3}) which satisfies the inequality (4.21):

$$\text{Mode 3:} \quad 2\overline{R}_4 + \overline{R}_3 \le \underline{R}_1$$

- *Step 3* : In Step 3, the firing strength associated with Rule 2 and 3 can be specified via the corresponding Zadeh AND operation.

1. *Step 3.1* : Since Step 2.1 divides the whole input space into two parts: the group M_{L1} for Mode 1 and the other region for Mode 2 and 3, the firing strength associated with Rule 3 can be respectively specified as follows:

   In the group M_{L1}, H_3 is weighted by its lower bound $\underline{R}_3$. By considering the Zadeh AND operation the firing strength for Rule 3 may be specified as

   $$R_3 = \underline{R}_3 = \min\{\underline{\mathbf{EN}}, \underline{\mathbf{RP}}\} \tag{4.28}$$

   In the other regions for Mode 2 and Mode 3, the corresponding upper bound $\overline{R}_3$ weights H_3 and thus the firing strength for Rule 3 can be specified as

   $$R_3 = \overline{R}_3 = \min\{\overline{\mathbf{EN}}, \overline{\mathbf{RP}}\} \tag{4.29}$$

2. *Step 3.2* : The whole input space is divided into two parts in Step 2.2: the group M_{L3} for Mode 3 and the other region for Mode 1 and Mode 2. Therefore, the firing strength for Rule 2 can be specified by using Zadeh AND operation to divide the input space as follows:

The upper bound $\overline{R}_2$ weights H_2 in the region M_{L3}. The regions M_{L3} is divided by the Zadeh AND operation as

$$R_2 = \overline{R}_2 = \min\{\overline{\mathbf{EP}}, \overline{\mathbf{RN}}\} \tag{4.30}$$

In the other regions, H_2 is weighted by the corresponding lower bound $\underline{R}_2$. The Zadeh AND operation divides the regions as

$$R_2 = \underline{R}_2 = \min\{\underline{\mathbf{EP}}, \underline{\mathbf{RN}}\} \tag{4.31}$$

- *Step 4* : Superimpose all the partitions obtained by considering the mode switch and Zadeh AND operations.

Similarly, the firing strength for the right endpoint $\triangle U_j^{max}$ can be specified. The procedures to derive the analytical structure of IT2 fuzzy PD controller can be summarized in Fig. 4.3.

4.3 Derivation of the Analytical Structure of non-symmetric IT2 Fuzzy PD Controller

By following the proposed algorithm presented in the last section, the analytical structure of the non-symmetric fuzzy PD controller will be derived in this section.

4.3.1 The Expressions of the Firing Strength for $\triangle U_j^{min}$ and $\triangle U_j^{max}$

The firing strength for the left endpoint $\triangle U_j^{min}$ can be specified by following the proposed algorithm in Section 4.2.2 as follows:

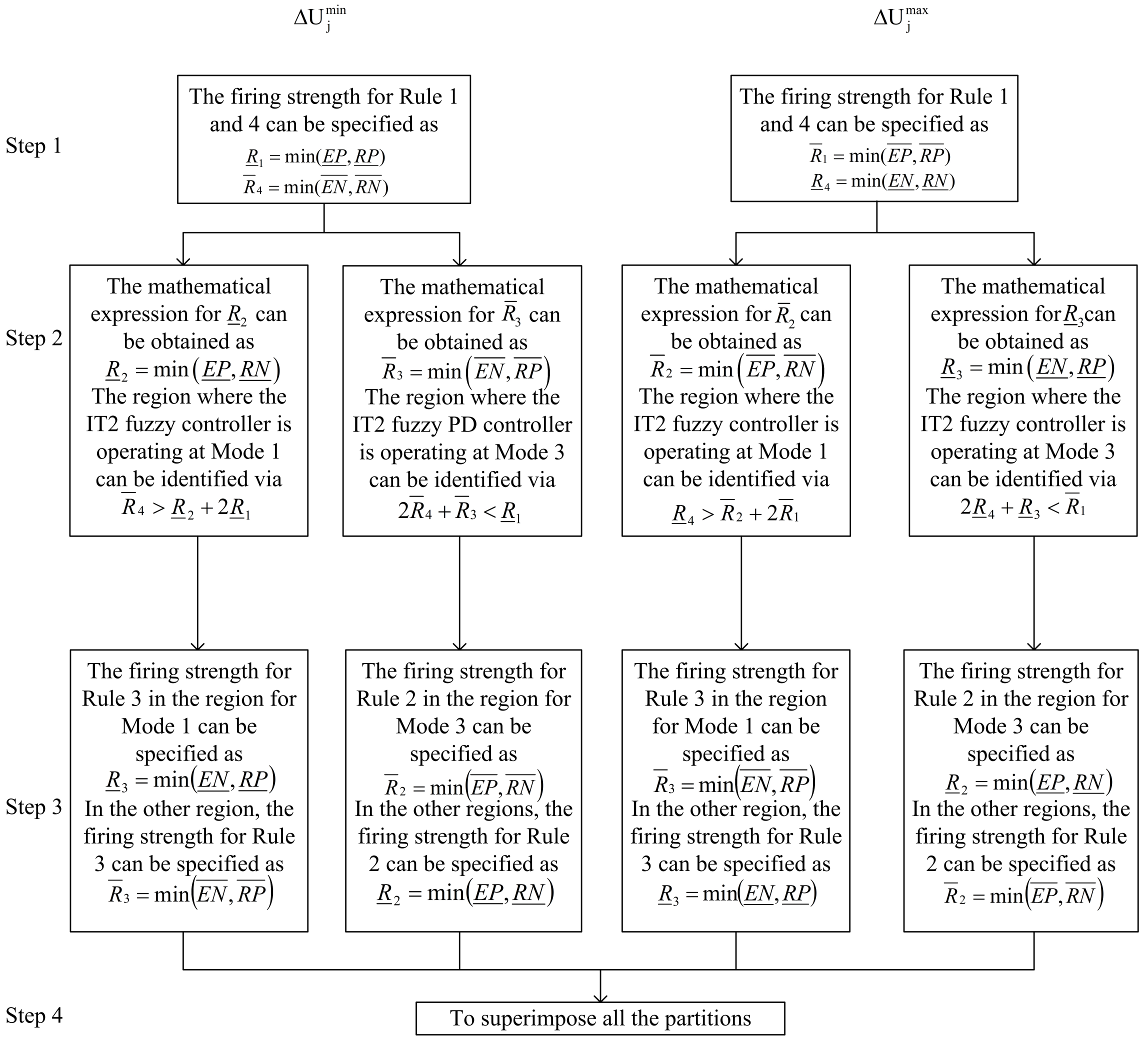

$\Delta U_j^{\min}$
$\Delta U_j^{\max}$
Step 1
Step 2
Step 3
Step 4
The firing strength for Rule 1 and 4 can be specified as $\underline{R}_1 = \min(\underline{EP}, \underline{RP})$ $\overline{R}_4 = \min(\overline{EN}, \overline{RN})$
The firing strength for Rule 1 and 4 can be specified as $\overline{R}_1 = \min(\overline{EP}, \overline{RP})$ $\underline{R}_4 = \min(\underline{EN}, \underline{RN})$
The mathematical expression for $\underline{R}_2$ can be obtained as $\underline{R}_2 = \min(\underline{EP}, \underline{RN})$ The region where the IT2 fuzzy controller is operating at Mode 1 can be identified via $\overline{R}_4 > \underline{R}_2 + 2\underline{R}_1$
The mathematical expression for $\overline{R}_3$ can be obtained as $\overline{R}_3 = \min(\overline{EN}, \overline{RP})$ The region where the IT2 fuzzy PD controller is operating at Mode 3 can be identified via $2\overline{R}_4 + \overline{R}_3 < \underline{R}_1$
The mathematical expression for $\overline{R}_2$ can be obtained as $\overline{R}_2 = \min(\overline{EP}, \overline{RN})$ The region where the IT2 fuzzy controller is operating at Mode 1 can be identified via $\underline{R}_4 > \overline{R}_2 + 2\overline{R}_1$
The mathematical expression for $\underline{R}_3$can be obtained as $\underline{R}_3 = \min(\underline{EN}, \underline{RP})$ The region where the IT2 fuzzy controller is operating at Mode 3 can be identified via $2\underline{R}_4 + \underline{R}_3 < \overline{R}_1$
The firing strength for Rule 3 in the region for Mode 1 can be specified as $\underline{R}_3 = \min(\underline{EN}, \underline{RP})$ In the other region, the firing strength for Rule 3 can be specified as $\overline{R}_3 = \min(\overline{EN}, \overline{RP})$
The firing strength for Rule 2 in the region for Mode 3 can be specified as $\overline{R}_2 = \min(\overline{EP}, \overline{RN})$ In the other regions, the firing strength for Rule 2 can be specified as $\underline{R}_2 = \min(\underline{EP}, \underline{RN})$
The firing strength for Rule 3 in the region for Mode 1 can be specified as $\overline{R}_3 = \min(\overline{EN}, \overline{RP})$ In the other region, the firing strength for Rule 3 can be specified as $\underline{R}_3 = \min(\underline{EN}, \underline{RP})$
The firing strength for Rule 2 in the region for Mode 3 can be specified as $\underline{R}_2 = \min(\underline{EP}, \underline{RN})$ In the other regions, the firing strength for Rule 2 can be specified as $\overline{R}_2 = \min(\overline{EP}, \overline{RN})$
To superimpose all the partitions

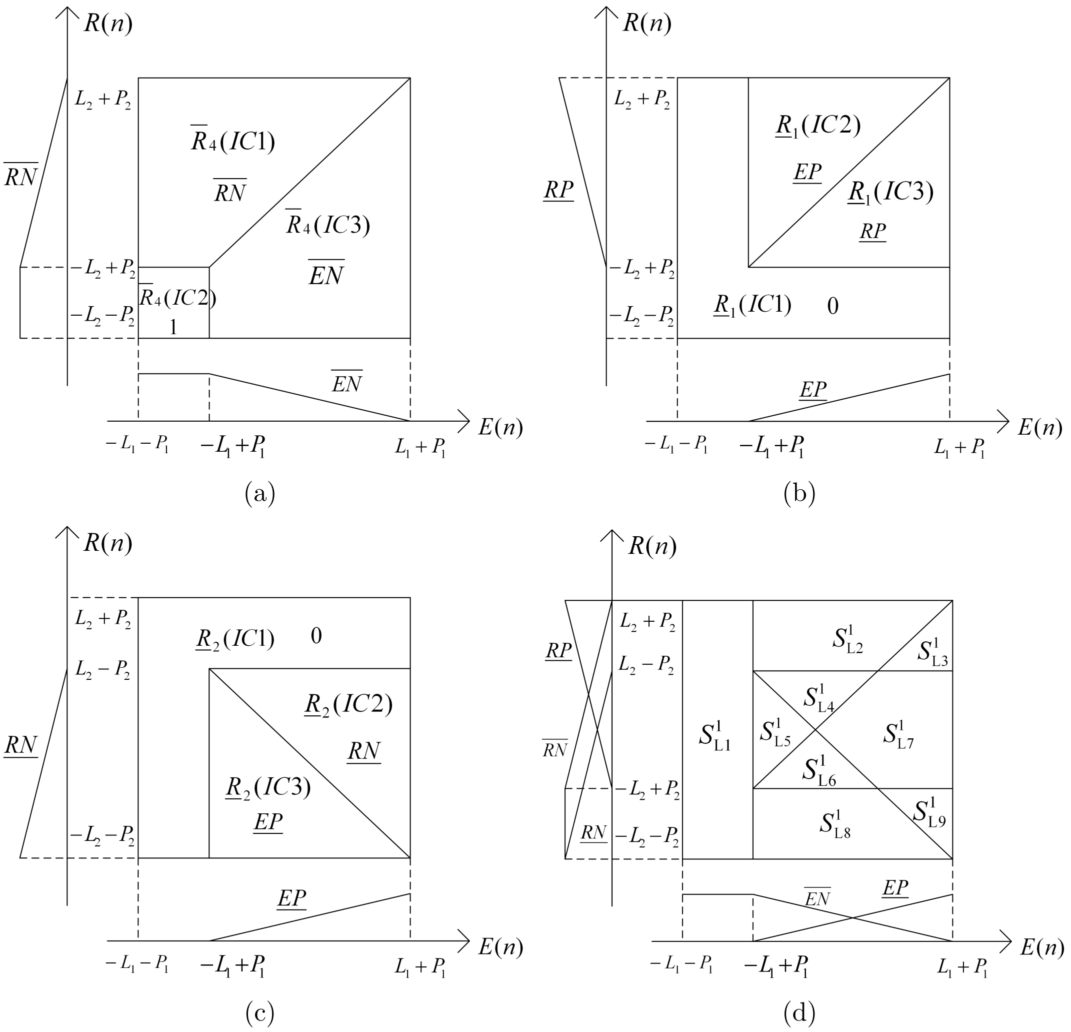
Figure 4.3: Flowchart of the algorithm to specify the firing strength of the non-symmetric IT2 fuzzy PD controller

1. Step 1 is to mathematically express the firing strength for Rule 1 ($\underline{R}_1$) and Rule 4 ($\overline{R}_4$) by dividing the input space into regions.

   - The firing strength of Rule 1 ($\underline{R}_1$) can be calculated via (4.24)

     $$R_1 = \underline{R}_1 = \min\{\underline{\mathbf{EP}}, \underline{\mathbf{RP}}\}.$$

     Hence, the input space is divided into three regions: $\underline{R}_1$(IC1, IC2 and IC3) in Fig. 4.4a. In these three regions, the firing strength $\underline{R}_1$ is 0, $\underline{EP}$ and $\underline{RP}$ respectively.



(a)

(b)

(c)

(d)

Figure 4.4: Partition of the input space by (a) $\bar{R}_4$. (b) $\underline{R}_1$. (c) $\underline{R}_2$. (d) The superimposition of $\bar{R}_4$, $\underline{R}_1$ and $\underline{R}_2$.

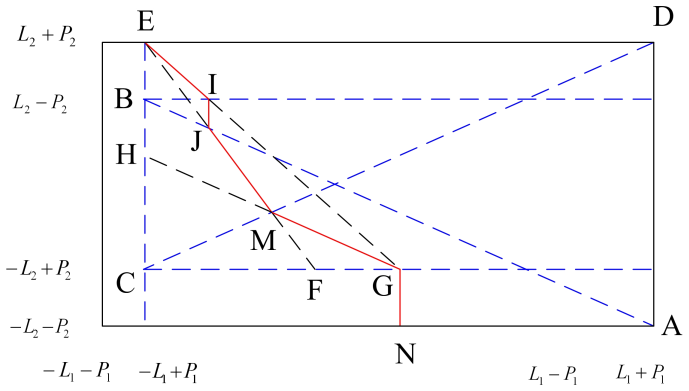


Figure 4.5: The region below the red line where the embedded T1 FLS in Mode 1 is used as $\triangle U_j^{min}$

- As the firing strength of Rule 4 is defined as (4.25)

$$R_4 = \overline{R}_4 = \min\{\overline{\mathbf{EN}}, \overline{\mathbf{RN}}\}$$

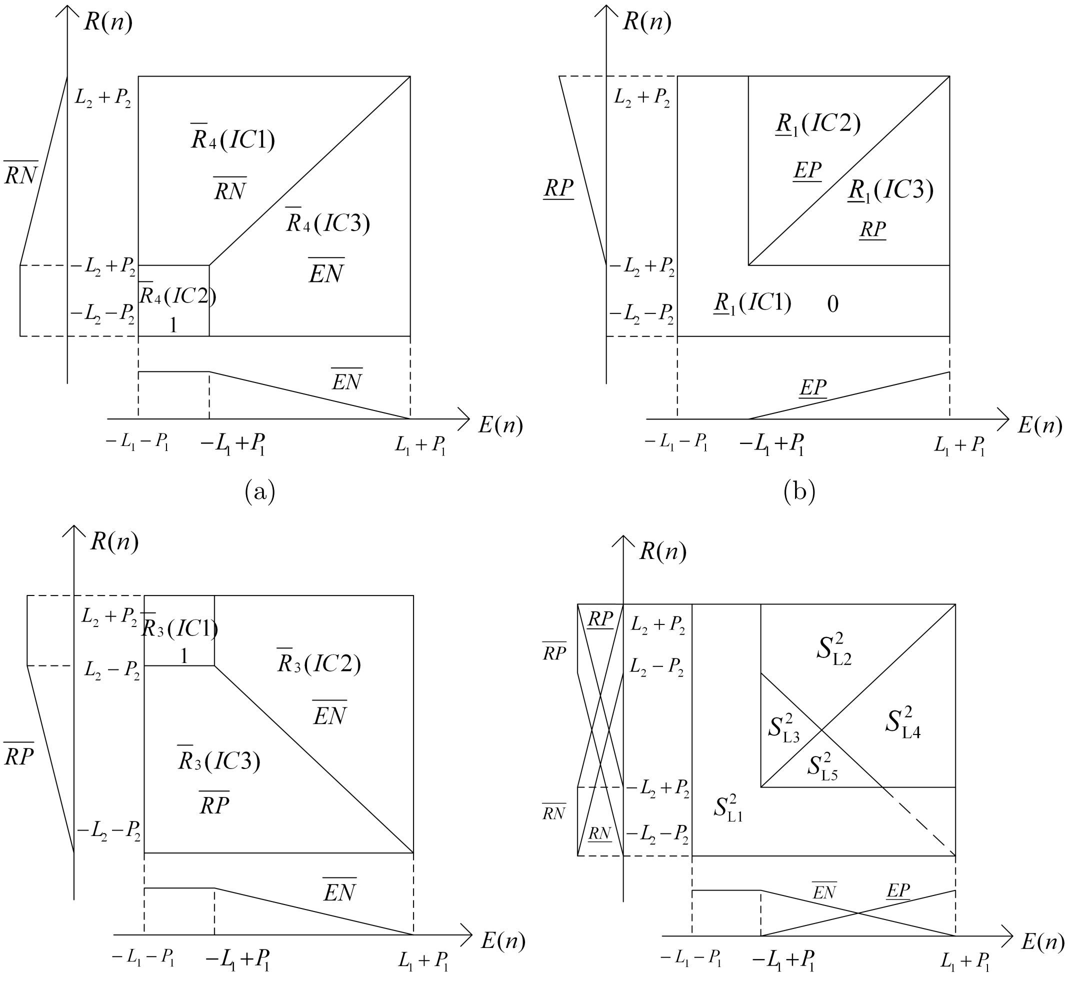
the input space is divided into three regions: $\overline{R}_4$(IC1, IC2 and IC3) in Fig. 4.4b. In these three regions, the firing strength $\overline{R}_4$ is $\overline{RN}$, 1 and $\overline{EN}$ respectively.

2. Step 2 is to sperate the input space into the three groups where the three embedded T1 FLSs in (4.9)-(4.11) are respectively used as the left endpoint $\triangle U_j^{min}$.

   (a) In Step 2.1 The lower bound $\underline{R}_2$ can be specified in the whole input space by considering the corresponding Zadeh operation:

$$\underline{R}_2 = \min(\underline{\mathbf{EP}}, \underline{\mathbf{RN}})$$

the input space is divided into three parts: $\underline{R}_2$ (IC1, IC2 and IC3) in Fig. 4.4(d). In these regions, $\underline{R}_2$ is 0, $\underline{RN}$ and $\underline{EP}$, respectively.



(a) (b) (c) (d)

Figure 4.6: Partition of the input space by (a) $\bar{R}_4$ (b) $\underline{R}_1$ (c) $\bar{R}_3$. (b) The superimposition of $\bar{R}_4$, $\underline{R}_1$ and $\bar{R}_3$.

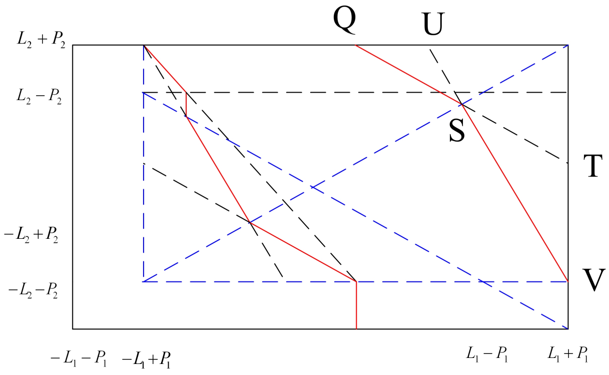


Figure 4.7: The region above the left red line where the embedded T1 FLS in Mode 3 is used as $\triangle U_j^{min}$

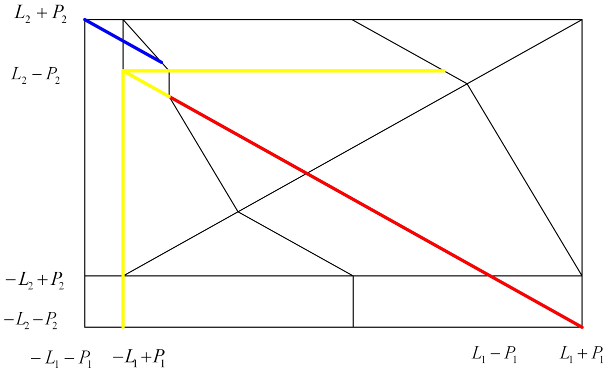


Figure 4.8: The partition of the input space for the left endpoint $\triangle U_j^{min}$

By superimposing the partitions derived in Fig. 4.4(a-c), nine regions in Fig. 4.4(d) are generated: S^1_{L1}-S^1_{L9}. Whether the IT2 fuzzy PD controller operates in Mode 1 (M_{L1}) depends on the relative values of $\overline{R}_4$, $\underline{R}_1$ and $\underline{R}_2$.

- Region S^1_{L1}: $\underline{R}_2 = \underline{R}_1 = 0$ and $\overline{R}_4 > 0$. Hence, $\overline{R}_4 > \underline{R}_2 + 2\underline{R}_1$ and thus for any input pair in Region S^1_{L1}, the IT2 fuzzy PD controller is operating in Mode 1. ($S_{L1} \in M_{L1}$)
- Region S^1_{L2}: $\overline{R}_4 = \overline{RN}, \underline{R}_2 = 0, \underline{R}_1 = \underline{EP}$. The inequality (4.15)

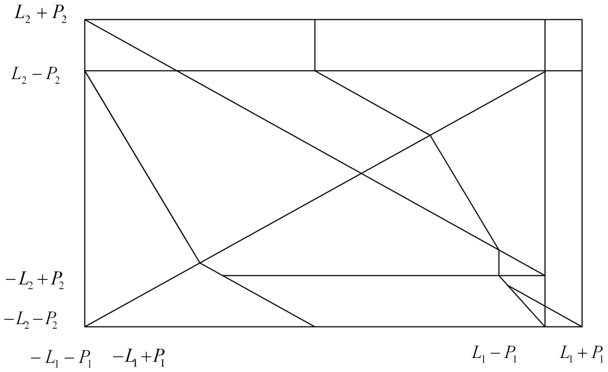


Figure 4.9: The partition of the input space for the right endpoint $\triangle U_j^{max}$

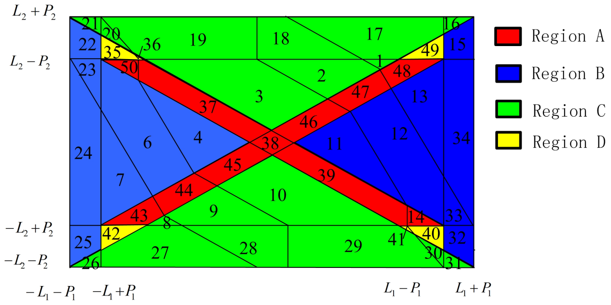


Figure 4.10: The partition of the input space for the IT2 FLC

can be simplified into $\overline{RN} \geq 2\underline{EP}$. As shown in Fig. 4.5, the condition $\overline{RN} = 2\underline{EP}$ is denoted by the red line EG.

- Region S^1_{L3}: $\overline{R}_4 = \overline{EN}, \underline{R}_2 = 0, \underline{R}_1 = \underline{RP}$. The fact $\overline{EN} < \overline{RN}$ and $\overline{RN} < \underline{RP}$ in Region S^1_{L3} lead to $\overline{EN} < \underline{RP}$. Hence, in Region S^1_{L3}, the property $\overline{R}_4 < \underline{R}_2 + 2\underline{R}_1$ holds and the IT2 fuzzy PD controller operates in Mode 2 or 3.

- Region S^1_{L4}: $\overline{R}_4 = \overline{RN}, \underline{R}_2 = \underline{RN}, \underline{R}_1 = \underline{EP}$. The inequality (4.15) leads to $\overline{RN} \geq \underline{RN} + 2\underline{EP}$ which is the left side of the red line IJ in Fig.4.5.

- Region S^1_{L5}: $\overline{R}_4 = \overline{RN}, \underline{R}_2 = \underline{EP}, \underline{R}_1 = \underline{EP}$ and then the condition

(4.15) can be simplified into $\overline{RN} > 3\underline{EP}$. The line EF includes all input pairs where $\overline{RN} = 3\underline{EP}$ holds in Region S_{L5}.

- Region S^1_{L6}: $\overline{R}_4 = \overline{EN}, \underline{R}_2 = \underline{EP}, \underline{R}_1 = \underline{RP}$. The condition in (4.15) leads to $\overline{EN} \geq \underline{EP} + 2\underline{RP}$. The boundary $\overline{EN} = \underline{EP} + 2\underline{RP}$ is denoted by the red line MG in Fig.4.5.
- Region S^1_{L7}: $\overline{R}_4 = \overline{EN}, \underline{R}_2 = \underline{RN}, \underline{R}_1 = \underline{RP}$. Due to the fact $\overline{R}_4 < \underline{R}_2 + 2\underline{R}_1$ in Region S_{L7}, the IT2 fuzzy PD controller is operating at Mode 2 or 3.
- Region S^1_{L8}: $\overline{R}_4 = \overline{EN}, \underline{R}_2 = \underline{EP}, \underline{R}_1 = 0$. The condition (4.15) can be reduced to $EN_U \geq EP_L$, which is the left side of the red line GN in Fig.4.5.
- Region S^1_{L9}: $\overline{R}_4 = \overline{EN}, \underline{R}_2 = \underline{RN}, \underline{R}_1 = 0$. In Region S_{L9}, there exists $\overline{R}_4 < \underline{R}_2 + 2\underline{R}_1$ and thus Region S^1_{L9} does not belong to M_{L1}.

(b) Step 2.2 seeks to identify the region M_{L3} where the IT2 fuzzy PD controller is operating at Mode 3. The upper bound $\overline{R}_3$ is defined as

$$\overline{R}_3 = \min(\overline{\mathbf{EN}}, \overline{\mathbf{RP}}) \tag{4.32}$$

the input space is divided into the three parts: $\overline{R}_3$ (IC 1, IC 2 and IC3) in Fig. 4.6. In these regions, $\overline{R}_3$ is 1, $\overline{EN}$ and $\overline{RP}$, respectively.

The superimposition of the partitions in Fig. 4.4(a-b) and those in Fig. 4.6(a) creates 5 regions: S^2_{L1}-S^2_{L5}. The relative values of $\underline{R}_1$, $\overline{R}_4$ and $\overline{R}_3$ determines whether in any of these 5 regions the IT2 fuzzy PD controller is operating at Mode 3 (M_{L3}).

- Region S^2_{L1}: $\underline{R}_1 = 0$ while the values of both $\overline{R}$ and $\overline{R}_4$ are above zero. Hence, in the region S^2_{L1}, $2\overline{R}_4 + \overline{R}_3 > \underline{R}_1$ and thus the fuzzy controller is operating at Mode 1 or Mode 2.

