Universidad de Granada

EXTENDING THE CONCEPTS OF TYPE-2 FUZZY LOGIC AND SYSTEMS

Presented by

GONZALO RUIZ GARCÍA

To apply for the
INTERNATIONAL PHD DEGREE BY
UNIVERSIDAD DE GRANADA

Advisors
IGNACIO ROJAS RUIZ
HÉCTOR POMARES CINTAS

Doctoral Program in
INFORMATION AND COMMUNICATION TECHNOLOGIES

Signed: Gonzalo Ruiz García

May 2017

Editor: Universidad de Granada. Tesis Doctorales
Autor: Gonzalo Ruiz García
ISBN: 978-84-9163-485-0
URI: http://hdl.handle.net/10481/48265

VISTO BUENO

El **Prof. Dr. D. Ignacio Rojas Ruiz**, Catedrático de Universidad, y el profesor **Prof. Dr. D. Héctor Pomares Cintas**, Catedrático de Universidad, del Departamento de Arquitectura y Tecnología de los Computadores de la Universidad de Granada,

CERTIFICAN:

Que la memoria titulada:

"Extending the Concepts of Type-2 Fuzzy Logic and Systems"

ha sido realizada por **D. Gonzalo Ruiz García** bajo nuestra dirección en el Departamento de Arquitectura y Tecnología de los Computadores de la Universidad de Granada para optar al grado de **Doctor en Ingeniería de Telecomunicación**.

En Granada, a 21 de Mayo de 2017.

Los Directores de la tesis doctoral:

Fdo. Ignacio Rojas Ruiz y Héctor Pomares Cintas

DECLARACIÓN

El doctorando Gonzalo Ruiz García y los directores de la tesis Ignacio Rojas Ruiz y Héctor Pomares Cintas garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, Mayo de 2017

Gonzalo Ruiz García	Ignacio Rojas Ruiz	Héctor Pomares Cintas

A mi hermana, el único y auténtico amor de mi vida.

ABSTRACT

The work presented in this dissertation is a contribution to the field of fuzzy sets and fuzzy logic systems theory.

Firstly, we will approach the controversial discussion that has been effusively debated among scholars of fuzzy logic for many years. Some authors argue that the ability of type-2 fuzzy logic systems to perform better than their type-1 counterparts relies on the higher number of parameters they need to be defined. On the other hand, other authors pose the argument that this ability is due to how those parameters are used, and how type-2 fuzzy sets model uncertainty in a more suitable way. Although other previous works have tackled this discussion, we propose a new approach based on a function approximation framework, using type-1 fuzzy logic systems with a varying number of parameters. This part of the work aims to support the previous findings related to this topic, and justify the further research on type-2 fuzzy sets and fuzzy logic systems in the rest of the dissertation.

Secondly, after shedding some light on the previous discussion, we will focus on the development of the theory about type-2 fuzzy sets and fuzzy logic systems. Traditionally, although type-2 fuzzy logic has proven to perform better than type-1, its use has been somehow limited. One of those reasons has been the limitation to operate with those sets; although the operations of intersection and union on these sets were defined at the same time that type-2 fuzzy sets themselves, the operations were computationally intensive, and closed formulas were only available for type-2 fuzzy sets having normal and convex secondary grades. The main contribution of this work to the fuzzy sets theory is to provide two new theorems for the intersection and union operations, regardless of the specific shape of the sets' secondary grades.

Those new theorems, which allow us to operate on type-2 fuzzy sets having non-convex secondary grades, are the keystone to further developing the theory of interval type-2 fuzzy logic systems. Interval type-2 fuzzy sets have been recently shown to be more general than interval-valued fuzzy sets, and can actually have non-convex secondary grades. Hence, a whole new theory needs to be developed in order to provide those fuzzy logic systems with the appropriate theoretical framework; we aim to do so in the last part of this dissertation.

RESUMEN

El trabajo presentado en esta tesis es una contribución en el ámbito de la teoría de conjuntos y sistemas difusos.

En primer lugar, se abordará una discusión controvertida y polémica que se ha debatido intensamente a lo largo de los años por los estudiosos de la lógica difusa. Algunos autores argumentan que la capacidad de los sistemas difusos tipo-2 para obtener un mejor rendimiento que sus homólogos tipo-1 subyace en el mayor número de parámetros que requieren para ser definidos. Por otra parte, otros autores exponen que dicha capacidad se debe a cómo se utilizan dichos parámetros, y cómo los conjuntos difusos tipo-2 modelan la incertidumbre de forma más adecuada. A pesar de que otros trabajos previos también han abordado esta problemática, en este documento se propone una nueva aproximación a la misma, basada en un marco de problemas de aproximación funcional, usando sistemas difusos tipo-1 con un número variable de parámetros. Esta parte del trabajo tiene por objetivo apoyar los resultados previos relacionados con esta temática, y justificar así la investigación posterior sobre conjuntos y sistemas difusos tipo-2 durante el resto de esta tesis.

En segundo lugar, tras arrojar algo de luz a la discusión anterior, nos centraremos en el desarrollo de la teoría de conjuntos y sistemas basados en lógica difusa de tipo-2. Tradicionalmente, y a pesar de las pruebas fehacientes del mejor rendimiento de los sistemas tipo-2 frente a los tipo-1, su uso ha sido, en cierto modo, limitado. Una de las razones ha sido la limitación para operar con estos conjuntos; a pesar de que las operaciones de intersección y unión para estos conjuntos fueron definidas a la vez que estos, dichas operaciones son computacionalmente muy costosas, y sólo se disponía de fórmulas cerradas para conjuntos difusos de tipo-2 con grados secundarios normales y convexos. La principal contribución de este trabajo a la teoría de conjuntos difusos es proporcionar dos nuevos teoremas para las operaciones de intersección y unión, independientemente de la forma específica de los grados secundarios.

Esos dos nuevos teoremas, que permiten operar sobre conjuntos difusos tipo-2 con grados secundarios no convexos, son la pieza clave para desarrollar un paso más la teoría de sistemas difusos intervalo tipo-2. Se ha demostrado recientemente que los conjuntos difusos intervalo tipo-2 son más generales que los conjuntos difusos con valores de intervalo. Por tanto, se hace nece-

sario desarrollar una nueva teoría al completo para proporcionar a estos sistemas difusos un marco teórico apropiado; este es el objetivo de esta última parte de esta tesis.

There is nothing worse than a sharp image
of a fuzzy concept.

— Ansel Adams

ACKNOWLEDGMENTS

First of all, and risking not to be according to standards, I would like to thank myself for all the hard work done all these years. I am well aware this might be a bit egocentric, but I really feel grateful to myself, and I truly think I deserve a thank you. Good work buddy, you did it!

Secondly, I would like to thank both my supervisors, Ignacio Rojas and Héctor Pomares, for their guidance, good advise, wisdom and eternal support to accomplish this dissertation.

In third place, I would like to *effusively* thank Professor Hani Hagras; I don't really think he is completely aware of how determinant he has been to me, both in my research career and my future yet to come. Thank you so much for all the time, effort and interest you have dedicated to me during these couple of years. For all that, I'll be forever grateful, and it will never be forgotten.

Next, I'd like to thank all friends, work mates and colleagues that have been around all these years: Curro, Fergu, Antares, Nolo, Juanma, Paloma and Antonio from the office; to my friends, "Los Morsalinos", for being so incredibly awesome and close to me. You could really prevent me from thinking about work when I needed so. I love you all. And also thank you to all those friends I just forgot to mention here. You are in my heart too.

And last but not least (and probably the most important), I'd like to thank my family for just being around and letting me be myself. It's much easier to go on with all your support. Thank you, Piedad, for paying more attention to me than I deserve. And especially, I'd like to thank both my sister Cristina (the one and only love of my life) and my father Antonio Miguel, the two people who made me the man I am today. I don't think I could properly describe in any language how much I love you both.

Once again, just thank you.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI Adaptive Integration

AIFS Atanassov's Intuitionistic Fuzzy Sets

BADD Basic Defuzzification Distributions

BOA Bisector Of Area

CDD Constraint Decision Defuzzification

COA Centre Of Area

COG Centre Of Gravity

EA Evolutionary Algorithm

ECOA Extended Centre Of Area

EQM Extended Quality Method

FCD Fuzzy Clustering Defuzzification

FL Fuzzy Logic

FLC Fuzzy Logic Controller

FLS Fuzzy Logic System

FM Fuzzy Mean

FOM First Of Maximum

FS Fuzzy Set

GA Genetic Algorithm

gfIT2FLS General Forms of Interval Type-2 Fuzzy Logic Systems

gfIT2FS General Forms of Interval Type-2 Fuzzy Sets

GLSD Generalised Level Set Defuzzification

GS Grey Sets

HFS Hesitant Fuzzy Sets

ICOG Indexed Centre of Gravity

IE Inference Engine

IV Influence Value or Interval Valued

IVAIFS Interval-Valued Atanassov's Intuitionistic Fuzzy Sets

IVFS Interval-Valued Fuzzy Sets

KW Kruskal-Wallis

LFS L-Fuzzy Sets

LOM Last Of Maximum

MeOM Mean Of Maxima

MF Membership Function

MIMO Multiple-Input-Multiple-Output

MISO Multiple-Input-Single-Output

MOM Middle Of Maximum

nSFS Non-Stationary Fuzzy Sets

QI Quality Indicator

QM Quality Method

RCOM Random Choice Of Maximum

SG Secondary Grades

SLIDE Semi-Linear Defuzzification

SM Secondary Membership

SS Shadowed Sets

SVFS Set Valued Fuzzy Sets

T1 Type-1

T2 Type-2

T2FS Type-2 Fuzzy Set

TS Takagi-Sugeno

TSK Takagi-Sugeno-Kang

WFM Weighted Fuzzy Mean

Part I

INTRODUCTION AND FUNDAMENTALS

1 UNA REVISIÓN HISTÓRICA DE LA LÓGICA DIFUSA

La lógica difusa es errónea, errónea, y perniciosa. Lo que necesitamos es más pensamiento lógico, no menos. El peligro de la lógica difusa es que anima a ese tipo de pensamiento impreciso que tantos problemas nos ha traído. La lógica difusa es la cocaína de la ciencia.

— Profesor William Kahan (UC Berkeley)

1.1 EL INICIO DE LA LÓGICA DIFUSA

La lógica difusa se introdujo en el año 1965 de mano de L. A. Zadeh, en su propuesta inicial de la teoría de conjuntos difusos [117]. Este autor intentaba proporcionar un marco de referencia para representar conceptos que son imprecisos o poco claros en sí mismos [104]. A pesar de que su trabajo fue criticado con severidad por la comunidad académica (debido principalmente al énfasis de la teoría en la imprecisión), continuó su trabajo a lo largo de la siguiente década, con algunos de sus artículos de referencia como [116], [113], [114] y [115], entre otros. En estos trabajos se introdujeron los conceptos de los conjuntos difusos (y las nociones de su extensión), la lógica y la inferencia difusas, el razonamiento aproximado y los conceptos básicos de las variables linguísticas.

No obstante, muchos otros investigadores a lo largo y ancho del mundo se sintieron atraídos por su propuesta y comenzaron a expandir la teoría de la lógica, sistemas y controladores difusos, aplicándolos en numerosos y diferentes campos. En 1974, E. H. Mamdani presentó el primer controlador difuso [66], en contraposición con las metodologías clásicas de control basadas en un modelado matemático preciso. En 1977, Dubois presentó un estudio exhaustivo en su tesis doctoral de las condiciones del tráfico utilizando conjuntos difusos [28]. Más tarde, entre 1976 y 1987, la lógica difusa y sus aplicaciones a la industria atrajeron una especial atención en Japón, donde en la ciudad de Sempai se implementó un controlador basado en lógica difusa para controlar una de sus líneas de tren [104]. Dicho éxito en aplicaciones reales revivió el interés en la lógica difusa en Estados Unidos a finales de la década de los 80, llevando a lo que se conoce como "fuzzy boom" a lo largo de todo el mundo, atrayendo la atención de numerosos investigadores.

Una de las principales críticas sobre la lógica difusa está relacionada con la construcción de las funciones de pertenencia (para conocer los detalles sobre éstas, por favor véase el capítulo 2). Si se supone que los conjuntos difusos modelan la incertidumbre, entonces algunos autores argumentan que asignar un número concreto a un valor de pertenencia no contiene incertidumbre alguna. Dicha crítica llevó a Zadeh [10] a introducir la noción de los conjuntos difusos de tipo-N en [118] y, como caso particular, los conjuntos difusos de tipo-2, en los que los valores de pertenencia son conjuntos difusos en sí mismos. La definición formal de tales conjuntos fue finalmente presentada en 1975 en [113].

Puesto que algunos autores claman que, paradójicamente, la lógica difusa tipo-1 es incapaz de lidiar con la incertidumbre, su extensión tipo-2 empezó a ganar relevancia debido a su capacidad para manejarla, en el sentido de *modelar y minimizar sus efectos* [78]. Por consiguiente, numerosos autores comenzaron a estudiar su definición, sus propiedades y sus operaciones teóricas entre conjuntos: Mizumoto y Tanaka [80], Dubois y Prade [29] y Mendel y Karnik [46] centraron sus esfuerzos en la definición de los conjuntos difusos tipo-2, así como en las operaciones entre ellos.

El interés creciente en la lógica difusa, especialmente en su versión tipo-2, motivó a los investigadores a afrontar de muchas maneras diferentes el manejo y el modelado de la incertidumbre, surgiendo variadas y numerosas propuestas de otros conjuntos difusos: L-conjuntos difusos [34], conjuntos difusos de valores de conjunto [35], conjuntos difusos intuicionistas de Atanassov [4] (y, posteriormente, su versión en intervalo, conjuntos difusos intuicionistas de Atanassov intervalados, [3]), conjuntos grises [22], conjuntos sombreados [82], conjuntos difusos titubeantes [93], conjuntos difusos no estacionarios [33], entre otros. Una revisión histórica de los diferentes tipos de conjuntos difusos, algunas de sus propiedades y sus relaciones puede encontrarse en [10]. Sin embargo, estudiar profundamente estas variedades de conjuntos difusos está fuera de los objetivos de esta tesis; aquí nos centraremos en los conjuntos tipo-1, tipo-2 y en su versión más específica tipo-2 intervalo.

Centrándonos en la lógica difusa tipo-2, tal y como se mencionó anteriormente, en 1975 Zadeh introdujo la noción de conjunto difuso tipo-2. En 1976, Mizumoto y Tanaka estudiaron las operaciones teóricas entre conjuntos, las propiedades de los valores de pertenencia de tipo-2 y ambas operaciones de suma y producto algebraico. Posteriormente, a finales de la década de los 90, Karnik y Mendel extendieron dicho trabajo, y presentaron fórmulas cerradas útiles para la intersección, unión y negación sobre conjuntos difusos tipo-2 [46], [49]; también introdujeron el concepto de centroide tipo-2 [48]. Muchos otros artículos, centrados en estudiar la composición de relaciones tipo-2, se publicaron entre 1978 and 1999 [26], [29]. Todo este trabajo previo llevó a Liang y Mendel a centrar sus esfuerzos en el desarrollo de una teoría completa sobre sistemas difusos tipo-2 intervalo.

A pesar de que la teoría de la lógica difusa iba progresando, durante los inicios de la década del 2000 la lógica difusa tipo-2 intervalo atesoró la mayor parte de la atención por parte de los investigadores, en detrimento de la lógica difusa tipo-2 general (a pesar de que no hay una definición formal para esta última, el término es ampliamente utilizado para diferenciarla explícitamente de su versión más simple tipo intervalo). Esta diferencia estaba fundamentada debido a su alta complejidad computacional, especialmente en la operación de reducción de tipo (una operación intermedia necesaria en un sistema difuso para completar el mapeo entre las entradas y la salida). Esta diferencia justificó una clara escisión en la investigación de la lógica difusa: el progreso teórico se centró en el desarrollo de la lógica difusa de tipo-2, mientras que las aplicaciones prácticas de sistemas reales tendían a utilizar la versión más simple tipo intervalo.

1.2 OBJETIVOS Y MOTIVACIÓN

La breve introducción histórica sobre la lógica difusa y su desarrollo nos permite acercarnos apropiadamente a los objetivos y motivaciones de esta tesis, que detallamos a continuación.

El primer debate que atrae nuestra atención ha sido discutido desde la misma aparición de la lógica difusa tipo-2. En general, diversos trabajos y estudios han establecido [13] que la lógica difusa tipo-2 general y la tipo-2 intervalo son capaces de obtener mejor rendimiento que sus homólogos tipo-1, especialmente cuando se enfrentan a altos niveles de incertidumbre [31], [38], [59]; sin embargo, por qué ocurre este rendimiento superior aún permanece sin explicación. Este hecho llevó a una polémica bien conocida [88], [11], en la que algunos autores declaran que una razón por la que los sistemas difusos tipo-2 muestran un mejor rendimiento estriba en el uso de parámetros extra en las funciones de pertenencia en tipo-2 general e intervalo, cuando se comparan con los tipo-1.

Algunos autores han propuesto marcos de referencia anteriormente para comparar los sistemas difusos tipo-1 y tipo-2 [45], [88]; además, otros trabajos [11] proponen una comparación entre tipo-1, non-singleton tipo-1 y singleton tipo-2 intervalo en varios problemas de aproximación funcional, proporcionando a cada sistema el mismo número de parámetros (cuando es posible), y dando a todos los sistemas las mismas oportunidades de optimización utilizando algoritmos de búsqueda semi-aleatoria. Sin embargo, tal y como se explicará posteriormente en esta tesis, algunos de esos parámetros (por ejemplo, los consecuentes) se obtienen de diferentes formas en los distintos tipos de sistemas, y por tanto, podría argumentarse que las opciones de optimización no son *estrictamente* iguales, sino más bien *aproximadamente* iguales.

Motivados por esta polémica sin fin aparente, la primera parte de este trabajo trata de contribuir a la solución de este debate desde una perspectiva nueva y diferente: si la discusión gira en torno al número de parámetros libres o grados de libertad disponibles para ajustar el sistema, entonces se propone dar un paso atrás hacia los sistemas difusos tipo-1, y utilizar un marco de referencia basado en aproximación funcional para comparar este tipo de sistemas difusos, pero utilizando un número diferente de parámetros por función de pertenencia, y compararlos para verificar si existen diferencias estadísticamente significativas en su rendimiento. Este trabajo trata de descubrir si el número de parámetros es o no *la clave* para mejorar el rendimiento de los sistemas difusos y, por tanto, probar o refutar si hay otros factores involucrados (por ejemplo, cómo esos parámetros se utilizan en los sistemas tipo-1 y tipo-2).

Las conclusiones extraídas en este estudio justificarán plenamente que nos centremos en la teoría de conjuntos y sistemas difusos tipo-2 en la parte ii. En ese sentido, es necesario considerar dos factores importantes en la bibliografía existente sobre lógica difusa tipo-2 general e intervalo, que se mencionan a continuación.

En primer lugar, algunos trabajos recientes [89] han demostrado que los conjuntos difusos tipo-2 intervalo son más generales que los conjuntos con valores intervalados (este y otros conceptos se definirán con precisión en el Capítulo 2). Una interpretación adecuada de la definición original de Mendel [78] revela que esa clase de conjuntos incluye una variedad más amplia de ellos, incluyendo conjuntos con formas no consideradas hasta ahora, algunos de los cuales pueden tener grados secundarios no convexos. Sin embargo, las operaciones matemáticas entre conjuntos involucradas en los sistemas tipo-2 intervalo suponen que dichos conjuntos tienen como pertenencia primaria un intervalo cerrado y conexo, una condición que ya no es válida. En consecuencia, las operaciones teóricas sobre estos conjuntos (esto es, las operaciones *join* y *meet*) necesitan ser revisadas para desarrollar adecuadamente la teoría relacionada con estos nuevos conjuntos difusos.

No obstante, tal y como se ha mencionado, esta nueva clase de conjuntos pueden tener grados secundarios no convexos, un hecho que puede ser problemático. A pesar de que las definiciones para las operaciones sobre estos conjuntos tipo-2 se establecieron cuando Zadeh introdujo su noción en [113], éstas se describieron utilizando el Principio de Extensión, y no se proporcionó ninguna fórmula cerrada para operar con conjuntos difusos tipo-2. Otros autores centraron sus esfuerzos más tarde [49] en obtener tales fórmulas cerradas, pero éstas estaban limitadas a conjuntos tipo-2 con grados secundarios normales y convexos. Así pues, el segundo obstáculo es que la bibliografía existente sobre las operaciones *join* y *meet* no puede utilizarse para desarrollar la teoría de los sistemas difusos tipo-2 intervalo que utilizan conjuntos no convexos.

Estas dos necesidades motivan el trabajo restante propuesto en esta tesis, y comprenden el resto de nuestra contribución: por un lado, la segunda parte del trabajo se centra en desarrollar las operaciones teóricas sobre conjuntos difusos tipo-2 generales, con el fin de obtener fórmulas cerradas para operar con estos conjuntos, en los que las restricciones sobre la normalidad y convexidad de sus grados secundarios dejan de ser necesarias. Por otra parte, estos resultados nos permitirán desarrollar más allá la teoría de los sistemas difusos tipo-2 intervalo: en la tercera parte de esta tesis, dichas ecuaciones para conjuntos tipo-2 generales serán particularizadas para conjuntos tipo-2 intervalo, obteniendo fórmulas cerradas para las operaciones *join* y *meet* cuando estos conjuntos tienen grados secundarios no convexos. Dichas operaciones nos permitirán definir un nuevo motor de inferencia, que es la pieza clave para definir las formas generales de los sistemas difusos tipo-2 intervalo.

1.3 ESTRUCTURA DE ESTA TESIS

Esta tesis está organizada en Capítulos, cuyo contenido se resume a continuación:

Capítulo 1. A Historical Overview On Fuzzy Logic: Se presenta una breve revisión histórica sobre lógica difusa, tanto tipo-1 como tipo-2, así como otros tipos de conjuntos difusos propuestos a lo largo de la historia. Este pequeño resumen nos permite justificar los objetivos y motivaciones de esta tesis, así como su contribución al campo. Además, se presenta la estructura de la misma.

Capítulo 2. Fundamentals of Fuzzy Logic: Este capítulo tiene el propósito de revisar los conceptos básicos relacionados con la teoría tanto de conjuntos como de sistemas difusos: las definiciones, las operaciones teóricas, el proceso de inferencia y el Principio de Extensión coparán la mayor parte de nuestra atención, pese a que también se mencionará la estructura de los sistemas difusos y su uso como aproximadores funcionales universales.

Capítulo 3. Effects of Extra Type-1 Fuzzy Set Parameters on the Performance of a Fuzzy System: Se propone un enfoque novedoso en este capítulo para dilucidar si el número de parámetros libres es la única clave para mejorar el rendimiento de los sistemas difusos. Para hacerlo, se propone utilizar nueve problemas de aproximación funcional diferentes, y comparar el rendimiento de dos clases de sistemas difusos tipo-1, que son esencialmente iguales pero difieren en el número de parámetros por función de pertenencia. Este marco de referencia utiliza algoritmos genéticos tanto mono-objetivo como multi-objetivo.

Capítulo 4. Join and Meet Operations for Type-2 Fuzzy Sets With Nonconvex Secondary Memberships: Este capítulo se centra en las limitaciones tradicionales de las operaciones entre conjuntos (unión e intersección) cuando se trata con conjuntos tipo-2.

Se presenta una breve revisión sobre el Principio de Extensión y las fórmulas existentes en la bibliografía. Además, se introducen dos nuevos teoremas para las operaciones de unión e intersección, a fin de eliminar las restricciones relativas a la normalidad y convexidad de los conjuntos tipo-2 al operar con los mismos. Se realizarán varios ejemplos para ambas operaciones.

Capítulo 5. Towards a Fuzzy Logic System Based on General Forms of Interval Type-2 Fuzzy Sets: los trabajos recientes que demuestran que los conjuntos difusos tipo-2 intervalo son más generales que los conjuntos difusos con valores intervalados hacen necesario el desarrollo completo de un nuevo marco teórico para estos sistemas difusos tipo-2 intervalo que pueden tener grados secundarios no convexos (a los cuales nos referiremos como "formas generales de conjuntos/sistemas difusos tipo-2 intervalo"). La estructura completa de tales sistemas es revisada, y se prestará especial atención al motor de inferencia (que utiliza las operaciones definidas en el capítulo anterior) y al proceso de reducción de tipo, los dos bloques que presentan diferencias significativas con otros sistemas difusos ampliamente conocidos. Se consideran ambos tipos de entradas a dichos sistemas, singleton y non-singleton, y se propondrán dos ejemplos diferentes sobre cómo usar estos conjuntos y sistemas.

Capítulo 6. Conclusions, main contributions and list of publications: El capítulo final resume las conclusiones extraídas durante todo el trabajo, y resalta las principales contribuciones de esta tesis. Además, se facilita una lista de publicaciones.

Finalmente, con la finalidad de facilitar la lectura de esta tesis, se han adoptado las siguientes convenciones:

- Las ecuaciones, definiciones, figuras y teoremas están numerados de acuerdo al número de sección, seguido de un número creciente representado el elemento concreto dentro de dicha sección; es decir, la ecuación/definición/figura/-teorema 2.1.2 es el segundo elemento de la sección 2.1

- Los conjuntos difusos, tanto tipo-2 como tipo-2, se representarán indistintamente utilizando letras mayúsculas, tales como A, B, F_1, F_2, y así. No obstante, si en algún contexto específico se requiere tratar con conjuntos tipo-1 y tipo-2 *al mismo tiempo*, las mayúsculas normales se utilizarán para representar a los tipo-1 (como en A o B), mientras que sus homólogos tipo-2 se representarán utilizando letras con tilde de la eñe (~) (como en $\tilde{A}$ o $\tilde{B}$).

- En general el término "lógica/conjuntos/sistemas difusos tipo-2" se utilizará en lugar de "lógica/conjuntos/sistemas difusos tipo-2 general". El motivo es que no existe definición formal para esta última. Sin embargo, en algunos contextos específicos se utilizará para diferenciarla explícitamente de la versión más específica de lógica/conjuntos/-sistemas difusos tipo-2 intervalo.

1

A HISTORICAL OVERVIEW ON FUZZY LOGIC

Fuzzy theory is wrong, wrong, and pernicious. What we need is more logical thinking, not less. The danger of fuzzy logic is that it will encourage the sort of imprecise thinking that has brought us so much trouble. Fuzzy logic is the cocaine of science.

— Professor William Kahan (UC Berkeley)

1.1 THE BEGINNING OF FUZZY LOGIC

Fuzzy logic was introduced as early as 1965 by L.A. Zadeh in his proposal of fuzzy sets (FS) theory [117]. He aimed to provide a complete and formal mathematical framework to represent concepts that are themselves imprecise or unclear [104]. Although his work was severely criticised by the academic community (due mostly to the theory's emphasis on imprecision), he continued to develop his work over the following decade, with some of his seminal papers such as [116], [113], [114] and [115], among many others. In these works he introduced the concepts of fuzzy sets (and the notion of their extension), fuzzy logic and inference, approximate reasoning and the basic concepts of liguistic variables.

Nonetheless, many other researchers around the world were attracted by his proposal and started expanding fuzzy logic (FL) theory, systems and controllers, applying it to many different fields. In 1974, E. H. Mamdani presented the very first fuzzy logic controller (FLC) [66], in contrast with the classical methodologies of control based in precise mathematical modeling. In 1977, Dubois presented a comprehensive study of traffic conditions using fuzzy sets in his PhD thesis [28]. Later, between 1976 and 1987, fuzzy logic and its application to industry attracted special attention in Japan, where even a train line in Sendai implemented a controller based on fuzzy logic [104]. That success in real world applications revived the interest in fuzzy logic in the United States in the late 80s, leading to a so called "fuzzy boom" world-wide, attracting the attention of many researchers around the world.

One of the main criticisms about fuzzy logic was related to the construction of membership functions (MFs; for details about MFs and its definition, please refer to Chapter 2). If fuzzy sets are supposed to model uncertainty, then some authors argue that assigning a crisp number to a membership degree has no uncertainty at all. This criticism led Zadeh [10] to introduce the notion of type-N fuzzy set in [118] and, therefore, type-2 fuzzy sets (T2FS), in which the membership degrees were themselves fuzzy sets. The formal definition of such sets was finally introduced in 1975 in [113].

As some authors argued that, paradoxically, type-1 (T1) fuzzy logic is unable to handle uncertainty, its type-2 (T2) extension started to gain prominence because of its ability to handle it, in the sense of *modelling and minimising its effects* [78]. Hence, many authors began to study their definition, properties and set theoretic operations: Mizumoto and Tanaka [80], Dubois and Prade [29] and Mendel and Karnik [46] focused their efforts in the definition of type-2 fuzzy sets, as well as in the operations on them.

The growing interest on FL, specially its type-2 version, motivated researchers to approach uncertainty handling and modeling in many different ways, and many other proposals of fuzzy sets arose: L-fuzzy sets (LFS) [34], set-valued fuzzy sets (SVFS) [35], Atanassov intuitionistic fuzzy sets (AIFS) [4] (and, later, the interval version of those sets, interval-valued Atanassov's intuitionistic fuzzy sets, IVAIFS [3]), grey sets (GS) [22], shadowed sets (SS) [82], hesitant fuzzy sets (HFS) [93], non-stationary fuzzy sets (nSFS) [33], among many others. A historical review of the different kinds of fuzzy sets, some of their properties and their relationships can be found in [10]. However, a deep study on these variety of fuzzy sets is out of the scope of this dissertation; we will focus on type-1, type-2 and the most specific version of interval type-2 fuzzy sets (IT2FS).

Focusing on type-2 fuzzy logic, as mentioned above, as early as 1975 Zadeh introduced the notion of type-2 FSs. In 1976, Mizumoto and Tanaka studied the set theoretic operations, the properties of membership grades of type-2 and both the operations of algebraic sum and product. Later, in the late 90s, Karnik and Mendel extended that work, and presented practical closed formulas for the intersection, union and negation on T2FSs [46], [49]; they also introduced the concept of a type-2 centroid [48]. Many other papers, studying mainly the composition of type-2 relations were published between 1978 and 1999 [26], [29]. All this previous work led to Liang and Mendel to focus their efforts on the development of a complete theory about IT2FLS.

Although the theory of T2FL was progressing, during the early 2000s IT2FL hoarded most of the researchers' attention to the detriment of general type-2 (GT2) FL (although there is no formal definition for the latter, the term is widely used in order to make explicit difference from the simpler version of IT2FSs/FLSs). This difference was motivated by its high computational complexity, specially in the type reduction operation (an intermediate operation required in a fuzzy logic system in order to accomplish the input-output mapping). This difference motivated a clear secession in the research of fuzzy logic: theoretical progress was focused on developing the theory of T2FL, whereas practical implementations of real world systems tended to use the simpler IT2FL.

1.2 GOALS AND MOTIVATIONS

The brief historical introduction about fuzzy logic and its development allows us to approach appropriately the goals and motivations of this dissertation, which we will itemise in the following.

The first debate that attracts our attention has been discussed since the very appearance of type-2 fuzzy logic. In general, it has been established by various works and studies [13] that GT2 and IT2FLSs are able to obtain a better performance than their type-1 counterparts, specially when facing high levels of uncertainty [31], [38], [59]; however, why this outperformance happens still remains unexplained. This fact led to a well known controversy, in which some authors state that one reason for type-2 FLSs to perform better is due to the use of extra parameters in each MF in GT2 and IT2 FLSs when compared to their T1 counterparts.

Some authors have previously proposed different frameworks to compare both type-1 and type-2 FLSs [45], [88]; furthermore, other works [11] propose a comparison between T1, non-singleton T1 and singleton IT2 FLSs in several function approximation problems, providing each system with the same number of parameters (when possible), and giving all systems equal opportunities of optimisation using semi-random search algorithms. However, as will be explained later in this dissertation, some of those parameters (i.e. the consequents) are obtained optimally but in different ways in the different kinds of systems, and thus, it could be argued that the opportunities of optimisation are not *strictly* the same, but rather *roughly* equal.

Motivated by this endless controversy, the first part of this dissertation aims to contribute to solve this debate from a different and novel approach: if the controversy revolves around the number of free parameters or degrees of freedom available to tune the system, then we propose to step back to T1FLSs, and use a function approximation framework to compare T1FLSs which are essentially the same, but use a different number of parameters per MF, and compare them to verify if there exist statistically significant differences in their performance. The work aims to find out whether the number of parameters is *the key* to improve FLSs' performance or not, and thus, to prove or disprove if some other factors (i.e., how those parameters are used differently in T1 and T2 FLSs) are involved.

Conclusions from the previous part will plenty justify our focus on type-2 fuzzy sets and systems theory in ii. On that regard, we need to consider two important facts about the existing literature of GT2 and IT2 FL, which are introduced in the following.

Firstly, recent works have shown [89] that IT2FSs are more general than interval-valued fuzzy sets (IVFS) (this and other concepts will be precisely defined in Chapter 2). A proper interpretation of the original definition by Mendel [78] revealed that class of sets include a broader variety of them, including sets with certain shapes that had not been considered until now, some of which might have non-convex secondary grades. However, the mathematical set operations involved in the classic IT2FLSs assumed those sets had closed and connected intervals as their primary membership, a condition that does not hold any more. Hence, the set theoretical operations on these sets (i.e, the *join* and *meet* operations) need to be revised in order to properly develop the theory related to the new IT2FLSs.

Nonetheless, as stated, this new class of sets might have non-convex secondary grades, a fact that can actually be troublesome. Although the definitions for those operations on type-2 fuzzy sets were established when Zadeh introduced their notion in [113], they were described using the Extension Principle, and no closed formula was provided to operate on type-2 fuzzy sets. Other authors focused their efforts later [49] to obtain closed formulas for these operations, but they were limited to T2FSs having normal and convex secondary grades. Thus, the second obstacle is that existing literature about the join and meet operations cannot be used to develop the theory of IT2FLSs using non-convex sets.

These two needs motivate the remaining work proposed in this dissertation, and they comprehend the rest of our contribution: on the one hand, the second part of this thesis will focus our efforts in devoloping the theoretical operations on GT2FSs, in order to obtain closed formulas to operate with these sets where the restrictions about the normality and convexity of the secondary grades are no longer required. On the other hand, these results will allow us to further develop the theory of IT2FLSs: in the third part of this work, those equations for GT2FSs will be particularised for IT2FSs, obtaining closed formulas for the join and meet operations when these sets have non-convex SG. Those operations will allow us to define a new inference engine, which is the keystone to defining the general forms of IT2FLSs.

1.3 STRUCTURE OF THIS DISSERTATION

This dissertation is organised in Chapters, which content is summarised in the following:

Chapter 1. A Historical Overview On Fuzzy Logic: A brief historical overview about fuzzy logic, both type-1 and type-2, is presented, as well as other kinds of fuzzy sets proposed throughout history. This brief summary allows us to justify the motivations and goals of this dissertation, as well as its contribution to the field. Moreover, the structure of the thesis is presented.

Chapter 2. Fundamentals of Fuzzy Logic: This chapter aims to review the basic concepts related to the theory of both fuzzy sets and fuzzy logic systems: definitions, set theoretic operations, the inference process and the Extension Principle will attract most of our attention, although the structure of fuzzy logic systems and their use as universal approximators will also be hightlighted.

Chapter 3. Effects of Extra Type-1 Fuzzy Set Parameters on the Performance of a Fuzzy System: An innovative approach is proposed in this chapter tu elucidate if the number of free parameters is the only keystone to improve the performance of fuzzy logic systems. To do so, we propose using nine different function approximation problems, and comparing the performance between two types of type-1 FLSs, which are essentially the same but have different number of parameters per MF. The framework uses both single-objective and multi-objective genetic algorithms (GAs).

Chapter 4. Join and Meet Operations for Type-2 Fuzzy Sets With Nonconvex Secondary Memberships: This chapter focuses in the traditional limitations of the set theoretic operations (union and intersection) when dealing with type-2 fuzzy sets. A brief overview on the Extension Principle and existing formulas in the literature is presented. Moreover, two new theorems for the union and intersection operations are introduced, in order to eliminate the restrictions regarding normality and convexity of type-2 sets when operating with them. Several examples about both operations will be carried out.

Chapter 5. Towards a Fuzzy Logic System Based on General Forms of Interval Type-2 Fuzzy Sets: recent works proving that IT2FSs are more general than IVFSs created the need of developing a whole new theoretical framework for this IT2FLSs which can have non-convex secondary grades (which we will be referring to as "general forms of interval type-2 fuzzy sets/logic systems, gfIT2FS/gfIT2FLSs). The whole structure of the systems is revisited, and especial attention is focused on the inference engine (which uses the set theoretic operations defined in the previous chapter) and the type-reduction process, the two blocks presenting significant differences with the other well-know FLSs. Both singleton and non-singleton gfIT2FLSs are considered, and two different examples about how to use these sets and systems are proposed.

Chapter 6. Conclusions, main contributions and list of publications: The final chapter summarises the conclusions obtained during all our work, and highlights the main contributions of this dissertation. Moreover, a list of publications is provided.

Finally, in order to ease the reading of this dissertation, the following conventions have been adopted:

- Equations, definitions, figures and theorems are numbered with the section number, followed by an increasing number representing the number of item within the section, i.e., equation/definition/figure/theorem 2.1.2 is the second one in section 2.1.

- Fuzzy sets, both type-1, type-2 and the more specific version interval type-2, will be represented indistinctly using capital letters, such as A, B, F_1, F_2, and so on. Nonetheless, if in some specific context we deal with type-1 and type-2 sets *at the same time*, regular capital letters will be used to represent type-1 fuzzy sets (as A or B), whereas their type-2 counterparts will be represented using the tilde symbol (as in $\tilde{A}$ or $\tilde{B}$).

- In general the term "type-2 fuzzy sets/logic/systems" will be used rather than "general type-2 fuzzy sets/logic/systems". The reason for that is that there is no formal definition for the latter. Nevertheless, in some specific contexts we will use it to make explicit difference with the more specific version of interval type-2 fuzzy sets/logic/systems.

2 FUNDAMENTALS OF FUZZY LOGIC

"Fuzzification" is a kind of scientific permissiveness. It tends to result in socially appealing slogans unaccompanied by the discipline of hard scientific work and patient observation.

— Professor Rudolf Kalman (University of Florida)

This chapter focuses in reviewing the concepts of fuzzy logic needed to completely comprehend the work presented in this dissertation. The first section is dedicated to type-1 fuzzy sets and logic, whereas the second one draws its attention on the type-2 counterpart. For both sections, we introduce the definitions of the corresponding sets as well as the set theoretic operations for the union and intersection, using the Extension Principle when needed. Subsequently, we will introduce the concepts of fuzzy rules and their role in fuzzy inference. Finally, we will briefly highlight fuzzy logic systems as universal approximators.

2.1 INTRODUCTION TO TYPE-1 FUZZY LOGIC

Although human beings are able to manage, process and understand cuantitative information (represented by numbers and/or mathematical equations), most of human knowledge is described using language and linguistic terms. This linguistic information is, by nature, imprecise: the statement "that person is tall" can be considered as true by some people, whereas others may think that given person is not tall at all; or, on the other hand, some others may answer *he is not very tall* or *not very short*. In addition, the appreciation of the concept *tall* might be strongly influenced by social factors, as nationality, culture and age, among others. This fact reveals that human knowledge is imprecise, vague or partially true, a fact that is perfectly represented in the fuzzy logic adage *words mean different things to different people*.

Zadeh's initial work about fuzzy sets [117] and linguistic variables [113] was aiming to provide a whole theoretical framework to properly represent uncertain or incomplete knowledge. In the following subsection, we introduce the concepts and definitions related to type-1 fuzzy sets and logic.

2.1.1 *The concept of type-1 fuzzy sets*

Fuzzy sets are named as opposition to *classic* or *crisp* sets, which are completely delimited by a well defined frontier. In this sense, a given element from a given universe of discourse belongs completely to the set, or it does not at all; partial memberships do not exist in this context.

Hence, each element x in the universe of discourse X, $x \in X$ has associated a membership degree (or membership value) to the set A, which is equal to 1 if the element belongs to the set, and 0 otherwise. Thus, we can define the *membership function of a crisp set* A, defined over a universe of discourse X, as:

$$\mu_A(x) = \begin{cases} 1 & \text{if} \quad x \in X \\ 0 & \text{if} \quad x \notin X \end{cases} \tag{2.1.1}$$

As the fuzzy set A is completely and univoquely determined and described by its membership function $\mu_A(x)$, we can stablish a mathematical equivalence between them, as knowing one of them implies knowing the other too. Usually, crisp sets are represented only by means of their members, it is, representing only the elements having a membership value equal to 1. For instance, the set of natural numbers less than or equal to 10 would be:

$$A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \quad (2.1.2)$$

Oppositely, fuzzy sets were defined by Zadeh [117] as sets whose members may have a *partial degree of membership*, allowing the membership values to be any number contained within the unit interval $[0, 1]$, i. e., $\mu_A(x) \in [0, 1] \ \forall x \in X$. The formal definition of a fuzzy set is as follows:

Definition 2.1.1. A fuzzy set A, defined over a universe of discourse X, is the set of all ordered pairs such that:

$$A = \{(x, \mu_A(x)) \mid x \in X, \ \mu_A : X \mapsto [0, 1]\} \quad (2.1.3)$$

As it can be deduced from Definition 2.1.1, if we restrict the membership values to be either 0 or 1, then a fuzzy set reduces to a crisp set, thus proving that the former are a generalisation of the latter.

Depending on the nature of the universe of discourse, fuzzy sets are represented in a different way. When X is discrete, then fuzzy sets are represented as in Equation (2.1.4), where the symbol "/" represents ordered pairs, and the symbol $+$ denotes union of all elements within the set.

$$A = \sum_i \mu_A(x_i)/x_i = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \mu_A(x_3)/x_3 + ... \quad (2.1.4)$$

On the other hand, if X is continuous, the sets are represented as in Equation (2.1.5), where union over all admisible values is noted as $\int_X$.

$$A = \int_X \mu_A(x)/x \quad (2.1.5)$$

This scheme is much more suitable to represent sets and concepts whose limits are not clearly defined. As an example, let us consider the set of *tall people*. We may agree that all people taller than 1.90m are definitely tall (i.e., membership value equal to 1), whereas all people shorter than 1.60m are short. Nonetheless, individuals having a height value between 1.60m and 1.90m *belong partially* to the set of *tall people*; someone being 1.65m tall could have a membership value of 0.15, whereas another person being 1.85m tall could be assigned a membership equal to 0.9.

2.1.2 Other definitions regarding type-1 fuzzy sets

In this subsection we present other basic definitions related to type-1 fuzzy sets, most of which will be used across this dissertation.

Definition 2.1.2. Fuzzy subset: let A and B be two fuzzy sets defined over the same universe of discourse X. Then, A is a subset of B or, equivalently, A is contained in B (which is denoted as $A \subseteq B$), if and only if:

$$A \subseteq B \iff \mu_A(x) \leqslant \mu_B(x) \qquad \forall x \in X \tag{2.1.6}$$

Definition 2.1.3. Convex fuzzy set: a type-1 fuzzy set is said to be *convex* if and only if it verifies the following condition:

$$\lambda\mu_A(x_1) + (1-\lambda)\mu_A(x_2) \geqslant \min(\mu_A(x_1), \mu_A(x_2)) \\ \forall x_1,\ x_2 \in X,\ \forall\lambda \in [0, 1] \tag{2.1.7}$$

Definition 2.1.4. Support: let A be a fuzzy set. Then, the *support* of A, which is denoted as $S(A)$, is defined as:

$$S(A) = \{x \in X \,|\, \mu_A(x) \geqslant 0\} \tag{2.1.8}$$

Definition 2.1.5. Core (or kernel): let A be a fuzzy set. Then, the *core* (also named *kernel* by other authors) is the subset of the domain having a membership value equal to 1, i.e.:

$$\mathrm{Core}(A) = \{x \in X \,|\, \mu_A(x) = 1\} \tag{2.1.9}$$

Definition 2.1.6. Amplitude: let A be a fuzzy set defined over $X \subseteq \mathbb{R}$, having a support given by $S(A)$. Then, the *amplitude* of A, denoted by $\mathrm{Amp}(A)$, is given by:

$$\mathrm{Amp}(A) = \mathrm{Sup}(S(A)) - \mathrm{Inf}(S(A)) \tag{2.1.10}$$

In many applications of fuzzy logic and fuzzy control, normal (*normality* is defined below) and convex sets are most widely used. In those cases, it is usual to use the terms *support* and *amplitude* as synonyms. However, in part of this thesis we will work with non-convex sets, whose support will be comprised in many cases by closed, connected and disjointed intervals; hence, it is importat to highlight that these terms will not be interchangeable in the context of this dissertation.

Definition 2.1.7. Height: let A be a fuzzy set. Then the *height* of A, denoted by $\mathrm{Height}(A)$, is given by:

$$\mathrm{Height}(A) = \sup\{\mu_A(x) \,|\, x \in X\} \tag{2.1.11}$$

Definition 2.1.8. Normality: let A be a fuzzy set. Then A is said to be a *normal fuzzy set* if there is at least one value $x \in X$ such that $\mu_A(x) = 1$. Analogously, we can also define A as *normal* if the following stands:

$$A \text{ is normal} \iff \mathrm{Heigth}(A) = 1 \tag{2.1.12}$$

Definition 2.1.9. α-cut: let A be a fuzzy set. Then, an α-cut of A, which is denoted by A_α, is a crisp set defined as [101]:

$$A_\alpha = \{x \in X \,|\, \mu_A(x) \geqslant \alpha\} \qquad \alpha \in [0,1] \tag{2.1.13}$$

It is worthwhile to highlight that, when the fuzzy sets considered are convex, the resulting α-cut is a closed and connected interval for any $\alpha \in [0,1]$; nonetheless, in this dissertation we consider non-convex and/or non-normal type-1 fuzzy sets. In such cases, the α-cuts may be comprised by the union of several closed, connected and disjointed intervals and/or *singletons*, which are subsequently defined.

Definition 2.1.10. Fuzzy singleton: a fuzzy set A is called a *fuzzy singleton* if its support is comprised by a single element $x \in X$.

$$\#S(A) = 1 \iff S(A) = x \in X \tag{2.1.14}$$

Where the symbol # denotes the number of elements within a set.

Definition 2.1.11. Fuzzy number: [113] a fuzzy set A is said to be a *fuzzy number* if it is defined over the real line $\mathbb{R}$ with a normal and convex membership function of bounded support.

2.2 OPERATIONS ON TYPE-1 FUZZY SETS

The set theoretic operations of *intersection* and *union* are well defined for crisp sets, and can be extended to operate on type-1 fuzzy sets. However, there is not a unique way of defining such operations. In [112], Zadeh proposed the following formulas for the union and intersection of fuzzy sets. Let A and B be two fuzzy sets. Hence:

- The *intersection* of A and B, denoted as $A \cap B$, is uniquely determined by its membership function $\mu_{A \cap B}(x)$, and is given by:

$$A \cap B \iff \mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x)) = = \mu_A(x) \wedge \mu_B(x) \tag{2.2.1}$$

  Where $\wedge$ denotes the minimum operation.

- The *union* of A and B, denoted as $A \cup B$, is uniquely determined by its membership function $\mu_{A \cup B}(x)$, and is given by:

$$A \cup B \iff \mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x)) = = \mu_A(x) \vee \mu_B(x) \tag{2.2.2}$$

  Where $\vee$ denotes the maximum operation.

- The *negation* of a set A, denoted usually as A' or $\bar{A}$, is uniquely determined by its membership function $\mu_{A'}(x)$ (or $\mu_{\bar{A}}$), and is given by:

$$\bar{A} \iff \mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{2.2.3}$$

The definitions of these operations are a generalisation of the union, intersection and negation on crisp sets, in the sense that if we restrict the membership values to $\{0, 1\}$ rather than $[0, 1]$, then Equations (2.2.1), (2.2.2) and (2.2.3) describe the union, intersection and negation of classic sets. Nevertheless, there is not a unique way of defining these operations in such a way they generalise their crisp counterparts, a topic we introduce subsequently.

2.2.1 *Intersection operation on type-1 fuzzy sets: T-norms*

In this subsection we focus our attention on the class of functions that are suitable to generalise the intersection operation on type-1 fuzzy sets from its crisp counterpart. That class of functions, referred to as *T-norms*, need to comply with a given set of axioms, which are presented subsequently:

Definition 2.2.1. Let $\mathcal{T}$ be a function $\mathcal{T} : [0,1]x[0,1] \mapsto [0,1]$. Then $\mathcal{T}$ is a *T-norm* if it verifies the following axioms [101]:

- **Axiom 1:** $\mathcal{T}(0,0) = 0$, $\mathcal{T}(a,1) = \mathbb{T}(1,a) = a$ (known as **boundary condition**).
- **Axiom 2:** $\mathcal{T}$ must be commutative, i.e., $\mathcal{T}(a,b) = \mathcal{T}(b,a)$.
- **Axiom 3:** $\mathcal{T}$ must be non-decreasing in both variables, i.e., if $a \leqslant a'$ and $b \leqslant b'$, then:

$$\mathcal{T}(a,b) \leqslant \mathcal{T}(a',b') \tag{2.2.4}$$

- **Axiom 4:** $\mathcal{T}$ must verify the associative condition, i.e.:

$$\mathcal{T}(\mathcal{T}(a,b),c) = \mathcal{T}(a,\mathcal{T}(b,c)) \tag{2.2.5}$$

Many different T-norms have been proposed in the literature [101], as:

- The drastic product:

$$\mathcal{T}_{DP}(a,b) = \begin{cases} a & \text{if } b = 1 \\ b & \text{if } a = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.2.6}$$

- The Einstein product:

$$\mathcal{T}_{ES}(a,b) = \frac{ab}{2-(a+b-ab)} \tag{2.2.7}$$

- The algebraic product:

$$\mathcal{T}_{AP}(a,b) = ab \tag{2.2.8}$$

- The minimum operation:

$$\mathcal{T}_{min}(a,b) = \min(a,b) = a \wedge b \tag{2.2.9}$$

In addtion, some authors have also defined complete families of functions to act as T-norms, obtaining a different function for each value of a given parameter. Some of them are presented in the following:

- The Dombi class [25]:

$$\mathcal{T}_\lambda = \frac{1}{1 + [(\frac{1}{a} - 1)^\lambda + (\frac{1}{b} - 1)^\lambda]^{1/\lambda}} \tag{2.2.10}$$
$$\lambda \in (0, \inf)$$

- The Dubois-Prade class [27]:

$$\mathcal{T}_\alpha(a, b) = \frac{ab}{max(a, b, \alpha)} \tag{2.2.11}$$
$$\alpha \in [0, 1]$$

- The Yager class [103]:

$$\mathcal{T}_\omega(a, b) = 1 - min[1, ((1 - a)^\omega + (1 - b)^\omega)^{1/\omega}] \tag{2.2.12}$$
$$\omega \in (0, \inf)$$

All these functions and classes are suitable to implement the operation of fuzzy intersection, among others. However, the most used intersection functions in the literature are the minimum and the product operations.

2.2.2 *Union operation on type-1 fuzzy sets: T-conorms*

In this subsection we study the class of functions that generalise the union operation on fuzzy sets from its crisp counterpart, which are called *T-conorms* (although other authors refer to them as *S-norms*). Those functions need to verify a set of conditions, which are itemised in the following definition:

Definition 2.2.2. Let $\mathcal{S}$ be a function $\mathcal{S} : [0, 1]x[0, 1] \mapsto [0, 1]$. Then $\mathcal{S}$ is a *T-conorm* if it verifies the following axioms [101]:

- **Axiom 1:** $\mathcal{S}(1, 1) = 1$, $\mathcal{S}(a, 0) = \mathcal{S}(0, a) = a$ (known as *boundary condition*).

- **Axiom 2:** $\mathcal{S}$ must be commutative, i.e., $\mathcal{S}(a, b) = \mathcal{S}(b, a)$.

- **Axiom 3:** $\mathcal{S}$ must be non-decreasing in both variables, i.e., if $a \leqslant a'$ and $b \leqslant b'$, then:

$$\mathcal{S}(a, b) \leqslant \mathcal{S}(a', b') \tag{2.2.13}$$

- **Axiom 4:** $\mathcal{S}$ must verify the associative condition, i.e.:

$$\mathcal{S}(\mathcal{S}(a, b), c) = \mathcal{S}(a, \mathcal{S}(b, c)) \tag{2.2.14}$$

Many different T-conorms have been proposed in the literature [101], as:

- The drastic sum:

$$\mathcal{S}_{DS}(a,b) = \begin{cases} a & \text{if } b = 0 \\ b & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.2.15}$$

- The Einstein sum:

$$\mathcal{S}_{ES}(a,b) = \frac{a+b}{1+ab} \tag{2.2.16}$$

- The algebraic sum:

$$\mathcal{S}_{AS}(a,b) = a+b-ab \tag{2.2.17}$$

- The maximum operation:

$$\mathcal{S}_{max}(a,b) = max(a,b) = a \vee b \tag{2.2.18}$$

Moreover, some authors defined complete families of functions to act as T-conorms. One different T-conorm is obtained for each value of a given parameter, as the ones listed subsequently:

- The Dombi class [25]:

$$\mathcal{S}_{\lambda} = \frac{1}{1+[(\frac{1}{a}-1)^{-\lambda}+(\frac{1}{b}-1)^{-\lambda}]^{-1/\lambda}} \qquad \lambda \in (0, \inf) \tag{2.2.19}$$

- The Dubois and Prade class [27]:

$$\mathcal{S}_{\alpha}(a,b) = \frac{a+b-ab-min(a,b,1-\alpha)}{max(1-a,1-b,\alpha)} \qquad \alpha \in [0,1] \tag{2.2.20}$$

- The Yager class [103]:

$$\mathcal{S}_{\omega}(a,b) = min[1,(a^{\omega}+b^{\omega})^{1/\omega}] \qquad \omega \in (0, \inf) \tag{2.2.21}$$

All these functions and classes are suitable to implement the operation of fuzzy union, among others. Nonetheless, the most used union functions in the literature are the maximum and the algebraic sum operations.

2.2.3 *Complement operation on type-1 fuzzy sets: negations*

The concept of *complement* of a given element in a set is easily and intuitively extended from the complement of an element from a crisp set; the only difference is that the operation is applied to the membership values of such given elements. The complement measures to which extent a given element $x \in X$ does not belog to the given set.

As in the previous subsections related to intersection and union, respectively, there are many different functions that can extend the concept of negation from crisp sets to fuzzy sets. For a given function to be considered a *fuzzy complement*, it must comply with a given set of axioms, which are detailed in the following:

Definition 2.2.3. Let $\mathcal{C}$ be a function $\mathcal{C} : [0,1] \mapsto [0,1]$. Then $\mathcal{C}$ is a *fuzzy negation* if it verifies the following axioms [101]:

- $\mathcal{C}(0) = 1$ and $\mathcal{C}(1) = 0$ (known as *boundary condition*).
- $\mathcal{C}$ must be non-increasing, i.e., $\forall a, b \in [0,1]$, if $a \leqslant b$, then $\mathcal{C}(a) \geqslant \mathcal{C}(b)$.

The most used negation function is the ordinary complement, which is the most frequently used in the fuzzy logic literature:

$$\mathcal{C}(a) = 1 - a \tag{2.2.22}$$

In addition, some classes of negation functions have been defined, where a different complement is obtained depending of the chosen parameter:

- The Sugeno class [90]:

$$\mathcal{C}_\lambda(a) = \frac{1-a}{1+\lambda a} \qquad \lambda \in (-1, \inf) \tag{2.2.23}$$

- The Yager class [103]:

$$\mathcal{C}_\omega(a) = (1 - a^\omega)^{\frac{1}{\omega}} \qquad \omega \in (0, \inf) \tag{2.2.24}$$

2.2.4 *T-norms and T-conorms associated through complements*

It is worthwhile to highlight that T-norms and T-conorms are related in pairs through a given negation. To be more precise, we can summarise that relation in the following definition:

Definition 2.2.4. Let $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{C}$ be a T-norm, a T-conorm and a fuzzy negation, respectively. Then, $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{C}$ are said to form an *associated class*, if they verify the following condition:

$$\mathcal{C}(\mathcal{S}(a, b)) = \mathcal{T}(\mathcal{C}(a), \mathcal{C}(b)) \tag{2.2.25}$$

Definition 2.2.4 is important as, when designing a type-1 fuzzy logic system, it is usual to implement the intersection and union by choosing a t-norm and a t-conorm belonging to the same associated class.

2.3 FUZZY REASONING AND INFERENCE OF TYPE-1

Propositional logic (or propositional calculus) is a formal system [7] concerned with the study of *propositions* (i.e. statements, concepts, etc.) formed by other propositions, related and connected through *logical connectives* (i.e., logical operations), and how their truth values (either *true* or *false*) depend on the truth values of their components. This logical framework accepts the *law of excluded middle*, which states that all propositions are either true or false; this is clearly opposite with the basis of fuzzy logic, which allows partial degrees of veracity. The logical connectives that allow us to combine propositions are presented subsequently: let p and q be two propositions.