- Region S^2_{L2}: $\underline{R}_1 = \underline{EP}$, $\overline{R}_4 = \overline{RN}$ and $\overline{R}_3 = \overline{EN}$. The boundary $2\overline{R}_4 + \overline{R}_3 = \underline{R}_1$ is shown by the red line QS in Fig. 4.7.
- Region S^2_{L3}: $\underline{R}_1 = \underline{EP}$, $\overline{R}_4 = \overline{RN}$ and $\overline{R}_3 = \overline{RP}$. In Region S_{L3}, $2\overline{R}_4 + \overline{R}_3 > \underline{R}_1$ and thus the fuzzy controller is operating in Mode 1 or Mode 2.
- Region S^2_{L4}: $\underline{R}_1 = \underline{RP}$, $\overline{R}_4 = \overline{EN}$ and $\overline{R}_3 = \overline{EN}$. The red line SV shows the boundary $2\overline{R}_4 + \overline{R}_3 = \underline{R}_1$ in Fig.4.7.
- Region S^2_{L5}: $\underline{R}_1 = \underline{RP}$, $\overline{R}_4 = \overline{EN}$ and $\overline{R}_3 = \overline{RP}$. Because in this region $2\overline{R}_4 + \overline{R}_3 > \underline{R}_1$ always holds, the fuzzy controller is operating in Mode 1 or Mode 2.

In this manner, the whole input space is divided into the three parts: M_{L1}, M_{L2} and M_{L3}, where the fuzzy controller is operating at Mode 1, Mode 2 or Mode 3, respectively.

3. Step 3 is to specify the firing strength for Rule 2 and Rule 3 by considering the corresponding Zadeh AND operation in the subregions for Mode 1-3

   (a) In the region M_{L1} for Mode 1, the firing strength for Rule 3 can be specified by considering the following Zadeh AND operation:

   $$R_3 = \underline{R}_3 = \min\{\underline{\mathbf{EN}}, \underline{\mathbf{RP}}\}$$

   while in the other regions M_{L2} and M_{L3}, the following Zadeh AND operation is considered to specify the firing strength for Rule 3:

$$R_3 = \overline{R}_3 = \min\{\overline{\mathbf{EN}}, \overline{\mathbf{RP}}\}$$

The red and blue lines in Fig. 4.8 show the divisions of the input space for Rule 3.

(b) As shown in Fig. 4.8, the red lines and yellow lines show the partition for Rule 2. In the subregion M_{L1} and M_{L2}, H_2 is weighted by the lower bound $\underline{R}_2$. Hence, the firing strength for Rule 2 in these regions can be specified by considering

$$R_2 = \underline{R}_2 = \min\{\underline{\mathbf{EP}}, \underline{\mathbf{RN}}\}$$

In the subregion M_{L3}, the firing strength for Rule 2 can be specified using

$$R_2 = \overline{R}_2 = \min\{\overline{\mathbf{EP}}, \overline{\mathbf{RN}}\}$$

4. At last, the partition of the input space for $\triangle U_j^{min}$ in Fig. 4.8 can be obtained by superimposing all the results derived in Step 1-3.

Similarly, the firing strength for $\triangle U_j^{max}$ can be specified by following the proposed algorithm and the partitions of the input space is shown in Fig. 4.9. By superimposing the partitions by $\triangle U_j^{min}$ and $\triangle U_j^{max}$ in Fig. 4.8 and 4.9, the partitions by the IT2 FLC can be obtained in Fig. 4.10. The corresponding firing strength for 50 subregions are listed in Table 4.1.

4.3.2 The Expressions for the non-symmetric IT2 Fuzzy PD Controller

By replacing each firing strength in (4.5) and (4.6) with their expressions in Table 4.1, the expressions for $\triangle U_j^{min}$ in each subregion can be derived. For illustration, the input condition IC1 is taken as an example to show the procedure. Table 4.1 shows that in IC1, $R_4 = R_2 = \overline{RN} = -\frac{1}{2L_2}R(n) + 0.5 + \theta_2$, $R_3 = \overline{EN} = -\frac{1}{2L_1}E(n) + 0.5 + \theta_1$, $R_1 = \underline{EP} = \frac{1}{2L_1}E(n) + 0.5 - \theta_1$. After simple manipulation, $\triangle U_j^{min}$ for IC 1 can be expressed into functions of $E(n)$ and $R(n)$ in (4.33). Since both $\triangle U_j^{min}$ and $\triangle U_j^{max}$ can be expressed into functions of $E(n)$ and $R(n)$, the IT2 fuzzy PD controller is equivalent to a nonlinear PD controller in each subregion.

$$
\begin{aligned}
\triangle U_{jIC1}^{min} &= \frac{R_4^* * H_4 + R_3^* * H_3 + R_2^* * H_2 + R_1^* * H_1}{R_4^* + R_3^* + R_2^* + R_1^*} \\
&= \frac{\overline{RN} * H_4 + \overline{EN} * H_3 + \overline{RN} * H_2 + \underline{EP} * H_1}{\overline{RN} + \overline{EN} + \overline{RN} + \underline{EP}} \\
&= \frac{[-\frac{1}{2L_2}R(n) + 0.5 + \theta_1](H_2 + H_4) + [-\frac{1}{2L_1}E(n) + 0.5 + \theta_1]H_3 + [\frac{1}{2L_1}E(n) + 0.5 - \theta_1]H_1}{2[-\frac{1}{2L_2}R(n) + 0.5 + \theta_2] + [-\frac{1}{2L_1}E(n) + 0.5 + \theta_1] + [\frac{1}{2L_1}E(n) + 0.5 - \theta_1]} \\
&= \frac{-L_1(H_2 + H_4)R(n) + L_2(H_1 - H_3)E(n)}{4L_1L_2(1 + \theta_2) - 2L_1R(n)} \\
&\qquad - \frac{2L_1L_2[(0.5 + \theta_1)(H_2 + H_4) + (0.5 - \theta_1)H_1 + (0.5 + \theta_1)H_3]}{4L_1L_2(1 + \theta_2) - 2L_1R(n)} \\
&= K_p^1E(n) + K_d^1R(n) + \delta^1
\end{aligned}
\tag{4.33}
$$

4.4 Characteristics of the non-symmetric IT2 fuzzy PD controllers

The focus of this section is on using the analytical structure and the equivalent proportional and derivative gains of the IT2 fuzzy PD controller, derived in previous sections, to compare the non-symmetric with the symmetric IT2 FLC and to highlight interesting characteristics of the IT2 fuzzy PD controller. The unique difference between the structure of the non-symmetric IT2 FLC and the symmetric IT2 FLC is one extra consequent set. Fig. 4.11 shows that the whole input space of the symmetric IT2 FLC is divided into 35 separate partitions, each of which is equivalent to a nonlinear PD controller:

$$\triangle {}^{s}U_{jICl} = {}^{s}K_p^l E(n) + {}^{s}K_d^l R(n) + {}^{s}\delta^l \tag{4.34}$$

where ${}^{s}K_p^l$, ${}^{s}K_d^l$ and ${}^{s}\delta^l$ are the proportional, derivative gains and the offset, respectively. As shown in Fig. 4.10, the input space of the IT2 fuzzy PD controller needs to be divided into 49 subregions before the IT2 FLC are expressed into equivalent nonlinear PD controllers:

$$\triangle U_{jICq} = K_p^q E(n) + K_d^q R(n) + \delta^q \tag{4.35}$$

Without losing any generality, the assumption that $\sum_{i=1}^{4} H_i = 0$ is made to simplify the following study. The proportional gain K_p^q, the derivative gain K_d^q and the offset δ^q are tabulated in Table 4.3 and 4.2.

Research results have shown that an IT2 FLC is better able to manage the trade-off between fast rise time and overshoot [88], [14], [19]. Since an IT2 FLC has the structure of a PD controller, its properties can be analyzed by comparing with the characteristics that will enable a PD controller to provide fast rise time with small overshoot following a step change. An ideal nonlinear PD would have the following characteristics: 1) the proportional and derivative gains around the $(E(n), R(n)) = (0, 0)$ origin are as small as possible; 2) the control efforts in external region is relatively large; 3) the control surface of an ideal controller is as smooth as possible. Hence, the following sub-sections focuses on analyzing the characteristics of the proportional and derivative gains of the non-symmetric IT2 FLC in the external regions and around the $(E(n), R(n)) = (0, 0)$ origin in order to draw inference about its control performance.

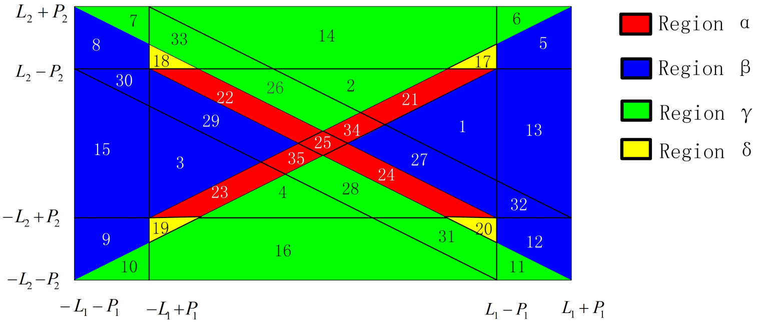


Figure 4.11: The partition of the input space for the symmetic IT2 FLC

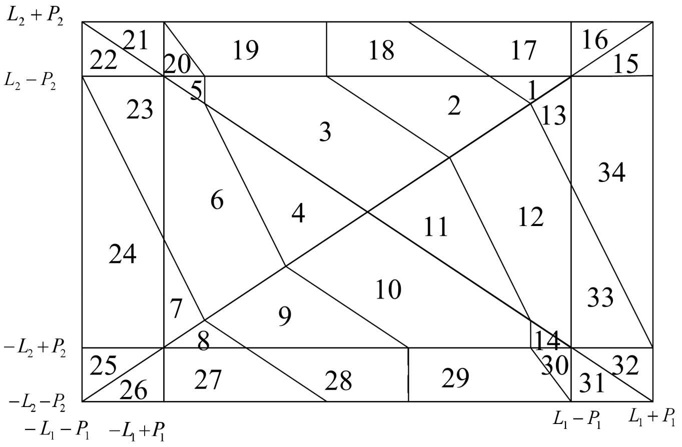


Figure 4.12: The partition of the input space for the IT2 FLC when $\theta_1 = \theta_2$

4.4.1 Comparison of the analytical structure of the non-symmetric IT2 FLC and the T1 FLC

A study of the analytical structure of the symmetric IT2 FLC in Chapter 3 has identified four interesting properties and these properties reveal that it can better alleviate the compromise between faster response and smaller overshoot by providing larger control effort and smoother surface compared to a T1 FLC. To examine whether these properties of the symmetric IT2 still hold true for the non-symmetric FLC, a comparative study of the analytical structure of the non-symmetric IT2 FLC and its T1 counterpart will be presented in the following part of this section. The configurations of the T1 FLC used in the comparative study are provided in the Appendix.

1. **Characteristics of the regions that exist only when $\theta_1 \neq \theta_2$**

   An observation of Fig. 4.10, Table 4.2 and 4.3 indicates that the proportional and derivative gains and the offset of Region B and Region C are functions of the input $E(n)$ and $R(n)$, respectively, while those gains of Region A except IC 38 are the functions of both the inputs $E(n)$ and $R(n)$. Noted that Region C only exist when $\theta_1 \neq \theta_2$ and its size increases as the value of $\theta_1 \neq \theta_2$ increases. As Region A connects Region B and Region C in Fig. 4.10, Region A whose gains are functions of both the inputs $E(n)$ and $R(n)$ may provide smoother transitions between Region B and Region C whose gains are functions of $E(n)$ and $R(n)$ respectively. This architecture enable the non-symmetric FLC to produce smooth surface and thus eliminate the overshoot despite greater control efforts provided by the IT2 FLCs. This property differs from Property 1 of the symmetric IT2 FLC in the subregions only:

Property 1 In the non-symmetric IT2 fuzzy PD controller, the gains K_p^q, K_d^q and δ^q of Region A except IC 38 in Fig. 4.10 are functions of both $E(n)$ and $R(n)$, while these gains of Region B and Region C are functions of only $E(n)$ or $R(n)$. The size of Region A increases as $|\theta_1 - \theta_2|$ increases.

2. **Gains relationship between internal regions and external regions**

Region B and Region C of Fig. 4.10 is the study focus of this subsection. Like the symmetric IT2 FLC, Region B and Region C may be classified into the following two categories:

(a) internal subregions comprising IC 1 - IC 13.

(b) external subregions consisting of the other regions i.e. IC 15 -IC 34.

To identify the relative gains of the internal and external subregions of the non-symmetric IT2 FLC, the property that corresponds to Property 2 for the symmetric IT2 FLC will be established as follows:

Property 2 In the non-symmetric IT2 fuzzy PD controller, the proportional gain of the internal subregions IC 12, 13 (IC 6, 7) is bigger than that of their adjacent external subregion IC 33, 34 (IC 23, 24) while the internal subregion IC 1,2,3,5 (IC 8,9,10,14) exhibit bigger derivative gain than their adjacent external subregion IC 17,18,19,20 (IC 27,28,29,30).

Proof of Property 2 is provided in the Appendix B.

3. **Comparative output values of the non-symmetric IT2 FLC and the T1 FLC**

To identify the relative output values of the non-symmetric IT2 FLC and its T1 counterpart, two properties that corresponds to Property 3 and Property 4 of the symmetric IT2 FLC will be established for the non-symmetric IT2 FLC as follows:

Property 3 Given any input pair in IC 11, 12, 13 (IC 4, 6, 7 due to symmetry in the input space), the non-symmetric IT2 fuzzy PD controller has bigger proportional gain than that of its T1 counterpart, while the non-symmetric IT2 fuzzy PD controller exhibit bigger derivative gain than its T1 counterpart for any input pair in its subregions IC 1, 2, 3, 5 (IC 8, 9, 10, 14). The differences between these gains of the IT2 FLC and the T1 FLC increases as θ_1 and θ_2 increase.

Proof of Property 3 is provided in the Appendix B.

Property 4 Any input pair $(E(n), R(n))$ in IC 1–IC 4, and IC 6 – IC 13 of the non-symmetric IT2 fuzzy PD controller will produce an output that is bigger in magnitude than that of its T1 counterpart when $\theta_1 = \theta_2$; and the difference increases as θ_1 and θ_2 increase.

Proof of Property 4 is provided in the Appendix B.

Comments: Property 3 and Property 4 slightly differ from the properties for the symmetric IT2 FLC in the subregions. Hence, it can be concluded that both the symmetric and non-symmetric IT2 FLC can produce control efforts that are larger in magnitude, compared to a T1 FLC. Consequently, both the IT2 FLCs can provide faster transient response.

4.4.2 Comparison of the analytical structure of the non-symmetric IT2 FLC and the symmetric IT2 FLC

Comparing with 35 subregions in Fig. 4.11 for the symmetric IT2 FLC, the input space of the IT2 fuzzy PD controller needs to be divided into 49 subregions in Fig. 4.10 before they can be expressed into nonlinear PD controllers. The 14 additional subregions with unique properties indicates that the IT2 fuzzy PD controller has the potential to outperform the symmetric IT2 FLC. To highlight the unique characteristics of the IT2 fuzzy PD controller, the rest of this subsection will establish two interesting properties.

Depending on the distance from the origin $(E(n), R(n)) = (0, 0)$, the internal subregions for the symmetric IT2 FLC including IC 1-4, IC 26-29 in Fig. 4.11 can be divided into two groups:

- The regions denoted by Group $\tilde{\alpha}$ consisting of IC 1-4
- The regions denoted by Group $\tilde{\beta}$ including IC 26-29

Results from Chapter 3 that the proportional and derivative gains of the subregion IC 26 (IC 27,28,29) in Group $\tilde{\beta}$ located around the origin $(E(n), R(n)) = (0, 0)$ are larger in magnitude than its adjacent subregion IC 2 (IC 1, 4, 3) in Group $\tilde{\alpha}$, respectively. Unlike the symmetric IT2 FLC, the internal subregions of the IT2 fuzzy PD controller comprising IC 1-4,6-13 in Fig. 4.10 may be divided into the following three categories depending on the distance from the original:

- The regions labeled by Group $\tilde{A}$ consisting of IC 1, 7,8,13

- The regions labeled by Group $\tilde{B}$ including IC 2,6,9,12

- The regions labeled by Group $\tilde{C}$ including IC 3,4,10,11

The relative relationship of the proportional and derivative gains of these three groups may be formally stated as:

Property 5 For the IT2 fuzzy PD controller, the proportional and derivative gains of the subregion IC 1(IC 7,8,13) are smaller in magnitude than the adjacent subregion IC 2 (IC 6,9,12), respectively while the proportional and derivative gains of the subregion IC 3 (IC 4,10,11) are smaller in magnitude than the adjacent subregion in IC 2 (IC 6,9,12) and IC 1 (IC 7,8,13), respectively.

Proof: Since the partitions of the IT2 fuzzy PD controller in Fig. 4.10 are symmetrical, the relationship of the gains between IC 1, IC 2 and IC 3 can be taken as an example to show this property. From Table 4.2, it can be found

$$\begin{aligned} K_p^1 &= \frac{L_2(H_1-H_3)}{8L_1L_2(1-\theta_2)-4L_1R(n)} + \frac{L_2(H_1-H_3)}{8L_1L_2(1+\theta_2)-4L_1R(n)} \\ &< \frac{L_2(H_1-H_3)}{8L_1L_2(1-\theta_2)-4L_1R(n)} + \frac{L_2(H_1-H_3))}{8L_1L_2-4L_1R(n)} = K_p^2 \end{aligned} \tag{4.36}$$

$$\begin{aligned} K_p^1 &= \frac{L_2(H_1-H_3)}{8L_1L_2(1-\theta_2)-4L_1R(n)} + \frac{L_2(H_1-H_3)}{8L_1L_2(1+\theta_2)-4L_1R(n)} \\ &> \frac{L_2(H_1-H_3)}{8L_1L_2-4L_1R(n)} + \frac{L_2(H_1-H_3)}{8L_1L_2-4L_1R(n)} = K_p^3 \end{aligned} \tag{4.37}$$

Another observation is

$$\begin{aligned} K_d^1 &= \frac{-(H_2+H_4)}{8L_2(1-\theta_2)-4R(n)} + \frac{-(H_2+H_4)}{8L_2(1+\theta_2)-4R(n)} \\ &< \frac{-(H_2+H_4)}{8L_2(1-\theta_2)-4R(n)} + \frac{-(H_2+H_4)}{8L_2-4R(n)} = K_d^2 \end{aligned} \tag{4.38}$$

$$\begin{aligned} K_d^1 &= \frac{-(H_2+H_4)}{8L_2(1-\theta_2)-4R(n)} + \frac{-(H_2+H_4)}{8L_2(1+\theta_2)-4R(n)} \\ &> \frac{-(H_2+H_4)}{8L_2-4R(n)} + \frac{-(H_2+H_4)}{8L_2-4R(n)} = K_d^3 \end{aligned} \tag{4.39}$$

Comments: For the symmetric IT2 FLC, the subregions around the zero feedback point, i.e. Group $\tilde{\beta}$, produce the larger proportional and derivative gains than their adjacent subregions. Unlike the symmetric IT2 FLC, the subregions of the non-symmetric IT2 FLC around the zero feedback point, i.e. Group $\tilde{C}$, generate the smallest gains among the internal subregions. Smaller proportional and derivative gains around the zero feedback point may produce a smaller overshoot. Hence, the non-symmetric IT2 FLC is able to achieve a smaller overshoot compared to the symmetric IT2 FLC.

It has been demonstrated that the proportional and derivative gains of the internal subregions for the symmetric IT2 FLC will increase in magnitude as either of the parameters θ_1 and θ_2 or both of them increase. Hence, increasing the parameters θ_1 and θ_2 can enlarge the control efforts in magnitude. But smaller proportional and derivative gains around the zero feedback point requires θ_1 and θ_2 to be small. Such a conflict may be alleviated by the non-symmetric IT2 FLC due to the characteristic presented as follows:

Property 6 For the IT2 fuzzy PD controller, the proportional and derivative gains of the subregions IC 3 and IC 10 are functions of $|\theta_1 - \theta_2|$, while the proportional and derivative gains of the subregions IC 4 and IC 11 are independent of θ_1 and θ_2. As a special case, when $\theta_1 = \theta_2$, the proportional and derivative gains of IC 3, 4, 10, 11 are independent of θ_1 and θ_2.

Comments: Based on the property stated above, varying the values of the parameters θ_1 and θ_2 simultaneously does not alter the surface around zero feedback point for the non-symmetric IT2 FLC. It means that increasing θ_1 and θ_2 can enlarge the control efforts of internal subregions but also maintain the smooth surface around the zero feedback point. Therefore, the non-symmetric IT2 FLC can better alleviate the amount of the compromise between fast response and small overshoot by varying the value of θ_1 and θ_2 simultaneously.

4.4.3 Discussion

Based on the results presented in Property 1-8, the characteristics of the non-symmetric IT2 FLC that may enable them to outperform the symmetric IT2 FLC by providing fast rise time and small overshoot may be summarized as follows:

1. Property 1-4 indicate that both the symmetric IT2 FLC and the non-symmetric IT2 FLC can outperform a T1 FLC by providing larger control efforts and smoother surface.

2. As stated in Property 5, another group of subregion whose proportional and derivative gains are smaller than the other adjacent internal subregions is introduced around the origin $(E(n), R(n)) = (0, 0)$ at the cost of one consequent set. This characteristic may enable the non-symmetric IT2 FLC to achieve smaller overshoot despite larger control efforts provided by the IT2 FLC for the external regions.

3. Unlike the symmetric IT2 FLC, the gains of the non-symmetric IT2 FLC around the origin $(E(n), R(n)) = (0, 0)$ depend on the parameters $|\theta_1 - \theta_2|$ only (Property 6). Hence, varying the value θ_1 and θ_2 simultaneously for the non-symmetric IT2 FLC can maintain small proportional and derivative gains around the origin $(E(n), R(n)) = (0, 0)$ and enlarge the control effort for the external regions.

4.5 Conclusion

This chapter presents the analytical structure of a class of IT2 fuzzy PD and PI controllers that uses the KM iterative algorithm for type-reduction. A methodology for identifying the input space boundaries where the KM algorithm uses a new switch point to compute the bounds of the type-reduced set is first established for the considered IT2 FLC. This result resolves one of the main issues hindering theoretical analysis of IT2 FLCs that employs the KM type reducer. Based on the finding, the analytical structure of an IT2 FLC is derived by treating each input region as a T1 FLS problem using the well-established techniques for T1 FLSs. A contribution is that the four properties identified for the symmetric IT2 FLC in Chapter 3 were established for the non-symmetric IT2 fuzzy FLC considered in this chapter. Furthermore, a comparative study of the analytical structure of the two IT2 FLCs identified the unique properties of the IT2 fuzzy PD controller. These properties provide further theoretical basis for explaining the ability of an IT2 FLC to alleviate the trade-off between fast rise time and small overshoot.

Table 4.1: The Firing strengths of four rules in $\triangle U_j^{min}$ and $\triangle U_j^{max}$

IC No.	$\triangle U_j^{min}$				$\triangle U_j^{max}$			
	Rule 4	Rule 3	Rule 2	Rule 1	Rule 4	Rule 3	Rule 2	Rule 1
1	$\overline{RN}$	$\overline{EN}$	$\overline{RN}$	$\underline{EP}$	$\underline{RN}$	$\underline{EN}$	$\underline{RN}$	$\overline{EP}$
2	$\overline{RN}$	$\overline{EN}$	$\underline{RN}$	$\underline{EP}$	$\underline{RN}$	$\underline{EN}$	$\underline{RN}$	$\overline{EP}$
3	$\overline{RN}$	$\overline{EN}$	$\underline{RN}$	$\underline{EP}$	$\underline{RN}$	$\underline{EN}$	$\overline{RN}$	$\overline{EP}$
4	$\overline{RN}$	$\overline{RP}$	$\underline{EP}$	$\underline{EP}$	$\underline{RN}$	$\underline{RP}$	$\overline{EP}$	$\overline{EP}$
5	$\overline{RN}$	$\underline{EN}$	$\underline{RN}$	$\underline{EP}$	$\underline{RN}$	$\underline{EN}$	$\overline{RN}$	$\overline{EP}$
6	$\overline{RN}$	$\underline{RP}$	$\underline{EP}$	$\underline{EP}$	$\underline{RN}$	$\underline{RP}$	$\overline{EP}$	$\overline{EP}$
7	$\overline{RN}$	$\underline{RP}$	$\underline{EP}$	$\underline{EP}$	$\underline{RN}$	$\overline{RP}$	$\overline{EP}$	$\overline{EP}$
8	$\overline{EN}$	$\underline{RP}$	$\underline{EP}$	$\underline{RP}$	$\underline{EN}$	$\overline{RP}$	$\overline{EP}$	$\overline{RP}$
9	$\overline{EN}$	$\underline{RP}$	$\underline{EP}$	$\underline{RP}$	$\underline{EN}$	$\underline{RP}$	$\overline{EP}$	$\overline{RP}$
10	$\overline{EN}$	$\overline{RP}$	$\underline{EP}$	$\underline{RP}$	$\underline{EN}$	$\underline{RP}$	$\overline{EP}$	$\overline{RP}$
11	$\overline{EN}$	$\overline{EN}$	$\underline{RN}$	$\underline{RP}$	$\underline{EN}$	$\underline{EN}$	$\overline{RN}$	$\overline{RP}$
12	$\overline{EN}$	$\overline{EN}$	$\underline{RN}$	$\underline{RP}$	$\underline{EN}$	$\underline{EN}$	$\underline{RN}$	$\overline{RP}$
13	$\overline{EN}$	$\overline{EN}$	$\overline{RN}$	$\underline{RP}$	$\underline{EN}$	$\underline{EN}$	$\underline{RN}$	$\overline{RP}$
14	$\overline{EN}$	$\overline{RP}$	$\underline{EP}$	$\underline{RP}$	$\underline{EN}$	$\underline{RP}$	$\underline{EP}$	$\overline{RP}$
15	$\overline{EN}$	$\overline{EN}$	$\overline{RN}$	$\underline{RP}$	0	0	0	1
16	$\overline{RN}$	$\overline{EN}$	$\overline{RN}$	$\underline{EP}$	0	0	0	1
17	$\overline{RN}$	$\overline{EN}$	$\overline{RN}$	$\underline{EP}$	0	$\underline{EN}$	0	$\overline{EP}$
18	$\overline{RN}$	$\overline{EN}$	0	$\underline{EP}$	0	$\underline{EN}$	0	$\overline{EP}$
19	$\overline{RN}$	$\overline{EN}$	0	$\underline{EP}$	0	$\underline{EN}$	$\overline{RN}$	$\overline{EP}$
20	$\overline{RN}$	$\underline{EN}$	0	$\underline{EP}$	0	$\underline{EN}$	$\overline{RN}$	$\overline{EP}$
21	$\overline{RN}$	$\underline{EN}$	0	0	0	$\underline{EN}$	$\overline{RN}$	$\overline{EP}$
22	$\overline{RN}$	$\underline{RP}$	0	0	0	$\underline{RP}$	$\overline{EP}$	$\overline{EP}$
23	$\overline{RN}$	$\underline{RP}$	0	0	$\underline{RN}$	$\underline{RP}$	$\overline{EP}$	$\overline{EP}$
24	$\overline{RN}$	$\underline{RP}$	0	0	$\underline{RN}$	$\overline{RP}$	$\overline{EP}$	$\overline{EP}$
25	1	0	0	0	$\underline{RN}$	$\overline{RP}$	$\overline{EP}$	$\overline{EP}$
26	1	0	0	0	$\underline{EN}$	$\overline{RP}$	$\overline{EP}$	$\overline{RP}$
27	$\overline{EN}$	0	$\underline{EP}$	0	$\underline{EN}$	$\overline{RP}$	$\overline{EP}$	$\overline{RP}$
28	$\overline{EN}$	0	$\underline{EP}$	0	$\underline{EN}$	0	$\overline{EP}$	$\overline{RP}$
29	$\overline{EN}$	$\overline{RP}$	$\underline{EP}$	0	$\underline{EN}$	0	$\overline{EP}$	$\overline{RP}$
30	$\overline{EN}$	$\overline{RP}$	$\underline{EP}$	0	$\underline{EN}$	0	$\underline{EP}$	$\overline{RP}$
31	$\overline{EN}$	$\overline{RP}$	$\underline{EP}$	0	0	0	$\underline{EP}$	$\overline{RP}$
32	$\overline{EN}$	$\overline{EN}$	$\underline{RN}$	0	0	0	$\underline{RN}$	$\overline{RP}$
33	$\overline{EN}$	$\overline{EN}$	$\underline{RN}$	$\underline{RP}$	0	0	$\underline{RN}$	$\overline{RP}$
34	$\overline{EN}$	$\overline{EN}$	$\overline{RN}$	$\underline{RP}$	0	0	$\underline{RN}$	$\overline{RP}$