- **Conjuntion:** denoted as $p \wedge q$. It is evaluated as true if both p and q are true.
- **Disjunction:** denoted as $p \vee q$. It is evaluated as true if either p or q are true.
- **Negation:** denoted as $\neg p$. It is evaluated as the opposite of p.
- **Implication:** denoted $p \to q$. In this connective, if p is true, so is q. It is used to construct *logical rules*, and is usually read as "if p is true, then q is true". This operation allows us to define *logical rules*, also named *if-then rules*. Implication describes a *relation* between these two variables.

In propositional logic, in the process of *inference*, a certain number of propositions are combined, using the connectives listed above, to create the *premises*, which are taken for granted. Afterwards, a *conclusion* is extracted from them using the inference rule known as *modus ponens*, which takes the form:

Premise 1:	$p \rightarrow q$
Premise 2:	p
Conclusion:	q

This example could be read as follows:

- Premise 1: if p is true, then q is true.
- Premise 2: p is true.
- Conclusion: q must be true.

An equivalent way of stating the previous premises and conclusions is using a *logical rule* as:

$$\text{IF} \quad p \quad \text{THEN} \quad q \tag{2.3.1}$$

Where both p and q can be simple or complex premises or propositions. An example of how propositional logic works using *modus ponens* could be as follows:

Premise 1:	All men are mortals.
Premise 2:	Gonzalo is a man.
Conclusion:	Gonzalo is mortal.

Nonetheless, this structure for reasoning does not hold in the context of fuzzy logic, where truth is a matter of degree. When dealing with fuzzy inference and reasoning, the propositions within a rule admit fuzzy membership, and such rules are named *fuzzy rules*, in which the concepts or propositions involved are not clearly defined, or partially true. Such a fuzzy rule can be of the form:

$$\text{IF} \quad \text{speed is high} \quad \text{THEN} \quad \text{kinetic energy is high} \tag{2.3.2}$$

Where *speed is high* is referred to as the *rule antecedent*, whereas *kinetic energy is high* is the *rule consequent*. A general way of expressing a fuzzy rule such as Equation (2.3.2) is as Equation (2.3.3):

$$\text{IF} \quad x \text{ is } A \quad \text{THEN} \quad y \text{ is } B \tag{2.3.3}$$

Expressing Equation (2.3.3) using the implication operator would lead us to $A \to B$, which express a relation R between the variables x and y. If we consider the crisp case, as stated in [52], "a *crisp relation* represents the presence or absence of association, interaction, or interconnectedness between the elements of two or more sets". Herein, we limit our example to binary relations; extending those concepts to the n-dimensional case is straightforward.

Let us denote $X x Y$ the Cartesian product of X and Y, i.e., $X x Y = \{(x, y) \mid x \in X \, and \, y \in Y\}$. Hence, the relation R between X and Y, usually denoted as $R(X, Y)$, is a subset of $X x Y$, which can be characterised by its membership function as follows:

$$\mu_R(x, y) = \begin{cases} 1 & iff \qquad (x, y) \in R(X, Y) \\ 0 & otherwise \end{cases} \tag{2.3.4}$$

When extending relations to their fuzzy counterparts [67], they "represent a *degree* of presence or absence of association, interaction, or interconectedness between the elements of two or more fuzzy sets". Hence, $R(X, Y)$ is a fuzzy subset of $X x Y$, where each element (x, y) is associated with a membership value $\mu_R(x, y)$. Formally:

$$R = A \to B = \int_{XxY} \mu_A(x) \to \mu_B(y)/(x, y) \tag{2.3.5}$$

Where $\to$ denotes the implication operator chosen, which can be, among others [5]:

- **S-Implications:** such as Kleen-Dienes, Reichenbach and Lukasiewicz or Largest S-implications.
- **R-Implications:** as Gödel, Goguen, Lukasiewicz and Largest R-implications.
- **QL-Implications:** as those by Zadeh, Klir and Yuan (1 and 2) and Kleen-Dienes.

Fuzzy inference is the process of obtaining a consequence (which can be a fuzzy set or a crisp number) from a set of fuzzy antecedents, based on a set of fuzzy rules. Although there are several options to perform the fuzzy inference process (such as the *Generalised Modus Tollens*), they are not widely used in engineering [67], and it is usually preferred the *Generalised Modus Ponens*, which is subsequently introuced:

Premise 1: IF x is A THEN y is B
Premise 2: x is A'
Conclusion: y is B'

Where A, A', B and B' are linguistic labels, represented by fuzzy sets. Intuitively, this inference process can be described as follows: Premise 1 is a standard fuzzy rule; nonetheless, Premise 2 states that x is *not* A, but rather A', which is not *exactly* A but *close to it*, it is, A' is A *to some degree*. Hence, our conclussion is that y is B', which is *not exactly* B, but rather B *to a certain extent*. Consequently, the *closer* A' is to A, then the *closer* B' will be to B.

To obtain the fuzzy set B' within the conclusion, the following operation has to be performed:

$$B' = A' \circ R \tag{2.3.6}$$

Where $\circ$ denotes *fuzzy composition*, and R represents the fuzzy relation expressed by Premise 1. To define fuzzy composition, we need to previously define the operations of *projection* and *cylindrical extension* [101].

Definition 2.3.1. Projection: Let R be a fuzzy relation in the space given by $X = \prod_{i=1}^{n} X_i = X_1 \text{ x } X_2 \text{ x} \ldots \text{x } X_n$, and let $\{i_1, i_2, \ldots, i_k\}$ be a subsequence of $\{1, 2, \ldots, n\}$. Hence, the *projection* of R on $X_{i_1} \text{ x } X_{i_2} \text{ x } \ldots \text{ x } X_{i_k} = \prod_{m=1}^{k} X_{i_m} \equiv X_I$ is a fuzzy relation $R_{Pr(X_I)}$ in X_I and is defined as follows:

$$\text{proj}(R) \text{ on } X_I = \int_{X_I} \sup_{(x_{j_1} \in X_{j_1}, \ldots, x_{j_{n-k}} \in X_{j_{n-k}})} \mu_R(x_1, \ldots, x_n) / (x_{i_1}, \ldots, x_{i_k}) \tag{2.3.7}$$

Where $\{j_1, \ldots, j_{n-k}\}$ is the complementary subsequence of $\{i_1, \ldots, i_k\}$ with respect to $\{1, \ldots, n\}$. That projection is characterised by its membership function, which is given by:

$$\mu_{R_{Pr(X_I)}}(x_{i_1}, x_{i_2}, \ldots, x_{i_k}) = \underset{(x_{j_1} \in X_{j_1}, \ldots, x_{j_{n-k}} \in X_{j_{n-k}})}{\arg\max} \mu_R(x_1, \ldots, x_n) \tag{2.3.8}$$

We can particularise this general definition to the case of binary fuzzy relations, which is more intuitive. If R is a binary fuzzy relation in $X \times Y$, then the projection of R on X, denoted by $R_{Pr(X)}$, is a fuzzy set defined on X, represented by Equation (2.3.9) and characterised by its membership function as in Equation (2.3.10):

$$\text{proj}(R) \text{ on } X = \int_X \sup_y \mu_R(x,y)/x \tag{2.3.9}$$

$$\mu_{R_{Pr(X)}}(x) = \arg\max_{y \in Y} \mu_R(x,y) \tag{2.3.10}$$

It is worthwhile to highlight that Equations (2.3.8) and (2.3.10) are valid even when R is a crisp relation.

Definition 2.3.2. Cylindrical extension: Let R_{X_I} be a fuzzy relation in $X_{i_1} \times X_{i_2} \times \ldots \times X_{i_k} = \prod_{m=1}^{k} X_{i_m} = X_I$, and $\{i_1, \ldots, i_k\}$ be a subsequence of $\{1, 2, \ldots, n\}$. Hence, the *cylindric extension* of R_{X_I} to $X_1 \times X_2 \times \ldots \times X_n = \prod_{i=1}^{n} X_i = X$ is a fuzzy relation $R_{Ex(X)}$ in X and is defined as follows:

$$\text{cyl}(R_{X_I}) \text{ on } X = \int_X \mu_{R_{X_I}}(x_{i_1}, \ldots, x_{i_k})/(x_1, \ldots, x_n) \tag{2.3.11}$$

That cylindrical extension is characterised by its membership function, which is given by:

$$\mu_{R_{Ex(X)}}(x_1, \ldots, x_n) = \mu_{R_{X_I}}(x_{i_1}, \ldots, x_{i_k}) \tag{2.3.12}$$

As a special situation for binary relations, if R_X is fuzzy set in X, then the cylindric extension of R_X to $X \times Y$ is a fuzzy relation $R_{Ex(X,Y)}$ in $X \times Y$ represented by Equation (2.3.13) and characterised by its membership function as in Equation (2.3.14):

$$\mu_{Ex(X,Y)}(x,y) = \int_{X \times Y} \mu_{R_X}(x)/(x,y) \tag{2.3.13}$$

$$\mu_{R_{Ex(X,y)}}(x,y) = \mu_{R_X}(x) \tag{2.3.14}$$

Once again, it is worthwhile mentioning that Equations (2.3.13) and (2.3.14) are also valid for crisp relations.

After introducing Definitions 2.3.1 and 2.3.2, we can introduce the concept of *fuzzy composition*, which is needed in order to obtain the set B' in Equation (2.3.6).

Definition 2.3.3. Fuzzy composition: Let A' be a fuzzy set defined on X, and let R be a fuzzy relation on $X \times Y$. Then, the composition of A' and R, denoted as $A' \circ R$, is a fuzzy set B' on Y given by:

$$B' = A' \circ R = \mathrm{proj}((\mathrm{cyl}(A') \text{ on } Y) \cap R) \text{ on } Y \tag{2.3.15}$$

Let us focus on a single explanation of Equation (2.3.15) step by step:

1. The cylindrical projection $\mathrm{cyl}(A')$ on Y is a fuzzy relation defined in $X \times Y$, say $S = S(X, Y)$.
2. The intersection $S \cap R = S(X, Y) \cap R(X, Y)$ is a fuzzy relation defined in $X \times Y$.
3. Finally, $S \cap R$ is projected on Y to obtain B'. This final set could be thought as *how much of* A' *goes to* B' *through the relation* R.

Expressing Equation (2.3.15) in terms of the membership functions involved, we obtain the following equation for the fuzzy *sup-star composition*:

$$\mu_{B'}(y) = \sup_x \mu_{B'}(x) \star \mu_R(x, y) \tag{2.3.16}$$

Where $\star$ denotes a given T-norm. If we choose $\star$ to be the minimum T-norm, then Equation (2.3.16) becomes the original inference rule proposed by Zadeh given in Equation (2.3.17), whereas if we set $\star$ to be the product T-norm, the retrieved Equation is as in (2.3.18):

$$\mu_{B'}(y) = \sup_x \min(\mu_{A'}(x), \mu_R(x, y)) \tag{2.3.17}$$

$$\mu_{B'}(y) = \sup_x \ (\mu_{A'}(x) \cdot \mu_R(x, y)) \tag{2.3.18}$$

If conjunctive premises are used in the *generalised modus ponens*, then the fuzzy inference process would be as follows:

Premise 1: IF x is A AND y is B THEN z is C
Premise 2: x is A' AND y is B'

Conclusion: z is C'

In this case, the rule given in Premise 1 can be expressed as a ternary relation R given by $R = A \times B \rightarrow C$ (instead of $A \rightarrow B$ when premises where not conjunctive), where $R = R(x, y, z)$, $A' = A'(x)$, $B' = B'(y)$ and $C' = C'(z)$. Thus, generalising Equations (2.3.6) and (2.3.15) to ternary relations, we have:

$$C' = (A' \times B') \circ R \tag{2.3.19}$$

Assuming we use the same T-norm $\star$ for implication, intersection and conjunction, it leads us to:

$$\begin{aligned}
\mu_{C'}(z) &= \sup_{x,y} \{[\mu_{A'}(x) \star \mu_{B'}(y)] \star [(\mu_A(x) \star \mu_B(y)) \star \mu_C(z)]\} \\
&= \sup_{x,y} \{\mu_{A'}(x) \star \mu_{B'}(y) \star \mu_A(x) \star \mu_B(y)\} \star \mu_C(z) \\
&= \left(\sup_{x} \{\mu_{A'}(x) \star \mu_A(x)\}\right) \star \left(\sup_{y} \{\mu_{B'}(y) \star \mu_B(y)\}\right) \star \mu_C(z) \\
&= (\alpha_A \star \alpha_B) \star \mu_C(z) = \alpha_{Rule} \star \mu_C(z)
\end{aligned} \tag{2.3.20}$$

In Equation (2.3.20) the term α_A indicates *how similiar* A' is to A, whereas term α_B measures the same between B' and B; jointly, α_{Rule} represents the final *rule firing strength*.

Real FLSs have multiple rules within their rule base, and each rule will produce a fuzzy set, similar to the one described in Equation (2.3.20). All those sets have to be combined using a T-conorm. Once combined, the final fuzzy set is the final result of the inference process.

2.4 TYPE-1 FUZZY LOGIC SYSTEM AS UNIVERSAL APPROXIMATORS

Rule-based fuzzy logic systems have been found to be very useful in engineering applications for a number of reasons. On the one hand [101], their IF-THEN rule-based structure can be specified using linguistic labels, which allows system designers to implement control strategies described using natural language.

On the other hand, functionally they are able to represent highly non-linear input-output mappings. Hence, they are useful for control applications, function approximation, decision making, pattern recognition among many other applications.

One interesting question to pose when dealing with non-linear mappings (such as neural networks or FLSs themselves) is to what extent are they able to approximate other non-linear functions. More precisely, it would be interesenting to know if FLSs can approximate other non-linear mappings defined over closed and bounded intervals to any extent, i.e., with arbitrary accuracy. In such case, rule-based FLSs would be *universal approximators*.

Many authors have dedicated their efforts to prove that certain types of FLSs are universal approximators:

- One of the first works approaching this topic was done by Wang [99] in 1992, where he proved that FLSs with product inference, product conjunction, centre of area defuzzification and Gaussian membership functions are able to approximate a real continuous function defined in a compact set with arbitrary accuracy. His proof was based in the Stone-Weierstrass Theorem [85].

- Later in 1992 [9] and 1993 [8] Buckley proved that a modified version of Sugeno type FLSs were universal controllers.

- In the same year, Wang and Mendel [100] extended their previous results using a fuzzy basis function representation for FLSs, crisp numbers as consequents and an orthogonal least-squares learning algorithm.

- In 1994 [53], Kosko based his proof for additive fuzzy systems using the concept of fuzzy patch, properly using enough MFs for each input.

- In 1995 [17] Castro extended this property to other FLSs, having triangular or trapezoidal MFs, conjunction modelled by any T-norm, and both implication and defuzzification need to satisfy some weak conditions.

- In 1999 [54] Kreinovich extended the function approximation capacity not only to the function itself, but also for its derivatives, for smooth functions.

Nevertheless, this universal approximation ability of FLSs has not been without criticism. In [51], Klement et al. posed a critical reflection on such previous claims by other authors, arguing that some crucial features are neglected (e. g. a boundary for the number of rules) and, hence, «fuzzy systems can only be universal approximators in a rather reduced sense». For more information about this argumentation, please refer to [51].

Moreover, it is important to highlight that all mentioned previous works provide a *justification* for using FLSs: no matter how complex a given input-output relation is, certain FLS having enough complexity can approximate it to any accuracy. However, these proofs only demonstrate the *existence* of those systems, but do not provide any practical algorithm to actually *obtain* such systems; this is why design methodologies regarding FLSs is a topic that still attracts attention nowadays.

Existence theorems are not the only relevant work that has been developed around the ability of type-1 fuzzy logic systems (T1FLSs) to perform as universal approximators. Other research has focused on studying their approximation accuracy [121] [122], necessary conditions [24] [110] and sufficient approximation conditions [57] [108] [109] [120].

2.5 STRUCTURE OF A TYPE-1 FUZZY LOGIC SYSTEM

Some of the main contributions of this dissertation are directly related to fuzzy logic systems and their structure. For instance, Chapter 4 presents two new theorems for the join and meet operations on general type-2 fuzzy sets, which are key to define the inference engine. In addition, Chapter 5 presents the structure of the general forms of interval type-2 fuzzy logic systems, including fuzzification, inference and type-reduction. Hence, it is important to introduce the simpler structure of type-1 FLSs, and later in Section 2.9, the more complex structure of type-2 FLSs, in order to properly contextualise our work.

In this section we intend to review the structure of a type-1 fuzzy logic system, describing all the blocks involved in the process of mapping each crisp input value into a crisp output. We focus our attention on multiple-input-single-output (MISO) FLSs. The reason to do so is that any multiple-input-multiple-output (MIMO) system can be decomposed into several MISO systems.

The structure of such a FLS is presented in Figure 2.5.1.

Each of the following subsections will focus on one specific part or block within the system, providing a detailed description about them.

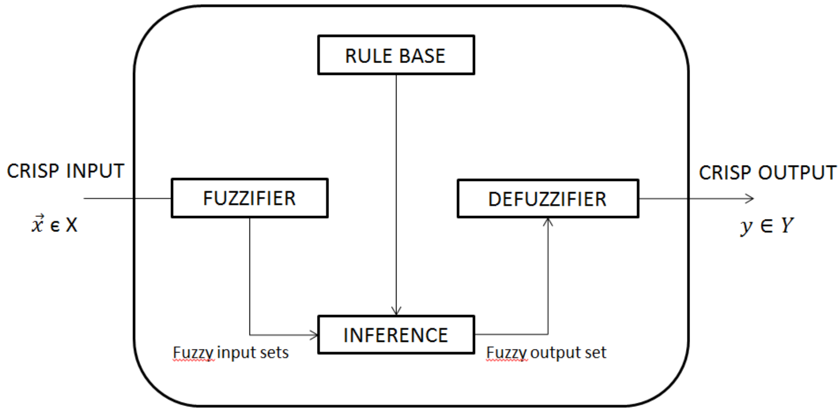

Figure 2.5.1
Structure of a type-1 fuzzy logic system.

2.5.1 *Crisp inputs to the system*

Each input to the sistem, denoted as x_i, will be defined in its own universe of discourse, X_i. Thus, the universe of discourse for the whole input space, considering p inputs, is given by the Cartesian product of all subspaces, as in the following equation:

$$X = X_1 \text{ x } X_2 \text{ x...x } X_p = \prod_{i=1}^{p} X_i \tag{2.5.1}$$

Hence, the input to the system, which is denoted as $\vec{x}$, is an element of X, i. e., $\vec{x} \in X$. Although, in general, X_i (and thus X) can be any set, in engineering problems all inputs are frequently related to numerical variables. Thus, usually X is a subset of $\mathbb{R}^n$, $X \subseteq \mathbb{R}^n$.

2.5.2 *The fuzzifier*

As it was introduced in Section 2.3, the fuzzy inference process within a T1FLS operates on type-1 fuzzy sets; nonetheless, the inputs to the system are usually real numbers. Hence, the function of the *fuzzifier* is to *map* each possible crisp input vale $\vec{x} \in X$ into a T1 fuzzy set A in X.

Depending on the type of fuzzy set each component x_i of the input vector $\vec{x}$ is mapped to, two different fuzzifiers can be distinguished:

- On the one hand, the *singleton fuzzifier* maps each input value x_i into a *fuzzy singleton*, which is a type-1 fuzzy set having a non-null membership value only at one point in the domain, $x_i = x_i'$; it is, if A_i is a fuzzy singleton in X_i, then its membership function $\mu_{A_i}(x_i)$ would be as:

$$\mu_{A_i}(x_i) = \begin{cases} 1 & \text{if } x_i = x_i' \\ 0 & \text{otherwise} \end{cases} \tag{2.5.2}$$

- On the other hand, the *non-singleton fuzzifier* maps each input component into an arbitrary fuzzy set. However, in real world applications those fuzzy sets are usually fuzzy numbers: if the crisp input value is x_i', then the associated input MF $\mu_{A_i}(x_i)$ has a membership value equal to 1 at x_i', and it decreases as x_i moves away from x_i'.

Some examples of input fuzzy sets, both singleton and non-singleton, are depicted in Figure 2.5.2.

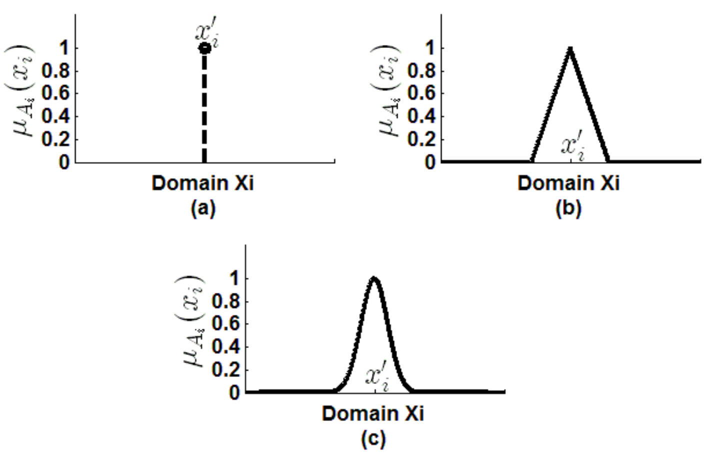

Figure 2.5.2
Different types of fuzzy inputs: (a) Fuzzy singleton. (b) Triangular MF. (c) Gaussian MF.

Once every single input has been fuzzified into a fuzzy set (whether it is singleton or non-singleton fuzzification) the resulting sets take a step forward to the inference engine. But firstly, we will take a look at the rule base.

2.5.3 *The rule base*

As it was introduced in Section 2.3, fuzzy logic systems (as human knowledge) can be specified by means of a set of IF-THEN rules, as in Equation(2.3.1), where p is the *antecedent* fuzzy proposion, and q is the fuzzy *consequent*, and both p, q can be compound fuzzy propositions built by means of logical connectives as explained in Section 2.3. For instance, assume a FLS having three inputs $x_1 \in X_1$, $x_2 \in X_2$ and $x_3 \in X_3$, and one output $y \in Y$; let A_1, A_2, A_3 and C be fuzzy sets in X_1, X_2, X_3 and Y, respectively. Thus, an example of a fuzzy rule using both the AND (intersection) and OR (union) connectives could be as follows:

$$\text{IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } A_2 \text{ OR } x_3 \text{ is } A_3 \text{ THEN } y \text{ is } C \quad (2.5.3)$$

However, in engineering applications it is customary to express the rule base using only the AND operator in the antecedents; hence, a FLS having p inputs, and a given rule base with M rules, the m-th rule (denoted as R^m), $m = 1, ..., M$, would be expressed in a standard format as:

$$R^m: \text{ IF } x_1 \text{ is } A_1^m \text{ AND } x_2 \text{ is } A_2^m \text{ AND ... AND } x_p \text{ is } A_p^m \text{ THEN } y \text{ is } C^m \quad (2.5.4)$$

Where $x_i \in X_i$, $i = 1, ..., p$ are each of the inputs to the system; A_i^m, $m = 1, ..., M$, is the antecedent related to input i for rule m; and C^m is the consequent for that rule.

It is worthwhile mentioning that different types of type-1 FLSs can be specified depending on the type of consequents used within the rules: Mamdani FLSs [66] use fuzzy sets as consequents (including fuzzy singletons), whereas Takagi-Sugeno-Kang (TSK) [91] systems use as consequents a polynomial function of the inputs. Hence, a rule from a TSK FLS would be as:

$$R^m: \text{ IF } x_1 \text{ is } A_1^m \text{ AND } x_2 \text{ is } A_2^m \text{ AND ... AND } x_p \text{ is } A_p^m \text{ THEN } y \text{ is } F(x_1, ..., x_p) \quad (2.5.5)$$

A TSK is said to be of order N if $F(x_1, ..., x_p)$ is a polynomial of order N in the input variables. It is intuitive to see that a zero-order TSK FLS and a Mamdani FLS using crips sets as consequents are completely equivalent.

2.5.4 *Inference engine*

The inference engine (IE) is the core part within a fuzzy logic system, as it is in charge of mapping the input fuzzy sets (coming from the fuzzification stage) into the output fuzzy set. This inference process is performed for each rule, and is comprised of two different steps:

1. Firstly, a matching process is carried out, to obtain the activation degree of each antecedent A_i^m. This operation basically consists of finding the intersection between the input fuzzy sets and their correspondint antecedents. Afterwards, those antecedent activation degrees are combined by means of a T-norm (implementing the AND operation) in order to obtain the *rule firing strength*, similarly to Equation (2.3.20).

2. Secondly, the rule firing strength is used along with the consequent set to perform the *implication operation*, as described in Section 2.3.

The result of the inference process is a fuzzy set (including fuzzy singletons or crisp numbers) per fired rule, usually denoted as B^l and characterised by its membership function $\mu_{B^l}(y)$. Those sets are later combined in the defuzzification stage before choosing a representative crisp number as the output of the system.

2.5.5 *Defuzzification*

As it has been discussed in the previous Section, the output of the inference engine is a fuzzy set per rule B^l, which is the result of applying the approximate reasoning process. However, real world systems usually require a crisp value as its output; hence, one more step is required in order to map these output T1 fuzzy sets into a crisp number, which is representative of the whole inference process. That operation is known as *defuzzification*.

The defuzzification operation produces a crisp output from the rule output fuzzy sets, and thus, it completes the input-output mapping performed by the fuzzy logic system. How these sets are combined to produce a single number which is representative of the whole inference process depends on the chosen defuzzification strategy.

Many different defuzzification procedures have been defined in the literature, many of them summarised in [56], also in [78] and [101], and itemised in the following in alphabetical order of acronyms:

- Adaptive integration (AI).
- Basic defuzzification distributions (BADD).
- Bisector of area (BOA).
- Constraint decision defuzzification (CDD).

- Centre of area (COA).
- Centre of gravity (COG).
- Extended centre of area (ECOA).
- Extended quality method (EQM).
- Fuzzy clustering defuzzification (FCD).
- Fuzzy mean (FM).
- First of maximum (FOM).
- Generalised level set defuzzification (GLSD).
- Indexed centre of gravity (ICOG).
- Influence value (IV).
- Last of maximum (LOM).
- Mean of maxima (MeOM).
- Middle of maximum (MOM).
- Quality method (QM).
- Random choice of maximum (RCOM).
- Semi-linear defuzzification (SLIDE).
- Weighted fuzzy mean (WFM).

According to [56], the maxima methods are good choices for fuzzy reasoning systems, as the point chosen is always one having the highest membership value, whereas on the other hand, distribution and area methods are more suitable for fuzzy control as they provide good propoerties in terms of continuity and smooth output surface. In our context we are mostly interested in the application of fuzzy logic systems to general engineering; hence, we are usually interested in defuzzification methods with one requirement: *computational simplicity*. In this regard, TSK FLSs usually implement the *weighted sum* and *weighted average* methods, wheras for Mamdani systems, some of the most relevant defuzzifiers are introduced subsequently.

2.5.5.1 *Centroid defuzzifier*

The centroid defuzzifier [78] combines the rule output fuzzy sets B^m using the fuzzy union, implemented by a given T-conorm, to provide the final output fuzzy set B, as in the following equation:

$$B = \bigcup_{m=1}^{M} B^m \iff \mu_B(y) = \mathcal{S}_{m=1}^{M} \mu_{B^m}(y) \tag{2.5.6}$$

The fuzzy set B, described by its membership function $\mu_B(y)$, is then discretised in its domain into N points, and the output provided by the centroid defuzzifier is as follows:

$$y_c(\vec{x}) = \frac{\sum_{i=1}^{N} y_i \mu_B(y_i)}{\sum_{i=1}^{N} \mu_B(y_i)} \tag{2.5.7}$$

Nonetheless, the centroid defuzzifier is computationally expensive as it requires to previously compute the fuzzy union given by Equation (2.5.6), which has led researchers to use different defuzzifiers.

2.5.5.2 *Centre-of-sums defuzzifier*

This method combines the rule output fuzzy sets by adding them (using fuzzy addition), i.e.:

$$\mu_B = \sum_{m=1}^{M} \mu_{B^m}(y) \tag{2.5.8}$$

And then computes the centroid of such set, using Equation (2.5.7). The final expression of this method can be presented as follows:

$$y_a(\vec{x}) = \frac{\sum_{m=1}^{M} c_{B^m} a_{B^m}}{\sum_{m=1}^{M} a_{B^m}} \tag{2.5.9}$$

Where c_{B^m} and a_{B^m} are the centroid and the area of the set B^m, respectively.

2.5.5.3 *Height defuzzifier*

The height defuzzifier replaces each rule output set by a singleton $\bar{y}^m$ at the point having maximum membership value (if there are more than one point having the maximum value, their mean can be taken as $\bar{y}^m$). Afterwards, the centroid of the fuzzy set comprised by these singletons is provided as the final output:

$$y_h(\vec{x}) = \frac{\sum_{m=1}^{M} \bar{y}^m \mu_{B^m}(\bar{y}^m)}{\sum_{m=1}^{M} \mu_{B^m}(\bar{y}^m)} \tag{2.5.10}$$

2.5.5.4 *Centre-of-sets defuzzifier*

This method replaces each rule consequent C^m by its centroid c^m (an operation that can be done ahead of time), and then the centroid of the fuzzy set comprised by these singletons is provided as the output:

$$y_{cos}(\vec{x}) = \frac{\sum_{m=1}^{M} c^m \mathcal{T}_{i=1}^{p} \mu_{A_i^m}(x_i)}{\mathcal{T}_{i=1}^{p} \mu_{A_i^m}(x_i)} \tag{2.5.11}$$

Where $\mathcal{T}$ represents the given T-norm chosen for the intersection operation.

2.6 INTRODUCTION TO TYPE-2 FUZZY LOGIC

One of the main criticisms that type-1 fuzzy sets theory received since it was firstly proposed by Zadeh was related to type-1 membership functions and how they are created. Despite the theory's efforts were mainly focused on the concepts of *uncertainty* and *imprecission*, many authors argued that assigning a crisp number (which is extremely precise) to a membership degree was not a proper way of representing uncertainty.

In [78], Mendel provides several plausible sources of uncertainties that can occur in a FLS:

- Fuzzy rules can be described by means of *linguistic terms*, i.e., words, which are inherently imprecise or vague, a fact that is well collected in the adage *words mean different things to different people*. This vagueness affects both antecedents and consequents.
- Uncertainty is also present during the measurement process to obtain the input values. This uncertainty is related both with input noise and errors associated with the measurement process itself.
- The last source of uncertainty is related to the data used to train or tune a FLS, which might be noisy, polluted or corrupted.

Also in [78] type-1 fuzzy logic systems are stated to be unable to handle uncertainty, in the sense of modelling and minimising its effect. Hence, in order to overcome all this limitations and criticisms about type-1 fuzzy sets theory, Zadeh introduced in [113] the concept of type-2 fuzzy sets, whose main characteristic is that their membership values are themselves fuzzy. These sets, their definitions and related concepts are extensively discussed in the subsequent subsections.

2.6.1 *The concept of type-2 fuzzy sets*

In [113] Zadeh firstly introduced "fuzzy sets with fuzzy membership functions". In his own words, considering these sets "is motivated by the close association which exists between the concept of a linguistic truth with truth-values" and "fuzzy sets in which the grades of membership are specified in linguistic terms". Having this thoughts in mind, he provided the first notion of a type-n fuzzy set [113]:

Definition 2.6.1. Type-n fuzzy set: a fuzzy set is of type n, $n = 2, 3, ...$ if its membership function ranges over fuzzy sets of type $n - 1$. The membership function of a fuzzy set of type-1 ranges over the interval $[0, 1]$.

Nevertheless, more modern and formal definitions have been proposed to introduce the concept of type-2 fuzzy sets, as the one given by Karnik and Mendel in [46]:

Definition 2.6.2. A T2FS A is characterised by a type-2 membership function $\mu_A(x, u)$, where $x \in X$, $u \in J_x \subseteq [0, 1]$:

$$A = \{(x, \mu_A(x, u)) \mid x \in X,\ u \in J_x \subseteq [0, 1]\} \tag{2.6.1}$$

Where x is the *primary variable*, u is the *secondary variable*, J_x is the *primary membership* of x, and $\mu_A(x, u)$ is referred to as the *secondary membership* of x.

Or the one provided by Bustince et al. in [10]:

Definition 2.6.3. Type-2 fuzzy set: let X be a non-empty universe of discourse, and let $FS([0,1])$ be the class of all type-1 fuzzy sets defined over the unit interval $[0,1]$. Hence, a type-2 fuzzy set A defined over X is a mapping as:

$$A : X \mapsto FS([0,1]) \tag{2.6.2}$$

All these notions and definitions lead to the same idea: type-2 fuzzy sets assign to each element in the universe of discourse $x \in X$ a fuzzy membership value in $[0,1]$. Other representations of type-2 fuzzy sets can be found in the literature, such as:

$$A = \sum_i \mu_A(x_i)/x_i \tag{2.6.3}$$

$$A = \int_{x \in X} \mu_A(x)/x = \int_{x \in X} \int_{u \in J_x} \mu_A(x, u)/(x, u) \tag{2.6.4}$$

Equation (2.6.3) is used for discrete universes of discourse, whereas Equation (2.6.4) is preferred when X is continuous. The operands $\sum$ and $\int$ denote union over all admisible values, respectively, and $\cdot/\cdot$ represents ordered pairs.

2.6.2 *Concepts and definitions related to type-2 fuzzy sets*

In this subsection attention is focused on other definitions related to type-2 fuzzy sets:

Definition 2.6.4. Secondary membership function: let A be a T2FS represented by its membership function $\mu_A(x, u)$. Then, at a given $x = x'$, the function $\mu_A(x = x', u)$, which depends only on u, i.e., $\mu_A(x = x')(u) = f_{x'}(u)$, is called the *secondary membership function* at $x = x'$.

$$\mu_A(x = x', u) = \mu_A(x') = \int_{u \in J'_x} f_{x'}(u)/u \qquad J'_x \subseteq [0,1] \tag{2.6.5}$$

Where $0 \leqslant f_{x'}(u) \leqslant 1$. This secondary membership function is usually referred to as a *vertical slice* of A at $x = x'$.

Definition 2.6.5. Secondary grade: a secondary membership function $\mu_A(x = x', u)$ evaluated at a given $u = u' \in J_x$ is called a *secondary grade* at (x', u'), it is, the membership function evaluated at $\mu_A(x = x', u = u')$. Secondary grades are also called frequently *secondary membership values*.

Definition 2.6.6. Footprint Of Uncertainty (FOU): let A be a T2FS defined over a universe of discourse X. Hence, the *footprint of uncertainty* of A, denoted as $\text{FOU}(A)$, is the union of the primary memberhips $\forall x \in X$, i.e.:

$$\text{FOU}(A) = \bigcup_{x \in X} J_x \tag{2.6.6}$$

Where the primary membership is as introduced in Definition 2.6.2.

The FOU is a very descriptive element of a T2FS, as it provides a very convenient representation of how the uncertainty is distributed within a set in terms of the support of the fuzzy set.

Definition 2.6.7. Embedded type-2 fuzzy set: let A be a T2FS defined over a universe of discourse X. Hence, an *embedded type-2 fuzzy set* of A, denoted as $\tilde{A}_e$, is as in Equation (2.6.7) and Equation (2.6.8) for continuous and discrete universes of discourse, respectively.

$$\tilde{A}_e = \int_{x \in X} [\mu_A(x, \theta)/\theta]/x \qquad \theta \in J_x \subseteq [0, 1] \tag{2.6.7}$$

$$\tilde{A}_e = \sum_{i} [\mu_A(x_i, \theta)/\theta]/x \qquad \theta \in J_x \subseteq [0, 1] \tag{2.6.8}$$

It is, at each value x, A_e has only one primary membership, namely θ, and one secondary grade associated, $\mu_A(x, \theta)$. Note that both Equations (2.6.7) and (2.6.8) *do not* perform the union over all admisible values in the primary membership. Besides, it is worhtwhile mentioning that A_e is a T2FS which is *embedded* (included) in A.

Definition 2.6.8. Embedded type-1 fuzzy set: let A be a T2FS defined over a universe of discourse X. Hence, an *embedded type-1 fuzzy set* of A, denoted as A_e, is as in Equation (2.6.9) and Equation (2.6.10) for continuous and discrete universes of discourse, respectively.

$$A_e = \int_{x \in X} \theta/x \qquad \theta \in J_x \subseteq [0, 1] \tag{2.6.9}$$

$$A_e = \sum_i \theta/x_i \qquad \theta \in J_x \subseteq [0,1] \tag{2.6.10}$$

Again, it is worthwhile to highlight that both Equations (2.6.9) and (2.6.10) *do not* perform the union over all admisible values of $x \in X$. Besides, it is easy to see that A_e is a T1FS.

Embedded type-1 and type-2 fuzzy sets are very useful to obtain some theoretical results regarding operations on type-2 fuzzy sets [78].

Definition 2.6.9. Type-1 fuzzy set represented as a type-2 fuzzy set: let A be a T1FS defined as in Equation (2.1.3) and characterised by its membership function $\mu_A(x)$. Hence, A can be represented as a type-2 fuzzy set as follows:

$$A = \int_{x \in X} [1/\mu_A(x)]/x \qquad x \in X \tag{2.6.11}$$

Or, in the discrete case:

$$A = \sum_i [1/\mu_A(x_i)]/x_i \qquad x \in X \tag{2.6.12}$$

It is, A is a type-2 fuzzy set having at each x one single point as the primary membership, given by $\mu_A(x)$, at which the secondary grade is equal to 1.

Definition 2.6.10. Type-2 singleton: a T2FS is said to be a singleton if it has only one point with non null membership value, i.e., $\mu_A(x) = 1/1$ if $x = x'$ and $\mu_A(x) = 1/0$ elsewhere.

Definition 2.6.11. Interval type-2 fuzzy set: let A be a T2FS as described in Equation (2.6.1). Hence, if $\mu_A(x,u) = 1$ $\forall x \in X$, $\forall u \in J_x$, then A is an interval type-2 fuzzy set (IT2FS) [78].

$$A = \{(x, \mu_A(x,u)) \,|\, x \in X,\ u \in J_x \subseteq [0,1],\ \mu_A(x,u) = 1\} \tag{2.6.13}$$

IT2FSs can be viewed as a particular case of T2FSs, which are usually referred to as "General T2FSs" in order to make explicit difference between the former and the latter. These IT2FSs have at each $x \in X$ a secondary membership function consisting of a closed and connected interval with height equal to 1. This interval represents that uncertainty is uniformly distributed across the primary membership. As the secondary grades are always equal to unity, it is redundant, and such an interval can be uniquely represented by its two extreme points, namely $[l_x, r_x]$. Hence, this interval representation of the primary membership leads us to the following definitions:

Definition 2.6.12. Upper and lower membership functions of an IT2FS: let A be an IT2FS as described in Equation (2.6.13), having at each x a primary membership given by an interval as $[l_x, r_x]$. Hence, the *upper membership function* of A (denoted as $UMF(A)$ or $\overline{\mu}(x)$) is given by Equation (2.6.14), whereas the *lower membership function* of A (denoted as $LMF(A)$ or $\underline{\mu}(x)$) is given by Equation (2.6.15)

$$UMF(A) = \overline{\mu}(x) = \{r_x \mid x \in X\} \tag{2.6.14}$$

$$LMF(A) = \underline{\mu}(x) = \{l_x \mid x \in X\} \tag{2.6.15}$$

The fact that the secondary grades of IT2FSs are always either 0 or 1 (depending whether the point belongs or not to the primary membership) significantly reduces the amount of computation effort required to operate with them. This has motivated researchers to focus their attention on IT2FLSs, to the detriment of their GT2 counterparts. Nonetheless, in this dissertation both GT2 and IT2 FLSs are considered.

Figure 2.6.1 depicts some of the T2FSs defined along this subsection. It is worthwhile mentioning that in Figure 2.6.1(d), as it represents an IT2FS, the third dimension is always equal to 1 and has not been plotted.

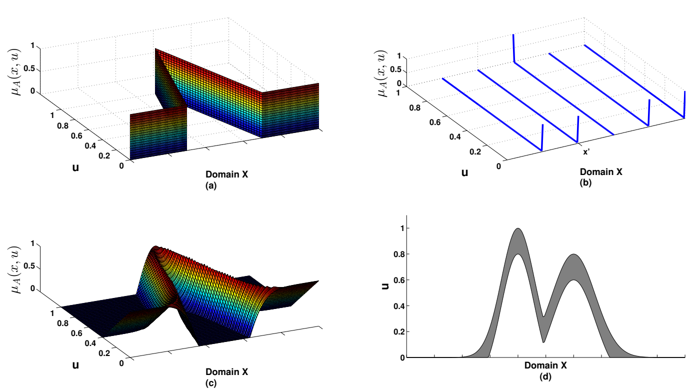

Figure 2.6.1
Different types of T2FSs: (a) T1FS represented as a T2. (b) T2 singleton. (c) Arbitrary T2FS. (d) IT2FS.

And last but not least, the definition of the α-plane of a T2FS is presented, which was introduced in [61] and later slightly corrected in [70]. This concept is an extension of the α-cut for T1FSs as presented in Definition 2.1.9, and is as follows:

Definition 2.6.13. α-plane: let A be a T2FS defined over a universe of discourse X and characterised by its membership function $\mu_A(x)$, $x \in X$, and let α be a real number such that $\alpha \in [0, 1]$. Hence, the α-plane of A, denoted as A_α, is the union of all primary membership whose secondary grades are greater than or equal to the specific value of α, i.e.:

$$A_\alpha = \bigcup_{x \in X} (x, u) \mid \mu_A(x, u) \geqslant \alpha \tag{2.6.16}$$

For the specific case in which $\alpha = 0$, then A_0 is defined as the FOU of A, it is:

$$A_0 = \mathrm{FOU}(A) \tag{2.6.17}$$

It is worthwhile to highlight that A_0 is also the closed support of A.

Each of these α-planes have associated a T2FS, which is as follows:

Definition 2.6.14. Associated type-2 fuzzy set of the α-plane A_α: let $I_{A_\alpha}(x, u)$ be the indicator function of the α-plane A_α, i.e.:

$$I_{A_\alpha}(x, u) = \begin{cases} 1 & \text{if} \quad (x, u) \in A_\alpha \\ 0 & \text{if} \quad (x, u) \notin A_\alpha \end{cases} \tag{2.6.18}$$

Hence, the associated type-2 fuzzy set of the α-plane A_α is given by:

$$A(\alpha) = (x, u),\ \mu_{A_\alpha}(x, u) \mid \forall x \in X,\ \forall u \in J_x \subseteq [0, 1] \tag{2.6.19}$$

Where $\mu_{A_\alpha}(x, u) = \alpha I_{A_\alpha}(x, u)$.

Definition 2.6.13 is quite important because type-2 fuzzy sets can be represented in terms of their α-planes through the *α-plane representation theorem*, which is as follows:

Theorem 2.6.1. ***α-plane Representation Theorem**: a T2FS A can be represented as the union of its associated type-2 fuzzy sets $A(\alpha)$ as:*

$$A = \bigcup_{\alpha \in [0,1]} A(\alpha) \tag{2.6.20}$$

In which

$$\mu_A(x, u) = \arg \max_{\alpha \in [0,1]} \{\mu_{A(\alpha)}(x, u)\} \tag{2.6.21}$$

For a complete proof ot Theorem 2.6.1, please refer to [61].

A concept similar to the α-planes are the zSlice based type-2 fuzzy sets, introduced in [95] and later extended in [97], where FLSs based on zSlices were also presented. These zSlices aim to represent type-2 fuzzy sets by *slicing* the third dimension (the z-axis).

2.7 OPERATIONS ON TYPE-2 FUZZY SETS

As it was introduced in the previous Section, type-2 fuzzy sets are an extension of type-1 fuzzy sets. Hence, in the same way operations on type-1 fuzzy sets were extended from their crisp counterparts, the same process can be taken one step further, and set theoretic operations on type-2 fuzzy sets can be extended from their type-1 analogues.

Zadeh provided a mathematical tool in order to obtain the set theoretic operations on T2FSs as an extension of their equivalent T1 operations. Such a tool is named the *Extension Principle*, which is introduced in the following, before tackling the union and intersection on T2FSs.

2.7.1 *The Extension Principle*

The *Extension Principle* is one of the most basic concepts and most useful mathematical tools in fuzzy sets theory. It can be used to generalise crisp mathematical concepts and operations to fuzzy sets. A vague notion was proposed by Zadeh in his original paper [117], which was later formalised and modified in [113], [119] and [27]. Its formal definition is as follows [124]:

Definition 2.7.1. The Extension Principle: let X be a universe of discourse given by the Cartesian product $X_1 \,x \dots x\, X_r$, and let $A_1, \dots, A_r$ be fuzzy sets in $X_1 \,x \dots x\, X_r$, respectively. Moreover, let $f : X \mapsto Y$ be a mapping such that $f(x_1, \dots, x_r) = y \in Y$. Hence, the *Extension Principle* allows us to *induce* a fuzzy set B on Y *through* f, i.e., $B = f(A_1, \dots, A_r)$ as:

$$B = \{(y, \mu_B(y)) \,|\, y = f(x_1, \dots, x_r),\; (x_1, \dots, x_r) \in X\} \tag{2.7.1}$$

Such that:

$$\mu_B(y) = \begin{cases} \sup\limits_{(x_1,...,x_r)\in f^{-1}(y)} (min\{\mu_{A_1}(x_1),...,\mu_{A_r}(x_r)\}) & \text{if} \quad f^-1(y) \neq \emptyset \\ 0 & otherwise \end{cases} \tag{2.7.2}$$

Where $f^{-1}(y)$ denotes all points $x_1 \in X_1,...,x_r \in X_r$ such that $y = f(x_1,...,x_r)$, and the suprema is usually implemented with the maximum t-conorm. It is worthwhile mentioning that the Extension Principle has been modified using sumation rather than suprema, and product rather than minimum in [27]. Nonetheless, it is usually used as in Definition 2.7.1.

The Extension Principle as presented above, when using the maximum operation to implement the suprema, and a general t-nom $\star$ instead of the minimum, can be written as follows:

$$f(A_1,...,A_r) = \int_{x_1\in X_1} ... \int_{x_r\in X_r} \mu_{A_1}(x_1)\star...\star\mu_{A_r}(x_r)/f(x_1,...,x_r) \tag{2.7.3}$$

This principle in Equation 2.7.3, allows us to extend the set theoretic operations of union and intersection to T2FSs. To do so, two new operators, named *join* and *meet*, are introduced, which are intensively described in [49].

In the following subsections, two T2FSs are considered, $F_1(x)$, and $F_2(x)$, both defined over the same universe of discourse X such that $x \in X$. Their membership values are given by Equations (2.7.4) and (2.7.5), respectively:

$$\mu_{F_1}(x) = \int_v f_1(v)/v \tag{2.7.4}$$

$$\mu_{F_2}(x) = \int_w f_2(w)/w \tag{2.7.5}$$

In the specific case of IT2FSs, Equations (2.7.4) and (2.7.5) become Equations (2.7.6) and (2.7.7), respectively.

$$\mu_{F_1}(x) = \int_{v\in[l_{F_1},r_{F_1}]} 1/v \tag{2.7.6}$$

$$\mu_{F_2}(x) = \int_{w\in[l_{F_2},r_{F_2}]} 1/w \tag{2.7.7}$$

2.7.2 *Intersection operation on type-2 fuzzy sets*

Using the Extension Principle as in Definition 2.7.1, the intersection between two T2FSs can be defined as follows:

Definition 2.7.2. Intersection of type-2 fuzzy sets: let F_1 and F_2 be two type-2 fuzzy sets as in Equations (2.7.4) and (2.7.5), respectively. Hence, the intersection of F_1 and F_2, denoted as $F_1 \cap F_2$ and characterised by its membership function $\mu_{F_1 \cap F_2}(x)$, is given by [49]:

$$F_1 \cap F_2 \iff \mu_{F_1 \cap F_2}(x) = \mu_{F_1}(x) \sqcap \mu_{F_2}(x) = \int_v \int_w (f_1(v) \star f_2(w))/(v \star w) \tag{2.7.8}$$

Where $\sqcap$ denotes the *meet* operator, $\star$ represents a given t-norm, and integrals indicate logical union over all admisible values.

A special case of Definition 2.7.2 arises when we consider the particular case of IT2FSs. For those sets, the intersection operation is obtained as:

Definition 2.7.3. Intersection of interval type-2 fuzzy sets: let F_1 and F_2 be two interval type-2 fuzzy sets as in Equations (2.7.6) and (2.7.7), respectively. Hence, the intersection of F_1 and F_2, denoted as $F_1 \cap F_2$ and characterised by its membership function $\mu_{F_1 \cap F_2}(x)$, is given by:

$$F_1 \cap F_2 \iff \mu_{F_1 \cap F_2}(x) = \mu_{F_1}(x) \sqcap \mu_{F_2}(x) = \int_{u = v \wedge w \in [l_{F_1} \star l_{F_2}, r_{F_1} \star r_{F_2}]} 1/u \tag{2.7.9}$$

Equation (2.7.9) reveals that performing the intersection operation on IT2FSs only requires to do simple operations between intervals, significantly reducing the computational complexity when compared with GT2FSs.

2.7.3 *Union operation on type-2 fuzzy sets*

Similarly to the intersection, the union operation on type-2 fuzzy sets can be defined via the Extension Principle, as subsequently presented:

Definition 2.7.4. Union of type-2 fuzzy sets: let F_1 and F_2 be two type-2 fuzzy sets as in Equations (2.7.4) and (2.7.5), respectively. Hence, the union of F_1 and F_2, denoted as $F_1 \cup F_2$ and characterised by its membership function $\mu_{F_1 \cup F_2}(x)$, is given by [49]:

$$F_1 \cup F_2 \iff \mu_{F_1 \cup F_2}(x) = \mu_{F_1}(x) \sqcup \mu_{F_2}(x) = \int_v \int_w (f_1(v) \star f_2(w))/(v \vee w)$$

(2.7.10)

Where $\sqcup$ denotes the *join* operator, $\star$ represents a given t-norm, $\vee$ depicts the maximum operation and integrals indicate logical union over all admisible values.

Definition 2.7.4 can be particularised to IT2FSs, in which case the union operation is as follows:

Definition 2.7.5. Union of interval type-2 fuzzy sets: let F_1 and F_2 be two interval type-2 fuzzy sets as in Equations (2.7.6) and (2.7.7), respectively. Hence, the union of F_1 and F_2, denoted as $F_1 \cup F_2$ and characterised by its membership function $\mu_{F_1 \cup F_2}(x)$, is given by:

$$F_1 \cup F_1 \iff \mu_{F_1 \cup F_2}(x) = \mu_{F_1}(x) \sqcup \mu_{F_2}(x) = \int_{u=v \vee w \in [l_{F_1} \vee l_{F_2}, r_{F_1} \vee r_{F_2}]} 1/u \tag{2.7.11}$$

As in the intersection case (given in Equation (2.7.9)), the union between IT2FSs represented in Equation (2.7.11) only requires performing simple operations between intervals, which is the reason why researchers have favoured IT2FLSs in detriment of their GT2 counterparts.

2.7.4 *Negation operation on type-2 fuzzy sets: complement*

The last operation on type-2 fuzzy sets that will be defined in this work using the Extension Principle is the *negation*, as presented below:

Definition 2.7.6. Negation of a type-2 fuzzy set: let F_1 be a type-2 as in Equation (2.7.4). Hence, the negation (or complement) of F_1, denoted as $c(F_1)$, $\neg F_1$ or $\overline{F}_1$ is given by [49]:

$$c(F_1) = \neg F_1 = \overline{F}_1 \iff \mu_{c(F_1)}(x) = \mu_{\neg F_1}(x) = \mu_{\overline{F}_1}(x) = \int_u f_1(u)/(1-u) \tag{2.7.12}$$

Once again, when dealing with the simpler version of IT2FSs, the negation reduces to:

Definition 2.7.7. Negation of an interval type-2 fuzzy set: let F_1 be an interval type-2 fuzzy set as in Equation (2.7.6). Hence, the negation (or complement) of F_1, denoted as $c(F_1)$, $\neg F_1$ or $\overline{F}_1$ is given by:

$$c(F_1) = \neg F_1 = \overline{F}_1 \iff \mu_{c(F_1)}(x) = \mu_{\neg F_1}(x) = \mu_{\overline{F}_1}(x) = \int_u 1/(1-u) \quad (2.7.13)$$

2.8 TYPE-2 FUZZY LOGIC SYSTEMS AS UNIVERSAL APPROXIMATORS

As introduced in Section 1.1, and summarised in [10], although T2FLSs were defined and theoretically studied since 1971, they did not gain significant relevance until the early 2000s, when Karnik and Mendel [46] [78] focused their efforts on devolping the theory and practical implementations of IT2FLSs. Hence, studying the properties of these systems as universal approximators has not been widely addressed. In words of Hao Ying [107], "an approximation theory for T2 fuzzy systems is still in its infancy".

Ying himself presented the first approach to this field in 2008 [106], where he proved that a general class of Mamdani IT2FLSs are universal approximators, in the sense that they can uniformly approximate any real continuous function defined on a compact domain to any degree of accuracy. In 2009, in [107], he also proved the same properties on IT2 Takagi-Sugeno (TS) FLSs having linear rule consequents. And last but not least, in 2010 [111] You and Ying provided the same proof for interval type-2 boolean fuzzy systems. Moreover, the same properties have been studied for some hybrid paradigms, such as interval type-2 fuzzy neural networks as in [14].

Nonetheless, there is still a need to further develop a formal approximation theory of T2FLSs as universal approximators.

2.9 STRUCTURE OF A TYPE-2 FUZZY LOGIC SYSTEM

In this section we present a general overview on the structure of type-2 fuzzy logic systems, describing all the blocks comprising them and paying special attention to those which present significant differences with their type-1 counterparts as presented in Secton 2.5. Once again, we focus our attention on multiple-input-single-output (MISO) FLSs. The general structure of such systems is depicted in Figure 2.9.1.

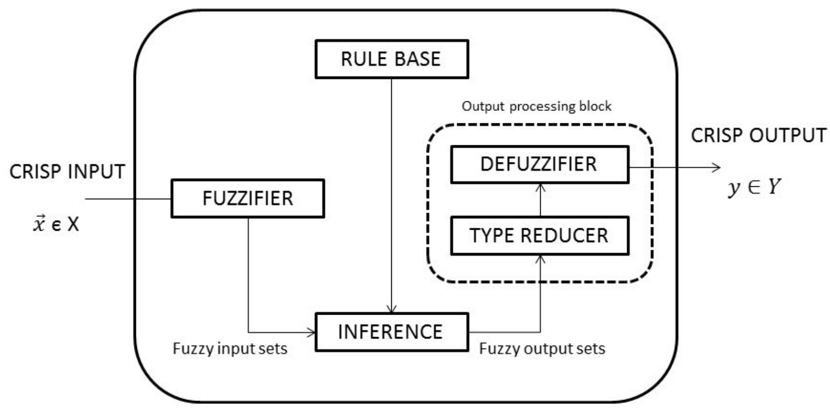

Figure 2.9.1
Structure of a type-2 fuzzy logic system.

In the following subsections we provide a detailed itemisation of each component within the system.

2.9.1 *Crisp inputs to the system*

Each input to the T2FLS, denoted as x_i, $i = 1, ...p$, is defined over its own universe of discourse, X_i, but are usually represented by a single vector as follows:

$$\vec{x} = (x_1, ..., x_p) \in X = X_1 x ... x X_p \tag{2.9.1}$$

Generally, from a theoretical point of view, each X_i can be any kind of set or universe of discourse; nevertheless, in engineering problems each x_i is usually associated with a real number, thus leading to X being a subset of $\mathbb{R}$, i.e., $X \subseteq \mathbb{R}^n$. The vector $\vec{x}$ is the input to the fuzzifier block.

2.9.2 *The fuzzifier*

As in their type-1 counterpart described in Section 2.5.2, the fuzzifier is the element in the system in charge of mapping each component x_i from the input vector $\vec{x}$ into a fuzzy set, usually denoted as $\mu_{X_i}(x_i)$, as required by the inference engine. Depending on which kind of fuzzy set we chose to map the inputs to, we can distinguish among different kinds of fuzzifiers in T2FLSs, which are presented in the following definitions:

Definition 2.9.1. A T2FLS is said to be a **singleton type-2 fuzzy logic system** if the fuzzifier block maps each input component x_i into a type-2 singleton as described in Equation (2.6.10) and depicted in Figure 2.6.1(b).

Definition 2.9.2. A given T2FLS is said to be a **type-1 non-singleton type-2 fuzzy logic system** when the fuzzifier block maps each component x_i from the input vector into a type-1 fuzzy set as described in Equation (2.1.3).

Definition 2.9.3. When the fuzzifier block within a T2FLS maps each input component x_i into an arbitraty type-2 set, which is neither a type-2 singleton or a type-1 fuzzy set (as described in Equation (2.6.1)), then it is said to be a **type-2 non-singleton type-2 fuzzy logic system**. A special case of type-2 non-singleton fuzzifier is the *interval type-2 non-singleton* case.