Table 4.2: The gains for the internal subregions

IC No.	K_p	K_d	δ
1	$\frac{L_2(H_1-H_3)E(n)}{8L_1L_2(1-\theta_2)-4L_1R(n)} + \frac{L_2(H_1-H_3)E(n)}{8L_1L_2(1+\theta_2)-4L_1R(n)}$	$\frac{-(H_2+H_4)R(n)}{8L_2(1-\theta_2)-4R(n)} + \frac{-(H_2+H_4)R(n)}{8L_2(1+\theta_2)-4R(n)}$	$\frac{L_2[\theta_1(H_1-H_3)-\theta_2(H_2+H_4)]}{4L_2(1-\theta_2)-2R(n)} + \frac{L_2[\theta_1(H_3-H_1)+\theta_2(H_2+H_4)]}{4L_2(1+\theta_2)-2R(n)}$
2	$\frac{L_2(H_1-H_3)E(n)}{8L_1L_2(1-\theta_2)-4L_1R(n)} + \frac{L_2(H_1-H_3)E(n)}{8L_1L_2-4L_1R(n))}$	$\frac{-(H_2+H_4)R(n)}{8L_2(1-\theta_2)-4R(n)} + \frac{-(H_2+H_4)R(n)}{8L_2-4R(n))}$	$\frac{L_2[\theta_1(H_1-H_3)-\theta_2(H_2+H_4)]}{4L_2(1-\theta_2)-2R(n)} + \frac{L_2[\theta_2(H_4-H_2)+\theta_1(H_3-H_1)]}{4L_2-2R(n))}$
3	$\frac{L_2(H_1-H_3)E(n)}{8L_1L_2-4L_1R(n)} + \frac{L_2(H_1-H_3)E(n)}{8L_1L_2-4L_1R(n))}$	$\frac{-(H_2+H_4)R(n)}{8L_2-4R(n)} + \frac{-(H_2+H_4)R(n)}{8L_2-4R(n))}$	$\frac{L_2[\theta_1(H_1-H_3)+\theta_2(H_2-H_4)]}{4L_2-2R(n)} + \frac{L_2[\theta_2(H_4-H_2)+\theta_1(H_3-H_1)]}{4L_2-2R(n))}$
4	$\frac{(H_2+H_1)E(n)}{8L_1(1-\theta_2+\theta_1)+4E(n)} + \frac{(H_2+H_1)E(n)}{8L_1(1+\theta_2-\theta_1)+4E(n)}$	$\frac{L_1(H_3-H_4)R(n)}{8L_1L_2(1-\theta_2+\theta_1)+4L_2E(n)} + \frac{L_1(H_3-H_4)R(n)}{8L_1L_2(1+\theta_2-\theta_1)+4L_2E(n)}$	$\frac{L_1[\theta_1(H_1+H_2)-\theta_2(H_3+H_4)]}{4L_1(1-\theta_2+\theta_1)+2E(n)} + \frac{L_1[\theta_2(H_3+H_4)-\theta_1(H_1+H_2)]}{4L_1(1+\theta_2-\theta_1)+2E(n)}$
5	$\frac{L_2(H_1-H_3)E(n)}{8L_1L_2-4L_1R(n)} + \frac{L_2(H_1-H_3)E(n)}{8L_1L_2(1-\theta_1)-4L_1R(n)}$	$\frac{-(H_2+H_4)R(n)}{8L_2-4R(n)} + \frac{-(H_2+H_4)R(n)}{8L_2(1-\theta_1)-4R(n)}$	$\frac{L_2[\theta_1(H_1-H_3)+\theta_2(H_2-H_4)]}{4L_2-2R(n)} \frac{+L_2[\theta_2(H_4-H_2)-\theta_1(H_1+H_3)]}{4L_2(1-\theta_1)-2R(n)}$
6	$\frac{(H_2+H_1)E(n)}{8L_1(1-\theta_2+\theta_1)+4E(n)} + \frac{(H_1+H_2)E(n)}{8L_1(1-\theta_1)+4E(n)}$	$\frac{L_1(H_3-H_4)R(n)}{8L_1L_2(1-\theta_2+\theta_1)+4L_2E(n)} + \frac{L_1(H_3-H_4)R(n)}{8L_1L_2(1-\theta_1)+4L_2E(n)}$	$\frac{L_1[\theta_1(H_1+H_2)-\theta_2(H_3+H_4)]}{4L_1(1-\theta_2+\theta_1)+2E(n)} + \frac{L_1[\theta_2(H_4-H_3)-\theta_1)(H_1+H_2)]}{4L_1(1-\theta_1)+2E(n)}$
7	$\frac{(H_1+H_2)E(n)}{8L_1(1+\theta_1)+4E(n)} + \frac{(H_1+H_2)E(n)}{8L_1(1-\theta_1)+4E(n)}$	$\frac{-L_1(H_4-H_3)R(n)}{8L_1L_2(1+\theta_1)+4L_2E(n)} + \frac{L_1(H_3-H_4)R(n)}{8L_1L_2(1-\theta_1)+4L_2E(n)}$	$\frac{L_1[\theta_1(H_1+H_2)+\theta_2(H_3-H_4)]}{4L_1(1+\theta_1)+2E(n)} + \frac{L_1[\theta_2(H_4-H_3)-\theta_1)(H_1+H_2)]}{4L_1(1-\theta_1)+2E(n)}$
8	$\frac{L_2(H_2-H_4)E(n)}{8L_1L_2(1+\theta_2)+4L_1R(n)} + \frac{L_2(H_2-H_4)E(n)}{8L_1L_2(1-\theta_2)+4L_1R(n)}$	$\frac{(H_3+H_1)R(n)}{8L_2(1+\theta_2)+4R(n)} + \frac{(H_3+H_1)R(n)}{8L_2(1-\theta_2)+4R(n)}$	$\frac{L_2[\theta_2(H_3+H_1)+\theta_1(H_2-H_4)]}{4L_2(1+\theta_2)+2R(n)} + \frac{L_2[\theta_1(H_4-H_2)-\theta_2(H_1+H_3)]}{4L_2(1-\theta_2)+2R(n)}$
9	$\frac{L_2(H_2-H_4)E(n)}{8L_1L_2+4L_1R(n)} + \frac{L_2(H_2-H_4)E(n)}{8L_1L_2(1-\theta_2)+4L_1R(n)}$	$\frac{(H_3+H_1)R(n)}{8L_2+4R(n)} + \frac{(H_3+H_1)R(n)}{8L_2(1-\theta_2)+4R(n)}$	$\frac{L_2[\theta_1(H_2-H_4)+\theta_2(H_1-H_3)]}{4L_2+2R(n)} + \frac{L_2[\theta_1(H_4-H_2)-\theta_2(H_1+H_3)]}{4L_2(1-\theta_2)+2R(n)}$
10	$\frac{L_2(H_2-H_4)E(n)}{8L_1L_2+4L_1R(n)} + \frac{L_2(H_2-H_4)E(n)}{8L_1L_2+4L_1R(n)}$	$\frac{(H_3+H_1)R(n)}{8L_2+4R(n)} + \frac{(H_3+H_1)R(n)}{8L_2+4R(n)}$	$\frac{L_2[\theta_1(H_2-H_4)+\theta_2(H_1-H_3)]}{4L_2+2R(n)} + \frac{L_2[\theta_1(H_4-H_2)+(0.5+\theta_2(H_3-H_1)]}{4L_2+2R(n)}$
11	$\frac{-(H_3+H_4)E(n)}{8L_1(1-\theta_1+\theta_2)-4E(n)} + \frac{-(H_4+H_3)E(n)}{8L_1(1+\theta_1-\theta_2)-2E(n)}$	$\frac{L_1(H_1-H_2)R(n)}{8L_1L_2(1-\theta_1+\theta_2)-4L_2E(n)} + \frac{L_1(H_1-H_2)R(n)}{8L_1L_2(1+\theta_1-\theta_2)-2L_2E(n)}$	$\frac{L_1[-\theta_1(H_4+H_3)+\theta_2(H_2+H_1)]}{4L_1(1-\theta_1+\theta_2)-2E(n)} + \frac{L_1[\theta_1(H_4+H_3)-\theta_2(H_2+H_1)]}{4L_1(1+\theta_1-\theta_2)-2E(n)}$
12	$\frac{-(H_4+H_3)E(n)}{8L_1(1-\theta_1)-4E(n)} + \frac{-(H_4+H_3)E(n)}{8L_1(1+\theta_1-\theta_2)-2E(n)}$	$\frac{L_1(H_1-H_2)R(n)}{8L_1L_2(1-\theta_1)-4L_2E(n)} + \frac{L_1(H_1-H_2)R(n)}{8L_1L_2(1+\theta_1-\theta_2)-2L_2E(n)}$	$\frac{L_1[-\theta_1(H_4+H_3)+\theta_2(H_1-H_2)]}{4L_1(1-\theta_1)-2E(n)} + \frac{L_1[\theta_1(H_4+H_3)-\theta_2(H_2+H_1)]}{4L_1(1+\theta_1-\theta_2)-2E(n)}$
13	$\frac{-(H_4+H_3)E(n)}{8L_1(1-\theta_1)-4E(n)} + \frac{-(H_3+H_4)E(n)}{8L_1(1+\theta_1)-4E(n)}$	$\frac{L_1(H_1-H_2)R(n)}{8L_1L_2(1-\theta_1)-4L_2E(n)} + \frac{L_1(H_1-H_2)R(n)}{8L_1L_2(1+\theta_1)-4L_2E(n)}$	$\frac{L_1[-\theta_1(H_4+H_3)+\theta_2(H_1-H_2)]}{4L_1(1-\theta_1)-2E(n)} + \frac{L_1[\theta_1(H_3+H_4)+\theta_2(H_2-H_1)]}{4L_1(1+\theta_1)-2E(n)}$
14	$\frac{L_2(H_2-H_4)E(n)}{8L_1L_2(1-\theta_1)+4L_1R(n)} + \frac{L_2(H_2-H_4)E(n)}{8L_1L_2+4L_1R(n)}$	$\frac{(H_1+H_3)R(n)}{8L_2(1-\theta_1)+4R(n)} + \frac{(H_3+H_1)R(n)}{8L_2+4R(n)}$	$\frac{L_2[-\theta_1(H_2+H_4)+\theta_2)(H_1-H_3)]}{4L_2(1-\theta_1)+2R(n)} + \frac{L_2[\theta_1(H_4-H_2)+(0.5+\theta_2(H_3-H_1)]}{4L_2+2R(n)}$

Table 4.3: The gains for the external subregions

IC No.	K_p	K_d	δ
15	$\frac{-(H_3+H_4)}{8L_1(1+\theta_1)-4E(n)}$	$\frac{L_1(H_1-H_2)}{8L_1L_2(1+\theta_1)-4L_2E(n)}$	$0.5H_1 + \frac{L_1[\theta_1(H_3+H_4)+\theta_2(H_2-H_1)]}{4L_1(1+\theta_1)-2E(n)}$
16	$\frac{L_2(H_1-H_3)}{8L_1L_2(1+\theta_2)-4L_1R(n)}$	$\frac{-(H_2+H_4)}{8L_2(1+\theta_2)-4R(n)}$	$0.5H_1 + \frac{L_2[\theta_1(H_3-H_1)+\theta_2(H_2+H_4)]}{4L_2(1+\theta_2)-2R(n)}$
17	$\frac{1}{4L_1}(H_1 - H_3)$ $+\frac{L_2(H_1-H_3)}{8L_1L_2(1+\theta_2)-4L_1R(n)}$	$\frac{-(H_2+H_4)}{8L_2(1+\theta_2)-4R(n)}$	$0.5[(0.5-\theta_1)H_3 + (0.5+\theta_1)H_1]$ $+\frac{L_2[\theta_1(H_3-H_1)+\theta_2(H_2+H_4)]}{4L_2(1+\theta_2)-2R(n)}$
18	$\frac{1}{4L_1}(H_1 - H_3)$ $+\frac{L_2(H_1-H_3)}{4L_1L2(1.5+\theta_2)-2L_1R(n)}$	$\frac{-H_4}{4L2(1.5+\theta_2)-2R(n)}$	$0.5[(0.5-\theta_1)H_3 + (0.5+\theta_1)H_1]$ $+\frac{L_2[(0.5+\theta_2)H_4+(0.5+\theta_1)H_3+(0.5-\theta_1)H_1]}{2L2(1.5+\theta_2)-R(n)}$
19	$\frac{L_2(H_1-H_3)}{2L_1L_2(1.5+\theta_2)-L_1R(n)}$	$\frac{-H_2}{4L_2(1.5+\theta_2)-2R(n)}$ $+\frac{-H_4}{4L2(1.5+\theta_2)-2R(n)}$	$\frac{L_2[(0.5+\theta_1)H_1+(0.5+\theta_2)H_2+(0.5-\theta_1)H_3]}{2L_2(1.5+\theta_2)-R(n)}$ $+\frac{L_2[(0.5+\theta_2)H_4+(0.5+\theta_1)H_3+(0.5-\theta_1)H_1]}{2L2(1.5+\theta_2)-R(n)}$
20	$\frac{L_2(H_1-H_3)}{4L_1L_2(1.5+\theta_2)-2L_1R(n)}$ $+\frac{L_2(H_1-H_3)}{4L_1L_2(1.5+\theta_2-2\theta_1)-2L_1R(n)}$	$\frac{-H_2}{4L_2(1.5+\theta_2)-2R(n)}$ $+\frac{-H_4}{4L_2(1.5+\theta_2-2\theta_1)-2R(n)}$	$\frac{L_2[(0.5+\theta_1)H_1+(0.5+\theta_2)H_2+(0.5-\theta_1)H_3]}{2L_2(1.5+\theta_2)-R(n)}$ $+\frac{L_2[(0.5+\theta_2)H_4+(0.5-\theta_1)(H_1+H_3)]}{2L_1L_2(1.5+\theta_2-2\theta_1)-L_1R(n)}$
21	$\frac{L_2(H_1-H_3)}{4L_1L_2(1.5+\theta_2)-2L_1R(n)}$ $+\frac{-L_2H_3}{4L_1L_2(1-\theta_1+\theta_2)-2L_1R(n)-2L_2E(n)}$	$\frac{-H_2}{4L_2(1.5+\theta_2)-2R(n)}$ $+\frac{-L_1H_4}{4L_1L_2(1-\theta_1+\theta_2)-2L_1R(n)-2L_2E(n)}$	$\frac{L_2[(0.5+\theta_1)H_1+(0.5+\theta_2)H_2+(0.5-\theta_1)H_3]}{2L_2(1.5+\theta_2)-R(n)}$ $+\frac{L_1L_2[(0.5+\theta_2)H_4+(0.5-\theta_1)H_3]}{2L_1L_2(1-\theta_1+\theta_2)-L_1R(n)-L_2E(n)}$
22	$\frac{L_2(H_1+H_2)}{4L_1L_2(1.5-\theta_2+2\theta_1)+2L_1R(n)+4L_2E(n)}$	$\frac{L_1H_3}{4L_1L_2(1.5-\theta_2+2\theta_1)+2L_1R(n)+4L_2E(n)}$ $-\frac{1}{4L_2}(H_4 - H_3)$	$\frac{L_1L_2[(0.5-\theta_2)H_3+(0.5+\theta_1)(H_1+H_2)]}{2L_1L_2(1.5-\theta_2+2\theta_1)+L_1R(n)+2L_2E(n)}$ $+0.5[(0.5+\theta_2)H_4 + (0.5-\theta_2)H_3]$
23	$\frac{(H_2+H_1)}{8L_1(1-\theta_2+\theta_1)+4E(n)}$	$\frac{L_1(H_3-H_4)}{8L_1L_2(1-\theta_2+\theta_1)+4L_2E(n)}$ $-\frac{1}{4L_2}(H_4 - H_3)$	$\frac{L_1[\theta_1(H_1+H_2)-\theta_2(H_3+H_4)]}{4L_1(1-\theta_2+\theta_1)+2E(n)}$ $+0.5[(0.5+\theta_2)H_4 + (0.5-\theta_2)H_3]$
24	$\frac{(H_1+H_2)}{8L_1(1+\theta_1)+4E(n)}$	$\frac{-L_1(H_4-H_3)}{8L_1L_2(1+\theta_1)+4L_2E(n)}$ $-\frac{1}{4L_2}(H_4 - H_3)$	$\frac{L_1[\theta_1(H_1+H_2)+\theta_2(H_3-H_4)]}{4L_1(1+\theta_1)+2E(n)}$ $+0.5[(0.5+\theta_2)H_4 + (0.5-\theta_2)H_3]$
25	$\frac{(H_1+H_2)}{8L_1(1+\theta_1)+4E(n)}$	$\frac{-L_1(H_4-H_3)}{8L_1L_2(1+\theta_1)+4L_2E(n)}$	$\frac{L_1[\theta_1(H_1+H_2)+\theta_2(H_3-H_4)]}{4L_1(1+\theta_1)+2E(n)} + 0.5H_4$
26	$\frac{L_2(H_2-H_4)}{8L_1L_2(1+\theta_2)+4L_1R(n)}$	$\frac{(H_3+H_1)}{8L_2(1+\theta_2)+4R(n)}$	$\frac{L_2[\theta_2(H_3+H_1)+\theta_1(H_2-H_4)]}{4L_2(1+\theta_2)+2R(n)} + 0.5H_4$
27	$\frac{L_2(H_2-H_4)}{8L_1L_2(1+\theta_2)+4L_1R(n)}$ $+\frac{1}{4L_1}(H_2 - H_4)$	$\frac{(H_3+H_1)}{8L_2(1+\theta_2)+4R(n)}$	$\frac{L_2[\theta_2(H_3+H_1)+\theta_1(H_2-H_4)]}{4L_2(1+\theta_2)+2R(n)}$ $+0.5[(0.5+\theta_1)H_4 + (0.5-\theta_1)H_2]$
28	$\frac{L_2(H_2-H_4)}{4L_1L_2(1.5+\theta_2)+2L_1R(n)}$ $+\frac{1}{4L_1}(H_2 - H_4)$	$\frac{H_1}{4L_2(1.5+\theta_2)+2R(n)}$	$\frac{L_2[(0.5-\theta_1)H_4+(0.5+\theta_1)H_2+(0.5+\theta_2)H_1]}{2L_2(1.5+\theta_2)+R(n)}$ $+0.5[(0.5+\theta_1)H_4 + (0.5-\theta_1)H_2]$
29	$\frac{L_2(H_2-H_4)}{2L_1L_2(1.5+\theta_2)+L_1R(n)}$	$\frac{H_1}{4L_2(1.5+\theta_2)+2R(n)}$ $+\frac{H_3}{4L_2(1.5+\theta_2)+2R(n)}$	$\frac{L_2[(0.5-\theta_1)H_4+(0.5+\theta_1)H_2+(0.5+\theta_2)H_1]}{2L_2(1.5+\theta_2)+R(n)}$ $+\frac{L_2[(0.5+\theta_1)H_4+(0.5+\theta_2)H_3+(0.5-\theta_1)H_2]}{2L_2(1.5+\theta_2)+R(n)}$
30	$\frac{L_2(H_2-H_4)}{4L_1L_2(1.5-2\theta_1+\theta_2)+2L_1R(n)}$ $+\frac{L_2(H_2-H_4)}{4L_1L_2(1.5+\theta_2)+2L_1R(n)}$	$\frac{H_1}{4L_2(1.5-2\theta_1+\theta_2)+2R(n)}$ $+\frac{H_3}{4L_2(1.5+\theta_2)+2R(n)}$	$\frac{L_2[(0.5-\theta_1)(H_2+H_4)+(0.5+\theta_2)H_1]}{2L_2(1.5-2\theta_1+\theta_2)+R(n)}$ $+\frac{L_2[(0.5+\theta_1)H_4+(0.5+\theta_2)H_3+(0.5-\theta_1)H_2]}{2L_2(1.5+\theta_2)+R(n)}$
31	$\frac{L_2H_2}{4L_1L_2(1-\theta_1+\theta_2)+2L_1R(n)+2L_2E(n)}$ $+\frac{L_2(H_2-H_4)}{4L_1L_2(1.5+\theta_2)+2L_1R(n)}$	$\frac{L_1H_1}{4L_1L_2(1-\theta_1+\theta_2)+2L_1R(n)+2L_2E(n)}$ $+\frac{H_3}{4L_2(1.5+\theta_2)+2R(n)}$	$\frac{L_1L_2[(0.5-\theta_1)H_2+(0.5+\theta_2)H_1]}{2L_1L_2(1-\theta_1+\theta_2)+L_1R(n)+L_2E(n)}$ $+\frac{L_2[(0.5+\theta_1)H_4+(0.5+\theta_2)H_3+(0.5-\theta_1)H_2]}{2L_2(1.5+\theta_2)+R(n)}$
32	$\frac{L_2(H_3+H_4)}{4L_2E(n)+2L_1R(n)-4L_1L_2(1.5+2\theta_1-\theta_2)}$	$\frac{1}{4L_2}(H_1 - H_2)+$ $\frac{L_1H_2}{4L_2E(n)+2L_1R(n)-4L_1L_2(1.5+2\theta_1-\theta_2)}$	$0.5[(0.5-\theta_2)H_2 + (0.5+\theta_2)H_1]$ $+\frac{-L_1L_2[(0.5+\theta_1)(H_3+H_4)+(0.5-\theta_2)H_2]}{2L_2E(n)+L_1R(n)-2L_1L_2(1.5+2\theta_1-\theta_2)}$
33	$\frac{-(H_4+H_3)}{8L_1(1+\theta_1-\theta_2)-2E(n)}$	$\frac{1}{4L_2}(H_1 - H_2)$ $+\frac{L_1(H_1-H_2)}{8L_1L_2(1+\theta_1-\theta_2)-2L_2E(n)}$	$0.5[(0.5-\theta_2)H_2 + (0.5+\theta_2)H_1]$ $+\frac{L_1[\theta_1(H_4+H_3)-\theta_2(H_2+H_1)]}{4L_1(1+\theta_1-\theta_2)-2E(n)}$
34	$\frac{-(H_3+H_4)}{8L_1(1+\theta_1)-4E(n)}$	$\frac{1}{4L_2}(H_1 - H_2)$ $+\frac{L_1(H_1-H_2)}{8L_1L_2(1+\theta_1)-4L_2E(n)}$	$0.5[(0.5-\theta_2)H_2 + (0.5+\theta_2)H_1]$ $+\frac{L_1[\theta_1(H_3+H_4)+\theta_2(H_2-H_1)]}{4L_1(1+\theta_1)-2E(n)}$

Chapter 5

Improved algorithms for Fuzzy Weighted Average and Linguistic Weighted Average

Chapter 3 and Chapter 4 addressed the interval type-2 fuzzy logic controller using the Karnik-Mendel(KM) type-reducer in which the KM iterative algorithm is applied to implement the KM type-reducer. Another important reasoning process in the theory of fuzzy logic is fuzzy weighted average (FWA) and linguistic weighted average (LWA) which are the widely adopted aggregation operators. The common characteristic between these two reasoning process, interval type-2 fuzzy logic system and the aggregation operators (FWA and LWA), is the application of the KM iterative algorithm or the EKM iterative algorithm in their implementation. However, even with the introduction of the KM/EKM iterative algorithm to assist with the necessary α-cut arithmetic, the FWA and LWA remain computationally intensive due to the iterative nature of the KM algorithm. Just because of the prohibitively high computational requirement, FWA and LWA may not be suitable for practical applications. To reduce the computational overhead required by the FWA and LWA, this chapter presents three proposed algorithms by optimizing the initialization and termination of the EKM iterative algorithm.

5.1 Introduction

Averaging is a technique that is commonly employed for combining, amalgamating or fusing information. As practical data tend to be uncertain, the fuzzy weighted average (FWA) and the linguistic weighted average (LWA)[44, 84] have been proposed to blend uncertain information modeled by fuzzy sets in many engineering applications such as multi-criteria decision making, sensor fusion, risk analysis, filtering etc [11, 26, 27, 52, 86]. The process of converting the general type-2 fuzzy set (T2 FS) generated by the inference engine of a general type-2 fuzzy system (T2 FLS) to a type-1 fuzzy set (T1 FS) is another application where FWA is used to combine the centroid of all consequent sets with their respective firing sets.

Formally, the FWA may be defined as follows:

$$Y_{FWA} = \frac{\sum_{i=1}^{n} W_i X_i}{\sum_{i=1}^{n} W_i} \tag{5.1}$$

where X_i and W_i are T1 FSs. The result Y_{FWA} is also a T1 FS. In a decision making process, $X_i, i = 1, 2, \cdots, n$ represent the possible opinions or attributes and W_i are the weights. In a general T2 FLS, X_i are the centroids of consequent sets and W_i are the corresponding firing sets. As a generalized version of the FWA, the linguistic weighted average (LWA) aggregates interval type-2 (IT2) FSs representing the opinions or attributes and their weights, i.e.

$$\tilde{Y}_{LWA} = \frac{\sum_{i=1}^{n} \tilde{X}_i \tilde{W}_i}{\sum_{i=1}^{n} \tilde{W}_i} \tag{5.2}$$

where $\tilde{X}_i$ and $\tilde{W}_i$ are IT2 FSs representing the opinions or attributes and the corresponding weights. LWA is mainly used as a perceptual computer to perform computing with words (CWW) where the word uncertainties are modeled as IT2 FSs. As the result of the LWA $\tilde{Y}_{LWA}$ is also an IT2 FS and an IT2 FS may be completely characterized by the upper membership function (UMF) and the lower membership function (LMF), computing the LWA essentially reduces to using the FWA to identify the LMF and UMF of the IT2 FS $\tilde{Y}_{LWA}$. In the computation of the LWA, finding the UMF or LMF of $\tilde{Y}_{LWA}$ may be equivalently treated as computing a FWA, respectively.