After the fuzzification process, the resulting sets, whether the are singletons, type-1 or type-2 fuzzy sets, move forward to the inference engine, whose operation is closely connected with the rule base.

2.9.3 *The rule base*

The structure of the rule base describing a T2FLS is identical to the one discribing a T1FLS, introduced in Section 2.5.3, where each rule R^m, $m = 1, ..., M$, is as stated in Equation (2.5.4). The only difference is that some or all the involved fuzzy sets, either in the antecedents and/or the consequents part, are of type-2 instead of type-1.

2.9.4 *The inference engine*

The inference process in T2FLSs is analogous to the one presented in Section 2.5.4 for T1FLSs, the main differences being that the involved sets are of type-2, and the operators used to implement the union, intersection and implication are the *meet* and *join* operations, rather than T-norms and T-conorms.

Consider a rule from the rule base as described in Equation (2.5.4), which is repeated here for the convenience of the reader:

$$R^m: \text{ IF } x_1 \text{ is } A_1^m \text{ AND } x_2 \text{ is } A_2^m \text{ AND ... AND } x_p \text{ is } A_p^m \text{ THEN } y \text{ is } C^m$$

The inference process is performed on each rule in two steps:

1. In order to obtain the activation degree of each antecedent A_i^m, a matching process is performed by means of the meet operation between the input fuzzy set for input i and the given antecedent, as follows:

$$\mu_{F_i^m}(x_i) = \mu_{X_i}(x_i) \sqcap \mu_{A_i^m}(x_i) \tag{2.9.2}$$

   Where $\mu_{F_i^m}$ is the antecedent activation degree, μ_{X_i} is the input fuzzy set for input i and $\mu_{A_i^m}$ is the antecedent for input i in rule m. Afterwards, all the antecedent activation degrees within a rule are combined in order to obtain the *rule firing strength*, denoted as F^m:

$$\mu_{F^m}(\vec{x}) = \mu_{F_1^m}(x_1) \sqcap ... \sqcap \mu_{F_p^m}(x_p) = \prod_{i=1}^{p} \left[\mu_{X_i}(x_i) \sqcap \mu_{A_i^m}(x_i)\right] \tag{2.9.3}$$

2. Afterwards, the rule firing strength obtained in the previous step is combined with the rule consequent C^m with the implication operator, which is also usually implemented with the meet operation, to obtain the *rule output fuzzy set*, denoted as $\mu_{B^m}(y)$:

$$\begin{aligned} \mu_{B^m}(y) &= \mu_{F^m}(\vec{x}) \sqcap \mu_{C^m}(y) = \\ &= \prod_{i=1}^{p} \left[\mu_{X_i}(x_i) \sqcap \mu_{A_i^m}(x_i)\right] \sqcap \mu_{C^m}(y) \end{aligned} \tag{2.9.4}$$

After the inference process is applied on each rule, all the resulting rule output fuzzy sets as in Equation (2.9.4) are provided to the output processing block, which will combine them somehow depending on the chosen type reduction and defuzzification strategy.

2.9.5 *Type reduction*

The output of the inference engine, as explained above and as detailed in Equation (2.9.4), is a T2FS per fired rule; however, a FLS usually requires its output to be a crisp number, in order to implement an input-output mapping and to be useful in real world applications. In the T1 counterpart, the output of the inference engine is a T1FS per rule, which are combined in the defuzzification stage, and the resulting set is mapped into a crisp output. Hence, when dealing with T2FLSs, and as depicted in Figure 2.9.1, one extra block is required before the defuzzification stage: the *type-reducer*.

This block is in charge of combining the rule ouput fuzzy sets B^m into a single T2FS B, and then mapping the latter into a T1FS which is *representative* of it. Actually, the type-reduction operation can be viewed as an *extension* [47] of the defuzzification process, via the Extension Principle. Hence, a type-reduction method can be specified for each defuzzification method defined.

For theoretical purposes, the most extended type-reduction method is the *centroid type-reducer*, which is an extension of the centroid defuzzifier; it was introduced in [48] and extensively detailed in [78]. This method combines each rule output set B^m, by means of the join operation, into a single output fuzzy set B as follows:

$$B = \bigcup_{m=1}^{M} B^m \iff \mu_B(y) = \bigsqcup_{m=1}^{M} \mu_{B^m}(y) \qquad (2.9.5)$$

Subsequently, the centroid of B is obtained. Nevertheless, this method is computationally very expensive as it requires to obtain the centroid of a huge number of enumerated embedded type-2 fuzzy sets (as in Definition 2.6.7). This drawback has motivated researchers to propose different type-reduction methods, some of which are enumerated in the following. Most of them are described in [46]:

- **Height type-reduction and modified height type-reduction:** these methods replace each rule output fuzzy set by a singleton, placed at the point having maximum primary memberhisp. The modified version includes a scaling factor to weigh those singletons according to the rules' firing strengths. Although they are computationally quite simple, both present problems when just one rule is fired [78].

- **Centre-of-sums type-reduction:** this type-reducer first uses fuzzy addition to combine all rule output fuzzy sets, and then the centroid is obtained. Its computational complexity is very similar to the centroid type-reducer.
- **Centre-of-sets (COS) type-reduction:** in this method, each rule consequent (which is in general a type-2 fuzzy set) is replaced by its centroid, an operation that can be done ahead of time. Then these centroids are combined with their respective rule firing strengths to obtain the type-1 reduced set. It has a reasonable computation complexity and, thus, it is widely used in FLSs' implementations.

It is worthwhile mentioning that the type-reduction operation significantly simplifies when dealing with IT2FLSs. Both the centroid of an IT2FS and the COS type-reducer use the well-known Karnik-Mendel (KM) algorithm, which was originally introduced in [48] and later improved in many works as [102], [87] and [72], among others.

The output of the type-reduction block is a T1FS, which is provided as the input to the final block.

2.9.6 *Defuzzification*

The defuzzification process is very similar to the stage within T1FLSs. The main difference is that this block in T2FLSs does not have to combine the rule output fuzzy sets B^m as this task has been performed in the type-reduction block. Hence, the only task needed to provide the final output of the system is choosing a crisp number which is representative of the whole type-1 reduced set. To do so, any defuzzification method, as those explained in Section 2.5.5, can be selected.

Special attention should be paid on the defuzzification stage when dealing with IT2FLSs. Whatever type-reduction method is chosen, the output of that block will be a T1FS consisting in an interval, characterised by its two endpoints $[y_l, y_r]$. Hence, the most used defuzzification method in such systems is taking the midpoint of that interval as the output, i.e.:

$$y = \frac{y_l + y_r}{2} \tag{2.9.6}$$

2.10 SUMMARY

In this Chapter, the main concepts and definitions related to both type-1 and type-2 fuzzy logic, sets and systems that are required in order to fully comprehend the rest of this dissertation have been introduced.

Each of the following Chapters will have its own introduction, in order to further contextualise the specific work presented within them; nonetheless, all related concepts and required definitions are contained here in Chapter 2.

Part II

DISCUSSION: IS TYPE-2 FUZZY LOGIC FULLY JUSTIFIED?

3

EFFECTS OF EXTRA TYPE-1 FUZZY SET PARAMETERS ON THE PERFORMANCE OF A FUZZY SYSTEM

Fuzziness is probability in disguise. I can design a controller with probability that could do the same thing that you could do with fuzzy logic.

— Professor Myron Tribus, on hearing of the fuzzy-logic control of the Sendai subway system IEEE Institute, May 1988

Many works have shown that IT2FLSs perform better than their type-1 counterparts, due to their ability to better handle and model encountered uncertainties. This has led to some controversy, in which it is argued that one reason for the superior performance of an IT2FLS over a T1FLS is that type-2 fuzzy sets have more parameters (design degrees of freedom) than do type-1 fuzzy sets. The associated hypothesis with this argument is that if the type-1 fuzzy sets were allowed to have the same number of parameters (and hence the same number of design degrees of freedom) as the type-2 fuzzy sets, and if both type-2 and type-1 fuzzy sets were given equal opportunities of optimisation, then both type-2 and type-1 FLSs should end up with equal performances. In this part of the dissertation, we address this claim in a novel way by investigating whether or not providing type-1 fuzzy sets with one extra parameter enables them to improve their function approximation and uncertainty handling performance totally within the framework of T1FLSs, because if the claim is not valid in that framework it cannot be valid for the IT2 versus T1 FLSs framework. We show, by means of extensive simulation studies for several function approximation problems, that adding one parameter to each of the antecedent fuzzy sets in a type-1 FLS (so that triangle membership functions become trapezoid membership functions) does not enable that FLS to improve its function approximation and uncertainty handling performances. In light of these results, it seems reasonable to conclude that the ability of IT2FLSs to perform better than their T1 counterparts is due to the way these systems handle uncertainty, rather than the number of degrees of freedom available in the system.

3.1 INTRODUCTION

Type-1 and type-2 fuzzy sets have been widely used in many different areas since their introduction by Zadeh [113]. Type-2 Fuzzy Logic Systems (FLSs), introduced by Karnik, et al. [50] and further explained in [78], include interval type-2 FLSs (IT2FLSs) and general type-2 FLSs (GT2FLSs). Although there is no formal definition for "GT2FLSs", the term is widely used to differentiate IT2FLSs (which use interval type-2 fuzzy sets) and the rest of type-2 FLSs, which use general (non-interval) shapes of type-2 fuzzy sets.

In recent years, T2FLSs have received increasing attention [75], [81], [18], [30], [43], [15], [16], [41], [60], [63], [68], [79] and have been applied successfully in many fields, due to their better ability to model and handle uncertainties than their type-1 counterparts. Although it has been shown in numerous simulation studies that IT2FLSs outperform their T1 counterparts, the reason for this superior performance has not been established mathematically. This has led to some controversy, in which it is argued that one reason for the superior performance of an IT2FLS over a T1FLS is that type-2 fuzzy sets have more parameters (design degrees of freedom) than do type-1 fuzzy sets. The associated hypothesis with this argument is that if the type-1 fuzzy sets were allowed to have the same number of parameters (and hence the same number of design degrees of freedom) as the type-2 fuzzy sets, and if both the type-2 and type-1 fuzzy sets were given equal opportunities of optimisation, then the type-2 and type-1 FLSs should end up with equal performances. The reasoning behind this argument is that there is a common thinking that increasing the number of available parameters to be optimised in a system always implies an improvement in system performance.

In previous work [11] Cara et al. compared a singleton IT2FLS with a non-singleton T1FLS for several function approximation problems (using multi-objective optimisation and genetic algorithms (GA)), and allowed both FLSs to have the same number of parameters to be optimised. They showed that, in spite of optimising the same number of parameters for the singleton IT2FLS and non-singleton T1FLS, the singleton IT2FLS outperformed the non-singleton T1FLS. Consequently, they concluded that the better performance of IT2FLSs is due to their structure and ability to model and handle uncertainties rather than the number of degrees of freedom available for optimisation [11].

In this Chapter, in order to plenty justify the use of T2FLSs in detriment of T1FLSs, we address this controversy further by investigating whether or not providing type-1 fuzzy sets with one extra parameter enables them to improve their function approximation and uncertainty handling performance. To do this, we use single and multi-objective GAs and optimise two different kinds of T1 FLSs for nine function approximation problems. The first T1FLS uses normal triangle membership functions (MFs) for the antecedents, whereas the second T1FLS uses normal trapezoid MFs for the antecedents.

The choice of GAs has not been without reason: we intended to use a semi-random search algorithm to optimise each FLS whose implementation would not favour any specific model, in order to guarantee impartiality in the comparison between systems. As we are comparing different systems uniquely represented by their parameters, the use of chromosomes to represent systems/individuals arises naturally, hence GAs where chosen over other bio-inspired algoritms.

The rest of the Chapter is organised as follows: Section 3.2 presents a brief overview of the T1FLSs that are used in the sequel; Section 3.3 describes the single and multi-objective experiments; Section 3.4 explains the procedure used to obtain the centroid consequent's parameters; Section 3.5 presents the experiments and results; and Section 3.6 presents conclusions and recommendations for future work. Details about our GA algorithms are in Appendix A.

3.2 A BRIEF OVERVIEW OF THE EMPLOYED TYPE-1 FUZZY LOGIC SYSTEMS

The T1FLSs used in this Chapter are Takagi-Sugeno-Kang (TSK) $A1 - C0$ systems [65], which means that type-1 fuzzy sets are used for the antecedent MFs, whereas singletons (crisp numbers) are used for the consequents. This structure is the same as using a Mamdani T1FLS with centre-of-sets or height defuzzifiers. Two different types of $A1 - C0$ FLSs are used: triangle and trapezoid, which employ triangle and trapezoid fuzzy sets, respectively. In the sequel, the T1FLS that uses triangle [trapezoid] MFs is referred to as a triangle [trapezoidal] T1FLS.

The triangle FLS uses normal triangle MFs in its rule-antecedents. These MFs are described by three parameters, because normality constrains the height of the triangle to be unity. Given a universe of discourse $X \subseteq \mathbb{R}$, a triangle MF describing a type-1 fuzzy set A in X is described by three real numbers, (a, b, c), such that $a \leqslant b \leqslant c$, i.e.:

$$\mu_A(x; a, b, c) = \begin{cases} 0 & \text{if} \quad x \leqslant a \\ \frac{x-a}{b-a} & \text{if} \quad a \leqslant x \leqslant b \\ \frac{c-x}{c-b} & \text{if} \quad b \leqslant x \leqslant c \\ 0 & \text{if} \quad x \geqslant c \end{cases} \tag{3.2.1}$$

For the trapezoid FLS, the normal MF $\mu_A(x)$ is described by four real numbers (a, b, c, d), such that $a \leqslant b \leqslant c \leqslant d$, i.e.:

$$\mu_A(x; a, b, c, d) = \begin{cases} 0 & \text{if} \quad x \leqslant a \\ \frac{x-a}{b-a} & \text{if} \quad a \leqslant x \leqslant b \\ 1 & \text{if} \quad b \leqslant x \leqslant c \\ \frac{d-x}{d-c} & \text{if} \quad c \leqslant x \leqslant d \\ 0 & \text{if} \quad x \geqslant d \end{cases} \tag{3.2.2}$$

In the sequel, $\left(a_i^j, b_i^j, c_i^j\right)$ $\left[\left(a_i^j, b_i^j, c_i^j, d_i^j\right)\right]$ denote the MF parameters of the i-th antecedent ($i = 1, ..., p$) and j-th rule ($j = 1, ..., N_r$) when triangle [trapezoid] MFs are used. A T1FLS with p inputs and N_r rules that uses triangle MFs has $3pN_r$ design parameters, whereas such a FLS that uses trapezoid MFs has $4pN_r$ design parameters.

To complete the description of the T1FLS, we have to define the operations for the *intersection*, *union* and *implication* within the inference engine, as well as the defuzzifier. In this study, we used *product T-norm* as intersection, *probabilistic OR* for union, *product* implication and a weighted average (centre-of-sets) defuzzifier. Because the consequents are crisp numbers (as explained above), the probabilistic OR and the OR operations are equivalent, and thus we can express the output of the T1FLS as the following well-known equation:

$$y = \frac{\sum_{j=1}^{N_r} \left[\prod_{i=1}^{p} \mu_i^j(x_i) \right] \cdot c_j}{\sum_{j=1}^{N_r} \left[\prod_{i=1}^{p} \mu_i^j(x_i) \right]} \tag{3.2.3}$$

In Equation (3.2.3), x_i is the i-th input, $\mu_i^j(x_i)$ is the antecedent MF of the i-th input in the j-th rule, and c_j are the crisp rule consequents. One benefit for using such consequents is [84] that they can be easily optimized by using least-squares, and do not have to be included in the GA portion of the optimizations, thereby reducing the dimension of the parameter vector that is used in the GA portion.

3.3 SINGLE-OBJECTIVE AND MULTI-OBJECTIVE EXPERIMENTS

In this Chapter we performe five experiments (simulations) for the nine functions that were used in [11] and are shown in Table 1.

Our comparison framework is based in nine function approximation problems, gathered in Table 3.3.1. Such functions will be approximated by two different types of FLSs: one using triangular MFs in the antecedents, and the other using trapezoidal MFs. These systems use antecedents (which will be optimised by the GA) that are essentially identical (piecewise linear functions), but use a different number of parameter per MF (triangular systems use three, whereas trapezoidal systems use four). For those two types of systems, we will compare their function approximation and uncertainty handling ability. By doing so, we intend to prove or disprove if providing a MF an extra parameter (i.

Table 3.3.1
Functions to be approximated by the FLSs.

Function	Input range
$f_1(x_1,x_2)=\sin(x_1x_2)$	$x_1,x_2\in[-2,2]$
$f_2(x_1,x_2)=\exp(x_1\sin(\pi x_2))$	$x_1,x_2\in[-1,1]$
$f_3(x_1,x_2)=\frac{a}{b+c}$ $a=40\cdot\exp\left[8\cdot\left((x_1-0.5)^2+(x_2-0.5)^2\right)\right]$ $b=\exp\left[8\cdot\left((x_1-0.2)^2+(x_2-0.2)^2\right)\right]$ $c=\exp\left[8\cdot\left((x_1-0.7)^2+(x_2-0.7)^2\right)\right]$	$x_1,x_2\in[0,1]$
$f_4(x_1,x_2)=\frac{1+\sin(2x_1+3x_2)}{3.5+\sin(x_1-x_2)}$	$x_1,x_2\in[-2,2]$
$f_6(x_1,x_2)=1.33[1.5(1-x_1)+\exp(2x_1-1)\sin\left(3\pi(x_1-0.6)^2\right)+$ $+\exp(3(x_2-0.5))\sin\left(4\pi(x_2-0.9)^2\right)]$	$x_1,x_2\in[0,1]$
$f_7(x_1,x_2)=1.9\left[1.35+\exp(x_1)\sin\left(13(x_1-0.6)^2\right)\exp(-x_2)\sin(7x_2)\right]$	$x_1,x_2\in[0,1]$
$f_8(x_1,x_2)=\sin\left(2\pi\sqrt{x_1^2+x_2^2}\right)$	$x_1,x_2\in[-1,1]$
$y_2(x_1,x_2)=0.5+64\cdot\frac{(x_1-0.5)(x_2-0.5)(x_1+0.2)}{1+(4x_1-2)^2+(4x_2-2)^2}$	$x_1,x_2\in[0,1]$
$y_5(x_1,x_2)=0.5\cdot[1+\sin(2\pi x_1)\cos(2\pi x_2)]$	$x_1,x_2\in[0,1]$

e., using trapezoidal system rather than triangular) allows the FLS to improve its performance: if it is improved, then we it will be proved that the number of parameters (or degrees of freedom) available in a FLS is key to enhance the FLSs; if it is not improved, then the number of parameters is not determinant. Hence, if the claim is not valid in T1FLSs' framework, then it cannot be valid when comparing IT2FLSs versus T1FLSs. This way we intend to shed some light on the controversial discussion presented at the beginning of the chapter.

The framework is based in five different experiments, each of them having its own goal, which will be explained in their corresponding subsection. Some of these experiments focus on a *single-objective* optimisation, in which a GA algorithm was used to optimise the antecedent MF parameters on noise-free measurements, by minimizing a normalized RMSE (NRMSE) function so as to obtain the best possible approximation to each function. Other experiments focused on a *multi-objective* optimisation, in which a GA algorithm was used to optimise MF parameters on noise-free measurements by simultaneously minimising the NRMSE and maximising the FLS's interpretability through reducing the number of rules (the size of the FLS's rule base). Additionally, one experiment used noisy measurements so as to assess the robustness of results in the presence of uncertain data. How the consequent parameters were optimized is explained in Section 3.4. Because there are many details associated with our

GA algorithms, they are collected at the end of this paper in Appendix A.

Recall that, in function approximation problems, when different MFs are allowed to be placed anywhere across the input space (not necessarily covering all of it), they are said to use a *scattered input space*, i.e., MFs are placed where they contribute the most to reducing the error. Consequently, some parts of the input space might not be covered. In all experiments, apart from experiment 4 and part of 5, *scattered input* space was used. On the other hand, when the input space is divided into some kind of *grid*, a MF is ensured in each delimited region within the grid, in which case the MFs are said to use a *partitioned input space*, i.e., the input space is divided into several subsets/grids, which cover the whole input space. In *partitioned input space*, we use a fixed number of MFs per input, N_{MAX}, so that the rules were determined by combining all possible MFs for each input, leading to a system comprised of $(N_{MAX})^p$ rules, where p is the number of inputs to the system. The *partitioned input space* was used for Experiment 4 and partially in 5.

3.4 COMPUTING OPTIMAL CONSEQUENTS

It is clear from Equation 3.2.3 that the output of the T1FLS, y, can be expressed as a linear combination of the scalar consequents, $c_1, c_2, ..., c_{N_r}$. A set of K pairs of inputs and outputs, is expressed here as $(\vec{x_i}, y_i)$, with $i = 1, ..., K$. Calling β_i^j the firing degree of the j-th rule for the i-th input-output pair ($j = 1, ..., N_r$), the K outputs can be collected in the following vector-matrix equation:

$$\beta \cdot \vec{C} = \begin{pmatrix} \beta_1^1(\vec{x}_1) & \cdots & \beta_1^{N_r}(\vec{x}_1) \\ \beta_2^1(\vec{x}_2) & \cdots & \beta_2^{N_r}(\vec{x}_2) \\ \vdots & \ddots & \vdots \\ \beta_{N_p}^1(\vec{x}_K) & \cdots & \beta_{N_p}^{N_r}(\vec{x}_K) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N_r} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \end{pmatrix} \tag{3.4.1}$$

Where β is the coefficient matrix for the K input points and $\vec{C}$ is a vector with the consequents. When $K \geqslant N_r$, Equation (3.4.1) is an over-determined system of linear equations, which can be solved to find the optimal consequents using different methods, including the Cholesky method [32] or the Singular Value Decomposition [105]. The method we used to obtain the optimal consequents, is described in Section A.2 of Appendix A. These optimal consequents were used in our evaluation functions to obtain the fitness of each individual when the GA described in Appendix A was used to optimize the MF parameters.

3.5 EXPERIMENTS AND RESULTS

This section explains the experimental setup used to establish if providing the T1FSs with one extra parameter per MF enables them to deliver improved performances for function approximation and uncertainty handling.

As mentioned in Section 3.3, we performed five experiments, whose explanations are given below and whose results are presented summarised in tabular format.

Due to the stochastic nature of the GA, each of the nine function approximation (FA) problems for each experiment was run 20 times, so that a statistical analysis could be performed to establish if there are significant differences between the performances of the triangle and trapezoidal FLSs. Statistical comparisons were done using the *Kruskal-Wallis* (KW) test for two classes (also known as the *Wilcoxon test*), which was chosen over its parametric counterpart ANOVA because the normality and homoscedasticity requirements [23] of ANOVA were not met by our data.

In our scenario, the KW test is used to compare two groups or populations (i.e., the scores obtained by triangular or trapezoidal FLSs in the 20 runs) and decide whether they belong to different distributions; it is, the KW test compares the means of these two groups and decides if such obtained means are different with statistical significance. Each comparison between two populations (20 triangular runs and 20 trapezoidal ones) is given a score, the p-value, which is compared with the chosen significance level α. If p-value $p \geqslant \alpha$, then the test cannot conclude the groups are different with statistical significance (for that given α); on the other hand, if the p-value $p \leqslant \alpha$, the test concludes both populations belong to different distributions with statistically significant differences. Hence, we can conclude if one system performs better than the other one. To decide if the differences are significant or not, we used a significance level of $\alpha = 0.05$ (5%).

In Experiments $1-4$, after the execution of each FA problem, the best triangle FLS and trapezoidal FLS individuals were tested using a different set of testing data points. We used a test dataset formed by 2025 noise-free points, randomly selected from a 45×45 grid division of the input space. In Experiment 5, noise was added to the testing data.

When using a *single-objective*, the cost function for evaluating each FA problem was the NRMSE, as is explained in Section A.2. On the other hand, when using a *multi-objective* GA the number of rules along with the NRMSE were optimised by comparing Pareto Fronts and (as in [11]) using a *Quality Indicator* (QI) that mapped each Pareto Front into a real number. The QI used in this work was the hypervolume measure [125], which is the portion of the objective space that is dominated by the Pareto Front (given a reference point); hence, the higher a QI, the better the Pareto Front which generated it. Although there is an equation for the hypervolume, presenting it here would require introducing many concepts and definitions that are out of the scope of this work, so for further information, please refer to [125].

3.5.1 *Experiment 1*

Experiment 1 is a single-objective scenario, and compared the triangle and trapezoidal T1FLSs in which the number of rules were fixed; thus, the measure to score the quality of each system is the NRMSE computed using a given set of testing points. This experiment was run for 5, 10, 15, 20, 25 and 30 rules, and its results are summarized in Table 3.5.1, in which:

1. Its first column gives the number of rules in each FLS.
2. Its second column states the three possible conclusions that can be drawn when comparing the triangle and trapezoidal FLSs.
3. Its third column gives the number of FA problems for which each conclusion occurred.
4. Its last column, for each of the first two possible conclusions (triangular better than trapezoidal or trapezoidal better than triangular), the % of improvement is shown.
5. The acronym "NA" in Table 3.5.1 and the rest of subsequent tables stands for "Not Applicable".

Table 3.5.1
Summary of results for Experiment 1

Rules	Possible conclusions	# of FA problems	% improvement
5	Triangle FLS better than trapezoid FLS	0	NA
	Trapezoid FLS better than triangle FLS	4	2.7(f_1), 30.32(f_2), 43.53(f_3), 29.99(y_5)
	No significant difference	5	NA
10	Triangle FLS better than trapezoid FLS	0	NA
	Trapezoid FLS better than triangle FLS	2	7.49(f_7), 28.3(y_5)
	No significant difference	7	NA
15	Triangle FLS better than trapezoid FLS	1	5.8(f_1)
	Trapezoid FLS better than triangle FLS	1	24.25(y_5)
	No significant difference	7	NA
20	Triangle FLS better than trapezoid FLS	1	7.88(f_1)
	Trapezoid FLS better than triangle FLS	1	4.72(f_4)
	No significant difference	7	NA
25	Triangle FLS better than trapezoid FLS	1	8(f_1)
	Trapezoid FLS better than triangle FLS	0	NA
	No significant difference	8	NA
30	Triangle FLS better than trapezoid FLS	0	NA
	Trapezoid FLS better than triangle FLS	1	10.25(f_6)
	No significant difference	8	NA

3.5.2 *Experiment 2*

Experiment 2 is a multi-objective scenario in which the NRMSE and the number of rules were simultaneously optimised, in order to consider the trade-off between accuracy and complexity (i.e., interpretability) in the T1FLSs. Results from this experiment are summarised in Table 3.5.2. Observe that results are quite even: triangle FLSs perform better in three FA problems, trapezoidal FLSs perform better in no FA problems, and there are no statistically significant differences in six FA problems. According to these results, it seems reasonable to conclude that using an extra parameter per MF for a T1FLS does not guarantee a better Pareto Front in the final solution.

This experiment has not included how much better one FLS is than the other because it is dependent on the reference point used to obtain the QI. In this work the chosen reference point is $[NRMSE = 1, \#Rules = 20]$. The reason for this is as follows: if a given function is replaced by its mean value, a $NRMSE = 1$ would be obtained; hence, any optimized FLS would be expected to have a value below unity. On the other hand, the maximum number of rules allowed for a system is 20; hence, it is not possible a higher value. This criterion ensures that the scores obtained by the best FLSs will be below these values in the two objectives.

These results demonstrate that in the multi-objective scenario it does not seem to matter which kind of MF is chosen, and hence, it seems natural to choose the one with fewest number of parameters, i.e., the triangle MF.

Table 3.5.2
Summary of results for Experiment 2

Possible conclusion	# of FA problem
Triangle FLS better than trapezoid FLS	$3(f_2, f_8, y_5)$
Trapezoid FLS better than triangle FLS	0
No significant difference	6

Table 3.5.3
Summary of NRMSE results for the testing data in Experiment 3

5 rules	Scenario 1	Scenario 2
Mean	0.41	0.41
STD	0.03	0.01
	p-value	
	0.89	

3.5.3 *Experiment 3*

Experiment 3, is a single-objective scenario that focused on FA problem #1 (function f_1) and used five-rule FLSs. Its goal is to see if one can obtain a significantly better result than was obtained in Experiment 1 (in which all individuals for the trapezoidal five-rule FLSs were initialized randomly) by first obtaining the optimal triangle five-rule FLS, and then smartly initializing the GA for the trapezoidal five-rule FLS, using the best solution from the optimal triangle five-rule FLS, X_{best} (encoded as a trapezoid). Speed of convergence is also examined.