As the FWA operator is useful in many application areas, the task of computing the FWA has been attracting much attention from the research community. Computing the FWA is essentially achieved by finding out the membership function (MF) of the T1 FS Y_{FWA}. The widely adopted approach is to construct the MF of the T1 FS Y_{FWA} by deriving its α-cuts, where each α-cut of the T1 FS Y_{LWA} is the weighted average among the corresponding α-cuts of T1 FSs X_i and W_i. However, it is challenging to compute the α-cuts because closed-form expressions for the left and right bound of each α-cut of the fuzzy set Y_{FWA} is not available. To overcome this challenge, an exhaustive search method was proposed to find the bounds for each α by calculating all possible weighted averages at each α level [13]. The exhaustive nature of this search method results in great computational overhead. To reduce the great computational burden of finding each α-cut of the

FS Y_{LWA}, several iterative algorithms were proposed by directly searching for the two bounds of each α-cuts iteratively. The first iterative algorithm by Liou and Wang [43] was presented based on the observation that X_i in (5.1) appears in the nominator. This iterative algorithm takes at most $n^2 + n + 2$ iterations before it reaches either bound of any chosen α-cut. Later, Lee and Park proposed an iterative algorithm that takes at most $2n\ln n$ iterations to complete the search of either bound of any α-cut [39]. Despite these efforts to improve the search efficacy, the computation cost required for the FWA operation remains large until the introduction of the Karnik-Mendel (KM) iterative algorithm, which was proposed for solving the following problem [85]:

$$y_l = \min_{\substack{x_i \in [\underline{x}_i, \overline{x}_i] \\ w_i \in [\underline{w}_i, \overline{w}_i]}} \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \tag{5.3}$$

$$y_r = \max_{\substack{x_i \in [\underline{x}_i, \overline{x}_i] \\ w_i \in [\underline{w}_i, \overline{w}_i]}} \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \tag{5.4}$$

When the intervals $[\underline{x}_i, \overline{x}_i]$ and $[\underline{w}_i, \overline{w}_i]$ in (5.3)-(5.4) are interpreted as α-cuts of the fuzzy sets X_i and W_i in the FWA operation, then the result of the KM algorithm $[y_l, y_r]$ represents the corresponding α-cut of Y_{FWA}. It has been demonstrated that the KM iterative algorithm takes at most n iterations for one search objective. In addition, the KM iterative algorithm is exponentially convergent [61], [46], which may further reduce the number of the iterations and thus speed up the search process. The enhanced KM (EKM) iterative algorithm is an enhanced version of the KM iterative algorithm by optimizing the initialization, computation process and termination of the search process.

Although the introduction of the EKM iterative algorithm has reduced the number of iterations required in the search process to determine each of the α-

cuts that make up Y_{FWA}, the computational overhead for computing the FWA and the LWA is still prohibitively high, especially as the required accuracy rises. This is because the required accuracy determines how many α-cuts need to be computed, and for each chosen α, the EKM iterative algorithm must be implemented twice to obtain the FWA and four times for LWA. As the high computational requirement of the EKM method is well documented, there have been efforts at improving the efficacy of the algorithm. One such work is the optimization of the EKM iterative algorithm for computing the centroid of a general T2 FS. Via the α-planes representation [62, 79], the centroid for a general T2 FS can be obtained by taking the union of the centroids of all the α-planes of the FS. Since an α-plane is an IT2 FS (shown as the dotted line in Fig. 5.1), its centroid is the IT1 FS $[c_l, c_r]$. The upper and lower bounds c_l and c_r may be expressed as

$$c_l = \min_{w_i \in [\underline{w}_i, \overline{w}_i]} \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \tag{5.5}$$

$$c_r = \max_{w_i \in [\underline{w}_i, \overline{w}_i]} \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \tag{5.6}$$

where x_i represents the discretized points of the primary variable and $\underline{w}_i$ and $\overline{w}_i$ are the lower and upper bound of the corresponding membership grade of an α-plane. The above equations for computing c_l and c_r is a special case of (5.3) and (5.4) when the variable x is a crisp value instead of an interval set, i.e. Eq. (5.3) and (5.4) reduced to Eq. (5.5) and (5.6) when $\underline{x}_i = \overline{x}_i = x_i$. To achieve lower computational overhead and a higher accuracy simultaneously, the centroid of the jth α-plane may be computed via the EKM iterative algorithm by setting the switch point for the first iteration as the switch point of the $(j-1)$th α-plane derived in previous computation [96]. This enhancement is based on the characteristic that

$x_i, i = 1, 2, \cdots, n$ are discretized points in the domain of the primary variable, and are fixed for a general T2 fuzzy set. However, for the FWA operation, $[\underline{x}_i, \overline{x}_i]$ in (5.3) are a collection of interval sets representing the α-cuts that constitute the T1 FS, X_i. Hence, it is not straightforward to extend the efficient EKM iterative algorithm for computing the centroid of a T2 FS to the FWA and LWA operations. To reduce the computational cost of the FWA and LWA operation, this chapter presents algorithms in which three methods are proposed to optimize the initialization of the EKM iterative algorithm by leveraging on the unique properties of the α-cuts of the FSs Y_{FWA} and $\tilde{Y}_{LWA}$ to reduce the number of iterations. In addition, a new termination condition is proposed to remove an unnecessary iteration in the termination of the EKM iterative algorithm. The rest of this chapter is organized as follows: Section II reviews the procedures for the FWA and LWA operation and the KM and EKM iterative algorithms; Section III covers the characteristics of the α-cuts of Y_{FWA} and $\tilde{Y}_{LWA}$ followed by the proposed algorithm for the FWA and LWA operation; Section IV presents a theoretical study to compare the proposed algorithms and KM/EKM iterative algorithm; numerical studies compare the proposed algorithm and the KM and EKM iterative algorithms in Section V. Section VI is the conclusion.

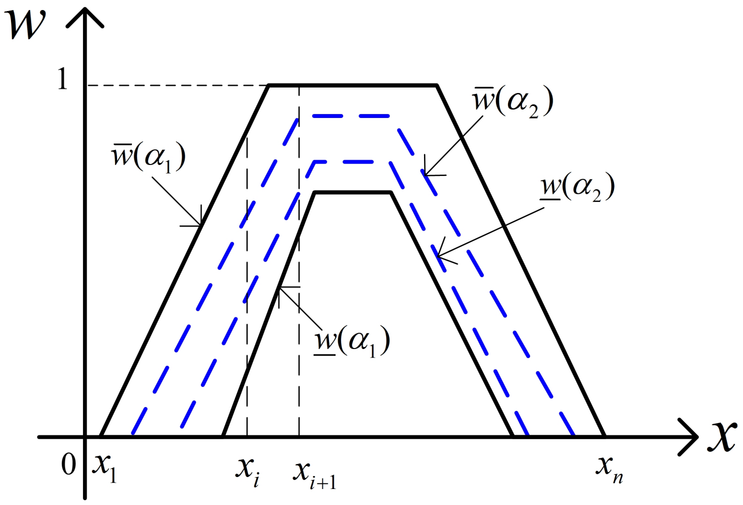


Figure 5.1: Two α-planes of a general T2 FS ($\alpha_1 < \alpha_2$).

5.2 Background

5.2.1 The α-cut Representation Theorem and the Extension Principle Theorem

For a T1 FS A, its α-cut A^α in which $\alpha \in [0, 1]$, is an interval:

$$A^\alpha = \{x \in X | \mu_A(x) \geq \alpha\} \tag{5.7}$$

where X is the universe of discourse, $\mu_A(x)$ is the membership of the FS A.

The α-cut Representation Theorem [35]: Let A be a T1 FS defined in X. Then

$$A = \bigcup_{\alpha \in [0,1]} \alpha I_{A^\alpha} / x \tag{5.8}$$

where I_{A^α} is an indicator function of the α-cut A^α, i.e.

$$I_{A^\alpha}(x) = \begin{cases} 1 & \forall x \in A^\alpha \\ \\ 0 & \forall x \notin A^\alpha \end{cases} \tag{5.9}$$

From the decomposition theorem, it can be observed that a T1 FS is determined once its α-cuts can be specified. Hence, a T1 FS can be identified by finding its α-cuts. This approach of identifying a T1 FS has been widely used in performing the FWA and LWA operation.

The Extension Principle Theorem [35]: Let $f(X_1, X_2, X_3, \cdots, X_n)$ be a function of fuzzy sets $X_i, i = 1, 2, \cdots, n$. Then,

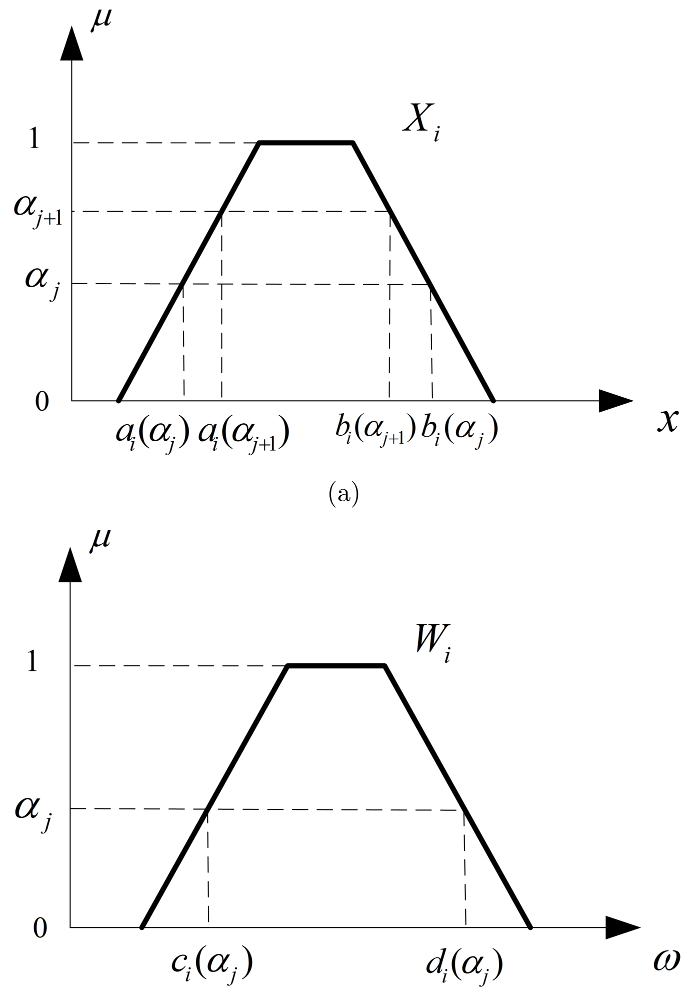
$$f^{\alpha}(X_1, X_2, \cdots, X_n) = f(X_1^{\alpha}, X_2^{\alpha}, \cdots, X_n^{\alpha}) \tag{5.10}$$

Computing the FWA and LWA operation are important applications of the extension principle theorem. Take the FWA operation as an example, each α-cut of the T1 output FS can be derived by performing the weighted average operation on the corresponding α-cut of all input fuzzy sets.

5.2.2 Computing FWA using the Karnik-Mendel Iterative Algorithm

The FWA can be computed by first discretizing the membership range $[0, 1]$ into m points: $\alpha_1 < \alpha_2 < \cdots < \alpha_m$. For each α_j, find the corresponding α cuts of X_i and W_i, denoted by $X_i(\alpha_j) = [a_i(\alpha_j), b_i(\alpha_j)]$ and $W_i(\alpha_j) = [c_i(\alpha_j), d_i(\alpha_j)]$ as shown in Fig. 5.2(a) and 5.2(b), respectively. For any particular α_j, the output of the FWA operation $Y_{FWA}(\alpha_j)$ is an interval, denoted by $[f_L(\alpha), f_R(\alpha)]$. According to the α-cut representation theorem, the output of the FWA operation $Y_{FWA}(\alpha_j)$ can be defined as

$$Y_{FWA}(\alpha_j) = \frac{\sum_{i=1}^{n} X_i(\alpha_j) W_i(\alpha_j)}{\sum_{i=1}^{n} W_i(\alpha_j)} = [f_L(\alpha_j), f_R(\alpha_j)] \tag{5.11}$$



(a)

(b)

Figure 5.2: Computing the FWA: (a) T1 FSs $X_i, i = 1, \cdots, n$. (b) T1 FSs $W_i, i = 1, \cdots, n$.

Based on (5.11), $f_L(\alpha_j)$ and $f_R(\alpha_j)$ may be defined as follows:

$$f_L(\alpha_j) = \min_{\substack{x_i \in [a_i(\alpha_j), b_i(\alpha_j)] \\ w_i \in [c_i(\alpha_j), d_i(\alpha_j)]}} \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \tag{5.12}$$

$$f_R(\alpha_j) = \max_{\substack{x_i \in [a_i(\alpha_j), b_i(\alpha_j)] \\ w_i \in [c_i(\alpha_j), d_i(\alpha_j)]}} \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \tag{5.13}$$

where x_i and w_i in (5.12) and (5.13) satisfy

$$x_i \in X_i(\alpha_j) = [a_i(\alpha_j), b_i(\alpha_j)] \tag{5.14}$$

$$w_i \in W_i(\alpha_j) = [c_i(\alpha_j), d_i(\alpha_j)] \tag{5.15}$$

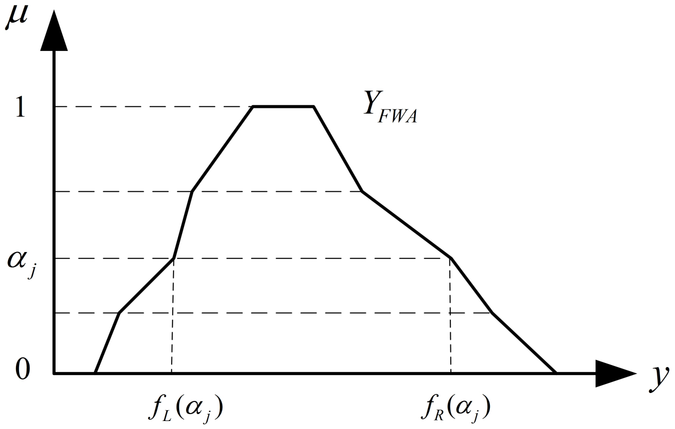


Figure 5.3: The output of the FWA: T1 FS Y_{FWA}

$f_L(\alpha_j)$ and $f_R(\alpha_j)$ defined in (5.12) and (5.13) may be modeled as y_l and y_r in (5.3) and (5.4), and hence $f_L(\alpha_j)$ and $f_R(\alpha_j)$ can be found using the KM iterative algorithm. Once the intervals $[f_L(\alpha_j), f_R(\alpha_j)]$ for all the chosen $\alpha_j, j = 1, 2, \cdots, m$ are found, the T1 FS Y_{FWA} can be constructed via

$$Y_{FWA} = \bigcup_{i=1}^{m} \alpha_j / Y_{LWA}(\alpha_j) = \bigcup_{j=1}^{m} \alpha_j / [f_L(\alpha_j), f_R(\alpha_j)] \tag{5.16}$$

5.2.3 Computing the LWA using the Karnik-Mendel Iterative Algorithm

For an IT2 FS, its membership is characterized by the footprint of uncertainty (FOU) bounded by the lower membership function (LMF) and the upper membership function (UMF). Take all IT2 FS $\tilde{X}_i$ and $\tilde{W}_i$ for example, it follows

$$\tilde{X}_i = 1/\text{FOU}(\tilde{X}_i) = 1/[\underline{X}_i, \overline{X}_i] \tag{5.17}$$

$$\tilde{W}_i = 1/\text{FOU}(\tilde{W}_i) = 1/[\underline{W}_i, \overline{W}_i] \tag{5.18}$$

where FOU$(\cdot)$ is the FOU of an IT2 FS. $\underline{X}_i$ and $\overline{X}_i$ ($\underline{W}_i$ and $\overline{W}_i$) are the LMF and the UMF of IT2 FS $\tilde{X}_i$ ($\tilde{W}_i$). Similarly, the output of the LWA operator, $\tilde{Y}_{LWA}$, may be represented as

$$\tilde{Y}_{LWA} = 1/\mathrm{FOU}(\tilde{Y}_{LWA}) = 1/[\underline{Y}_{LWA}, \overline{Y}_{LWA}] \tag{5.19}$$

where $\underline{Y}_{LWA}$ and $\overline{Y}_{LWA}$ are the LMF and the UMF of IT2 FS $\tilde{Y}_{LWA}$. Since an IT2 FS is determined once its LMF and UMF is known, computing the LWA is equivalent to computing the LMF $\underline{Y}_{LWA}$ and the UMF $\overline{Y}_{LWA}$.

As shown in Fig. 5.4, the LMF of IT2 FSs X_i and W_i have different height (maximum value of LMF) denoted by $h_{\underline{X}_i}$ and $h_{\underline{W}_i}$, respectively. Noted that the LMF $\underline{Y}_{LWA}$ has been proven to be limited to $\alpha_j \in [0, h_{min}]$ where h_{min} is the minimum height of all X_i and W_i, i.e.

$$h_{min} = \min\{h_{\underline{X}_i}, h_{\underline{W}_i}, i = 1, 2, \cdots, n\} \tag{5.20}$$

To find out $\underline{Y}_{LWA}$ and $\overline{Y}_{LWA}$, the range of the membership $[0, 1]$ is discretized into m points: $\alpha_1 < \alpha_2 < \cdots < \alpha_m$ where $\alpha_p \leq h_{min} < \alpha_{p+1}$ and then for each α_j, the α cuts of all $\tilde{X}_i$ and $\tilde{W}_i$ are found. Fig. 5.4(a) and 5.4(b) show the following variables to denote the α cuts of $\tilde{X}_i$ and $\tilde{W}_i$:

$$\underline{X}_i(\alpha_j) = [a_{ir}(\alpha_j), b_{il}(\alpha_j)], \quad \overline{X}_i(\alpha_j) = [a_{il}(\alpha_j), b_{ir}(\alpha_j)] \tag{5.21}$$

$$\underline{W}_i(\alpha_j) = [c_{ir}(\alpha_j), d_{il}(\alpha_j)], \quad \overline{W}_i(\alpha_j) = [c_{il}(\alpha_j), d_{ir}(\alpha_j)] \tag{5.22}$$

The interval $[f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)]$ and $[f_{Lr}(\alpha_j), f_{Rl}(\alpha_j)]$ in Fig. 5.5 are used to denote the α cut of the LMF $\underline{Y}_{LWA}$ and the UMF $\overline{Y}_{LWA}$, respectively. Applying the α-cut decomposition theorem to (5.2) leads to the following equations:

$$\overline{Y}_{LWA}(\alpha_j) = \frac{\sum_{i=1}^{m} \overline{X}_i(\alpha_j)\overline{W}_i(\alpha_j)}{\sum_{i=1}^{m} \overline{W}_i(\alpha_j)} = [f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)] \tag{5.23}$$

$$\underline{Y}_{LWA}(\alpha_j) = \frac{\sum_{i=1}^{m} \underline{X}_i(\alpha_j)\underline{W}_i(\alpha_j)}{\sum_{i=1}^{m} \underline{W}_i(\alpha_j)} = [f_{Lr}(\alpha_j), f_{Rl}(\alpha_j)] \tag{5.24}$$

(5.23) and (5.24) indicate that computing $\underline{Y}_{LWA}$ and $\overline{Y}_{LWA}$ is equivalent to computing a FWA, respectively. The LMF $\underline{Y}_{LWA}(\alpha_j)$ and the UMF $\overline{Y}_{LWA}(\alpha_j)$ may be computed as follows:

1. The interval $[f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)]$ for the UMF $\overline{Y}_{LWA}(\alpha_j)$: $0 < \alpha_j \leq 1$

   Based on (5.23), $f_{Ll}(\alpha_j)$ and $f_{Rr}(\alpha_j)$ may be defined as:

$$f_{Ll}(\alpha_j) = \min_{\substack{x_i \in [a_{il}(\alpha_j), b_{ir}(\alpha_j)] \\ w_i \in [c_{il}(\alpha_j), d_{ir}(\alpha_j)]}} \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \tag{5.25}$$

$$f_{Rr}(\alpha_j) = \max_{\substack{x_i \in [a_{il}(\alpha_j), b_{ir}(\alpha_j)] \\ w_i \in [c_{il}(\alpha_j), d_{ir}(\alpha_j)]}} \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \tag{5.26}$$

   where x_i and w_i in (5.25) and (5.26) satisfy

$$x_i \in \overline{X}_i(\alpha_j) = [a_{il}(\alpha_j), b_{ir}(\alpha_j)] \tag{5.27}$$

$$w_i \in \overline{W}_i(\alpha_j) = [c_{il}(\alpha_j), d_{ir}(\alpha_j)] \tag{5.28}$$

   $f_{Ll}(\alpha_j)$ and $f_{Rr}(\alpha_j)$ defined in (5.25) and (5.26) may be modeled as y_l and y_r in (5.3) and (5.4), respectively, and hence $f_{Ll}(\alpha_j)$ and $f_{Rr}(\alpha_j)$ can be found using the KM iterative algorithm.

2. The interval $[f_{Lr}(\alpha_j), f_{Rl}(\alpha_j)]$ for the LMF $\underline{Y}_{LWA}(\alpha_j)$: $0 < \alpha_j \leq h_{min}$

   Based on (5.24), $f_{Lr}(\alpha_j)$ and $f_{Rl}(\alpha_j)$ may be defined as:

$$f_{Lr}(\alpha_j) = \min_{\substack{x_i \in [a_{ir}(\alpha_j), b_{il}(\alpha_j)] \\ w_i \in [c_{ir}(\alpha_j), d_{il}(\alpha_j)]}} \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \tag{5.29}$$

$$f_{Rl}(\alpha_j) = \max_{\substack{x_i \in [a_{ir}(\alpha_j), b_{il}(\alpha_j)] \\ w_i \in [c_{ir}(\alpha_j), d_{il}(\alpha_j)]}} \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \tag{5.30}$$

   where x_i and w_i in (5.29) and (5.30) satisfy

$$x_i \in \underline{X}_i(\alpha_j) = [a_{ir}(\alpha_j), b_{il}(\alpha_j)] \tag{5.31}$$

$$w_i \in \underline{W}_i(\alpha_j) = [c_{ir}(\alpha_j), d_{il}(\alpha_j)] \tag{5.32}$$

$f_{Lr}(\alpha_j)$ and $f_{Rl}(\alpha_j)$ defined in (5.29) and (5.30) may be interpreted as y_l and y_r in (5.3) and (5.4), and hence $f_{Lr}(\alpha_j)$ and $f_{Rr}(\alpha_j)$ can be found using the KM iterative algorithm.

Once $f_{Lr}(\alpha_j)$, $f_{Rl}(\alpha_j)$, $f_{Ll}(\alpha_j)$ and $f_{Rr}(\alpha_j)$ are available, the UMF and LMF $\underline{Y}_{LWA}$ and $\overline{Y}_{LWA}$ can be constructed via the following equations:

$$\underline{Y}_{LWA} = \bigcup_{j=1}^{m} \alpha_j/\underline{Y}_{LWA}(\alpha_j) = \bigcup_{j=1}^{m} \alpha_j/[f_{Lr}(\alpha_j), f_{Rl}(\alpha_j)] \tag{5.33}$$

$$\overline{Y}_{LWA} = \bigcup_{j=1}^{m} \alpha_j/\overline{Y}_{LWA}(\alpha_j) = \bigcup_{j=1}^{m} \alpha_j/[f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)] \tag{5.34}$$

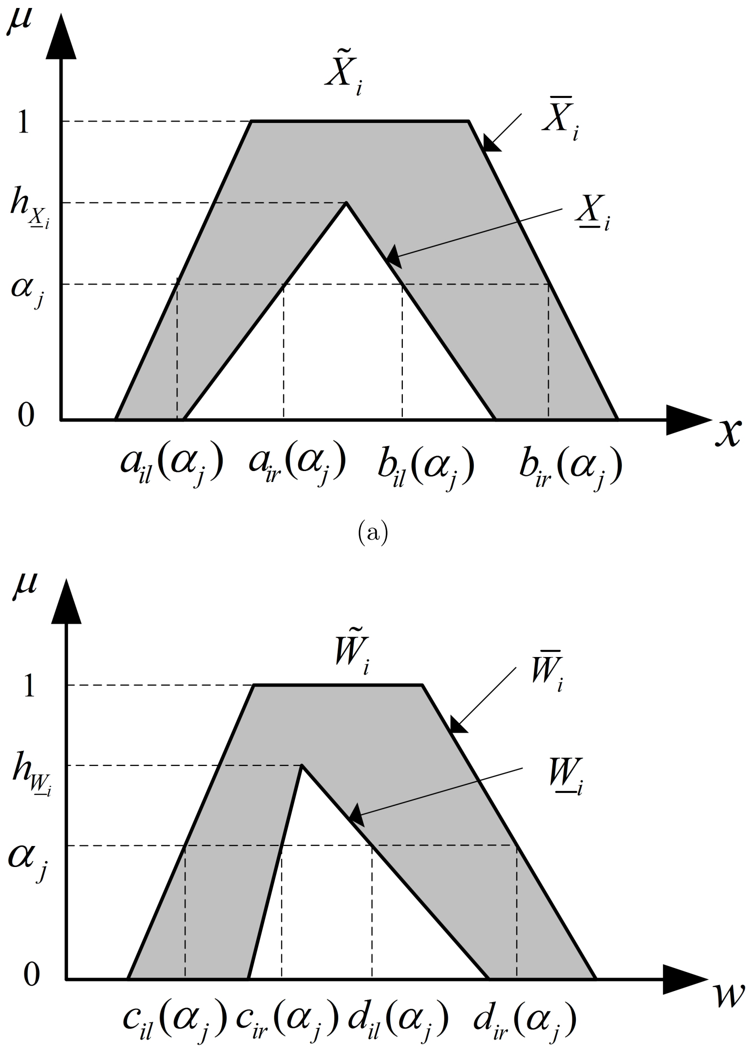
5.2.4 The KM Iterative Algorithm and the EKM Iterative Algorithm

In steps of computing the FWA and LWA, the endpoints of the α-cut, $Y_{FWA} = [f_L(\alpha_j), f_R(\alpha_j)]$, $\underline{Y}_{LWA} = [f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)]$ and $\overline{Y}_{LWA} = [f_{Rl}(\alpha_j), f_{Lr}(\alpha_j)]$ can be modelled as the problem defined in (5.3) and (5.4) and computed using the KM iterative algorithm or the EKM iterative algorithm. The KM and EKM iterative algorithm solves the problem defined in (5.3) and (5.4) based on the the following results [54]:

Since x_i appears only in the nominator of (5.3) and (5.4), y_l depends solely on the left bound $\underline{x}_i$ while y_r may be expressed using only the right bound $\overline{x}_i$. Hence, (5.3) and (5.4) can also be written as

$$y_l = \min_{w_i \in [\underline{w}_i, \overline{w}_i]} \frac{\sum_{i=1}^{n} \underline{x}_i w_i}{\sum_{i=1}^{n} w_i} \tag{5.35}$$

$$y_r = \max_{w_i \in [\underline{w}_i, \overline{w}_i]} \frac{\sum_{i=1}^{n} \overline{x}_i w_i}{\sum_{i=1}^{n} w_i} \tag{5.36}$$



(a)

(b)

Figure 5.4: Computing the LWA: (a) IT2 FS X_i. (b) IT2 FS W_i.