Scenario 1 (Table 3.5.3) is for independently designed five-rule triangle and trapezoidal FLSs. Scenario 2 is for an initially-designed five-rule triangle FLS whose best solution is then included in the initial population for the five-rule trapezoidal FLS. In that Table, "Mean" and "STD" are the mean and standard deviations of the NRMSE over the 20 runs of this experiment. Observe, from Table 3.5.3, that neither the trapezoidal nor the triangular systems NRMSEs change significantly. As in both previous experiments, these 20 runs were tested using the KW test, in order to check if the difference in the means in Table 3.5.3 are statistically significant. As it can be observed, including the best triangular individual in the trapezoidal initial population does not improve the quality of the final solution obtained, as the p-value obtained when comparing the trapezoidal FLSs in Scenarios 1 and 2 is far greater than the significance level $\alpha = 0.05$.

Figures 3.5.1(a) and 3.5.1(b) depict the evolution of the best fitness for each generation of Scenario 1, for both triangle and trapezoidal FLSs, whereas Figures 3.5.1(c) and 3.5.1(d) depict the same for Scenario 2. Observe that including a very good individual in the initial trapezoidal FLS population does not significantly reduce the best NRMSE, although it does speed up finding the optimal solution.

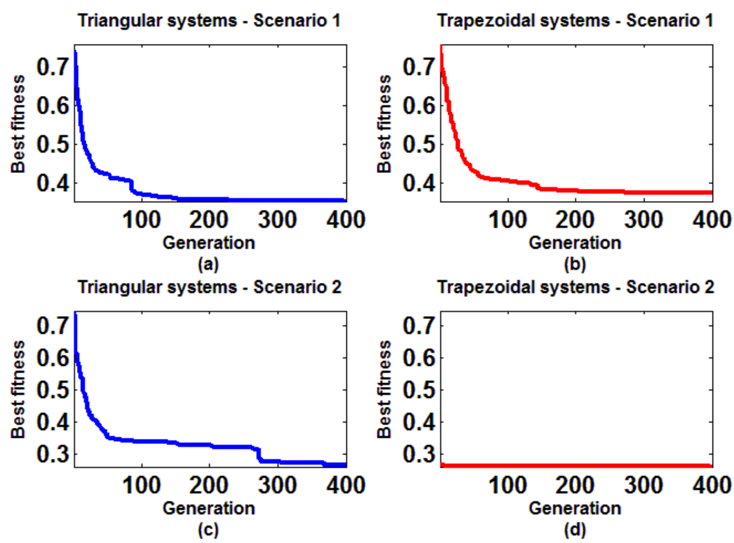

Figure 3.5.1
Evolution of the best fitness. (a) Triangle FLS, Scenario 1; (b) Trapezoidal FLS, Scenario 1; (c) Triangle FLS, Scenario 2; and, (d) Trapezoidal FLS, Scenario 2

3.5.4 *Experiment 4*

This experiment used a partitioned input space unlike the other experiments that used a scattered input space. The goal of this experiment is to find out if using a scattered or partitioned input space has significant impact on the quality of the final solution. In this experiment $N_{MAX} = 7$ and $p = 2$, so that there are $N_r = 7^2 = 49$ rules. We have repeated the setup of Experiment 2 for all nine FA problems but with the partitioned input space of 49 rules and the results are summarized in Table 3.5.4. Observe, that once again, the results indicate that there is no statistically significant difference in the performance between triangle and trapezoidal FLSs.

Table 3.5.4
Summary of results for Experiment 4 (the partitioned input-space scenario)

Possible conclusion	# of FA problem	% improvement (FA problem)
Triangle FLS better than trapezoid FLS	0	NA
Trapezoid FLS better than triangle FLS	1	18.82%(f_1)
No significant difference	8	NA

3.5.5 *Experiment 5*

Experiment 5 compared the performance of the triangle and trapezoidal FLSs, using both fixed and varying number of rules, when noise was added to the testing points (as in [11]). In this experiment, noise power was specified as a fraction of the testing point's variance, σ_{test}, i.e., $\sigma_{noise} = k \cdot \sigma_{test}$, where k is a real number in $(0, 1]$.

The best individuals obtained in Experiments 2 (varying number of rules) and 4 (fixed number of rules, 49) were used according to the following two-step procedure:

1. For each FA problem, the FLSs associated with the best individual/Pareto fronts obtained in these two experiments were retrieved.

2. Using these FLSs, the NRMSE was re-computed, using the set of noisy testing points for $k = \{0.01, 0.02, 0.05, 0.1, 0.2\}$.

Figure 3.5.2 and Table 3.5.5 summarize the results obtained for this experiment in the single objective setup followed in Experiment 1. In Figure 3.5.2, each point in each of its sub-figures represents the mean NRMSE obtained in 20 runs, for both the triangle FLS (solid line with circle markers) and trapezoidal FLS (dotted line with cross markers). A quick look at the figures reveals that both systems perform almost equally when dealing with noisy data. When the noise level increases, the performance of both T1FLSs deteriorates in the same way. This demonstrates that providing the T1FLS with one extra parameter does not enable it to deliver better performance when the data are noisy.

For all points in Figure 3.5.2, the respective p-values are presented in Table 3.5.5, so as to test if there are statistically significant differences between both FLSs. Each p-value in Table 3.5.5 represents the comparison between the 20 runs from each kind of system, triangular or trapezoidal (the same applies for Table 3.5.6). For instance for function f_3 and $k = 0.02$, the p-value is $p = 0.1$; hence, when comparing the 20 runs of both systems, the KW test concluded there are no statistically significant differences between them. For all 45 possible p-values, only four different cases are below the significance level of $\alpha = 0.05$. They are

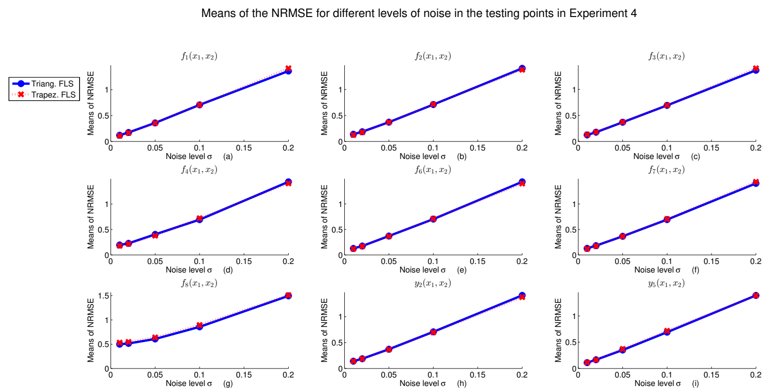

Figure 3.5.2
Structure of a type-1 fuzzy logic system.

Table 3.5.5
P-values for each 20 runs of each FA problem using different levels of noise

Noise fraction power	P values f_1	f_2	f_3	f_4	f_6	f_7	f_8	y_2	y_5
$k = 0.01$	**0.05**	0.13	0.21	0.11	0.15	0.55	0.50	0.53	0.70
$k = 0.02$	**0.02**	0.10	0.39	0.27	0.19	0.32	0.53	0.91	0.43
$k = 0.05$	0.39	0.70	0.57	**0.01**	0.81	0.89	0.39	0.43	**0.01**
$k = 0.1$	0.73	0.79	0.81	0.15	0.68	0.23	0.24	0.57	0.18
$k = 0.2$	0.08	0.52	0.19	0.52	0.32	0.33	0.59	0.27	0.83

shown in bold-face in Table 3.5.5, and are for function f_1, using $k = 0.01$ and $k = 0.02$ (in both these cases, trapezoidal FLSs performed slightly better than did the triangle FLSs), for function f_4, using $k = 0.05$ (in this case, the triangle FLSs performed better than the trapezoidal FLSs), and function y_5 with $k = 0.05$ (triangular FLS better than trapezoidal). In the 41 remaining cases, the statistical analysis demonstrates that there are no statistically significant differences between the performances of both types of FLSs when different amounts of noise corrupt the testing points.

Table 3.5.6
P-values for each 20 runs of each FA problem using different levels of noise

Noise fraction power	P values f_1	f_2	f_3	f_4	f_6	f_7	f_8	y_2	y_5
$k = 0.01$	0.22	**0.05**	0.74	**0.05**	**0.02**	0.41	**0.00**	0.62	**0.01**
$k = 0.02$	0.07	**0.02**	0.24	0.07	**0.05**	0.40	**0.00**	0.20	**0.01**
$k = 0.05$	0.50	0.23	0.35	0.11	0.07	0.59	0.98	0.78	**0.01**
$k = 0.1$	0.84	0.49	**0.03**	0.08	0.28	0.07	**0.04**	**0.04**	0.16
$k = 0.2$	0.44	0.23	0.09	0.14	**0.05**	0.39	**0.00**	0.87	0.51

Figure 3.5.3 and Table 3.5.6 summarise the results obtained for this experiment in the multi objective setup followed in Experi-

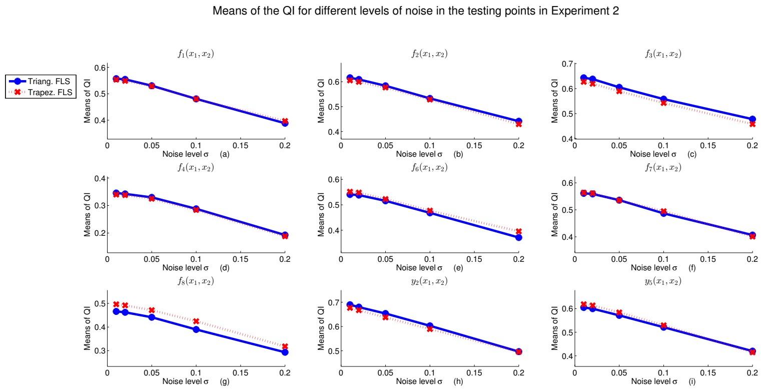

Figure 3.5.3
Structure of a type-1 fuzzy logic system.

ment 2. The only difference to Figure 3.5.2 and Table 3.5.5 is that the NRMSE is replaced by the quality indicator (QI), for which a higher value means better performance.

For all points in Figure 3.5.3, the respective p-values are presented in Table 3.5.6, so as to test if there are statistically significant differences between both FLSs. For all 45 possible p-values, 15 of them (shown bold-faced) have a p-value below $\alpha = 0.05$, indicating statistically significant differences. In five of them triangle FLSs performed better than the respective trapezoidal FLSs (f_2, using $k = 0.01$ and $k = 0.02$; f_3, using $k = 0.1$; f_4, using $k = 0.01$; and, y_2 using $k = 0.1$), whereas in ten of them trapezoidal FLSs performed better (f_6, using $k = 0.01$,0.02 and 0.2; f_8, using $k = 0.01$, 0.02, 0.1 and 0.2; and, y_5, using $k = 0.01$, 0.02 and 0.05). The remaining 30 cases showed no performance differences.

An examination of Figure 3.5.3 reveals that the QI deteriorates in the same way and at about the same level for both the triangle and trapezoidal FLSs. This again shows that giving the T1FLS one extra parameter usually does not enable the T1FLS to deliver better performance when noise is present.

3.6 CONCLUSIONS AND FUTURE WORK

This Chapter has examined the claim that one reason for the better performance of T2FLSs is their use of the extra parameters that describe type-2 fuzzy sets. Those who make this claim hypothesize that if the type-1 fuzzy sets were allowed to have the same number of parameters (and hence the same number of design degrees of freedom) as the type-2 fuzzy sets, and if both the type-2 and type-1 fuzzy FLSs were given equal opportunities of optimisation, then the type-2 and type-1 FLSs should end up with equal performances. In this work, we have addressed this claim in a novel way by investigating whether or not providing type-1 fuzzy sets with one extra parameter enables them to improve their function approximation and uncertainty handling performance totally within the framework of T1FLSs, because if the claim is not valid in that framework it cannot be valid for the IT2 versus T1FLS framework.

We have shown, by means of extensive simulation studies for nine function approximation problems, that adding one parameter to each of the antecedent fuzzy sets in a type-1 FLS does not enable that FLS to improve its function approximation and uncertainty handling performances. These findings are consistent with those in [11]. In light of these results, it seems reasonable to conclude that the ability of T2FLSs to perform better than their T1FLS counterparts is due to the way these systems handle uncertainty, rather than the number of degrees of freedom available in the system.

Future work relating this topic should focus on extending the approach that has been taken in this work to IT2FLSs. More specifically, suppose, e.g., that the FOU of an IT2FS is for a Gaussian primary MF with certain mean μ, and uncertain standard deviation, i.e., $\sigma \in [\sigma_1, \sigma_2]$, so that it is described by three parameters. This FOU can be allowed to have one extra parameter by, for instance, letting the mean also be uncertain, i.e., $\mu \in [\mu_1, \mu_2]$. The question to be studied is: does this extra parameter for the FOU (applied to all of the antecedents in an IT2FLS) allow an IT2FLS to improve its function approximation and uncertainty handling performances?

Part III

DEVELOPMENT OF TYPE-2 FUZZY SETS AND FUZZY LOGIC THEORY

4

JOIN AND MEET OPERATIONS FOR TYPE-2 FUZZY SETS WITH NONCONVEX SECONDARY MEMBERSHIPS

Fuzzy set theory has certainly been used to solve problems, but the fuzzy set theorists have failed to prove the existence of uncertain events which cannot be represented probabilistically.

— Michael Laviolette, John W. Seaman

Previously in this dissertation (more precisely in Part ii, Chapter 3), we approached a discussion about type-1 and type-2 fuzzy logic, and whether their abilities of modelling and handling uncertainties lied on the number of parameters each of them have or not. We shed some light on that discussion, and concluded that the number of parameters is not just the only key to improve the performance of fuzzy logic systems. Hence, type-2 fuzzy logic is plenty justified, and we shall focus our efforts on it from now on, trying to further improve fuzzy sets and fuzzy logic systems theory.

In order to contribute to the development of type-2 fuzzy logic theory, having a close look at the state of the art seems to be a good idea. Recent works have broadened the perception the research community had about IT2FSs. In [89], Bustince et al. proved that IT2FSs are actually more general than IVFSs, and, in addition, that some of these sets might have nonconvex secondary grades. This fact posed two new needs: on the one hand, the classic join and meet operations on IVFSs required the secondary grades to be closed and connected intervals, a condition that does not necessarily hold any more; hence, in order to be able to operate on these sets, new equations for the join and meet operations should be obtained. On the other hand, these new IT2FSs can help us represent uncertainty and imprecission on ways the classic IVFSs cannot, and thus, it is important to also explore FLSs using these sets, paying special attention to those blocks presenting significant differences with other well-known FLSs in the literature.

In this Chapter, our efforts are focused on the first need, and we will present two theorems for the join and meet operations for general type-2 fuzzy sets with arbitrary secondary memberships, which can be nonconvex and/or nonnormal type-1 fuzzy sets. These results will be used to derive the join and meet operations of the more general descriptions of interval type-2 fuzzy sets presented in [89], where the secondary grades can be nonconvex. Hence, this study will help to explore the potential of type-2 fuzzy logic systems which use the general forms of interval type-2 fuzzy sets which are not equivalent to interval-valued fuzzy sets. Several examples for both general type-2 and the more general forms of interval type-2 fuzzy sets are presented.

4.1 INTRODUCTION

General type-2 fuzzy sets (GT2FSs) are characterized by secondary memberships, which take any value between 0 and 1 (unlike interval type-2 fuzzy sets (IT2FSs), whose secondary memberships are either 0 or 1). The meet and join operations for GT2FSs, which represent the intersection and union for these sets (as presented in Sections 2.7.3 and 2.7.2), respectively, are based on the Extension Principle by Zadeh [113] as a generalization of the intersection and union for type-1 fuzzy sets (Section 2.7.1). In 2001, the initial work by Karnik and Mendel presented in [49] a simplified procedure to compute these operations for GT2FSs, although it depended on the condition that the secondary grades of type-2 fuzzy sets were normal and convex type-1 fuzzy sets. This work was later generalized by Coupland

and John [20] to incorporate nonnormal sets by borrowing some methods (Weiler-Atherton, Modified Weiler Atherton, Bentley-Ottmann Plane Sweep Algorithm, etc.) from the field of computational geometry; yet, convexity remained a necessary condition. More recent works [71] [92] [73] studied the geometrical properties of some GT2FSs to find closed formulas or approximations for the join and meet operations in some specific cases. However, to the best of the authors' knowledge, considering arbitrary secondary grades, which can be nonconvex, has not been addressed to date.

Recent developments in type-2 fuzzy logic have changed the perception researchers have of IT2FSs. IT2FSs are type-2 fuzzy sets whose uncertainty is equally distributed in the third dimension (also called secondary membership), and thus, these secondary membership are either 0 or 1, unlike GT2FSs, whose uncertainty in the third dimension is not equally weighted and the distribution can be an arbitrary type-1 fuzzy set. When IT2FSs were initially defined in [58], all the theory and operations were based on the specific case where IT2FSs are equivalent to interval-valued fuzzy sets (IVFSs). However, it has been recently shown that IT2FSs are more general than IVFSs [89]. Hence, in order to derive the theory of these general forms of interval type-2 fuzzy logic systems (gfIT2FLSs) (which employ IT2FSs which are not equivalent to IVFSs), it is necessary to develop the meet and join operations of GT2FSs with nonconvex secondary memberships and, then, particularize it to the case of IT2FSs, which have secondary grades equal to either 0 or 1.

Hence, in this Chapter, we will be finding the join and meet operations for GT2FSs where secondary memberships are arbitrary type-1 sets and, hence, can be nonconvex and/or nonnormal. This will be used to derive the join and meet operations of IT2FSs where the secondary grades are nonconvex sets.

The structure of this Chapter is as follows: Section 4.2 will present preliminaries in order to provide some basic background. Section 4.3 will present two theorems for the join and meet operations for GT2FSs making no assumptions about their normality or convexity. In Section 4.4, we will apply our results to the general forms of IT2FSs presented in [89]. Section 4.5 will present examples of the two theorems applied to GT2FSs with normal and convex secondary memberships, GT2FSs with nonconvex and nonnormal secondary memberships, and all general forms of IT2FSs (including type-1 sets and IVFSs) as presented in [89]. Conclusions and future work are presented in Section 4.6.

Note that, along this chapter, we will usually refer to type-1 fuzzy sets with capital letters, whereas type-2 fuzzy sets will be denoted by capital letters with tilde.

4.2 PRELIMINARIES

Part of the content of this Section was discussed in Chapter 2, and will be briefly repeated here for the convenience of the reader.

Type-2 fuzzy sets are an extension of type-1 fuzzy sets. While a type-1 fuzzy set F is characterised by a type-1 MF $\mu_F(x)$ (where $x \in X$ and $\mu_F(x) \in [0,1]$), a type-2 set $\tilde{F}$ is characterised by a type-2 MF $\mu_{\tilde{F}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e. [58] [78]:

$$\tilde{F} = \{((x,u), \mu_{\tilde{F}}(x,u)) \mid \forall x \in X,\ \forall u \in J_x \subseteq [0,1]\} \tag{4.2.1}$$

The set $\tilde{F}$ can also be expressed as follows [58]:

$$\tilde{F} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{F}}(x,u)/(x,u) \qquad J_x \subseteq [0,1] \tag{4.2.2}$$

Where the integrals denote aggregation over all admisible x and u. J_x is called the primary membership of x in $\tilde{F}$. At each value of x, say $x = x'$, the 2-D plane whose axes are u and $\mu_{\tilde{F}}(x',u)$ is called a *vertical slice* of $\tilde{F}$, as introduced in Definition 2.6.4. A secondary MF is a vertical slice of $\tilde{F}$. It is, $\mu_{\tilde{F}}(x = x', u)$, for $x' \in X$ and $\forall u \in J_{x'} \subseteq [0,1]$, [113], i.e.:

$$\mu_{\tilde{F}}(x = x', u) \equiv \mu_{\tilde{F}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u \qquad J_{x'} \subseteq [0,1] \tag{4.2.3}$$

Because $\forall x' \in X$, the prime notation on $\mu_{\tilde{F}}(x')$ is dropped and $\mu_{\tilde{F}}(x)$ is referred to as a secondary MF [58], [78] (see); it is a type-1 fuzzy set which is also referred to as a secondary set [58]. If $\forall x \in X$, the secondary MF is an interval type-1 set, where $f_x(u) = 1\ \forall u \in J_x$, (i.e., $\mu_{\tilde{F}}(x,u) = 1$) , the type-2 set $\tilde{F}$ is referred to as an IT2FS (see Definition 2.6.11). It should be noted that the notation we use here does not imply that J_x should only consider the values where $f_x(u)$ is greater than zero. In this study, we consider $J_x = [0,1]$ to simplify the representation.

4.3 JOIN AND MEET OPERATIONS FOR GT2FSS WITH NONCONVEX SECONDARY MEMBERSHIPS

In this section, two theorems for the join and meet operations on GT2FSs with nonconvex secondary memberships are presented.

4.3.1 *Join operation*

Definition 4.3.1. Let $\tilde{F}_1$ and $\tilde{F}_2$ be two type-2 fuzzy sets in a universe of discourse X. Let $\mu_{\tilde{F}_1}(x)$ and $\mu_{\tilde{F}_1}(x)$ denote membership grades of $\tilde{F}_1$ and $\tilde{F}_2$, respectively, at $x \in X$. Then, for each $x \in X$, using *minimum T-norm* and *maximum T-conorm*, the union set $\tilde{F}_1 \cup \tilde{F}_2$, which is characterised by its membership grade $\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta)$, is given by the *join* operation on $\mu_{\tilde{F}_1}(x)$ and $\mu_{\tilde{F}_2}(x)$ and is as follows[1]:

$$\begin{aligned}\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) &= \Big(\mu_{\tilde{F}_1}(x) \sqcup \mu_{\tilde{F}_2}(x)\Big)(\theta) = \\ &= \sup_{v \in [0,\theta]} \{f_1(v)\} \wedge (f_1(\theta) \vee f_2(\theta)) \wedge \sup_{w \in [0,\theta]} \{f_2(w)\}\end{aligned} \tag{4.3.1}$$

Such that $v \vee w = \theta$.

Theorem 4.3.1. *The union operation on type-2 fuzzy sets defined in [49] using minimum T-norm is equivalent to the union defined in Equation (4.3.1).*

Proof. Let the join operation be performed on two type-2 fuzzy sets, denoted $\tilde{F}_1$ and $\tilde{F}_2$, in a universe of discourse X. The membership grades at $x \in X$ of $\tilde{F}_1$ and $\tilde{F}_2$ are denoted as $\mu_{\tilde{F}_1}(x)$ and $\mu_{\tilde{F}_2}(x)$, respectively, which are fuzzy sets defined in $V, W \subseteq [0, 1]$, and are as in Equation (4.2.3). According to [49], the union of two type-2 fuzzy sets, denoted as $\tilde{F}_1 \cup \tilde{F}_2$, is given by the join operation as follows:

$$\begin{aligned}\tilde{F}_1 \cup \tilde{F}_2 \iff& \mu_{\tilde{F}_1 \sqcup \tilde{F}_2}(x) = \\ &= \int_{v \in V} \int_{w \in W} (f_1(v) \star f_2(w)) / (v \vee w) \qquad x \in X\end{aligned} \tag{4.3.2}$$

Where $\star$ indicates the minimum T-norm (hence, $\star$ will be replaced by $\wedge$ in the rest of the Chapter) and $\vee$ indicates the maximum T-conorm.

1 It should be noted that Equation (4.3.1) has some similarity to Equation (10) in [98] (see also [39]) as both equations refer to the join operation on GT2FSs. However, the representation of Equation (4.3.1) is quite different to simplify the computations and analysis

Thus, any element $\theta = (v \vee w)$ in the primary membership of $\tilde{F}_1 \cup \tilde{F}_2$ can be obtained by any of the following cases.

1. **Case 1:** If v is any value between 0 and θ, and $w = \theta$, i.e., $\{(v, w) \mid v \leqslant \theta \text{ and } w = \theta\} \to (v \vee w) = (v \vee \theta) = \theta$. This condition is equivalent to state that $v \in [0, \theta]$ and $w = \theta$.

2. **Case 2:** If w is any value between 0 and θ, and $v = \theta$, i.e., $\{(v, w) \mid w \leqslant \theta \text{ and } v = \theta\} \to (v \vee w) = (\theta \vee w) = \theta$. This condition is equivalent to state that $w \in [0, \theta]$ and $v = \theta$.

The membership value associated with θ can be obtained by applying the minimum T-norm on the secondary grades $f_1(v)$ and $f_2(w)$, where v and w are as described in Cases 1 and 2; hence, $\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = f_1(v) \wedge f_2(w)$.

It is important to note that if more than one pair $\{v, w\}$ result in the same $\theta = (v \vee w)$ but with different membership grade $\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = f_1(v) \wedge f_2(w)$, then we keep the maximum membership grade obtained from all $\{v, w\}$ pairs. Hence, $\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = f_1(v) \wedge f_2(w)$ is obtained by the following steps:

1. **Step 1:** Calculate $\phi_1(\theta)$, where:

$$\phi_1(\theta) = \sup_{v \in [0, \theta]} \{f_1(v) \wedge f_2(\theta)\} \tag{4.3.3}$$

According to the notation used in [98], Equation (4.3.3) would be $f_1^L(\theta) \wedge f_2(\theta)$. See [98] for this notation.

2. **Step 2:** Calculate $\phi_2(\theta)$, where:

$$\phi_2(\theta) = \sup_{w \in [0, \theta]} \{f_1(\theta) \wedge f_2(w)\} \tag{4.3.4}$$

According to the notation used in [98], Equation (4.3.4) would be $f_1(\theta) \wedge f_2^L(\theta)$.

3. **Step 3:** Calculate $\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta)$, where:

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = \phi_1(\theta) \vee \phi_2(\theta) \tag{4.3.5}$$

$f_1(\theta)$ and $f_2(\theta)$ are fixed as θ is fixed. Hence, $f_1(\theta)$ and $f_2(\theta)$ will not be considered in the suprema calculation. Consequently, we can rewrite Equations (4.3.3) and (4.3.4) as Equations (4.3.6) and (4.3.7), respectively, and combine them in Equation (4.3.8):

$$\phi_1(\theta) = \sup_{v \in [0, \theta]} \{f_1(v)\} \wedge f_2(\theta) \tag{4.3.6}$$

$$\phi_2(\theta) = f_1(\theta) \wedge \sup_{w \in [0,\theta]} \{f_2(w)\} \tag{4.3.7}$$

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = \left(\sup_{v \in [0,\theta]} \{f_1(v)\} \wedge f_2(\theta) \right) \vee \left(f_1(\theta) \wedge \sup_{w \in [0,\theta]} \{f_2(w)\} \right) \tag{4.3.8}$$

Using four labels denoted as A_1, B_1, C_1 and D_1 as illustrated in Equation (4.3.9), Equation (4.3.8) can be rewritten as Equation (4.3.10).