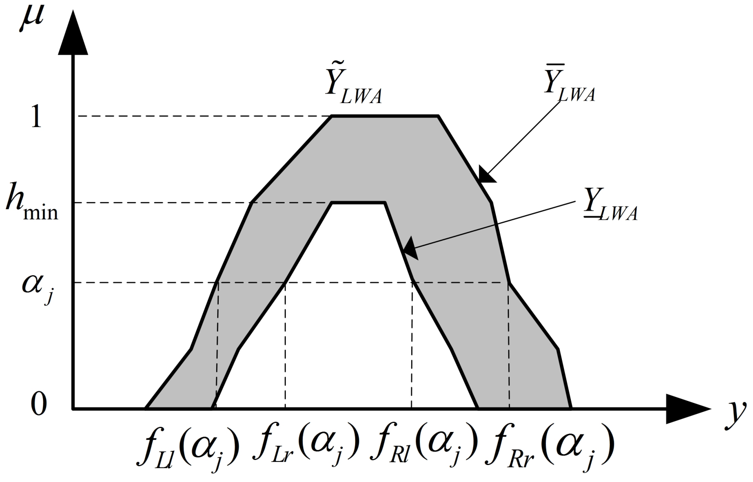


Figure 5.5: The result of the LWA: IT2 FS $\tilde{Y}_{LWA}$

It has been proven that y_l and y_r defined in (5.3) and (5.4) involve one switch point between the upper bound and the lower bound of the weights $[\underline{w}_i, \overline{w}_i]$:

$$y_l = \frac{\sum_{i=1}^{L} \underline{x}_i \overline{w}_i + \sum_{i=L+1}^{n} \underline{x}_i \underline{w}_i}{\sum_{i=1}^{L} \overline{w}_i + \sum_{i=L+1}^{n} \underline{w}_i} \tag{5.37}$$

$$y_r = \frac{\sum_{i=1}^{R} \overline{x}_i \underline{w}_i + \sum_{i=R+1}^{n} \overline{x}_i \overline{w}_i}{\sum_{i=1}^{R} \underline{w}_i + \sum_{i=R+1}^{n} \overline{w}_i} \tag{5.38}$$

where L and R are the switch points satisfying

$$\underline{x}_L \leq y_l < \underline{x}_{L+1} \tag{5.39}$$

$$\overline{x}_R \leq y_r < \overline{x}_{R+1} \tag{5.40}$$

Noted that in (5.37) and (5.38), $\underline{x}_i$ and $\overline{x}_i, i = 1, 2, \cdots, n$ have been ranked in an increasing order, i. e.

$$\underline{x}_1 \leq \underline{x}_2 \leq \cdots \leq \underline{x}_n \tag{5.41}$$

$$\overline{x}_1 \leq \overline{x}_2 \leq \cdots \leq \overline{x}_n \tag{5.42}$$

The KM iterative algorithm computes y_l or y_r by updating the switch point iteratively until the actual one L (R) is reached. The main steps of the KM iterative algorithm include initialization, update and termination, and the detailed procedures of the KM iterative algorithm are as follows:

1. Step 1: Sort x_i in an increasing order (x_i denotes $\underline{x}_i$ for y_l or $\overline{x}_i$ for y_r) and label them as $x_1 < x_2 < \cdots < x_n$. Let $[\underline{w}_i, \overline{w}_i]$ be the corresponding weight of x_i.

2. Step 2: Set w_i as

$$w_i = \frac{\underline{w}_i + \overline{w}_i}{2} \tag{5.43}$$

and then compute

$$y = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{5.44}$$

3. Step 3: Find the switch point k such as

$$x_k \leq y \leq x_{k+1} \tag{5.45}$$

4. Step 4: Set w_i as:

   (a) y_l:

$$w_i = \begin{cases} \overline{w}_i & \text{for } i \leq k \\ \\ \underline{w}_i & \text{for } i > k \end{cases} \tag{5.46}$$

   (b) y_r:

$$w_i = \begin{cases} \underline{w}_i & \text{for } i \leq k \\ \\ \overline{w}_i & \text{for } i > k \end{cases} \tag{5.47}$$

   and compute

$$y' = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{5.48}$$

5. Step 5: If $y' = y$, stop. k is the actual switch point L (R) and $y_l = y$ ($y_r = y$). Otherwise, set $y = y'$ and go to Step 3.

The EKM iterative algorithm is an enhanced version of the KM iterative algorithm through optimizing the initial switch point, the terminal condition and the computational process [85]. The strategies used to optimize the initialization, termination and computation are as follows:

1. Initialization: The initialization of the KM iterative algorithm is inefficient and thus may cause a large number of iterations. To reduce the number of iterations required in the search process, the switch point is initialized as $k = [n/2.4]$ for y_l or $k = [n/1.7]$ for y_r ($[\cdot]$ denotes the nearest integer.), where n is the number of x_i.

2. Computation: In the KM iterative algorithm, to compute y in (5.48) in each iteration, $\sum_{i=1}^{m} w_i$ and $\sum_{i=1}^{n} x_i w_i$ need to be computed. To reduce computational cost, the new computation technique is introduced by utilizing results from the last iteration. Suppose that in the jth iteration, the switch point k, $\sum_{i=1}^{m} w_i$ and $\sum_{i=1}^{n} x_i w_i$ are denoted by k_j, $(\sum_{i=1}^{m} w_i)_j$ and $(\sum_{i=1}^{n} x_i w_i)_j$, respectively. $(\sum_{i=1}^{m} w_i)_{j+1}$ can be computed by adding the difference between $(\sum_{i=1}^{m} w_i)_j$ and $(\sum_{i=1}^{m} w_i)_{j+1}$ to $(\sum_{i=1}^{m} w_i)_j$. Similarly, $(\sum_{i=1}^{m} w_i x_i)_{j+1}$ can be computed by adding the difference between $(\sum_{i=1}^{m} w_i x_i)_j$ and $(\sum_{i=1}^{m} w_i x_i)_{j+1}$.

3. Termination: In the KM iterative algorithm, the termination is identified by comparing the output of the current iteration with the last iteration, indicating that another iteration is needed although the actual switch point is found. To avoid the computation in the unnecessary iteration, the termination of iterations is proposed to be identified by comparing the switch point of the current iteration with that of the last iteration.

5.3 Improved Algorithms for the FWA and the LWA

Although the introduction of the KM iterative algorithm and EKM iterative algorithm has reduced the computational cost required in calculating the FWA and the LWA, computing the FWA and the LWA remains computationally intensive due to the iterative nature of the KM and EKM iterative algorithms. To further reduce their computational overhead, the following subsection proposes possible strategies employed for optimizing the KM and EKM iterative algorithms, followed by detailed descriptions of the proposed FWA and LWA algorithms in the next subsection.

5.3.1 Strategies for Optimizing the KM / EKM Iterative Algorithm for Computing FWA and LWA

The result of the FWA, Y_{FWA}, can be constructed from its α-cut $Y_{FWA}(\alpha_j) = [f_L(\alpha_j), f_R(\alpha_j)]$ via (5.16) where the endpoints $f_L(\alpha_j)$ and $f_R(\alpha_j)$ may be computed using the KM iterative algorithm or the EKM iterative algorithm. However, the KM and EKM iterative algorithms initialize the search process in a fixed way regardless of the actual position of the switch point. Such initialization methods may unnecessarily increase a large number of iterations and thus require greater computational overhead. As an effort to reduce the number of iterations in the KM /EKM iterative algorithm, the efficient method for computing the centroid of a T2 FS initializes the search process by setting the switch point of the last

α-plane as the switch points of the first iteration. This initialization was proposed based on the property that the switch points for its α-plane will converge to those of the $\alpha = 1$ plane as the value of α increases, and thus the switch point of successive α-planes are close. Unlike the steps for computing the centroid of a general T2 FS where the domain of the primary variable $x_i, i = 1, 2, \cdots, n$ is fixed, (5.3) and (5.4) and Fig. 5.2(a) shows the FWA computation involves the bound representing the α-cuts of X_i, i.e. $[\underline{x}_i, \overline{x}_i] = [a_i, b_i]$, that vary as the value of α varies. As a result, in computing the FWA or LWA, the switch points may not monotonously converge to that for $\alpha = 1$ cut, and thus the initialization proposed for computing the centroid of a T2 FS may not be directly extended to computing the FWA and LWA. Instead, new strategies need to be proposed based on the following properties about the endpoints, $f_L(\alpha_j)$ and $f_R(\alpha_j)$:

Theorem 5.1 *The output Y_{FWA} of a FWA operation defined in (5.1) is a T1 FS which has continuous membership if T1 FSs X_i and W_i have continuous MFs.*

A proof of Theorem 5.1 is provided in Appendix C. Theorem 5.1 indicates that for the FWA, the variables $f_L(\alpha_j)$ and $f_R(\alpha_j)$ in Fig. 5.3 are continuous with respect to α_j.

Theorem 5.2 *Assume the T1 FSs X_i and W_i in (5.1) have convex membership grade. Then $Y_{FWA}(\alpha_j) = [f_L(\alpha_j), f_R(\alpha_j)], j = 1, \cdots, m$ defined in (5.12)-(5.13) satisfy the following inequality:*

$$Y_{FWA}(\alpha_m) \subseteq Y_{FWA}(\alpha_{m-1}) \subseteq \cdots Y_{FWA}(\alpha_{j+1}) \subseteq Y_{FWA}(\alpha_j) \cdots \subseteq Y_{FWA}(\alpha_1) \tag{5.49}$$

Equivalently speaking,

$$f_L(\alpha_1) < f_L(\alpha_2) < \cdots < f_L(\alpha_j) < f_L(\alpha_{j+1}) < \cdots < f_L(\alpha_m) \tag{5.50}$$

$$f_R(\alpha_1) > f_R(\alpha_2) > \cdots > f_R(\alpha_j) > f_R(\alpha_{j+1}) > \cdots > f_R(\alpha_m) \tag{5.51}$$

A proof of Theorem 5.2 is provided in Appendix C. From Theorem 5.2, it can be observed that in the FWA, the variable $f_L(\alpha_j)$ ($f_R(\alpha_j)$) will increase (decrease) as the value of α_j increases.

Example 1 Consider a FWA example, in which three fuzzy attributes $X_j, j = 1, 2, 3$ and their corresponding weights are shown in Fig. 5.6(a)-5.6(b), respectively. Fig. 5.6(c) shows the output Y_{FWA}. An observation of Fig. 5.6(c) indicates that the MF of Y_{FWA} is continuous and the α-cuts of the FWA output converge to the $\alpha = 1$ cut as the value of α increases. Since the FS Y_{FWA} is a weighted average of X_i, the domain y of Y_{FWA} and the domain of X_i have the same range and they share the same unit.

As shown by (5.23) and (5.24) in Section 5.2.3, the result of a LWA operation, $\tilde{Y}_{LWA}$, may be obtained by using the FWA to determine the bounds $\underline{Y}_{LWA}$ and $\overline{Y}_{LWA}$. Consequently, Theorem 5.1 and Theorem 5.2 may be restated in the following form for the LWA operation:

Theorem 5.3 *The output $\tilde{Y}_{LWA}$ of a LWA operation defined in (5.2) is an IT2 FS. Its LMF $\underline{Y}_{LWA}$ and UMF $\overline{Y}_{LWA}$ are continuous if IT2 FSs $\tilde{X}_i$ and $\tilde{W}_i$ have continuous MFs.*

Theorem 5.4 $\underline{Y}_{LWA}(\alpha_j) = [f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)]$*, $j = 1, \cdots, p$ defined in (5.25)-(5.26) and* $\overline{Y}_{LWA}(\alpha_j) = [f_{Lr}(\alpha_j), f_{Rl}(\alpha_j)]$*, $j = 1, \cdots, m$ defined in (5.29)-(5.30)* *satisfy the following inequalities:*

$$\underline{Y}_{LWA}(\alpha_p) \subseteq \underline{Y}_{LWA}(\alpha_{p-1}) \subseteq \cdots \underline{Y}_{LWA}(\alpha_{j+1}) \subseteq \underline{Y}_{LWA}(\alpha_j) \cdots \subseteq \underline{Y}_{LWA}(\alpha_1) \qquad (5.52)$$

$$\overline{Y}_{LWA}(\alpha_m) \subseteq \overline{Y}_{LWA}(\alpha_{m-1}) \subseteq \cdots \overline{Y}_{LWA}(\alpha_{j+1}) \subseteq \underline{Y}_{LWA}(\alpha_j) \cdots \subseteq \overline{Y}_{LWA}(\alpha_1) \qquad (5.53)$$

Equivalently speaking,

$$f_{Lr}(\alpha_1) < f_{Lr}(\alpha_2) < \cdots < f_{Lr}(\alpha_j) < f_{Lr}(\alpha_{j+1}) < \cdots < f_{Lr}(\alpha_p) \qquad (5.54)$$

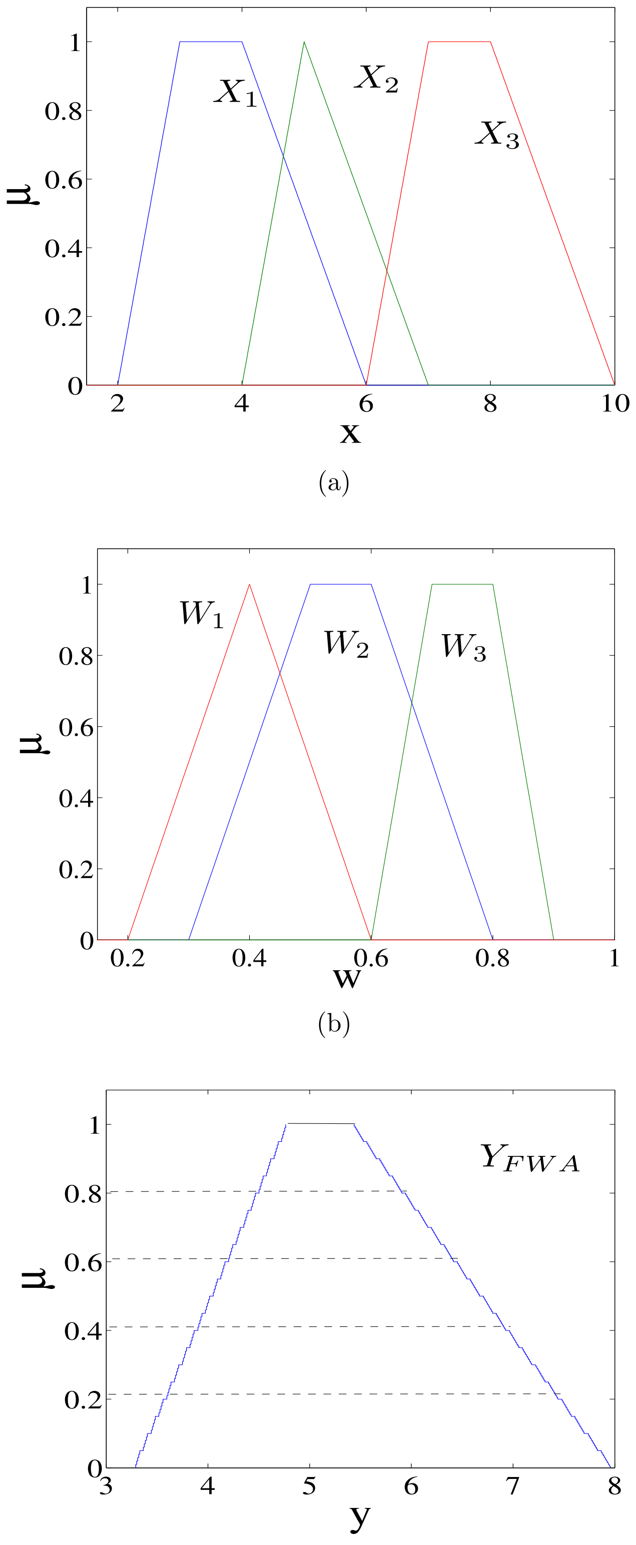
$$f_{Rl}(\alpha_1) > f_{Rl}(\alpha_2) > \cdots > f_{Rl}(\alpha_j) > f_{Rl}(\alpha_{j+1}) > \cdots > f_{Rl}(\alpha_p) \qquad (5.55)$$

$$f_{Ll}(\alpha_1) < f_{Ll}(\alpha_2) < \cdots < f_{Ll}(\alpha_j) < f_{Ll}(\alpha_{j+1}) < \cdots < f_{Ll}(\alpha_m) \qquad (5.56)$$

$$f_{Rr}(\alpha_1) > f_{Rr}(\alpha_2) > \cdots > f_{Rr}(\alpha_j) > f_{Rr}(\alpha_{j+1}) > \cdots > f_{Rr}(\alpha_m) \qquad (5.57)$$

Example 2 Consider a LWA example. Fig. 5.7(a) shows four IT2 FSs $\tilde{X}_i$ representing the attributes, and the corresponding weights $\tilde{W}_i$ are shown in Fig. 5.7(b). Fig. 5.7(c) shows the output of the FWA operation, $\tilde{Y}_{LWA}$. Its LMF $\underline{Y}_{LWA}$ is limited to $[0, h_{min}]$ where $h_{min} = 0.7$. From Fig. 5.7(c), it can be observed that both its LMF and UMF are continuous and their α cuts converge to their $\alpha = 0.7$ and $\alpha = 1$ cut, respectively. This observation is consistent with Theorem 4.

Since the FWA is the foundation of the LWA computation, the efficient KM algorithm for computing the centroid of a general T2 FS [96] may be extended to lower the computational cost of FWA/LWA operations by using the properties of FWA. Theorem 5.1 shows that both $f_L(\alpha_j)$ and $f_R(\alpha_j)$ are continuous with respect to α_j, while Theorem 5.2 indicates that $f_L(\alpha_j)$ and $f_R(\alpha_j)$ monotonously vary as α_j increases. The implication of the two theorems is that the endpoints of the jth

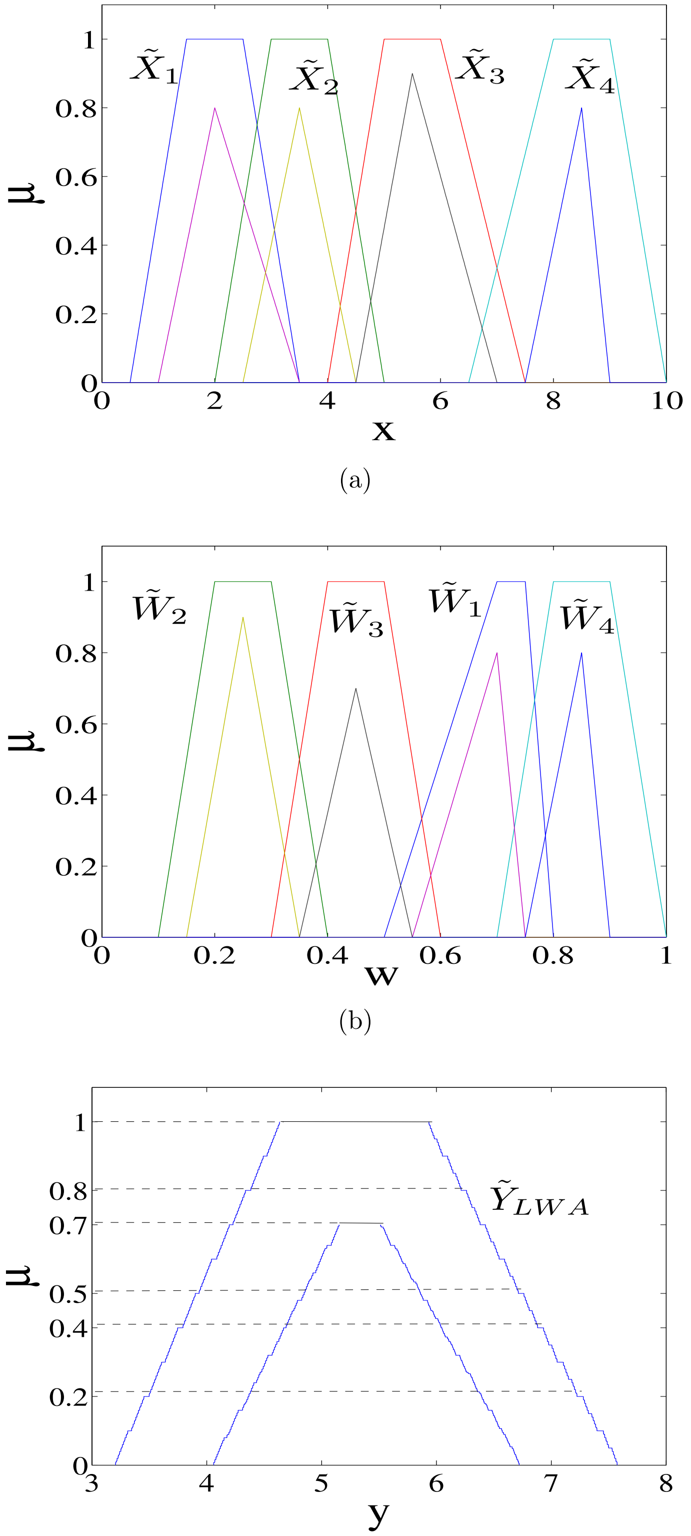


(a)

(b)

(c)

Figure 5.6: A FWA example: (a) T1 FSs $X_i, i = 1, 2, 3$. (b) T1 FSs $W_i, i = 1, 2, 3$. (c) the output T1 FS Y_{FWA}.

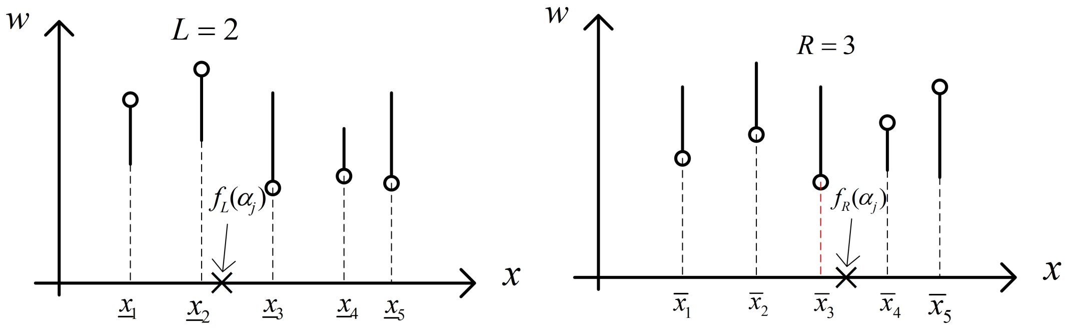


(a)

(b)

(c)

Figure 5.7: A LWA example(a) IT2 FSs $\tilde{X}_i, i = 1, 2, 3, 4$. (b) IT2 FSs $\tilde{W}_i, i = 1, 2, 3, 4$. (c) the output IT2 FS $\tilde{Y}_{LWA}$.



(a)

(b)

Figure 5.8: $f_L(\alpha_j)$ and $f_R(\alpha_j)$(y_l and y_r in (5.37) and (5.38)): (a) $f_L(\alpha_j)$ ($L = 2$). (b) $f_R(\alpha_j)$ ($R = 3$). (The solid vertical lines show the weights $[\underline{w}_i, \bar{w}_i]$ for $\underline{x}_i$ / $\bar{x}_i$, $i = 1, 2, 3, 4, 5$; The membership grades used to calculate $f_L(\alpha_j)$ and $f_R(\alpha_j)$ are labelled by circles.)

and $(j + 1)$th α-cuts, i. e. $f_L(\alpha_j)$ and $f_L(\alpha_{j+1})$ ($f_R(\alpha_j)$ and $f_R(\alpha_{j+1})$), are very close. As the update law of the KM or EKM iterative algorithm is monotonously convergent, starting the iterative search using a switch point which is close to the search objective may result in a reduction of the number of iterations. Hence, one possible initialization strategy that reduces the number of iterations is to use the endpoints of the adjacent α-cut, $f_L(\alpha_j)$ and $f_R(\alpha_j)$, to initialize the KM iterative algorithm for the α_{j+1}-cut. On the other hand, (5.39) and (5.40) indicate that the endpoints for an α-cut, $f_L(\alpha_j)$ and $f_R(\alpha_j)$, are derived from an unique switch point L (R) and correspond to unique values in the universe of discourse for attributes, $\underline{x}_L$ ($\overline{x}_R$). Fig. 5.8 illustrates the relationship between the endpoint $f_L(\alpha_j)$ ($f_R(\alpha_j)$), switch point L (R) and variable $\underline{x}_L$ ($\overline{x}_R$). Since the endpoints of the successive α-cuts are close, the corresponding switch point L (R) and variables $a_L(\alpha_j)$ ($b_R(\alpha_j)$) of the successive α-cuts will also be close. This observation leads to alternative strategies for initializing the KM iterative algorithm using the switch point and variables of the last α-cut. Details of the proposed algorithms will be summarized in the next subsection.

Another important factor that determines the computation requirement of the KM algorithm is the termination condition. In the original KM iterative algorithm, the decision to terminate computation is made by examining whether the result of the current iteration is equal to that of the last iteration. This termination method is based on the observation that the result of each iteration will vary until the actual endpoint is found. As a result, the algorithm will terminate one iteration after the one in which the actual endpoint is found. The implication is that an additional iteration is needed, even though the actual endpoint has been found. To remove this unnecessary iteration, the EKM iterative algorithm identifies the termination of iterations by examining whether the switch point is identical to the one of the last iteration. Although this termination condition of the EKM iterative algorithm avoids the calculation of the unnecessary iteration, the switch point for this unnecessary iteration still needs to be identified.

To completely remove the unnecessary iteration, a new terminal condition is introduced based on the update law of the KM iterative algorithm in Step 3. Suppose y is calculated using k as the switch point. Then, a new switch point, k', is generated via the update law $\underline{x}_{k'} < y < \underline{x}_{k'+1}$. Then it follows

- If $\underline{x}_k < y < \underline{x}_{k+1}$, $k = k'$ and the iterations end.
- If $y > \underline{x}_{k+1}$ or $y < \underline{x}_k$, $k \neq k'$ and the the switch point continues to update.

Hence, the termination of iterations can be identified by examining whether $\underline{x}_k < y < \underline{x}_{k+1}$ holds. Using this terminal condition to identify the end of iterations, the unnecessary iteration can be completely avoided.

5.3.2 The Proposed Algorithms for the FWA and the LWA

In summary, the proposed FWA algorithm starts by using KM /EKM iterative algorithm to compute $f_L(\alpha_1)$ and $f_R(\alpha_1)$, the bounds of the α_1-cut, and then increase the value of α_j. For subsequent α-cuts, the bounds, $f_L(\alpha_j)$ and $f_R(\alpha_j)$, are computed using the improved KM (IKM) iterative algorithm that is based on the proposed initialization strategies and the proposed termination strategy. Since three possible initialization strategies using the bounds, the variable and the switch point for the last α-cut are proposed in Section 5.3.1, there are three versions of IKM iterative algorithm. For ease of description, the IKM iterative algorithm initialized using the bounds, the variable and the switch point are denoted by IKMA-b, IKMA-x, IKMA-s, respectively. The flowchart of the proposed FWA algorithm is shown in Fig. 5.9. The detailed procedure of the IKM iterative algorithm may be described as

1. y_l

   - Step 1: Sort $\underline{x}_i, i = 1, 2, \cdots, n$ in an increasing order. Let $[\underline{w}_i, \overline{w}_i]$ be the corresponding weight of x_i.

   - Step 2:

     (a) IKMA-b: Find the switch point k such as

     $$\underline{x}_k \leq f_L(\alpha_{j-1}) \leq \underline{x}_{k+1}$$

     (b) IKMA-x: Find the switch point such that $\underline{x}_k = a_h(\alpha_j), 1 \leq h \leq n$ where in the IKM or EKM algorithm for the α_{j-1}-cut, $\underline{x}_L$ represents $a_h(\alpha_{j-1})$.

(c) IKMA-s: Initialize the switch point k as $k = L$ where L is the actual switch point for the α_{j-1}-cut.

- Step 3: Compute

$$a = \sum_{i=1}^{k} \underline{x}_i \overline{w}_i + \sum_{i=k+1}^{n} \underline{x}_i \underline{w}_i$$
$$b = \sum_{i=1}^{k} \overline{w}_i + \sum_{i=k+1}^{n} \underline{w}_i$$

and compute

$$y = \frac{a}{b} \tag{5.58}$$

- Step 4: if $\underline{x}_{k+1} > y > \underline{x}_k$, stop, $y_l = y$. Otherwise, continue.
- Step 5: Find the switch point k' such as

$$\underline{x}_{k'} \leq y \leq \underline{x}_{k'+1} \tag{5.59}$$

- step 6: Compute $s = \text{sign}(k' - k)$ and

$$a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} \underline{x}_i (\overline{w}_i - \underline{w}_i)$$
$$b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} (\overline{w}_i - \underline{w}_i)$$
$$y' = \frac{a'}{b'}$$

- Step 7: Set $y = y'$, $a = a'$, $b = b'$ and $k = k'$. Go to Step 3.