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = \left(\underbrace{\sup_{v \in [0,\theta]} \{f_1(v)\}}_{A_1} \wedge \underbrace{f_2(\theta)}_{B_1} \right) \vee \left(\underbrace{f_1(\theta)}_{C_1} \wedge \underbrace{\sup_{w \in [0,\theta]} \{f_2(w)\}}_{D_1} \right) \tag{4.3.9}$$

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = (A_1 \wedge B_1) \vee (C_1 \wedge D_1) \tag{4.3.10}$$

The distributive property of minimum and maximum operations allows us to rewrite the right-hand side of Equation (4.3.10) as follows:

$$\begin{aligned} &(A_1 \wedge B_1) \vee (C_1 \wedge D_1) = \\ &\quad = (A_1 \vee C_1) \wedge (A_1 \vee D_1) \wedge (B_1 \vee C_1) \wedge (B_1 \vee D_1) \end{aligned} \tag{4.3.11}$$

By substituting Equation (4.3.11) into (4.3.10) and replacing the labels denoted as A_1, B_1, C_1 and D_1, Equation (4.3.11) can be written as follows:

$$\begin{aligned} \mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = &\left(\sup_{v \in [0,\theta]} \{f_1(v)\} \vee f_1(\theta) \right) \wedge \\ &\wedge \left(\sup_{v \in [0,\theta]} \{f_1(v)\} \vee \sup_{w \in [0,\theta]} \{f_2(w)\} \right) \wedge \\ &\wedge (f_2(\theta) \vee f_1(\theta)) \wedge \left(f_2(\theta) \vee \sup_{w \in [0,\theta]} \{f_2(w)\} \right) \end{aligned}$$

(4.3.12)

Using four labels denoted as A_2, B_2, C_2 and D_2 as illustrated in Equation (4.3.13), (4.3.12) can be rewritten as Equation (4.3.14):

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = \underbrace{\left(\sup_{\nu \in [0,\theta]} \{f_1(\nu)\} \vee f_1(\theta) \right)}_{A_2} \wedge \\ \wedge \underbrace{\left(\sup_{\nu \in [0,\theta]} \{f_1(\nu)\} \vee \sup_{w \in [0,\theta]} \{f_2(w)\} \right)}_{B_2} \wedge \\ \wedge \underbrace{(f_2(\theta) \vee f_1(\theta))}_{C_2} \wedge \underbrace{\left(f_2(\theta) \vee \sup_{w \in [0,\theta]} \{f_2(w)\} \right)}_{D_2} \tag{4.3.13}$$

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = A_2 \wedge B_2 \wedge C_2 \wedge D_2 \tag{4.3.14}$$

It is worthwhile to analyse two of the terms in Equation (4.3.13), which are A_2 and D_2, separately:

$$A_2 = \sup_{\nu \in [0,1]} \{f_1(\nu)\} \vee f_1(\theta) \tag{4.3.15}$$

In the term A_2 shown in Equation (4.3.15), it is important to note that the value $f_1(\theta)$ is *included* in the value $\sup_{\nu \in [0,1]} \{f_1(\nu)\}$, as the value $\nu = \theta$ belongs to the interval $\nu \in [0,\theta]$. Hence, the maximum $f_1(\theta) \vee \sup_{\nu \in [0,1]} \{f_1(\nu)\}$ will always be represented in the value $\sup_{\nu \in [0,1]} \{f_1(\nu)\}$, regardless of the value of θ and the shape of the function $f_1(\nu)$. Consequently, term A_2 in Equation (4.3.15) can be written as A_2' as shown below:

$$A_2 = \sup_{\nu \in [0,\theta]} \{f_1(\nu)\} \vee f_1(\theta) = \sup_{\nu \in [0,\theta]} \{f_1(\nu)\} = A_2' \\ \to f_1(\theta) \in \{f_1(\nu) \mid \nu \in [0,\theta]\} \to f_1(\theta) \leqslant \sup_{\nu \in [0,\theta]} \{f_1(\nu)\} \tag{4.3.16}$$

Similarly, we will use the aforementioned approach for the term D_2 in Equation (4.3.13):

$$D_2 = f_2(\theta) \vee \sup_{w \in [0,\theta]} \{f_2(w)\} \tag{4.3.17}$$

Analogously, D_2 is equivalent to D_2':

$$\begin{aligned} D_2 = \sup_{w \in [0,\theta]} \{f_2(w)\} \vee f_2(\theta) = \sup_{w \in [0,\theta]} \{f_2(w)\} = D_2' \\ \rightarrow f_2(\theta) \in \{f_2(w) \mid w \in [0,\theta]\} \rightarrow f_2(\theta) \leqslant \sup_{w \in [0,\theta]} \{f_2(w)\} \end{aligned} \tag{4.3.18}$$

By using A_2' instead of A_2, and using D_2' instead of D_2 in Equation (4.3.14), we have Equation (4.3.19) as follows:

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = A_2' \wedge B_2 \wedge C_2 \wedge D_2' \tag{4.3.19}$$

Substituting each label A_2', B_2, C_2 and D_2' with their corresponding content, we obtain:

$$\begin{aligned} \mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = \sup_{v \in [0,\theta]} \{f_1(v)\} \wedge \\ \wedge \left(\sup_{v \in [0,\theta]} \{f_1(v)\} \vee \sup_{w \in [0,\theta]} \{f_2(w)\} \right) \wedge \\ \wedge (f_2(\theta) \vee f_1(\theta)) \wedge \sup_{w \in [0,\theta]} \{f_2(w)\} \end{aligned} \tag{4.3.20}$$

In order to simplify the notations, we will again label each term in (4.3.20) separatedly as shown below:

$$\begin{aligned} \mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta) = \underbrace{\sup_{v \in [0,\theta]} \{f_1(v)\}}_{A_3} \wedge \\ \wedge \left(\underbrace{\sup_{v \in [0,\theta]} \{f_1(v)\}}_{A_3} \vee \underbrace{\sup_{w \in [0,\theta]} \{f_2(w)\}}_{B_3} \right) \wedge \\ \wedge \underbrace{(f_2(\theta) \vee f_1(\theta))}_{C_3} \wedge \underbrace{\sup_{w \in [0,\theta]} \{f_2(w)\}}_{B_3} \end{aligned} \tag{4.3.21}$$

Using three labels denoted as A_3, B_3 and C_3 as illustrated in Equation (4.3.21), Equation (4.3.20) can be expressed as:

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = A_3 \wedge (A_3 \vee B_3) \wedge C_3 \wedge B_3 \tag{4.3.22}$$

We will focus on the partial expression $A_3 \wedge (A_3 \vee B_3)$ in (4.3.22). Using the fact that $a \wedge (a \vee b) = a$ for any real numbers a and b, then $A_3 \wedge (A_3 \vee B_3) = A_3$, and Equation 4.3.22 becomes 4.3.23. Substituting each label by its content, we have 4.3.24:

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = A_3 \wedge (A_3 \vee B_3) \wedge C_3 \wedge B_3 = A_3 \wedge C_3 \wedge B_3 \tag{4.3.23}$$

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = \sup_{v \in [0,\theta]} \{f_1(v)\} \wedge (f_1(\theta) \vee f_2(\theta)) \wedge \sup_{w \in [0,\theta]} \{f_2(w)\} \tag{4.3.24}$$

Equation (4.3.24) is the same as Equation (4.3.1), and this concludes the proof of Theorem 4.3.1. This equation is the final result for the join operation performed on two type-2 fuzzy sets, $\tilde{F}_1$ and $\tilde{F}_2$, for each $x \in X$. It is important to note that this result is obtained without any assumption regarding the normality or convexity of the secondary grades, denoted $f_1(v)$ and $f_2(w)$, that belong to the fuzzy sets $\tilde{F}_1$ and $\tilde{F}_2$, respectively.

□

4.3.2 *Meet operation*

Definition 4.3.2. Let $\tilde{F}_1$ and $\tilde{F}_2$ be two type-2 fuzzy sets in a universe of discourse X. Let $\mu_{\tilde{F}_1}(x)$ and $\mu_{\tilde{F}_2}(x)$ denote the membership grades of $\tilde{F}_1$ and $\tilde{F}_2$, respectively, at $x \in X$. Then, using *minimum T-norm*, the intersection set $\tilde{F}_1 \cap \tilde{F}_2$, which is characterised by its membership grade $\mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta)$, is given by the *meet* operation on $\mu_{\tilde{F}_1}(x)$ and $\mu_{\tilde{F}_2}(x)$ and is as follows[2]:

$$\begin{aligned} \mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta) &= \left(\mu_{\tilde{F}_1}(x) \sqcap \mu_{\tilde{F}_2}(x)\right)(\theta) = \\ &= \sup_{v \in [\theta],1} \{f_1(v)\} \wedge (f_1(\theta) \vee f_2(\theta)) \wedge \sup_{w \in [\theta,1]} \{f_2(w)\} \end{aligned} \tag{4.3.25}$$

Such that $v \wedge w = \theta$.

2 It should be noted that Equation (4.3.25) has some similarity to Equation (11) in [98] as both equations refer to the meet operation on GT2FSs. However, the representation of (4.3.25) is different to simplify the computation and analysis

Theorem 4.3.2. *The intersection operation on type-2 fuzzy sets defined in [49] using minimum T-norm is equivalent to the intersection defined in Equation (4.3.25).*

Proof. Proof of Theorem 4.3.2 is very similar to the proof of Theorem 4.3.1. In this case, any element θ in the primary membership of $\tilde{F}_1 \cap \tilde{F}_2$ is of the form $\theta = (v \wedge w)$, and can be obtained by any of the following two cases:

1. **Case 1:** $v \in [\theta, 1]$ and $w = \theta$.
2. **Case 2:** $w \in [\theta, 1]$ and $v = \theta$.

The rest of the proof is exactly the same as the one for the join operation, but changing the intervals $v \in [0, \theta]$ and $w \in [0, \theta]$ by $v \in [\theta, 1]$ and $w \in [\theta, 1]$, respectively. The final result will be as in Equation (4.3.25). □

4.4 JOIN AND MEET OPERATIONS FOR THE GENERAL FORMS OF INTERVAL TYPE-2 FUZZY SETS

In this section, we will focus on the particular case where $f_1(v)$ and $f_2(w)$ *are either* 0 *or* 1 *and their supports are non-empty closed sets*. In other words, we will focus on the general forms of IT2FSs, as presented in [89]. We will obtain specific versions of Equations (4.3.1) and (4.3.25) when sets are gfIT2FSs. It is important to note that all examples in [89] satisfy that the supports of $f_1(v)$ and $f_2(w)$ are non-empty closed sets.

Let $g_1(\theta) = \sup_{v \in [0,1]} \{f_1(v)\}$. For a given value of θ, $g_1(\theta)$ is the maximum value that the function $f_1(v)$ has attained for all values of v lower than or equal to θ, i.e., $\forall v \leqslant \theta$. Let v_1 be the infimum of the support of f_1. Hence, for all $\theta < v_1$:

$$g_1(\theta) = \sup_{v \in [0,\theta]} \{f_1(v)\} = \sup_{v \in [0,\theta]} \{0\} = 0 \qquad \forall \theta < v_1 \tag{4.4.1}$$

For values $\theta \geqslant v_1$, as $f_1(v_1) = 1$, the following stands:

$$\begin{aligned} g_1(\theta) &= \sup_{v \in [0,\theta]} \{f_1(v)\} = f_1(v_1) \vee \sup_{v \in [0,\theta], v \neq v_1} \{f_1(v)\} = \\ &= 1 \vee \sup_{v \in [0,\theta], v \neq v_1} \{f_1(v)\} = 1 \end{aligned} \tag{4.4.2}$$

Hence, combining Equations (4.4.1) and (4.4.2):

$$g_1(\theta) = \sup_{v \in [0,\theta]} \{f_1(v)\} = \begin{cases} 0 & \forall \theta < v_1 \\ 1 & \forall \theta \geqslant v_1 \end{cases} \tag{4.4.3}$$

Analogously, let $g_2(\theta) = \sup_{w \in [0,1]} \{f_2(w)\}$ and let w_1 be the infimum of the support of f_2. Hence:

$$g_2(\theta) = \sup_{w \in [0,\theta]} \{f_2(w)\} = \begin{cases} 0 & \forall \theta < w_1 \\ 1 & \forall \theta \geqslant w_1 \end{cases} \tag{4.4.4}$$

Let $g(\theta) = g_1(\theta) \wedge g_2(\theta)$. Combining Equations (4.4.3) and (4.4.4):

$$g(\theta) = \begin{cases} 0 & \forall \theta < \max(v_1, w_1) \\ 1 & \forall \theta \geqslant \max(v_1, w_1) \end{cases} \tag{4.4.5}$$

Considering the definition of $g(\theta)$, we can rewrite Equation (4.3.1) as:

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = \begin{cases} 0 & \forall \theta < \max(v_1, w_1) \\ f_1(\theta) \vee f_2(\theta) & \forall \theta \geqslant \max(v_1, w_1) \end{cases} \tag{4.4.6}$$

Let v_{end} and w_{end} be the supremum of the supports of f_1 and f_2, respectively. Hence, $f_1(v) = 0 \; \forall v > v_{end}$ and $f_2(w) = 0 \; \forall w > w_{end}$. Consequently, we can rewrite Equation (4.4.6) as:

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = \begin{cases} f_1(\theta) \vee f_2(\theta) & \forall \theta \in [\max(v_1, w_1), \max(v_{end}, w_{end})] \\ 0 & elsewhere \end{cases} \tag{4.4.7}$$

Now, let us consider the case of the *meet* operation. In this case, let $g_1(\theta) = \sup_{v \in [\theta,1]} \{f_1(v)\}$. Given a value of $\theta > v_{end}$:

$$g_1(\theta) = \sup_{v \in [\theta,1]} \{f_1(v)\} = \sup_{v \in [\theta,1]} \{0\} = 0 \qquad \forall \theta > v_{end} \tag{4.4.8}$$

For values $\theta \leqslant v_{end}$, as $f_1(v_{end}) = 1$, the following stands:

$$\begin{aligned} g_1(\theta) &= \sup_{v\in[\theta,1]} \{f_1(v)\} = f_1(v_{end}) \vee \sup_{v\in[\theta,1], v\neq v_{end}} \{f_1(v)\} = \\ &= 1 \vee \sup_{v\in[\theta,1], v\neq v_{end}} \{f_1(v)\} = 1 \end{aligned} \tag{4.4.9}$$

Hence, combining Equations (4.4.8) and (4.4.9):

$$g_1(\theta) = \sup_{v\in[\theta,1]} \{f_1(v)\} = \begin{cases} 1 & \forall\theta \leqslant v_{end} \\ 0 & \forall\theta > v_{end} \end{cases} \tag{4.4.10}$$

A similar expression can be found for $g_2(\theta) = \sup_{w\in[\theta,1]} \{f_2(w)\}$.

$$g_2(\theta) = \sup_{w\in[\theta,1]} \{f_2(w)\} = \begin{cases} 1 & \forall\theta \leqslant w_{end} \\ 0 & \forall\theta > w_{end} \end{cases} \tag{4.4.11}$$

Let $g(\theta) = g_1(\theta) \wedge g_2(\theta)$. We can rewrite Equation (4.3.25) as:

$$\mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta) = g(\theta) \wedge (f_1(\theta) \vee f_2(\theta)) \tag{4.4.12}$$

Considering $g(\theta) = g_1(\theta) \wedge g_2(\theta)$ and using Equations (4.4.10) and (4.4.11), we can rewrite (4.3.25) as follows:

$$\mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta) = \begin{cases} f_1(\theta) \vee f_2(\theta) & \forall\theta \leqslant \min(v_{end}, w_{end}) \\ 0 & \forall\theta > \min(v_{end}, w_{end}) \end{cases} \tag{4.4.13}$$

By definition of v_1 and w_1, $f_1(v) = 0 \ \forall v < v_1$ and $f_2(w) = 0 \ \forall w < w_1$. Therefore, the term $f_1(\theta) \vee f_2(\theta)$ will be $0 \ \forall\theta < \min(v_1, w_1)$. Consequently, we can rewrite Equation (4.4.13) as:

$$\mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta) = \begin{cases} f_1(\theta) \vee f_2(\theta) & \forall\theta \in [\min(v_1, w_1), \min(v_{end}, w_{end})] \\ 0 & elsewhere \end{cases} \tag{4.4.14}$$

It is worthwhile to highlight that Equations (4.4.7) and (4.4.14) lead to the well-known results of the join and meet when the involved sets are type-1 sets or IVFSs. For the join in type-1 sets, as $v_1 = v_{end}$ and $w_1 = w_{end}$, then θ is nonzero only when $\theta = \max(v_1, w_1)$; therefore, $f_1(\theta) \vee f_2(\theta)$ is a singleton placed at this value $\theta = max(v_1, w_1)$. An analogous reasoning for the meet operation, given $\theta = \min(v_1, w_1)$, leads to a singleton placed at this $\theta = \min(v_1, w_1)$.

For the case of IVFSs, as f_1 and f_2 have continuous supports, then $f_1(\theta) \vee f_2(\theta)$ will also be continuous in $\theta \in [\max(v_1, w_1), \max(v_{end}, w_{end})]$ for the join, and $\theta \in [\min(v_1, w_1), \min(v_{end}, w_{end})]$ for the meet, regardless of the relative positions of v_1, w_1, v_{end} and w_{end}, thus leading to the well-known equations for IVFSs:

$$\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x, \theta) = \begin{cases} 1 & \theta \in [\max(v_1, w_1), \max(v_{end}, w_{end})] \\ 0 & elsewhere \end{cases} \tag{4.4.15}$$

$$\mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta) = \begin{cases} 1 & \theta \in [\min(v_1, w_1), \min(v_{end}, w_{end})] \\ 0 & elsewhere \end{cases} \tag{4.4.16}$$

4.5 EXAMPLES OF THE JOIN AND MEET OPERATIONS

In this section, we will present several examples of the join and meet operations on different kinds of type-2 fuzzy sets.

4.5.1 *Examples of the join and meet operations for general type-2 fuzzy sets with normal and convex secondary memberships*

In this section, we will show that our approach for the join and meet operations on two GT2FSs presented in Equations (4.3.1) and (4.3.25) give consistent results when compared with the existing approaches where secondary grades are normal and convex type-1 fuzzy sets. We have used as a benchmark the examples presented in [95] (Figure 5, p. 493), which are shown in

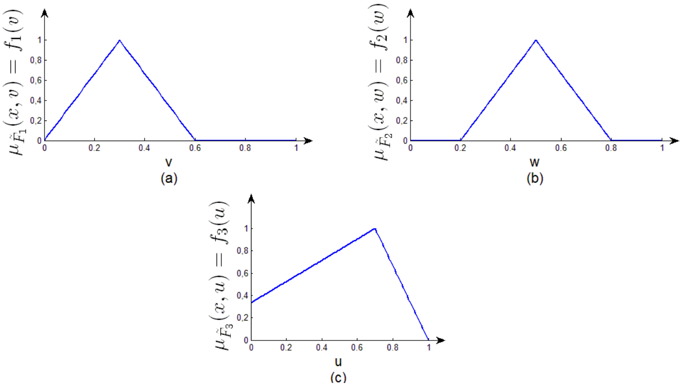

Figure 4.5.1
Vertical slices of three GT2FSs to perform the join operation.

Figure 4.5.1, so that we can compare the results achieved by Theorems 4.3.1 and 4.3.2 to the results achieved in [95].

First of all, we will perform the join operation on the first two sets given in Figure 4.5.1(a) and 4.5.1(b); second, we will perform the join operation on the resulting set and the set given in Figure 4.5.1(c).

Let $g_1(\theta) = \sup_{v\in[0,\theta]} \{f_1(v)\}$. It can be proven that, for any *convex* and *normal* secondary grade f_1 having its maximum value at $v = v_{max}$, the associated $g_1(\theta) = \sup_{v\in[0,\theta]} \{f_1(v)\}$ is as follows:

$$g_1(\theta) = \sup_{v\in[0,\theta]} \{f_1(v)\} = \begin{cases} f_1(\theta) & \forall\theta \leqslant v_{max} \\ 1 & \forall\theta > v_{max} \end{cases} \tag{4.5.1}$$

Analogously, we can obtain the term $g_2(\theta) = \sup_{w\in[0,\theta]} \{f_2(w)\}$:

$$g_2(\theta) = \sup_{w\in[0,\theta]} \{f_2(w)\} = \begin{cases} f_2(\theta) & \forall\theta \leqslant w_{max} \\ 1 & \forall\theta > w_{max} \end{cases} \tag{4.5.2}$$

These terms are illustrated in Figures 4.5.2 and 4.5.3, respectively. The only term in Equation (4.3.1) yet to be analysed is $(f_1(\theta) \vee f_2(\theta))$, which is depicted in Figure 4.5.4, along with $f_1(\theta)$ and $f_2(\theta)$. The final join result, which is as in Equation (4.3.1), is illustrated in Figure 4.5.5 as dashed line. It is important to note that the resulting $\mu_{\tilde{F}_1 \cup \tilde{F}_2}(x,\theta)$ is identical to f_2. Although this result may be surprising, we can get to the same conclusion using the equations by Karnik and Mendel in [49].

We now repeat operations between the resulting set (depicted in Figure 4.5.5 as dashed line) and the set illustrated in Figure

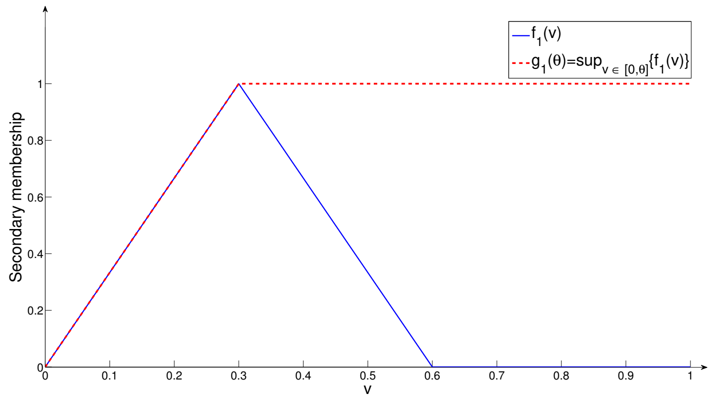

Figure 4.5.2
$f_1(v)$ and $g_1(\theta)$.

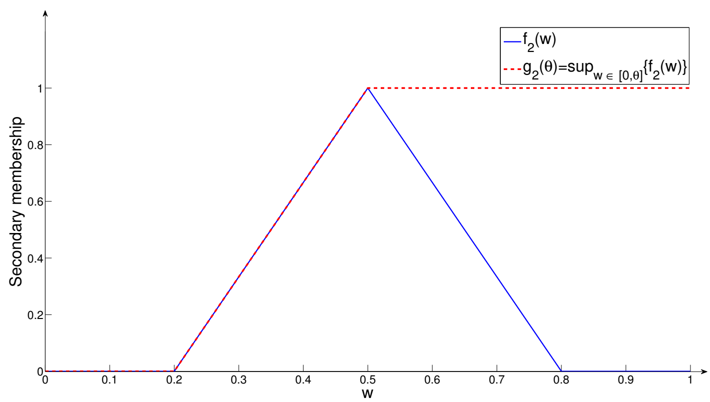

Figure 4.5.3
$f_2(w)$ and $g_2(\theta)$.

4.5.1(c). We obtain $g_2(\theta)$, $g_3(\theta)$, $(f_2(\theta) \vee f_3(\theta))$, and the minimum of all these quantities. Results are illustrated in Figures 4.5.6, 4.5.7, 4.5.8 and 4.5.9.

It is important to note that the final result displayed in Figure 4.5.9 is the same as he one presented in [95] (Figure 5, p. 493), and thus, Equation (4.3.1) is consistent with the specific case where secondary grades are normal and convex type-1 sets.

Now, we will perform the meet operation on the same three sets depicted in Figure 4.5.1. Doing a similar analysis to the one that led to Equations (4.5.1) and (4.5.2), it can be proven that:

$$g_1(\theta) = \sup_{v \in [\theta, 1]} \{f_1(v)\} = \begin{cases} 1 & \forall \theta \leqslant v_{max} \\ f_1(\theta) & \forall \theta > v_{max} \end{cases} \qquad (4.5.3)$$

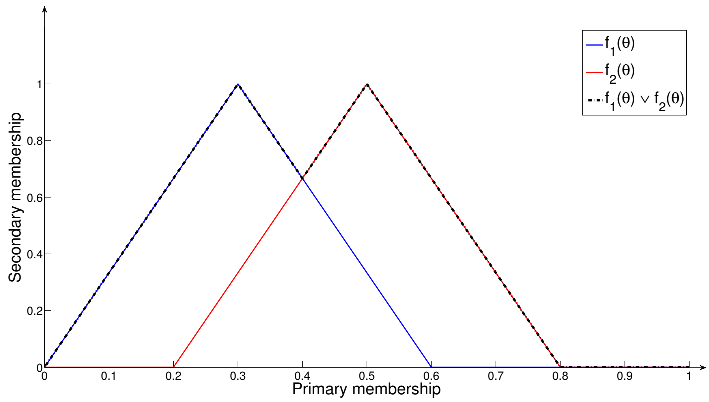

Figure 4.5.4
$f_1(\theta)$, $f_2(\theta)$ and $f_1(\theta) \vee f_2(\theta)$.

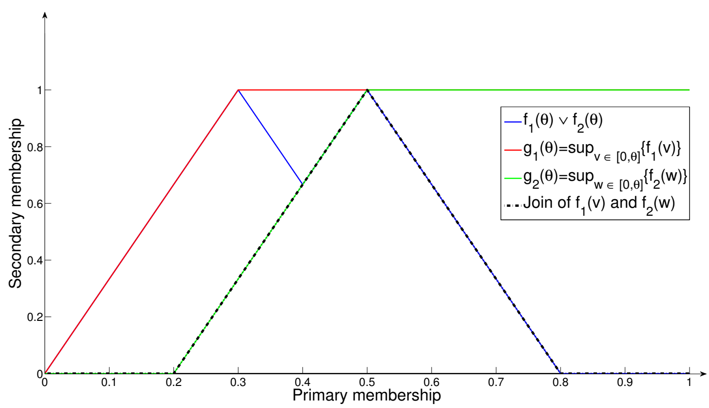

Figure 4.5.5
All terms and join result.

$$g_2(\theta) = \sup_{w \in [\theta, 1]} \{f_2(w)\} = \begin{cases} 1 & \forall \theta \leqslant w_{max} \\ f_2(\theta) & \forall \theta > w_{max} \end{cases} \qquad (4.5.4)$$

These terms are depicted in Figure 4.5.10 and 4.5.11. The only term in Equation (4.3.25) yet to be analysed is the second one, $(f_1(\theta) \vee f_2(\theta))$, which is the same as in the join operation and is depicted in Figure 4.5.12, along with $f_1(\theta)$ and $f_2(\theta)$. The final meet result, which is as in Equation (4.3.25), is illustrated in Figure 4.5.13.

It is important to note that the resulting $\mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta)$ is identical to f_1. Although this result may be surprising, we can get to the same conclusion using the equations by Karnik and Mendel in [49]. We now repeat all operations between the resulting set (depicted in Figure 4.5.13) and the set illustrated in Figure 4.5.1(c).

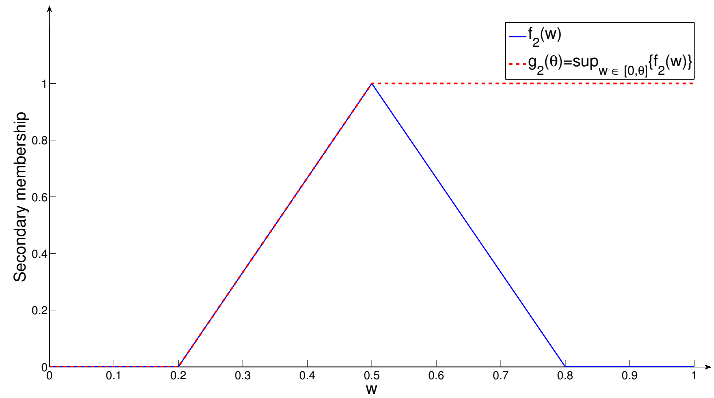

Figure 4.5.6
$f_2(w)$ and $g_2(\theta)$.

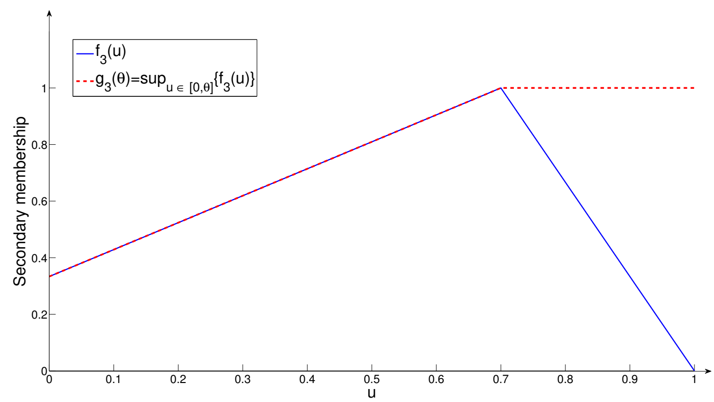

Figure 4.5.7
$f_3(u)$ and $g_3(\theta)$.

We obtain $g_1(\theta)$, $g_3(\theta)$, $f_1(\theta) \vee f_3(\theta)$, and the minimum of all these quantities. Results are illustrated in Figures 4.5.14, 4.5.15, 4.5.16 and 4.5.17. It is important to note that the final result displayed in Figure 4.5.17 is the same as the one presented in [95] (Figure 7, p. 495), and thus, Equation (4.3.25) is consistent with the specific case where secondary grades are normal and convex type-1 sets.