2. y_r

- Step 1: Sort $\overline{x}_i, i = 1, 2, \cdots, n$ in an increasing order. Let $[\underline{w}_i, \overline{w}_i]$ be the corresponding weight of $\overline{x}_i$.

- Step 2:

  (a) IKMA-b: Find the switch point k such as

  $$\overline{x}_k \leq f_R(\alpha_{j-1}) \leq \overline{x}_{k+1}$$

  (b) IKMA-x: Find the switch point such that $\overline{x}_k = b_h(\alpha_j), 1 \leq h \leq n$ where in the IKM or EKM algorithm for the α_{j-1}-cut, $\overline{x}_R$ represents $b_h(\alpha_{j-1})$.

  (c) IKMA-s: Initialize the switch point k as $k = R$ where R is the actual switch point for the α_{j-1}-cut.

- Step 3: compute

$$a = \sum_{i=1}^{k} \overline{x}_i \underline{w}_i + \sum_{i=k+1}^{n} \overline{x}_i \overline{w}_i$$
$$b = \sum_{i=1}^{k} \underline{w}_i + \sum_{i=k+1}^{n} \overline{w}_i$$

and compute

$$y = \frac{a}{b} \tag{5.60}$$

- Step 4: if $\overline{x}_{k+1} > y > \overline{x}_k$, stop, $y_r = y$. Otherwise, continue.

- Step 5: Find the switch point k' such as

$$\overline{x}_{k'} \leq y \leq \overline{x}_{k'+1}$$

- step 6: Compute $s = \text{sign}(k' - k)$ and

$$a' = a - s \sum_{i=\min(k,k')+1}^{\max(k,k')} \overline{x}_i(\overline{w}_i - \underline{w}_i)$$
$$b' = b - s \sum_{i=\min(k,k')+1}^{\max(k,k')} (\overline{w}_i - \underline{w}_i)$$
$$y' = \frac{a'}{b'}$$

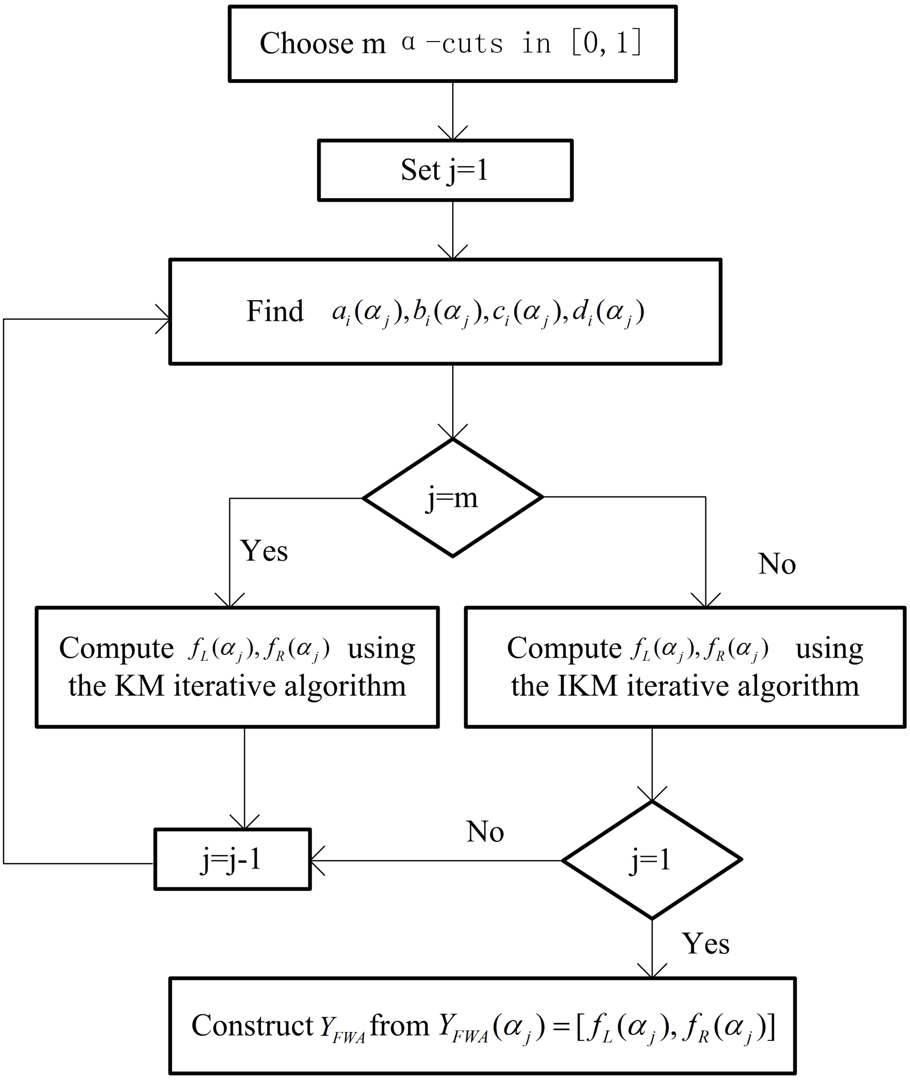

Choose m α-cuts in [0, 1]
Set j=1
Find $a_i(\alpha_j), b_i(\alpha_j), c_i(\alpha_j), d_i(\alpha_j)$
j=m
Yes
No
Compute $f_L(\alpha_j), f_R(\alpha_j)$ using the KM iterative algorithm
Compute $f_L(\alpha_j), f_R(\alpha_j)$ using the IKM iterative algorithm
j=j-1
No
j=1
Yes
Construct Y_{FWA} from $Y_{FWA}(\alpha_j) = [f_L(\alpha_j), f_R(\alpha_j)]$

Figure 5.9: The flowchart of the proposed FWA algorithm

- Step 7: Set $y = y'$, $a = a'$, $b = b'$ and $k = k'$. Go to Step 3.

Since the computation of the LWA uses the FWA as foundation, the proposed FWA algorithms can be extended to computing the LWA. The flowchart of the proposed LWA algorithms are showed in Fig. 5.10.

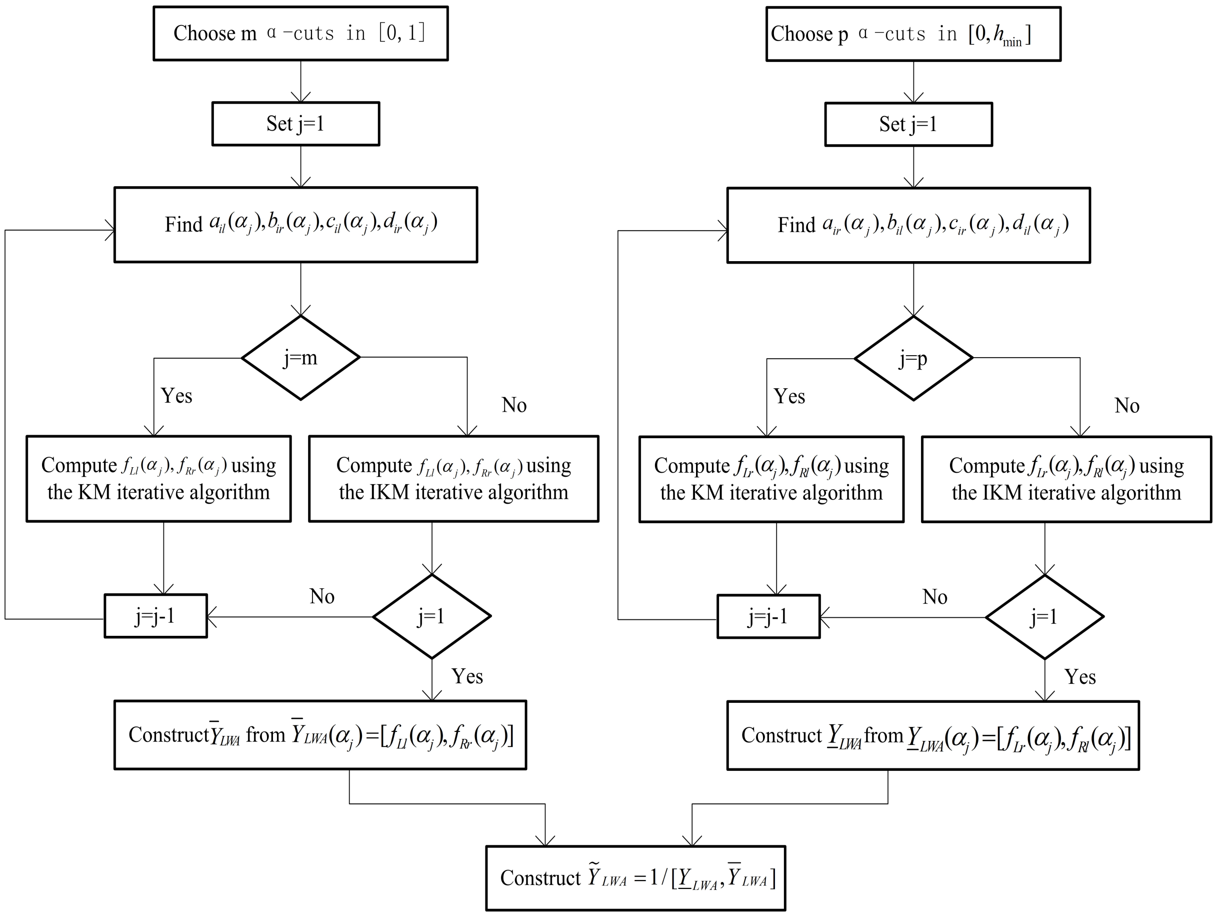

Choose m α -cuts in [0, 1]
Set j=1
Find $a_{il}(\alpha_j), b_{ir}(\alpha_j), c_{il}(\alpha_j), d_{ir}(\alpha_j)$
j=m
Yes
No
Compute $f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)$ using the KM iterative algorithm
Compute $f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)$ using the IKM iterative algorithm
j=j-1
No
j=1
Yes
Construct $\overline{Y}_{LWA}$ from $\overline{Y}_{LWA}(\alpha_j) = [f_{Ll}(\alpha_j), f_{Rr}(\alpha_j)]$
Choose p α -cuts in $[0, h_{\min}]$
Set j=1
Find $a_{ir}(\alpha_j), b_{il}(\alpha_j), c_{ir}(\alpha_j), d_{il}(\alpha_j)$
j=p
Yes
No
Compute $f_{Lr}(\alpha_j), f_{Rl}(\alpha_j)$ using the KM iterative algorithm
Compute $f_{Lr}(\alpha_j), f_{Rl}(\alpha_j)$ using the IKM iterative algorithm
j=j-1
No
j=1
Yes
Construct $\underline{Y}_{LWA}$ from $\underline{Y}_{LWA}(\alpha_j) = [f_{Lr}(\alpha_j), f_{Rl}(\alpha_j)]$
Construct $\widetilde{Y}_{LWA} = 1/[\underline{Y}_{LWA}, \overline{Y}_{LWA}]$

Figure 5.10: The flowchart of the proposed LWA algorithm

5.4 Theoretical Analysis of Computational Overhead of the Proposed FWA and LWA Algorithm

The above analysis indicates that the proposed FWA and LWA algorithms can achieve smaller computational overhead. To further demonstrate the effectiveness of the proposed algorithms, this section theoretically studies the computational overhead by comparing the number of iterations required by the proposed algorithm with that of the EKM iterative algorithm. Since computing the LWA is equivalent to computing two FWAs, the following analysis focuses on the FWA.

To compute the FWA using the EKM iterative algorithm, the EKM iterative algorithm needs to be implemented once for either bound of each α-cut of the output Y_{FWA}. Since the proposed IKM iterative algorithm updates the switch point in the same manner as the EKM iterative algorithm, the study can be conducted by considering the worst case in which all possible switch points need to be examined. In each implementation of the EKM iterative algorithm, all n possible switch points, $\underline{x}_i, i = 1, 2, \cdots, n$ ($\overline{x}_i, i = 1, 2, \cdots, n$), need to be examined in the worst case. As a result, at most $2n$ iterations are required for computing the bounds of each α-cut of the output Y_{FWA}. Suppose that in performing the FWA operation, m α-cuts are chosen, and then computing the FWA using the EKM iterative algorithm requires at most $2nm$ iterations.

The proposed FWA algorithm starts from α_1 (α_m) and increases (decreases) the value of α_j. For the first α-cut, the EKM iterative algorithm needs to be implemented twice and thus the maximum number of iterations for the first α-cut of the output Y_{FWA} is $2n$. For the other chosen α_j, the IKM iterative algorithm is used to search for the bounds $f_L(\alpha_j)$ and $f_R(\alpha_j)$. Here the IKM iterative algorithm initializing using the bounds of the adjacent α-cut of Y_{FWA} (IKMA-b) is considered. Because the IKM iteration algorithm for $f_L(\alpha_j)$ ($f_R(\alpha_j)$) starts its search from that of the last α-cut, the search process for the other chosen α_j takes at most $2n$ iterations. In total, computing the FWA using the proposed algorithm takes at most $4n$ iterations.

Let C_1 and C_2 be the maximum number of iterations needed by the EKM iterative algorithm and the proposed algorithm, respectively, i. e. $C_1 = 2nm$ and $C_2 = 4n$. $\triangle C$ is defined as the difference between C_1 and C_2, i.e.

$$\begin{aligned} \triangle C &= C_1 - C_2 \\ &= 2nm - 4n = 2n(m-2) \geq 0 \end{aligned} \tag{5.61}$$

An observation of (5.61) indicates that the proposed algorithm takes less iterations than the EKM iterative algorithm, regardless of the value of m ($m > 1$). More importantly, the number of the reduced iterations, $\triangle C$, will increase as the number of the chosen α cuts, m, increases. This may be attributed to the property that the maximum number of iterations required by the EKM iterative algorithm is linearly related with m, i. e. C_1 is $O(mn)$, while the maximum number of iterations required by the proposed FWA algorithm is independent of the value of m, i. e. C_2 is $O(n)$. Since the number of the chosen α cuts, m, determines the accuracy of the output Y_{LWA}, the proposed algorithm is able to achieve a high accuracy and a lower computational overhead simultaneously.

5.5 Numerical Study

This section will present a comparative study of the computational performance of the KM iterative algorithm, the EKM iterative algorithm and the three proposed algorithms through numerical simulations. Since computing a LWA is equivalent to computing two FWAs, the numerical studies focus on the FWA. In the following numerical studies, the platform is a Thinkpad X201i laptop with Intel CPU I5 and 4 GB memory, running Matlab 7.0.

In the numerical study, the parameters of the T1 FSs X_i and W_i, $i = 1, 2, \cdots, n$ were randomly generated by Monte Carlo method. In each simulation, the MFs of X_i and W_i, $i = 1, 2, \cdots, n$ were uniformly distributed. In addition, the range of the primary variable of X_i and W_i were constrained to the range $[0, 10]$ and $[0, 1]$, respectively. The number of X_i and W_i, n, are chosen as 20, 60 and 100. To access how the required accuracy affect the computational overhead for the FWA operator, the number of α-cuts, m, was increased by 5 from 5 to 20, ($m = 5, 6, , 7, \cdots, 20$) 100, and then was increased in step of 5 from 25 to 50 ($m = 25, 30, 35, \cdots, 50$) in the simulation. For each m, 100 Monte Carlo simulations were done. The mean and the standard deviation (STD) of the number of iterations and the computational time were recorded in Fig. 5.11-5.14 to evaluate the computational performance of the KM iterative algorithm (KMA), the EKM algorithm (EKMA), and the three proposed algorithms (IKMA-b, IKMA-x, IKMA-s).

5.5.1 The Mean and STD of the Number of Iterations

Fig. 5.11 shows the mean and the standard deviation (STD) of the number of iterations. Fig. 5.11(a)-5.11(c) show the results for triangular-shaped X_i and W_i while Fig. 5.11(d)-5.11(f) show the results for gaussian-shaped X_i and W_i. The following can be observed from Fig. 5.11:

1. The average number of iterations using the KMA ,EKMA and the proposed algorithms approximately increase linearly as the number of the chosen α-cuts,m, increases, regardless of the value of n. However, the rates of the increase for the proposed algorithms are smaller than the KMA and EKMA. In addition, the mean number of iterations for the proposed algorithms are smaller than those for the KMA and EKMA. This is consistent with the theoretical analysis in Section IV that the number of iterations for the proposed algorithm do not rely on the number of m as heavily as the KMA and EKMA.

2. Among the three proposed algorithms, the IKMA-s method requires the smallest iterations. The computation reduction is especially significant when the number of X_i and W_i, n, is between 60 and 100. This indicates that the switch point initialized based on the switch point is closer to the actual switch point and thus unnecessary iterations can be avoided.

3. To investigate the dependence of the variation in the number of iterations needed by the proposed algorithms on the value of attributes X_i and weights W_i in a FWA operation, the standard deviation (STD) of the number of iterations needed for different values of X_i and W_i is used as a criteria. When

$n = 20$, the STD for the KMA and the EKMA linearly increases as m increases while the STD for the three proposed algorithms are smaller and independent of the value of m. When the value of n is increased between 60 and 100, the STD for the proposed algorithm, especially the IKMA-s, are still smaller than the KMA and EKMA. This reveals that the number of iterations for the KMA and EKMA heavily rely on the position and parameters of T1 FSs X_i and W_i while the proposed algorithms do not, primarily because the proposed initialization strategy can predict an initial switch point which is close to the actual one, regardless of the parameters of X_i and W_i.

4. A comparative observation of Fig. 5.11(a)-5.11(c) and Fig. 5.11(d)-5.11(f) reveals that the mean and STD of iterations for triangular-shaped T1 FSs representing X_i and W_i are similar to those for gaussian-shaped ones when the proposed algorithms are applied.

Fig. 5.12 shows the percentage of the mean of the number of iterations reduced by the proposed algorithms over the average number of iterations taken by the KMA and the EKMA. From Fig. 5.12, it can be found that the proposed algorithms can achieve a roughly 60% reduction in the number of iterations required by the KMA and roughly 40% of those needed by the EKMA when m, is sufficiently large. Among the three proposed algorithms, the IKMA-s achieves the largest reduction percentage, regardless of the number and the shape of X_i and W_i, indicating the superiority of the initialization method employed by the IKMA-s.

5.5.2 The Mean and STD of the Computational Time

The proposed algorithms have been shown to require less iterations. However, less iterations may not necessarily result in less computational overhead as the calculations and mathematical manipulation in the initialization may be relatively high. To further examine the superiority of the proposed algorithms, this subsection focuses on using computational time as a criteria to compare the computational overhead of the KMA, the EKMA and the proposed algorithms. Fig. 5.13 shows the mean and STD of the computational time for the KMA, the EKMA and the proposed algorithms. An observation of Fig. 5.13 reveals that

1. Firstly, the mean computational time for the KMA, the EKMA and the proposed algorithms is approximately proportional to the number of the chosen α cuts, m. The growth rates of the computational time for the proposed algorithm are smaller than those for KMA and EKMA. This may be attributed to smaller iterations required by the proposed algorithms due to the proposed initialization method and termination condition. Among the proposed three algorithms, the IKMA-s takes the least computational time, primarily because it takes the smallest number of iterations shown in Fig. 5.11.

2. Secondly, the STD of the computational time for KMA and EKMA algorithms linearly increase as m increases while those for the proposed algorithms slightly vary. More importantly, the STD for the proposed algorithms are smaller than the KMA and EKMA. As a result, the iterations required by the proposed algorithms does not rely on the position or parameters of

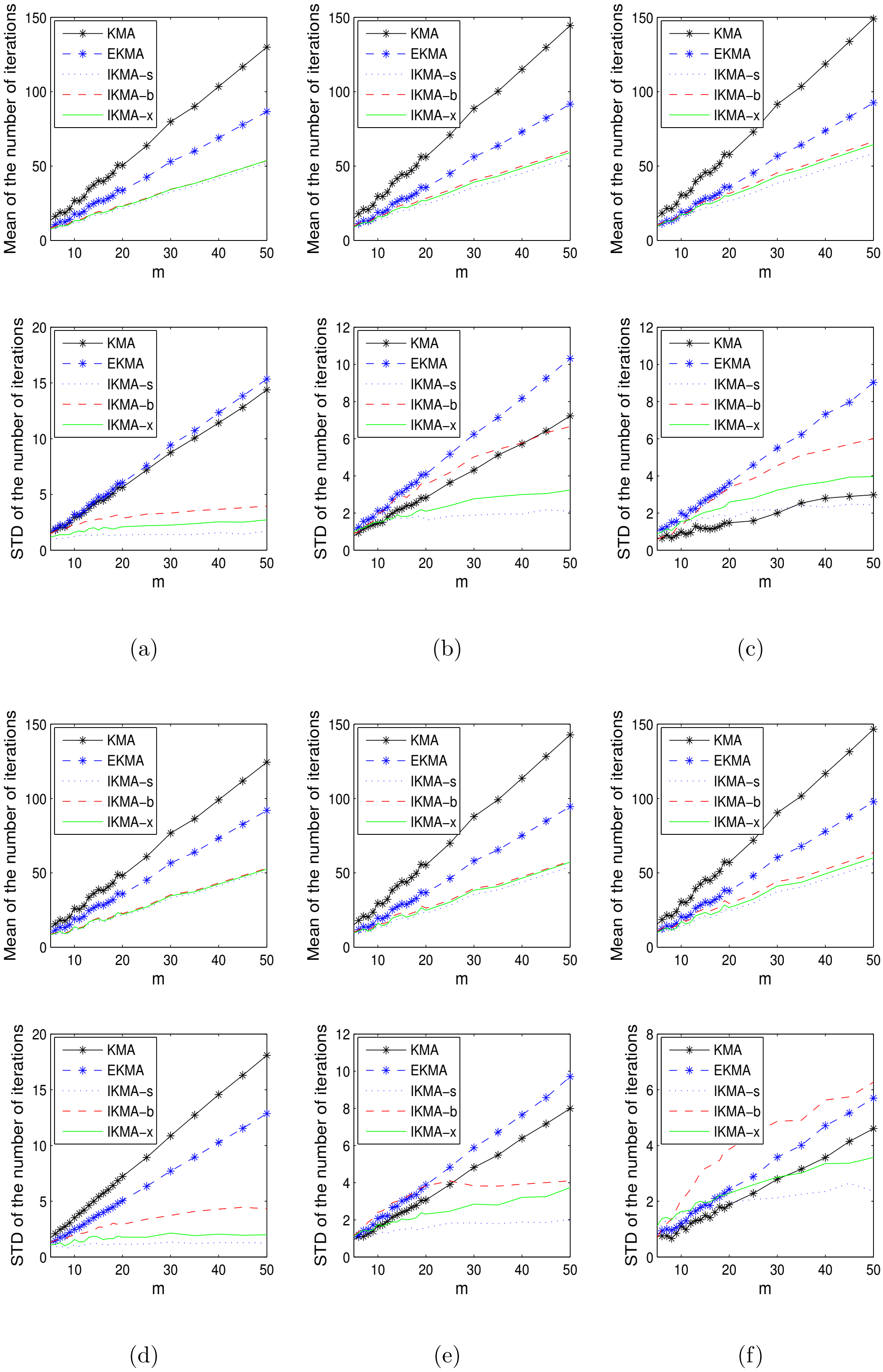
X_i and W_i. Among the three algorithms, the STD of the iterations for the IKMA-s is slightly smaller.

3. Lastly, when the proposed algorithms are employed, the mean and STD of the computational time for triangular X_i and W_i are similar to those for gaussian T1 FSs X_i and W_i, indicating that the computational performance of the proposed algorithms are independent of the shape of X_i and W_i.

Fig. 5.14 shows that the percentage the reduction of the average computational time by the proposed algorithm over the KMA and EKMA. An observation of Fig. 5.14 reveals that among the three proposed algorithms, the IKMA-s achieves the largest computational reduction and the IKMA-s reduces about 60% computational time required by the KMA and 40% computational time required by the EKMA.

5.6 Conclusion

This chapter proposed three algorithms for performing FWA/LWA that optimize the initialization and termination of the KM/EKM iterative algorithm. The three proposed initialization strategies reduce the number of iterations required in the search process by initializing the search process using the endpoint, the variable and the switch point of the computations for an adjacent α-cut. On the other hand, the proposed termination condition reduces the computational time required in the last iteration by avoiding the step of updating the switch point which is a necessary step to identify the end of iterations in the KM/EKM algorithm. Numerical experiments show that the three proposed algorithms can



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.11: Mean of the number of iterations: Triangle T1 FSs X_i and W_i (a) $n = 20$ (b) $n = 60$ (c) $n = 100$; Gaussian T1 FSs X_i and W_i (d) $n = 20$ (e) $n = 60$ (f) $n = 100$

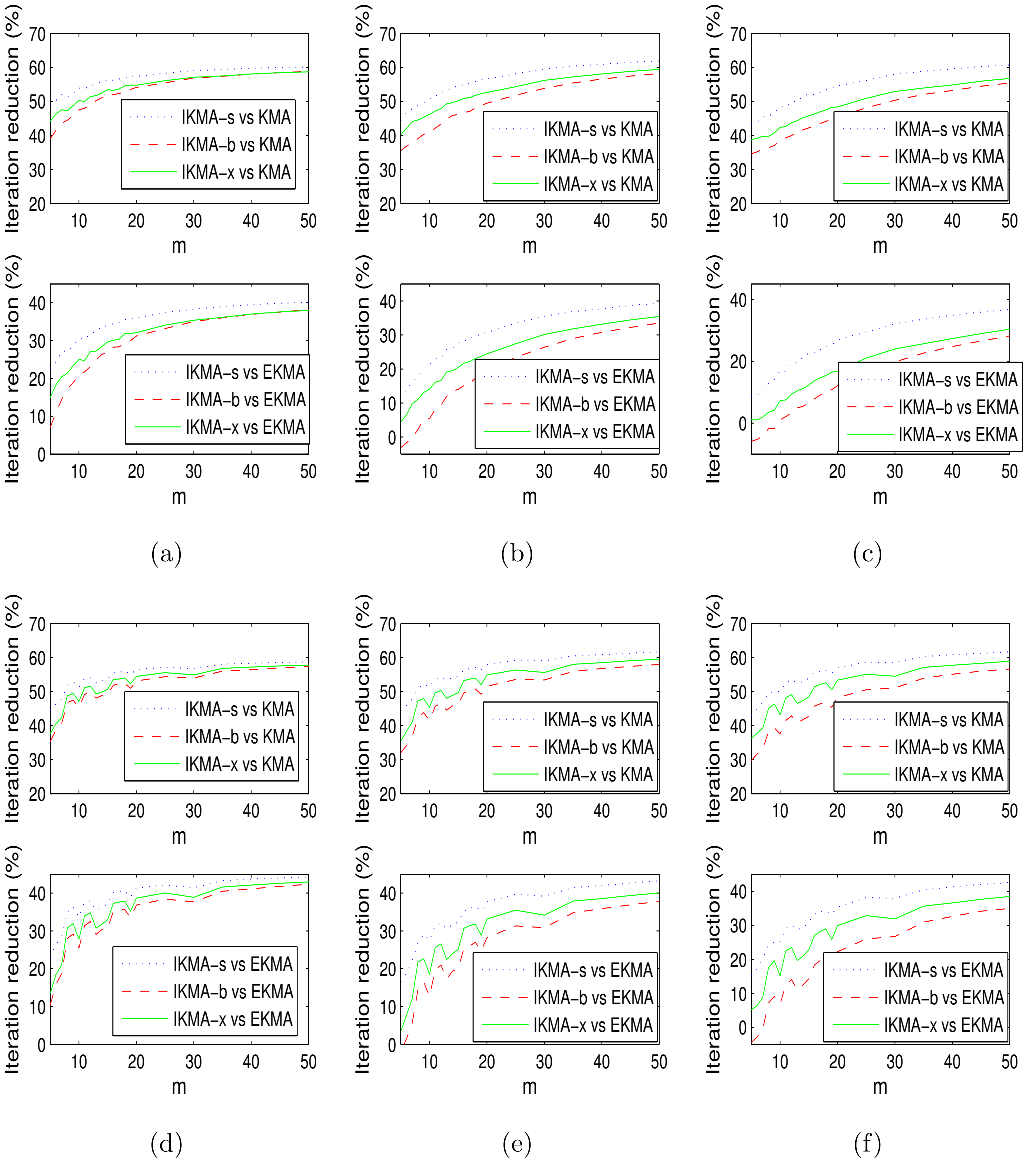
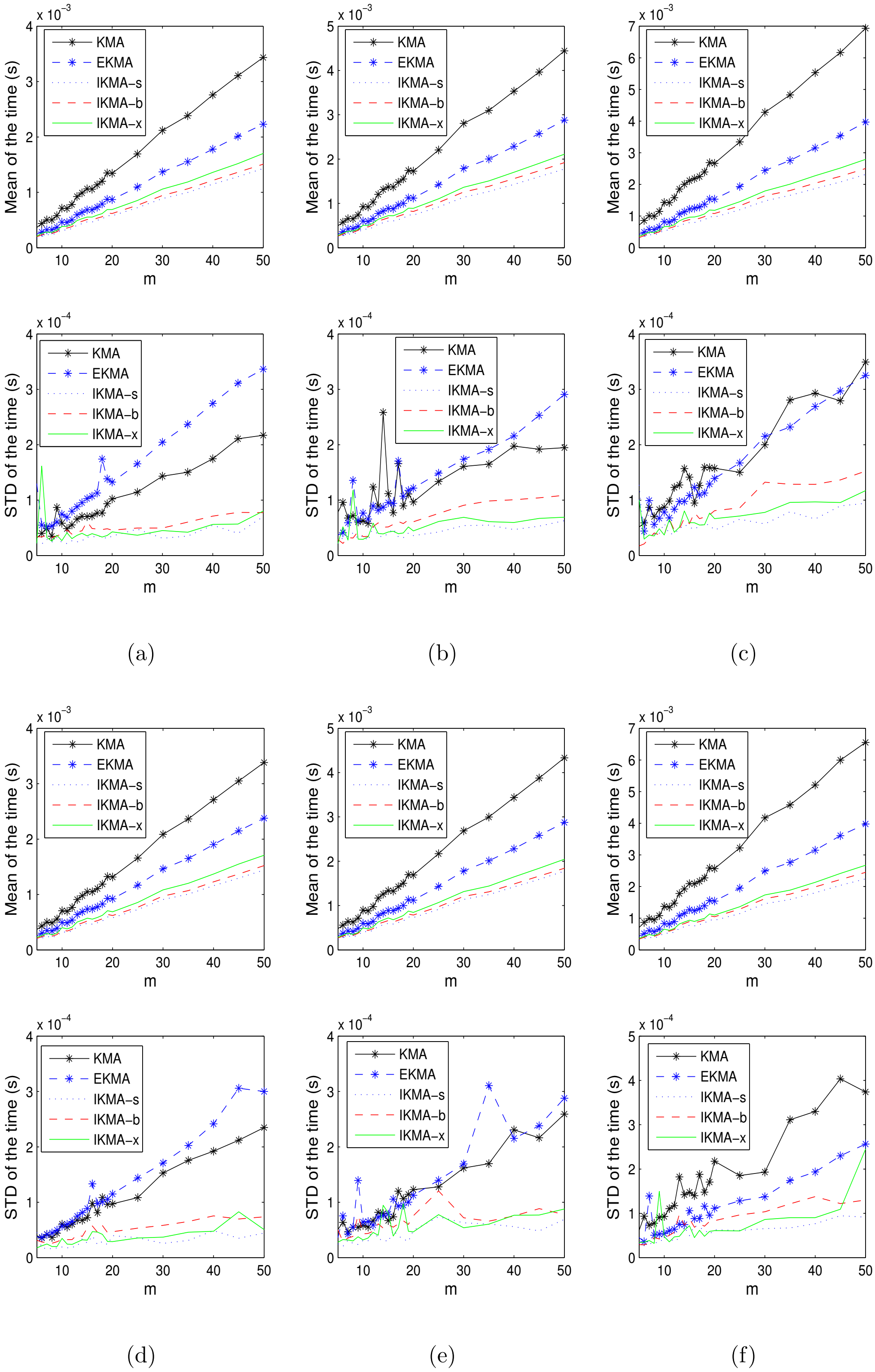


Figure 5.12: Iteration reduction: Triangle T1 FSs X_i and W_i (a) $n = 20$ (b) $n = 60$ (c) $n = 100$; Gaussian T1 FSs X_i and W_i (d) $n = 20$ (e) $n = 60$ (f) $n = 100$.



(a)

(b)

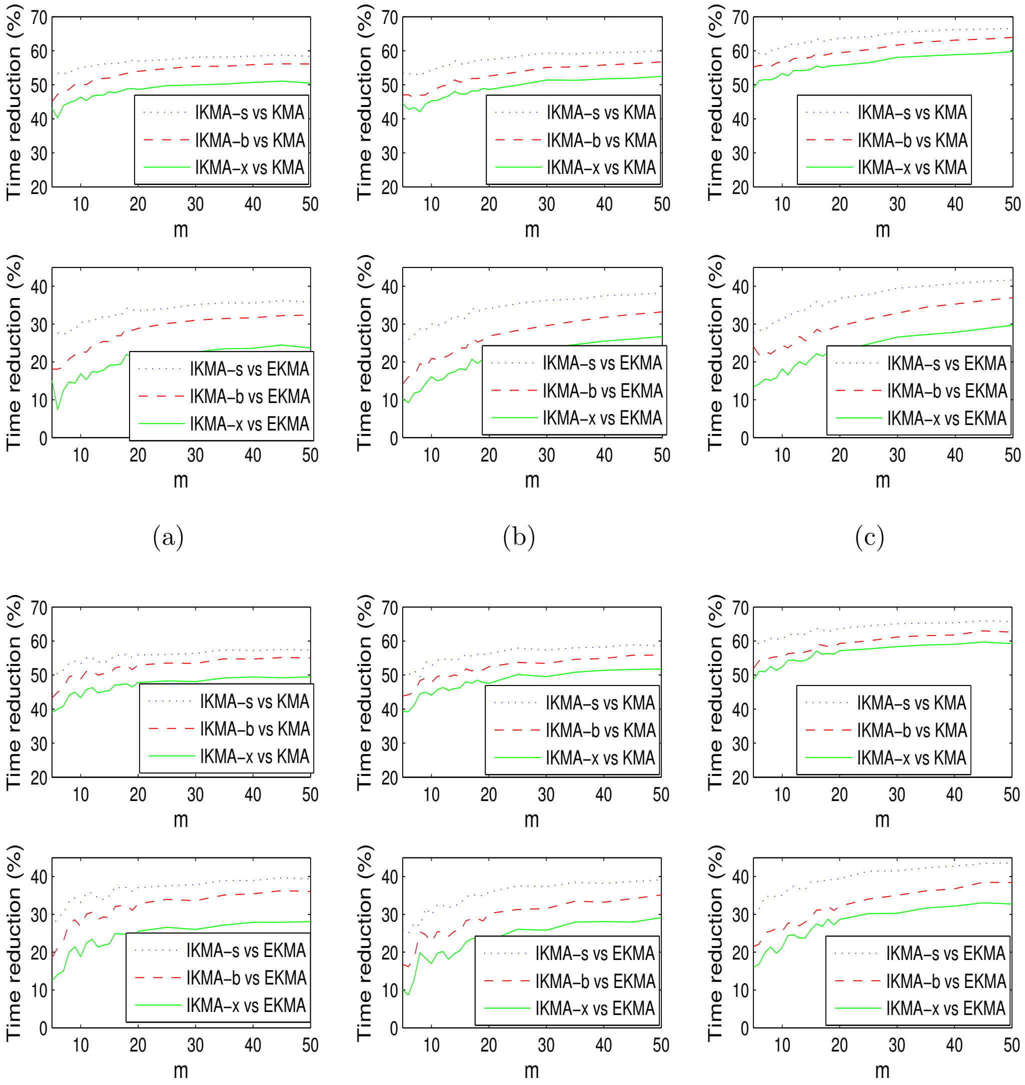
(c)

(d)

(e)

(f)

Figure 5.13: Mean and STD of the computational time: Triangle T1 FSs X_i and W_i (a) $n = 20$ (b) $n = 60$ (c) $n = 100$; Gaussian T1 FSs X_i and W_i (d) $n = 20$ (e) $n = 60$ (f) $n = 100$.



(a) (b) (c)

(d) (e) (f)

Figure 5.14: Computational time reduction: Triangle T1 FSs X_i and W_i (a) $n = 20$ (b) $n = 60$ (c) $n = 100$; Gaussian T1 FSs X_i and W_i (d) $n = 20$ (e) $n = 60$ (f) $n = 100$.

reduce the number of iterations required in the search process and the computational time required by the FWA and LWA operator. Among the three proposed algorithms, the proposed algorithm which employs the strategy of initializing the search process using the switch point of the adjacent α-cut requires the smallest computational time, achieving an approximately 60% reduction in computational time of the KM iterative algorithm and an approximately 40% reduction of the EKM iterative algorithm.

Chapter 6

Conclusions and Future work

6.1 Conclusions

Although a large number of publications on interval type-2 fuzzy logic have emerged, many questions on interval type-2 fuzzy logic still remain unsolved. The challenges that impede the study of interval type-2 fuzzy logic primarily arise from operations implemented using the Karnik-Mendel(KM) iterative algorithm or the enhanced Karnik-Mendel (EKM) iterative algorithm. To be specific, there are two major challenges. Firstly, there lack closed from expressions for the operations implemented using the KM iterative algorithm and thus it is challenging to perform theoretical analysis. Secondly, these operations are computationally intensive due to the iterative nature of the KM algorithm and thus they are not suitable for practical implementation in real systems. In view of these challenging issues, this thesis delved into the the operations implemented using the KM iterative algorithm: interval type-2 fuzzy controller using the KM type-reducer, fuzzy weighted average and linguistic weighted average.

In chapter 3, a class of symmetric interval type-2 fuzzy PI/PD controller using the KM type-reducer was investigated. To overcome the limitation of no closed form expression relating the output and the firing strength of the system, the input-output relationship of the symmetric interval type-2 fuzzy controllers was established by dividing the input space into parts and deriving the mathematical expression for each subregion. These mathematical expressions lay the foundation for the theoretical study of interval type-2 fuzzy logic controller. A larger number of the partitions for the symmetric interval type-2 fuzzy controller indicates that it has the potential to outperform type-1 fuzzy logic controller. By comparing the derived expressions with those of its type-1 counterpart, four properties were identified to address the potential advantage of the interval type-2 fuzzy controller over the type-1 fuzzy controller. These properties revealed that the interval type-2 fuzzy controller could better alleviate the compromise between faster response and smaller overshoot by providing larger control efforts and smaller overshoot simultaneously. Results from numerical experiments conducted to verify these results are consistent with the theoretical analysis.

Although the results in Chapter 3 shed some lights on the potential advantage of interval type-2 fuzzy controller, these results are limited to the symmetric interval type-2 fuzzy PD/PI controller. To generalize these results, a class of interval type-2 fuzzy PD/PI controller with more general configurations was studied in Chapter 4. Using the strategy of dividing the whole input space into parts presented in Chapter 3, the mathematical expressions relating the output and the inputs of the interval type-2 fuzzy PD/PI controller were established. These mathematical equations provide a platform for theoretical study of more general interval type-2 fuzzy controllers. By comparing the derived partitions of the input space and the corresponding expressions with those of the symmetric interval type-2 fuzzy controller and its type-1 counterpart, two aspects of results were derived. Firstly, it have been shown that the four properties for the symmetric interval type-2 fuzzy controller still hold true for general interval type-2 fuzzy controller. Secondly, the biggest advantage offered by an interval type-2 fuzzy controller over the symmetric one is the introduction of another class of subregion with unique characteristics to enable the interval type-2 fuzzy controller to better balance the compromise between fast rise time and smaller overshoot. These results revealed how the configurations of an interval type-2 fuzzy controller affect its characteristics and the control performance.

Another obstacle limiting the usefulness of the KM iterative algorithm is the intensive computational overhead, especially when the algorithm is used to compute the fuzzy weighted average (FWA) and linguistic weighted average (LWA). Chapter 5 proposed three algorithms to perform FWA and LWA in which the initialization and the termination of the KM iterative algorithm/ the EKM iterative algorithm are optimized. Three strategies of optimizing the initialization were proposed to reduce the number of iterations required by the search process by initializing using the endpoint, variable and the switch point from the previous computation for an adjacent α-cut. In addition, a better termination condition was proposed to reduce the required computational overhead by avoiding the step of updating the switch point in the last iteration which is a necessary step to identify the end of iterations in the KM iterative algorithm. Numerical studies verified that the three proposed algorithms can reduce the computational overhead required by the KM iterative algorithm / the EKM iterative algorithm. In particular, the proposed algorithm which employs the strategy of initializing the search process using the switch point available for an adjacent α cut needs the least computational time, achieving an approximately 60% reduction in the computational time of the KM iterative algorithm and an approximately 40% reduction of the EKM iterative algorithm. The large amount of reduction in the computational overhead by applying the proposed algorithms may enhance the application of FWA and LWA.

6.2 Future work

It should be noted that the derived mathematical expressions in Chapter 3 and Chapter 4 are limited to interval type-2 fuzzy controller with specific configurations such as symmetrical antecedent sets, equally spaced consequent sets. A possible extension of this work is to derive the mathematical input-output relationship for interval type-2 fuzzy controllers with more complex configurations. Whether the properties presented in Chapter 3 and Chapter 4 still hold true for more general interval type-2 fuzzy logic controller is unknown. Hence, it would be interesting to examine whether these properties still hold true or to establish whether there are any other unique properties. The mathematical equations for interval type-2 fuzzy controller lay the foundation for its theoretical study. Thus, it would be helpful to provide guidelines for designing interval type-2 fuzzy logic controller through these theoretical study tools.

It also should be noted that the strategies of optimizing the initialization of the

KM/EKM iterative algorithm in Chapter 5 may not be limited to the computation of FWA and LWA. These strategies may be applied to reduce the computational overhead of other operators or systems implemented using the KM /EKM iterative algorithm. Another direction of the future work is to develop more efficient strategies to further improve the computational efficacy of the KM/EKM iterative algorithm.

Appendix A

Proof of Theorem 3.1

For a T1 FLS using the center-of-sets defuzzification method, the defuzzified output y_c can be expressed as

$$y_c = \frac{\sum_{i=1}^{m} f_i y_i}{\sum_{i=1}^{m} f_i} \tag{A.1}$$

where $f_i, i = 1, 2, \cdots, m$ is the firing strength of the ith rule and y_i is the centroid of the corresponding consequent set. When all consequent sets of the T1 FLS are shifted horizontally by δ, i.e. $y_i \to y_i + \delta$, the defuzzified output y_c^* of the resulting T1 FLS can be computed as

$$y_c^* = \frac{\sum_{i=1}^{m} f_i(y_i + \delta)}{\sum_{i=1}^{m} f_i} = \frac{\sum_{i=1}^{m} f_i y_i}{\sum_{i=1}^{m} f_i} + \delta = y_c + \delta \tag{A.2}$$

The above results may be extended for IT2 FLSs that use the Karnik-Mendel (KM) type-reduction method. By employing the wavy slice representation to interpret the IT2 FS generated by the inference engine, the output of the KM type-reducer is the interval set $[y_l, y_r]$.

$$y_l = \min\{y_e^j, j = 1, 2 \cdots, v\}$$

$$y_r = \max\{y_e^j, j = 1, 2 \cdots, v\}$$

$$y_e^j = \frac{\sum_{i=1}^m f_i' y_i}{\sum_{i=1}^m f_i'}, j = 1, 2 \cdots, v$$

where y_e^j is an embedded IT2 FS of the output of the fuzzy inference process, and f_i' is any value in the firing set $[\underline{f}_i, \overline{f}_i]$. Suppose all consequent sets of an IT2 FLS are shifted by the amount δ, i.e. $y_i \rightarrow y_i + \delta$. Then the centroid y_j^* of an embedded IT2 FS that form the output of the new IT2 FLS is $y_e^j + \delta$. Since shifting all consequent sets by the amount of δ causes the centroids of all embedded IT2 FLS to shift by the same amount, the type-reduced set formed by these centroids will be $[y_l + \delta, y_r + \delta]$. As a result, the new defuzzified output y^* of the IT2 FLS will be $y^* = y + \delta$.

Appendix B

Proof of Property 2-4 of the non-symmetric IT2 fuzzy PD controller

This appendix aims to examine whether Property 2-4 of the symmetric IT2 FLC still hold true for the non-symmetric IT2 fuzzy PD controller by comparing the analytical structure of the IT2 fuzzy PD controller, derived in Section 4.3, with its T1 counterpart. Fig. B.1(a) and B.1(b) show the antecedent sets of the T1 FLC for inputs $E(n)$ and $R(n)$ used in this study. They are constructed by replacing every IT2 FS with a T1 FS such that both the IT2 and T1 FLC have the same input space. Fig. B.2 shows the input space is partitioned into four subregions before the T1 FLS can be equivalently expressed as nonlinear PD controller as follows:

$$\triangle u_{jICh} = k_p^h E(n) + k_d^h R(n) \tag{B.1}$$

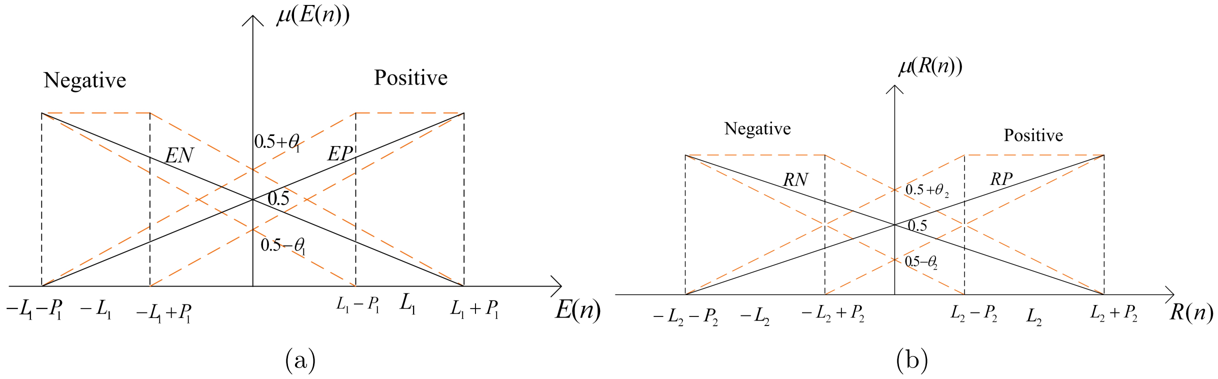


Figure B.1: (a) T1 FSs EN and EP (solid lines) as antecedent sets for the input $E(n)$. (b)T1 FSs RN and RP (solid lines) as antecedent sets for the input $R(n)$.)

where the proportional gain k_p^h and the derivative gain k_d^h are listed in Table B.1.

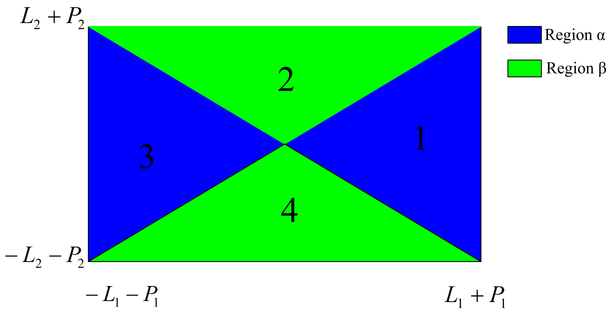


Figure B.2: The partitions of the input space by T1 FLS

B.1 Proof of Property 2

The symmetry of the antecedent sets and consequent sets allows IC 6 and 7 to be taken as an example to illustrate this property. The observation are

$$
\begin{aligned}
K_p^7 &= \frac{H_2+H_1}{8L_1(1-\theta_1)+4E(n)} + \frac{H_2+H_1}{8L_1(1-\theta_1)+4E(n)} \\
&> \frac{H_2+H_1}{8L_1(1+\theta_1)+4E(n)} = K_p^{24} \\
K_p^6 &= \frac{H_2+H_1}{8L_1(1-\theta_2+\theta_1)+4E(n)} + \frac{H_2+H_1}{8L_1(1-\theta_1)+4E(n)} \\
&> \frac{H_2+H_1}{8L_1(1-\theta_2+\theta_1)+4E(n)} = K_p^{23}
\end{aligned}
$$

Table B.1: The output of the T1 fuzzy PD controller $\triangle u_j(n)$ for IC 1 to IC 4

IC No.	$\triangle u_j(n)$
1	$\dfrac{L_1(1+2\theta_1)(H_1-H_2)R(n)-L_2(1+2\theta_2)(H_3+H_4)E(n)}{-2L_2(1+2\theta_2)E(n)+4L_1L_2(1+2\theta_1)(1+2\theta_2)}$
2	$\dfrac{-L_1(1+2\theta_1)(H_2+H_4)R(n)+L_2(1+2\theta_2)(H_1-H_3)E(n)}{-2L_1(1+2\theta_1)R(n)+4L_1L_2(1+2\theta_1)(1+2\theta_2)}$
3	$\dfrac{L_1(1+2\theta_1)(H_3-H_4)R(n)+L_2(1+2\theta_2)(H_1+H_2)E(n)}{2L_2(1+2\theta_2)E(n)+4L_1L_2(1+2\theta_1)(1+2\theta_2)}$
4	$\dfrac{L_1(1+2\theta_1)(H_1+H_3)R(n)+L_2(1+2\theta_2)(H_2-H_4)E(n)}{2L_1(1+2\theta_1)R(n)+4L_1L_2(1+2\theta_1)(1+2\theta_2)}$

This shows that in the direction of $E(n)$, IC 6 and 7 exhibits bigger slope than IC 23 and 24, respectively.

B.2 Proof of Property 3

It is sufficient to consider IC 11, 12, 13 (subset of IC 1 for the T1 controller) due to the symmetry of the two fuzzy PD controllers. The relationships between the proportional gains of the IT2 fuzzy PD controller and the T1 FLC are

$$\begin{aligned}
K_p^{11} &= \frac{-(H_3+H_4)}{8L_1(1-\theta_1+\theta_2)-4E(n)} + \frac{-(H_3+H_4)}{8L_1(1+\theta_1-\theta_2)-4E(n)} \\
&> \frac{-(H_3+H_4)}{8L_1-4E(n)} + \frac{-(H_3+H_4)}{8L_1-4E(n)} = \frac{-H_3-H_4}{4L_1-2E(n)} > \frac{-H_3-H_4}{4L_1(1+2\theta_1)-2E(n)} = k_p^1 \\
K_p^{12} &= \frac{-(H_3+H_4)}{8L_1(1-\theta_1)-4E(n)} + \frac{-(H_3+H_4)}{8L_1(1+\theta_1-\theta_2)-4E(n)} \\
&> \frac{-(H_3+H_4)}{8L_1(1-\theta_1)-4E(n)} + \frac{-(H_3+H_4)}{8L_1(1+\theta_1)-4E(n)} > \frac{-H_3-H_4}{4L_1-2E(n)} \\
&> \frac{-H_3-H_4}{4L_1(1+2\theta_1)-2E(n)} = k_p^1 \\
K_p^{13} &= \frac{-(H_3+H_4)}{8L_1(1-\theta_1)-4E(n)} + \frac{-(H_3+H_4)}{8L_1(1+\theta_1)-4E(n)} > \frac{-H_3-H_4}{4L_1-2E(n)} = k_p^1
\end{aligned}$$

The above inequalities show the proportional gain of IT2 fuzzy PD controller in IC 11, 12 and IC 13 is bigger than its T1 counterpart.

B.3 Proof of Property 4

Due to the symmetry in the IT2 fuzzy PD controller and its T1 counterpart, the output value property for IC 1–IC 4, and IC 6 – IC 13 will be illustrated by comparing the equivalent gains expressions for IC 1– IC 3 (subset of IC 2 of the T1 counterpart). From Table 4.3, 4.2 and B.1, the relationship between the outputs of the two fuzzy PD controllers can be established as

$$
\begin{aligned}
&\triangle U_{jIC1} = [\frac{L_2(H_1-H_3)}{8L_1L_2(1+\theta_2)-4L_1R(n)} + \frac{L_2(H_1-H_3)}{8L_1L_2(1-\theta_2)-4L_1R(n)}]E(n) \\
&+[\frac{-(H_2+H_4)}{8L_2(1+\theta_2)-4R(n)} + \frac{-(H_2+H_4)}{8L_2(1-\theta_2)-4R(n)}]R(n)+\delta_1 \\
&= [\frac{1}{8L_1L_2(1+\theta_1)-4L_2E(n)} + \frac{1}{8L_1L_2(1-\theta_1)-4L_2E(n)}] \\
&\qquad \times[-(H_2+H_4)L_1R(n)+(H_1-H_3)L_2E(n)]+\delta_1
\end{aligned}
$$

$$
\begin{aligned}
&\triangle u_{jIC1} = \frac{-L_1(1+2\theta_1)(H_2+H_4)R(n)+L_2(1+2\theta_2)(H_1-H_3)E(n)}{4L_1L_2(1+2\theta_1)(1+2\theta_2)-2L_1(1+2\theta_1)R(n)} \\
&= \frac{1}{4L_1L_2(1+2\theta_1)-2L_1R(n)}[-(H_2+H_4)L_1R(n)+(H_1-H_3)L_2E(n)] \\
&= (\frac{1}{8L_1L_2(1+2\theta_1)-4L_1R(n)} + \frac{1}{8L_1L_2(1+2\theta_1)-4L_1R(n)}) \\
&\qquad \times[-(H_2+H_4)L_1R(n)+(H_1-H_3)L_2E(n)]
\end{aligned}
$$

Because

$$
\begin{aligned}
\frac{1}{8L_1L_2(1-\theta_1)-4L_2E(n)} &> \frac{1}{8L_1L_2(1+2\theta_1)-4L_2E(n)} > 0 \\
\frac{1}{8L_1L_2(1+\theta_1)-4L_2E(n)} &> \frac{1}{8L_1L_2(1+2\theta_1)-4L_2E(n)} > 0
\end{aligned}
$$

and $\delta_1 > 0$, $-(H_2+H_4)L_1R(n)+(H_1-H_3)L_2E(n) > 0$ in IC 1, then $\triangle U_{jIC1} > \triangle u_{jIC2} > 0$ and their difference increases as θ_1 increases because θ_1 appears in the denominator.

Another observation is

$$\begin{aligned}\triangle U_{jIC3} &= \frac{L_2(H_1-H_3)}{4L_1L_2-2L_1R(n)}E(n)+\frac{-(H_2+H_4)}{4L_2(1+\theta_2)-2R(n)}R(n)\\ &= \frac{1}{4L_1L_2-2L_1R(n)}[-(H_2+H_4)L_1R(n)+(H_1-H_3)L_2E(n)]\end{aligned}$$

Since $L_1R(n)+L_2E(n)>0$ in IC 3 and

$$\frac{1}{4L_1L_2-2L_2R(n)} > \frac{1}{4L_1L_2(1+2\theta_1)-2L_2E(n)} > 0$$

then $\triangle U_{jIC1}$ and $\triangle u_{jIC2}$ have the same sign and $\triangle U_{jIC1}$ is larger than $\triangle u_{jIC2}$ in magnitude and their difference increases as θ_1 increases because θ_1 appears in the denominator. For any input pair in the region IC 2 of the IT2 fuzzy controller, the value calculated using the left endpoint of IC 2 is equivalent to that of IC3 and the value calculated using the right endpoint of IC 2 is larger than that of IC3. Hence, the value calculated using $\triangle U_{jIC2}$ is larger than that using $\triangle U_{jIC3}$ for any input pair of IC2 and thus it follows $\triangle U_{jIC2} > \triangle u_{jIC2} > 0$ for any input pair in IC 2 of the IT2 fuzzy controller. Furthermore, the difference increases as θ_1 and θ_2 increase.

Appendix C

Proof of Theorem 5.1 and Theorem 5.2

C.1 Proof of Theorem 5.1

For the FWA operation defined in (5.1), the output Y_{FWA} may completely constructed based on $f_L(\alpha_j)$ and $f_R(\alpha_j)$ defined in (5.12) and (5.13), respectively. (5.12) and (5.13) reveal that the left and right bound $f_L(\alpha_j)$ and $f_R(\alpha_j)$ are continuous functions of $c_i(\alpha_j)$, $d_i(\alpha_j)$, $a_i(\alpha_j)$ and $b_i(\alpha_j)$. For continuous X_i and W_i, these four variables $c_i(\alpha_j)$, $d_i(\alpha_j)$, $a_i(\alpha_j)$ and $b_i(\alpha_j)$ are continuous functions about α_j. Hence, $f_L(\alpha_j)$ and $f_R(\alpha_j)$ are continuous about α_j and thus the output fuzzy set Y_{FWA} has a continuous membership function.

C.2 Proof of Theorem 5.2

This theorem can be proved by showing $Y_{FWA}(\alpha_{j+1}) \subseteq Y_{FWA}(\alpha_j)$ where $j = 1, 2, \cdots, m-1$, i.e. $f_L(\alpha_j) < f_L(\alpha_{j+1})$ and $f_R(\alpha_j) > f_R(\alpha_{j+1})$.

1. From (5.12), the left bounds of the α_j cut and the α_{j+1} cut in (5.12) can be written as

$$f_L(\alpha_j) = \min_{\forall w_i \in [c_i(\alpha_j), d_i(\alpha_j)]} \frac{\sum_{i=1}^n w_i a_i(\alpha_j)}{\sum_{i=1}^n w_i} \tag{C.1}$$

$$f_L(\alpha_{j+1}) = \min_{\forall w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]} \frac{\sum_{i=1}^n w_i a_i(\alpha_{j+1})}{\sum_{i=1}^n w_i} \tag{C.2}$$

Since $[c_i(\alpha_j), d_i(\alpha_j)]$ are an α-cut of W_i, the weights in the above equations satisfy $[c_i(\alpha_{j+1}), d_i(\alpha_{j+1})] \subseteq [c_i(\alpha_j), d_i(\alpha_j)]$. Hence,

$$\begin{aligned} f_L(\alpha_j) &= \min_{\forall w_i \in [c_i(\alpha_j), d_i(\alpha_j)]} \frac{\sum_{i=1}^n w_i a_i(\alpha_j)}{\sum_{i=1}^n w_i} \\ &< \min_{\forall w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]} \frac{\sum_{i=1}^n w_i a_i(\alpha_j)}{\sum_{i=1}^n w_i} \end{aligned} \tag{C.3}$$

Because in the above inequality a_i always appear in the nominator and $a_i(\alpha_j) < a_i(\alpha_{j+1})$, for any $w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]$

$$\frac{\sum_{i=1}^n w_i a_i(\alpha_j)}{\sum_{i=1}^n w_j} < \frac{\sum_{i=1}^n w_i a_i(\alpha_{j+1})}{\sum_{i=1}^n w_i} \tag{C.4}$$

Hence,

$$\begin{aligned} f_L(\alpha_j) &< \min_{\forall w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]} \frac{\sum_{i=1}^n w_i a_i(\alpha_j)}{\sum_{i=1}^n w_i} \\ &< \min_{\forall w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]} \frac{\sum_{i=1}^n w_i a_i(\alpha_{j+1})}{\sum_{i=1}^n w_i} = f_L(\alpha_{j+1}) \end{aligned} \tag{C.5}$$

2. According to (5.13), the right bounds of the α_j cut and the α_{j+1} cut can be described as

$$f_R(\alpha_j) = \max_{\forall w_i \in [c_i(\alpha_j), d_i(\alpha_j)]} \frac{\sum_{i=1}^n w_i b_i(\alpha_j)}{\sum_{i=1}^n w_i}$$

$$f_R(\alpha_{j+1}) = \max_{\forall w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]} \frac{\sum_{i=1}^n w_i b_i(\alpha_{j+1})}{\sum_{i=1}^n w_i}$$

Since $[c_i(\alpha_{j+1}), d_i(\alpha_{j+1})] \subseteq [c_i(\alpha_j), d_i(\alpha_j)]$, the following inequality can be derived:

$$\begin{aligned} f_R(\alpha_j) &= \max_{\forall w_i \in [c_i(\alpha_j), d_i(\alpha_j)]} \frac{\sum_{i=1}^n w_i b_i(\alpha_j)}{\sum_{i=1}^n w_i} \\ &> \max_{\forall w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]} \frac{\sum_{i=1}^n w_i b_i(\alpha_j)}{\sum_{i=1}^n w_i} \end{aligned} \tag{C.6}$$

Because in the nominator $b_i(\alpha_j) > b_i(\alpha_{j+1})$, the above inequality can be written as

$$\begin{aligned} f_R(\alpha_j) &> \max_{\forall w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]} \frac{\sum_{i=1}^n w_i b_i(\alpha_j)}{\sum_{i=1}^n w_i} \\ &> \max_{\forall w_i \in [c_i(\alpha_{j+1}), d_i(\alpha_{j+1})]} \frac{\sum_{i=1}^n w_i b_i(\alpha_{j+1})}{\sum_{i=1}^n w_i} = f_R(\alpha_{j+1}) \end{aligned} \tag{C.7}$$

Author's Publications

The author has contributed to the following publications:

Journal Papers

[1] M. Nie and W. W. Tan, "Stable Adaptive Fuzzy PD plus PI controller for Nonlinear Uncertain Systems", *Fuzzy Sets and Systems*, vol. 179. pp. 1-19, 2011.

[2] M. Nie and W. W. Tan, "Analytical Structure and Characteristics of Symmetric Karnik-Mendel Type-Reduced Interval Type-2 Fuzzy PI and PD Controllers", accepted by *IEEE Trans. Fuzzy Syst.*

[3] M. Nie and W. W. Tan, "Improved algorithms for Fuzzy Weighted Average and Linguistic Weighted Average", manuscript submitted to *IEEE Trans. Fuzzy Syst.*

[4] M. Nie and W. W. Tan, "Analytical Structure and Characteristics of a class of Karnik-Mendel Type-Reduced Interval Type-2 Fuzzy PI and PD Controllers", manuscript submitted to *IEEE Trans. Fuzzy Syst.*

[5] M. Nie and W. W. Tan,"The Nie-Tan Type-reduction Method for Interval Type-2 Fuzzy Set and Interval Type-2 Fuzzy Logic System", manuscript

submitted to *IEEE Trans. Fuzzy Syst.*

Conference Papers

[1] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," *Proc. FUZZ-IEEE Conference*, Hong Kong, 2008.

[2] M. Nie and W. W. Tan, "Extension of fuzzy adaptive laws to IT2 fuzzy systems," *Proc. FUZZ-IEEE Conference*, Korea, 2009.

[3] M. Nie and W. W. Tan, "Derivation of the analytical structure of symmetrical IT2 fuzzy PD and PI controllers," *Proc. FUZZ-IEEE Conference*, Barcelona, 2010.

[4] M. Nie and W. W. Tan, "Derivation of the analytical structure of a class of IT2 fuzzy PD and PI controllers," *Proc. FUZZ-IEEE Conference*, Taipei, 2011.

Bibliography

[1] R. H. Abiyev and O. Kaynak. Type 2 fuzzy neural structure for identification and control of time-varying plants. *IEEE Trans. Ind. Electron.*, 57(12):4147–4159, 2010.

[2] H. A. Amin and H. Ying. Structural and stability analysis of fuzzy controllers with nonlinear input fuzzy sets with relation to nonlinear PID control with variable gains. *Automatica*, 40:1551–1559, 2004.

[3] A. Baai, M.J. Castro-Sitiriche, and A. R. Ofoli. Design and implementation of parallel fuzzy PID controller for high-performance brushless motor drives: An integrated environment for rapid control prototyping. *IEEE Trans. Ind. Applicat.*, 44:1090–1098, 2008.

[4] C. Bartolomeo and G. Mose. Type-2 fuzzy control of a bioreactor. In *Proc. IEEE Int. Conf. Intell. Comput. and Intell. Syst.*, pages 700–704, Cairo, Egypt, 2009.

[5] D. Ben-Arieh and Z. Chen;. Linguistic-labels aggregation and consensus measure for autocratic decision making using group recommendations. *IEEE Trans. Syst., Man, Cybern. A*, 36:558–568, 2006.

[6] M. Biglarbegian, W. W. Melek, and J. M. Mendel. On the stability of interval type-2 tsk fuzzy logic control systems. *IEEE Trans. Syst., Man, Cybern. B*, 40(3):798–818, 2010.

[7] M. Biglarbegian, W. W. Melek, and J. M. Mendel. Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: Theory and experiments. *IEEE Trans. Ind. Electron.*, 58(4):1371–1384, 2011.

[8] M. Biglarbegiana, W. Meleka, and J. M. Mendel. On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling. *Inf. Sci.*, 181(7):1325–1347, 2010.

[9] J. J. Buckley and H. Ying. Linear fuzzy controller: It is a linear non-fuzzy controller. *Inf. Sci.*, 51:183–192, 1990.

[10] O. Castillo, L. Aguilar, N. Cazarez, and S. Cardenas. Systematic design of a stable type-2 fuzzy logic controller. *Appl. soft comput.*, 8:1274–1279, 2008.

[11] P. Chang, K. Hung, K. Lin, and C. Chang. A comparison of discrete algorithms for fuzzy weighted average. *IEEE Trans. Fuzzy Syst.*, 14:663–675, 2006.

[12] H. Chaoui and W. Gueaieb. Type-2 fuzzy logic control of a flexible-joint manipulator. *J. Intell. Robot. Syst.: Theory Appl.*, 51:159–186, 2008.

[13] W. M. Dong and F. S. Wong. Fuzzy weighted averages and implementation of the extension principle. *Fuzzy Sets Syst.*, 21(2):183–199, 1987.

[14] X. Du and H. Ying. Control peformance comparison between a type-2 fuzzy controller and a comparable conventional mamdani fuzzy controller. In *Proc.*

of North Amer. Fuzzy Inf. Process. Soc. Conf., pages 100–105, San Diego, CA, 2007.

[15] 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.*, 8:802–814, 2010.

[16] Herrera-Viedma E., S. Alonso, F. Chiclana, and F.; Herrera. A consensus model for group decision making with incomplete fuzzy preference relations. *IEEE Trans. Fuzzy Syst.*, 15(5):863–877, 2007.

[17] G. Feng. A survey on analysis and design of model-based fuzzy control systems. *IEEE Trans. Fuzzy Syst.*, 14(5):676–697, 2006.

[18] M. M. Gupta. A survey of process control applications of fuzzy set theory. In *Proc. IEEE Int. Conf. Dec. and Contr.*, pages 1454–1461, 1978.

[19] H. Hagras. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Trans. Fuzzy Syst.*, 12:524–539, 2004.

[20] H. Hagras, F. Doctor, V. Callaghan, and A. Lopez. An incremental adaptive life long learning approach for type-2 fuzzy embedded agents inambient intelligent environments. *IEEE Trans. Fuzzy Syst.*, 15:41–55, 2007.

[21] S. Han and J. M. Mendel. Evaluating location choices using perceptual computer approach. In *Proc. IEEE Int. Conf. Fuzzy Syst*, pages 1–8, 2010.

[22] M. Y. Hsiao, T. H. S. Li, J. Z. Lee, C. H. Chao, and S. H. Tsai. Design of interval type-2 fuzzy sliding-mode controller. *Inf. Sci.*, 178:1696–1716, 2008.

[23] Y. Huang and S. Yasunobu. A general practical design method for fuzzy PID control from conventional PID control. In *Proc. IEEE Int. Conf. Fuzzy Syst*, pages 969–972, 2000.

[24] V. N. Huynh, Y. Nakamori, and J. Lawry. A probability-based approach to comparison of fuzzy numbers and applications to target-oriented decision making. *IEEE Trans. Fuzzy Syst.*, 16:371–387, 2008.

[25] E. A. Jammeh, M. Fleury, C. Wagner, H. Hagras, and M. Ghanbari. Interval type-2 fuzzy logic congestion control for video streaming across IP networks. *IEEE Trans. Fuzzy Syst.*, 17:1123–1142, 2009.

[26] C. Jing, J. Yang, and R. Ding. Fuzzy weighted average filter. In *Proc. IEEE Int. Conf. Signal Processing*, pages 525–528, 2000.

[27] C. Kao and S. T. Liu. Competitiveness of manufacturing firms: an application of fuzzy weighted average. *IEEE Trans. Fuzzy Syst.*, 29:661–667, 1999.

[28] N. N. Karnik and J. M. Mendel. Introduction to type-2 fuzzy logic systems. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 915–920, 1998.

[29] N. N. Karnik and J. M. Mendel. Type-2 fuzzy logic systems: type-reduction. In *Proc. IEEE Int. Conf. Syst., Man, and Cybern.*, pages 2046–2051, 1998.

[30] N. N. Karnik and J. M. Mendel. Applications of type-2 fuzzy logic systems: handling the uncertainty associated with surveys. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 1546–1551, 1999.

[31] N. N. Karnik and J. M. Mendel. Applications of type-2 fuzzy logic systems to forecasting of time-series. *Inf. Sci.*, 120:89–111, 1999.

[32] N. N. Karnik and J. M. Mendel. Centroid of a type-2 fuzzy set. *Inf. Sci.*, 132(5):195–220, 2001.

[33] N. N. Karnik and J. M. Mendel. Operations on type-2 fuzzy sets. *Fuzzy Sets and Syst.*, 122(1):327–348, 2001.

[34] N. N. Karnik, J. M. Mendel, and Q. Liang. Type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.*, 7(6):643–658, 1999.

[35] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications.* Upper Saddle River, NJ, 1995.

[36] C. Kobashikawa, Y. Hatakeyama, F. Dong, and K. Hirota. Fuzzy algorithm for group decision making with participants having finite discriminating abilities. *IEEE Trans. Syst., Man, Cybern. A*, 39(1):86–95, 2009.

[37] H. K. Lam and L. D. Seneviratne. Stability analysis of interval type-2 fuzzy-model-based control systems. *IEEE Trans. Syst., Man, Cybern. B*, 38(3):617–628, 2008.

[38] C. S. Lee and M. H. Wang. A fuzzy expert system for diabetes decision support application. *IEEE Trans. Syst., Man, Cybern. B*, 41(1):139–153, 2011.

[39] D. H. Lee and D. Park. An efficient algorithm for fuzzy weighted average. *Fuzzy Sets Syst.*, 87(1):39–45, 1997.

[40] R. Liao, H. Zheng, and S. Grzybowski. An integrated decision-making model for condition assessment of power transformers using fuzzy approach and evidential reasoning. 26:1111–1118, 2011.

[41] F. J. Lin and P. H. Chou. Adaptive control of two-axis motion control system using interval type-2 fuzzy neural network. *IEEE Trans. Ind. Electron.*, 56(1):178–193, 2009.

[42] P. Z. Lin, C. F. Hsu, and T. T. Lee. Type-2 fuzzy logic controller design for buck DC-DC converters. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 365–370, Reno, NV, 2005.

[43] T. S. Liou and M. J. Wang. Fuzzy weighted average: An improved algorithm. *Fuzzy Sets Syst.*, 49(3):307–315, 1992.

[44] F. Liu and J. M. Mendel. Aggregation using the fuzzy weighted average as computed by the karnik-mendel algorithms. *IEEE Trans. Fuzzy Syst.*, 16:1–12, 2008.

[45] F. Liu and J. M. Mendel. Encoding words into interval type-2 fuzzy sets using an interval approach. *IEEE Trans. Fuzzy Syst.*, 16(6):1503–1521, 2008.

[46] X. Liu and J. M. Mendel. ‘connect karnik-mendel algorithms to root-finding for computing the centroid of an interval type-2 fuzzy set. *IEEE Trans. Fuzzy Syst.*, 2011.

[47] Z. Liu, Y. Zhang, and Y. Wang. A type-2 fuzzy switching control system for biped robots. *IEEE Trans. Syst., Man, Cybern. C*, 37:1202–1213, 2007.

[48] J. Lu, J. Ma, G. Zhang, Y. Zhu, X. Zeng, and L. Koehl. Theme-based comprehensive evaluation in new product development using fuzzy hierarchical criteria group decision-making method. *IEEE Trans. Ind. Electron.*, 58:2236–2246, 2011.

[49] E. H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. In *Proc. of the Instit. of Electrical Engineers*, pages 1585–1588, 1974.

[50] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Mach. Studies*, 7(1):1–13, 1975.

[51] G.K.I. Mann, B. G. Hu, and R. G. Gosine. Analysis of direct action fuzzy PID controller structures. *IEEE Trans. Syst., Man, Cybern. B*, 29:371–388, 1999.

[52] M. Marimin, M. Umano, I. Hatono, and H. Tamura. Hierarchical semi-numeric method for pairwise fuzzy group decision making. *IEEE Trans. Syst., Man, Cybern. B*, 32:691–700, 2002.

[53] F. Mata, L. Martinez, and E. Herrera-Viedma. An adaptive consensus support model for group decision-making problems in a multigranular fuzzy linguistic context. *IEEE Trans. Fuzzy Syst.*, 17(2):279–290, 2009.

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

[55] J. M. Mendel. Computing derivatives in interval type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.*, 12(1):84–98, 2004.

[56] J. M. Mendel. On a 50% savings in the computation of the centroid of a symmetrical interval type-2 fuzzy set. *Inf. Sci.*, 172(3):417–430, 2005.

[57] J. M. Mendel. Advances in type-2 fuzzy sets and systems. *Inf. Sci.*, 177:84–110, 2007.

[58] J. M. Mendel. Computing with words and its relationship with fuzzistics. *Inf. Sci.*, 177(4):988–1006, 2007.

[59] J. M. Mendel. A quantitative comparison of interval type-2 and type-1 fuzzy logic systems: First results. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 1–8, 2010.

[60] J. M. Mendel and R. I. John. Type-2 fuzzy sets made simple. *IEEE Trans. Fuzzy Syst.*, 10:117–127, 2002.

[61] J. M. Mendel and F. Liu. Super-exponential convergence of the karnik-mendel algorithms for computing the centroid of an interval type-2 fuzzy set. *IEEE Trans. Fuzzy Syst.*, 15:309–320, 2007.

[62] J. M. Mendel, F. Liu, and D. Zhai. α-plane representation for type-2 fuzzy sets: Theory and applications. *IEEE Trans. Fuzzy Syst.*, 17:1189–1207, 2009.

[63] J. M. Mendel and H. Wu. Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems. *IEEE Trans. Fuzzy Syst.*, 14(6):781–792, 2006.

[64] J. M. Mendel and H. Wu. New results about the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule. *Inf. Sci.*, 177(2):360–377, 2007.

[65] J. M. Mendel and H. Wu. Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Forward problems. *IEEE Trans. Fuzzy Syst.*, 15(5):916–930, 2007.

[66] J. M. Mendel and H. Wu. Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 2, inverse problems. *IEEE Trans. Fuzzy Syst.*, 15(2):301–308, 2007.

[67] C. P. Pappis and E. H. Mamdani. A fuzzy logic controller for a traffic junction. *IEEE Trans. Syst., Man, Cybern. A*, 17:707–717, 1977.

[68] W. Pedrycz and M. Song. Analytic hierarchy process (AHP) in group decision making and its optimization with an allocation of information granularity. *IEEE Trans. Fuzzy Syst.*, 19:527–539, 2011.

[69] M. Petrov, I. Ganchev, and A. Taneva. Fuzzy PID control of nonlinear plants. In *Proc. IEEE Int. Conf. Intell. syst.*, pages 30 – 35, 2002.

[70] A. Rubaai, M. J. Castro-Sitiriche, and A.R. Ofoli. DSP-based laboratory implementation of hybrid fuzzy-PID controller using genetic optimization for high-performance motor drives. *IEEE Trans. Ind. Applicat.*, 44(6):1977–1986, 2008.

[71] R. Sepulveda, O.Castillo, P. Melin, A. Rodriguez-Diaz, and O. Montiel. Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. *Inf. Sci.*, 117:2023–2048, 2007.

[72] G. H. Shakouri and M.B. Menhaj. A systematic fuzzy decision-making process to choose the best model among a set of competing models. *IEEE Trans. Syst., Man, Cybern. A*, 38:1118–1128, 2008.

[73] N. K. Sinha and J. D. Wright. Application of fuzzy control to a heat exchanger system. In *Proc. IEEE Int. Conf. Dec. and Contr.*, pages 1424–1428, 1977.

[74] W. W. Tan and J. Lai. Development of a type-2 fuzzy proportional controller. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 1305–1310, Budapest, Hungary, 2004.

[75] K. Tanaka, T. Ikeda, and H.O. Wang. Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stabilizability, h control theory, and linear matrix inequalities. *IEEE Trans. Fuzzy Syst.*, 4(1):1–13, 1996.

[76] C. W. Tao, J. Taur, C. C. Chuang, C. W. Chang, and Y. H. Chang. An approximation of interval type-2 fuzzy controllers using fuzzy ratio switching type-1 fuzzy controllers. *IEEE Trans. Syst., Man, Cybern. B*, 41(3):828–839, 2011.

[77] B. Chen; X. Liu; S. Tong. Adaptive fuzzy output tracking control of MIMO nonlinear uncertain systems. *IEEE Trans. Fuzzy Syst.*, 15(2):287–300, 2007.

[78] C. S. Tseng and B. S. Chen. Robust fuzzy observer-based fuzzy control design for nonlinear discrete-time systems with persistent bounded disturbances. *IEEE Trans. Fuzzy Syst.*, 17(3):711–723, 2009.

[79] C. Wagner and H. Hagras. Toward general type-2 fuzzy logic systems based on zslices. *IEEE Trans. Fuzzy Syst.*, 18:637–660, 2010.

[80] J. W. Wang, H. N. Wu, and H. X. Li. Distributed fuzzy control design of nonlinear hyperbolic PDE systems with application to nonisothermal plug-flow reactor. *IEEE Trans. Fuzzy Syst.*, 19(3):514–526, 2011.

[81] L. X. Wang. *A Course in Fuzzy Systems and Control.* Upper Saddle River, NJ, 1997.

[82] W. J. Wang, Y. J. Chen, and C. H. Sun. Relaxed stabilization criteria for discrete-time T-S fuzzy control systems based on a switching fuzzy model and piecewise lyapunov function. *IEEE Trans. Syst., Man, Cybern. B*, 37(3):551–559, 2007.

[83] A. Wu and P. K. S. Tam. Stable fuzzy neural tracking control of a class of unknown nonlinear systems based on fuzzy hierarchy error approach. *IEEE Trans. Fuzzy Syst.*, 10(6):779–789, 2002.

[84] D. Wu and J. M. Mendel. Aggregation using the linguistic weighted average and interval type-2 fuzzy sets. *IEEE Trans. Fuzzy Syst.*, 15(6):1145–1161, 2007.

[85] D. Wu and J. M. Mendel. Enhanced karnik-mendel algorithms. *IEEE Trans. Fuzzy Syst.*, 17:923–934, 2009.

[86] D. Wu and J. M. Mendel. Computing with words for hierarchical decision making applied to evaluating a weapon system. *IEEE Trans. Fuzzy Syst.*, 18(3):441–460, 2010.

[87] D. Wu and J. M. Mendel. On the continuity of type-1 and interval type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.*, 19(1):179–192, 2011.

[88] D. Wu and W. W. Tan. A type-2 fuzzy logic controller for the liquid-level process. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 953–958, Budapest, Hungary, 2004.

[89] D. Wu and W. W. Tan. Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers. *Eng. Appl. Artif. Intell.*, 19:829–841, 2006.

[90] D. Wu and W. W. Tan. A simplified type-2 fuzzy controller for real-time control. *ISA Trans.*, 45(4):503–516, 2007.

[91] D. Wu and W. W. Tan. Interval type-2 fuzzy pi controllers: Why they are more robust. In *Proc. IEEE Int. Conf. Granr. and Comput.*, pages 802–807, 2010.

[92] 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.*, 10(5):622–639, 2002.

[93] Z. S. Xu. Multiple-attribute group decision making with different formats of preference information on attributes. *IEEE Trans. Syst., Man, Cybern. B*, 37(6):1500–1511, 2007.

[94] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Syst., Man, Cybern.*

[95] R. R. Yager and D.P. Filev. Induced ordered weighted averaging operators. *IEEE Trans. Syst., Man, Cybern. B*, 29(2):141–150, 1999.

[96] C. Yeh, W. R. Jeng, and S. Lee. An enhanced type-reduction algorithm for type-2 fuzzy sets. *IEEE Trans. Fuzzy Syst.*, 19(2):227–240, 2011.

[97] H. Ying. The simplest fuzzy contollers using different inference methods are different nonlinear proportinal-integral controller with variable gains. *Automatica*, 26(6):1579–1589, 1993.

[98] H. Ying. Practical design of nonlinear fuzzy controllers with stability analysis for regulating processes with unknown mathematical models. *Automatica*, 30(7):1185–1195, 1994.

[99] H. Ying. An analytical study on structure, stability and design of general takagi-sugeno fuzzy control systems. *Automatica*, 34:1617–1623, 1998.

[100] H. Ying. Constructing nonlinear variable gain controllers via the takagi-sugeno fuzzy control. *IEEE Trans. Fuzzy Syst.*, 6:226–234, 1998.

[101] H. Ying. Analytical analysis and feedback linearization tracking control of the general takagi-sugeno fuzzy dynamic systems. *IEEE Trans. Syst., Man, Cybern. C*, 29:290–298, 1999.

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

[103] H. Ying. A general technique for deriving analytical structure of fuzzy controllers that use arbitrary trapezoidal/triangular input fuzzy sets and zadeh fuzzy logic AND operator. *Automatica*, 39:1171–1184, 2003.

[104] H. Ying. Deriving analytical input-output relationship for fuzzy controllers using arbitrary input fuzzy sets and zadeh fuzzy AND operator. *IEEE Trans. Fuzzy Syst.*, 14(5):654–662, 2006.

[105] H. Ying, W. Siler, and J. J. Buckley. Fuzzy control theory: a nonlinear case. *Automatica*, 26(3):513–520, 1990.

[106] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci.*, 8:199–249, 1975.

[107] H. Zhang and X. Xie. Relaxed stability conditions for continuous-time T-S fuzzy-control systems via augmented multi-indexed matrix approach. *IEEE Trans. Fuzzy Syst.*, 19(3):478–492, 2011.

[108] X. X. Zhang, H. X. Li, and C. K. Qi. Spatially constrained fuzzy-clustering-based sensor placement for spatiotemporal fuzzy-control system. *IEEE Trans. Fuzzy Syst.*, 18(5):946–957, 2010.

[109] F. Zheng, Q. G. Wang, and T. H. Lee. Output tracking control of MIMO fuzzy nonlinear systems using variable structure control approach. *IEEE Trans. Fuzzy Syst.*, 10(6):686–697, 2002.