4.5.2 *Examples of the join and meet operations for general type-2 fuzzy sets with nonnormal nonconvex secondary grades*

In this section we will apply our approach presented in Equations (4.3.1) and (4.3.25) to type-2 fuzzy sets whose secondary grades are arbitrary, i.e., secondary grades are neither convex

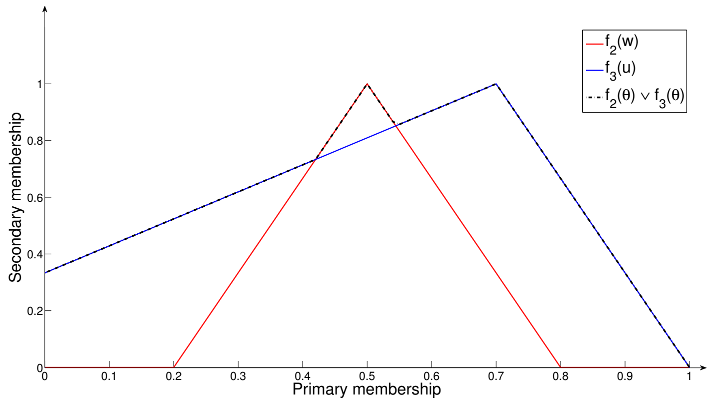

Figure 4.5.8
$f_2(\theta)$, $f_3(\theta)$ and $f_2(\theta) \vee f_3(\theta)$.

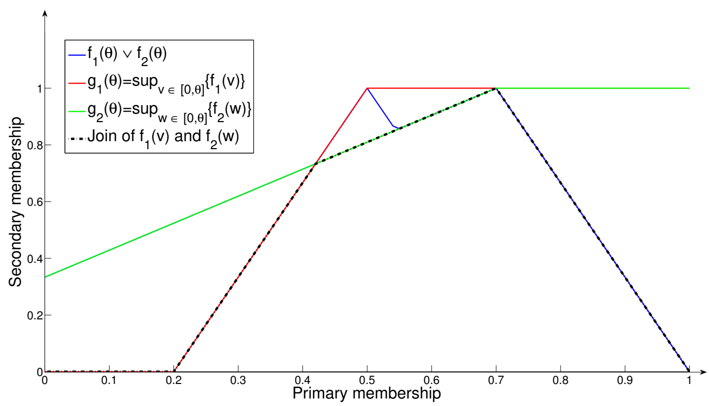

Figure 4.5.9
All terms and join result.

nor normal. The chosen secondary grades to perform this operation are illustrated in Figure 4.5.18 and 4.5.19, respectively, as solid lines, along with their terms $g_1(\theta) = \sup_{v \in [0,\theta]} \{f_1(v)\}$ and $g_2(\theta) = \sup_{w \in [0,\theta]} \{f_2(w)\}$. It is important to note that these terms $g_1(\theta)$ and $g_2(\theta)$ cannot be computed using Equations (4.5.1) and (4.5.2) as the sets are not convex. Figure 4.5.20 depicts the term $(f_1(\theta) \vee f_2(\theta))$. Figure 4.5.21 illustrates all terms involved along with the final join result.

The procedure to obtain the meet result for the same sets is analogous, but changing the definitions of $g_1(\theta) = \sup_{v \in [\theta,1]} \{f_1(v)\}$ and $g_2(\theta) = \sup_{w \in [\theta,1]} \{f_2(w)\}$. All figures related to the meet opera-

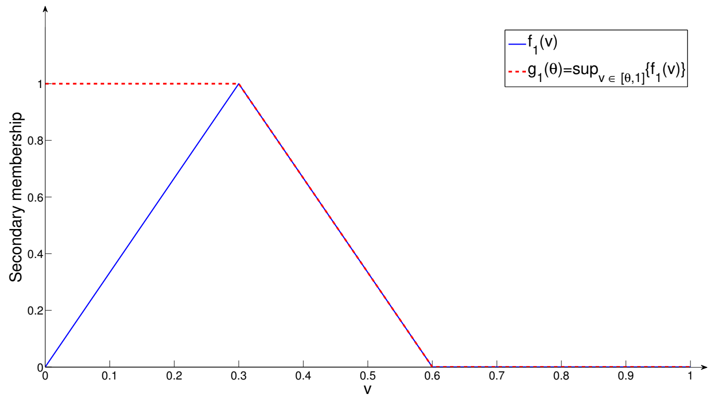

Figure 4.5.10
$f_1(v)$ and $g_1(\theta)$.

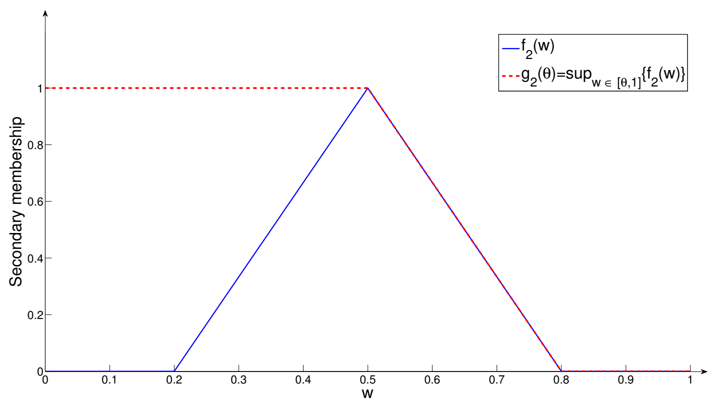

Figure 4.5.11
$f_2(w)$ and $g_2(\theta)$.

tion on the sets depicted in Figure 4.5.22 and 4.5.23 are illustrated in Figures from 4.5.22 to 4.5.25.

4.5.3 *Examples of the join and meet operations for the general forms of interval type-2 fuzzy sets*

In this section we will focus on the particular case where $f_1(v)$ and $f_2(w)$ *are either* 0 *or* 1 *and their supports are closed sets*. In other words, we will focus on the general forms of IT2FSs, as presented in [89]. We will use Equations (4.4.7) and (4.4.13) [which are specific versions of (4.3.1) and (4.3.25)] to compute the join and meet, respectively, when the secondary grades are either 0 or 1, to all the cases of gfIT2FSs. For simplicity, as in [89], we are going to work with a finite referential set X of cardinal m. However, our approach will also be valid for non-finite referential sets. In all

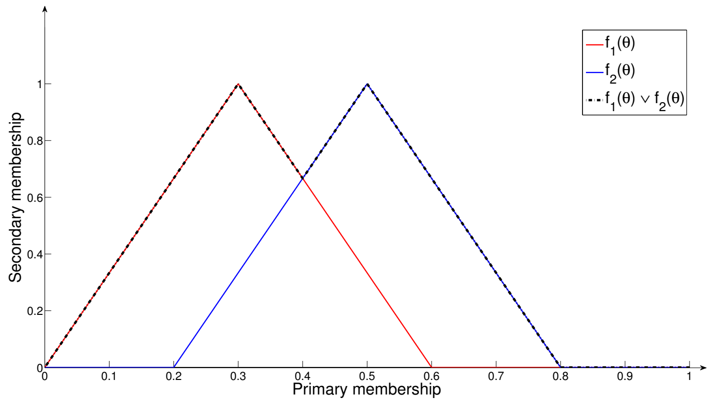

Figure 4.5.12
$f_1(\theta)$, $f_2(\theta)$ and $f_1(\theta) \vee f_2(\theta)$.

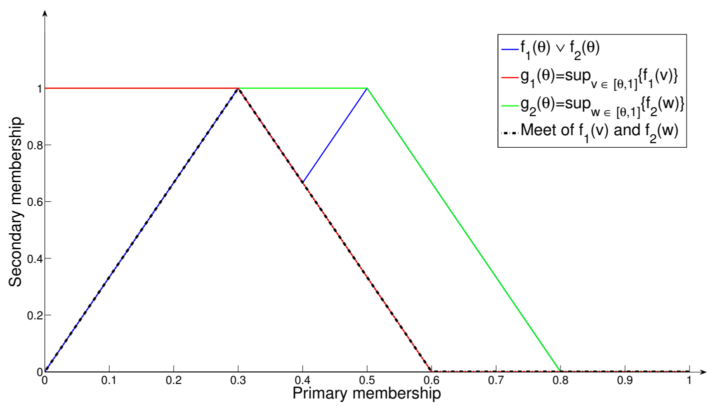

Figure 4.5.13
All terms and meet result.

cases, we will present both gfIT2FSs, and will perform the join and meet operations on the vertical slices placed at $x = x_4$.

1. **Case A: primary memberships are singletons (type-1 fuzzy sets):** the sets to perform the join and meet operations are depicted in Figures 4.5.26(a) and 4.5.26(b), respectively. The vertical slices we are going to operate with are depicted in Figures 4.5.27(a) and (b); Figure 4.5.27(c) illustrates the join operation as given by Equation (4.4.7), whereas Figure 4.5.27(d) illustrates the meet operation as in Equation (4.4.13).

2. **Case B: primary memberships are intervals (IVFSs):** the sets to perform the join and meet operations are depicted in Figures 4.5.28(a) and 4.5.28(b). The vertical slices we are going to operate with are depicted in Figures 4.5.29(a) and

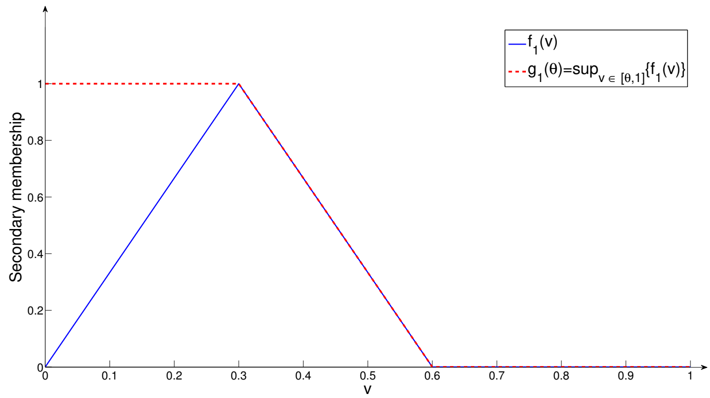

Figure 4.5.14
$f_1(v)$ and $g_1(\theta)$.

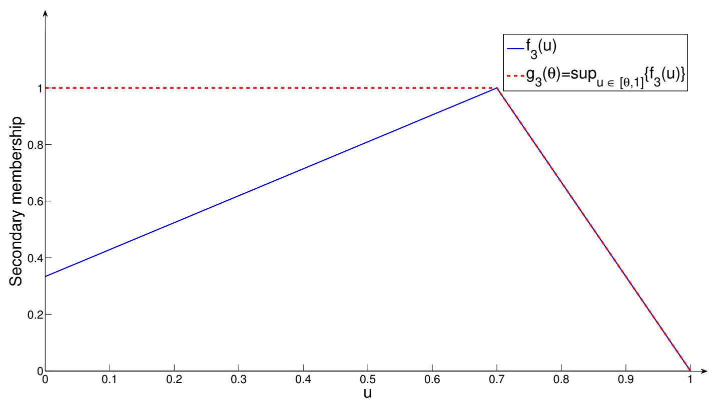

Figure 4.5.15
$f_3(u)$ and $g_3(\theta)$.

(b); Figure 4.5.29(c) illustrates the join operation as given in Equation (4.4.7), whereas Figure 4.5.29(d) illustrates the meet operation as in Equation (4.4.13).

3. **Case C: primary memberships are several singletons:** the sets to perform the join and meet operations are depicted in Figures 4.5.30(a) and 4.5.30(b), respectively. The vertical slices we are going to operate with are depicted in Figures 4.5.31(a) and (b); Figure 4.5.31(c) illustrates the join operation as given in Equation (4.4.7), whereas Figure 4.5.31(d) illustrates the meet operation as in Equation (4.4.13).

From [89], it is stated that this example may correspond to a setting in which anonymous users from a website score different objects and/or services within it. In this situation,

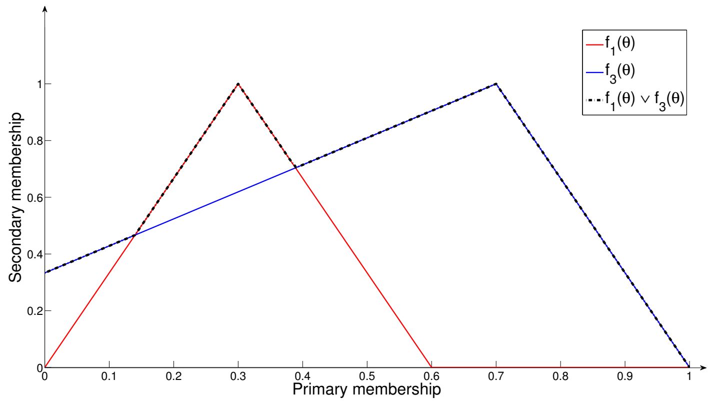

Figure 4.5.16
$f_1(\theta)$, $f_3(\theta)$ and $f_1(\theta) \vee f_3(\theta)$.

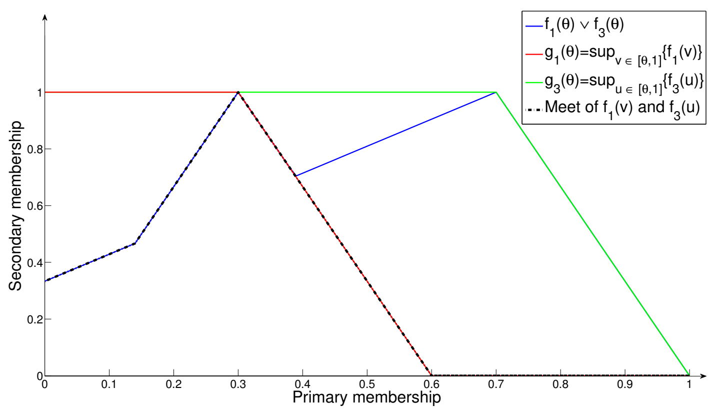

Figure 4.5.17
All terms and meet result.

and considering both anonymity and that not every user will score every object, we would obtain sets as in case C.

4. **Case D: primary memberships are several intervals:** the sets to perform the join and meet operations are depicted in Figures 4.5.32(a) and 4.5.32(b), respectively. The vertical slices we are going to operate with are depicted in Figures 4.5.33(a) and (b); Figure 4.5.33(c) illustrates the join operation as given in Equation (4.4.7), whereas Figure 4.5.33(d) illustrates the meet operation as in Equation (4.4.13).

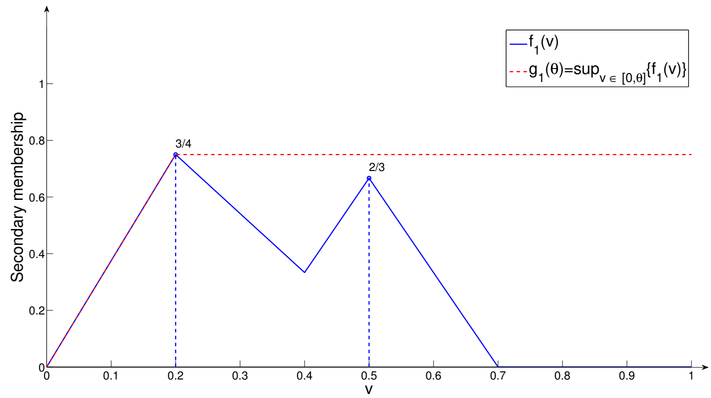

Figure 4.5.18
$f_1(v)$ and $g_1(\theta)$.

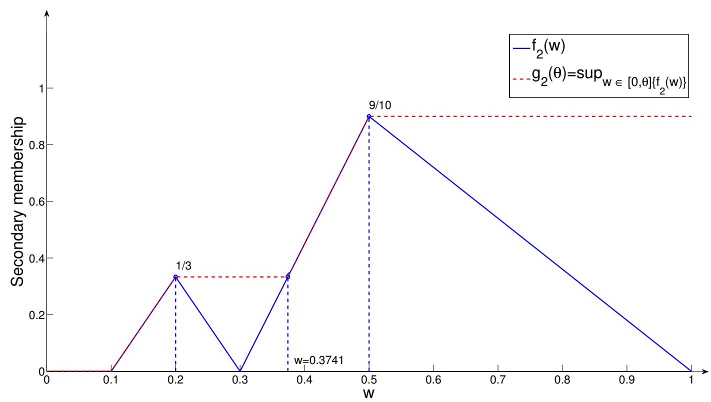

Figure 4.5.19
$f_2(w)$ and $g_2(\theta)$.

Several intervals could be used as follows: consider we have a fuzzy logic system (FLS), which contains the following rules for inputs x_1 and x_2:

$$\begin{aligned} R^l &: \text{ IF } x_1 \text{ is HIGH AND } x_2 \text{ is HIGH THEN } y \text{ is } Y \\ R^p &: \text{ IF } x_1 \text{ is HIGH AND } x_2 \text{ is LOW THEN } y \text{ is } Y \end{aligned} \tag{4.5.5}$$

Where, e.g., $\text{HIGH} = [0.9, 1]$ and $\text{LOW} = [0, 0.1]$. This could be modelled using a nonstandard rule:

$$R^{l'} : \text{ IF } x_1 \text{ is HIGH AND } x_2 \text{ is HIGH or LOW THEN } y \text{ is } Y \tag{4.5.6}$$

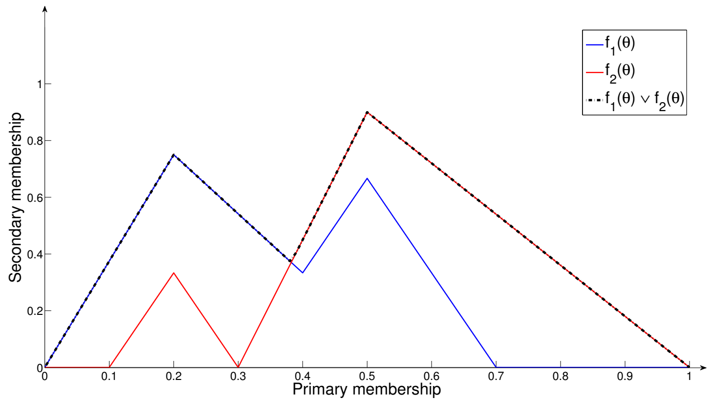

Figure 4.5.20
$f_1(\theta)$, $f_2(\theta)$ and $f_1(\theta) \vee f_2(\theta)$.

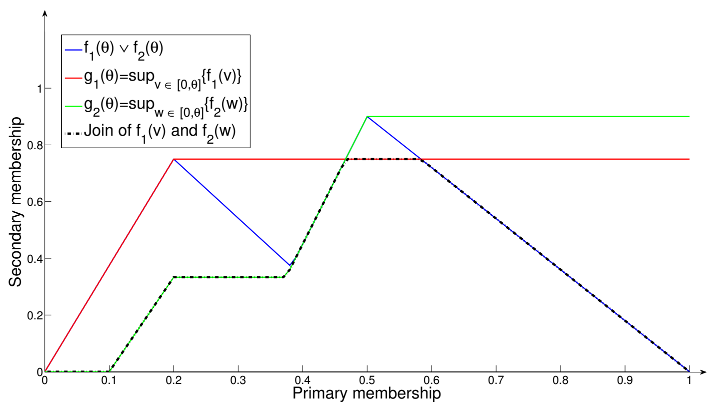

Figure 4.5.21
All terms and join result.

Or by using a standard rule with a set EXTREME $= [0, 0.1] \cup [0.9, 1]$ as follows:

$$R^{l'} : \text{IF } x_1 \text{ is HIGH AND } x_2 \text{ is EXTREME THEN } y \text{ is } Y \quad (4.5.7)$$

This way, the complexity of the FLS can be reduced and remain having a standard rule base.

5. **Case E: primary memberships are combinations of singletons and intervals:** the sets to perform the join and meet operations are depicted in Figures 4.5.34(a) and 4.5.34(b), respectively. The vertical slices we are going to operate with are depicted in Figures 4.5.35(a) and (b); Figure 4.5.35(c) illustrates the join operation as given in Equation (4.4.7),

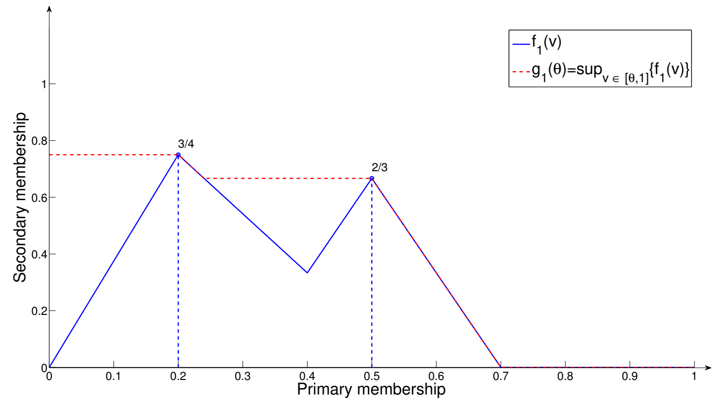

Figure 4.5.22
$f_1(v)$ and $g_1(\theta)$.

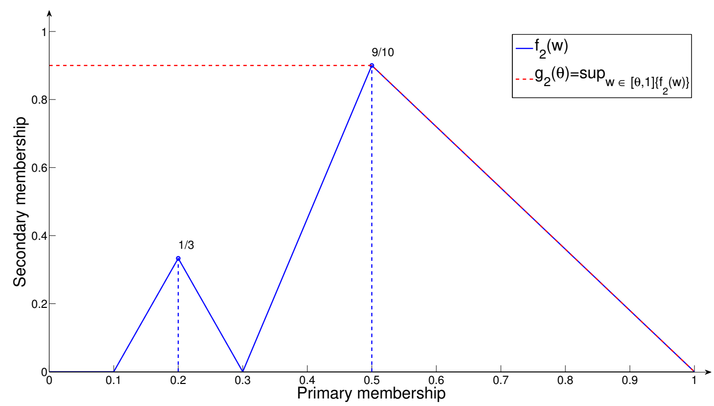

Figure 4.5.23
$f_2(w)$ and $g_2(\theta)$.

whereas Figure 4.5.35(d) illustrates the meet operation as in Equaton 4.4.13.

Considering again the scoring system in a website, we could use these type of sets when the number of scores provided increases significantly; hence, those regions of the interval $[0, 1]$, which are very crowded could be replaced by an interval, whereas the most isolated values could remain as singletons.

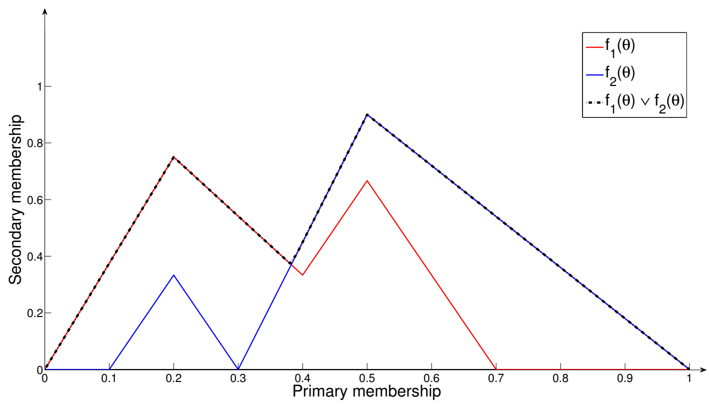

Figure 4.5.24
$f_1(\theta)$, $f_2(\theta)$ and $f_1(\theta) \vee f_2(\theta)$.

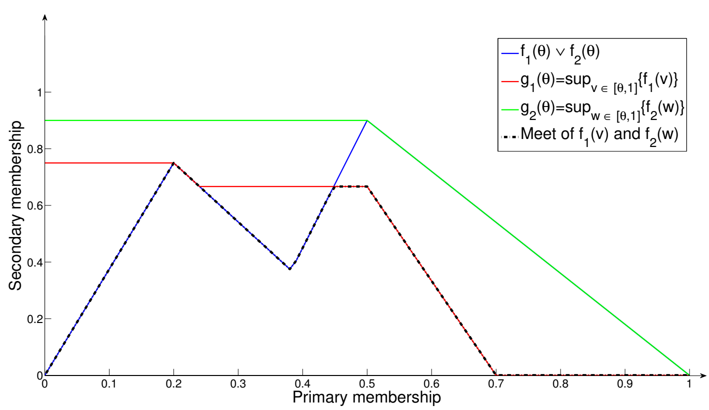

Figure 4.5.25
All terms and meet result.

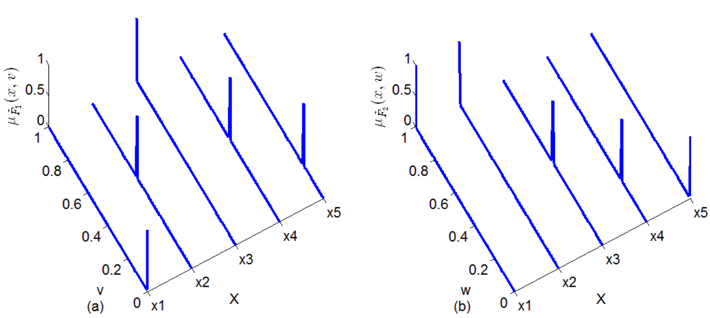

Figure 4.5.26
Sets $\tilde{F}_1$ and $\tilde{F}_2$ when gfIT2FSs are equivalent to type-1 sets.

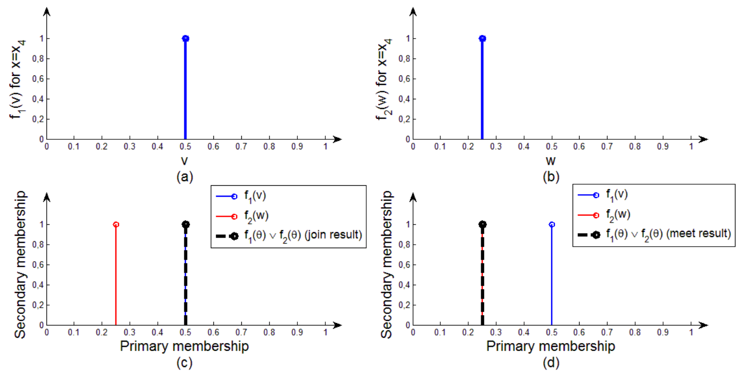

Figure 4.5.27
(a) $f_1(v)$. (b) $f_2(w)$. (c) Sets and join result. (d) Sets and meet result.

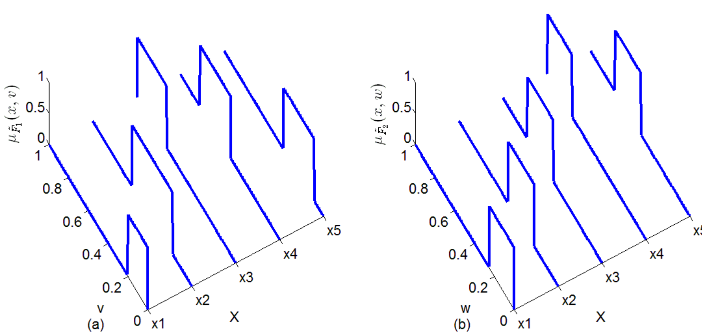

Figure 4.5.28
IVFSs to perform the join and meet operations.

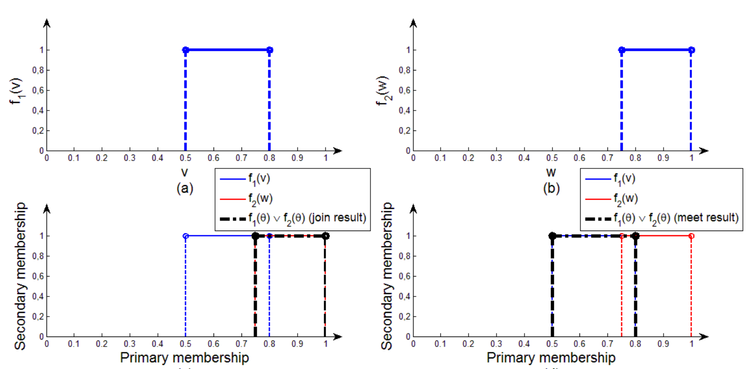

Figure 4.5.29
(a) $f_1(v)$. (b) $f_2(w)$. (c) Sets and join result. (d) Sets and meet result.

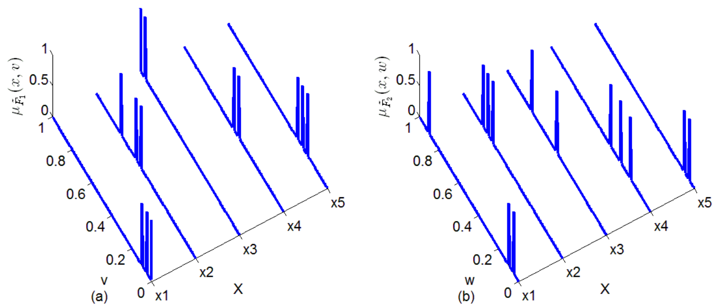

Figure 4.5.30
Multisingleton IT2FSs to perform the join and meet operations.

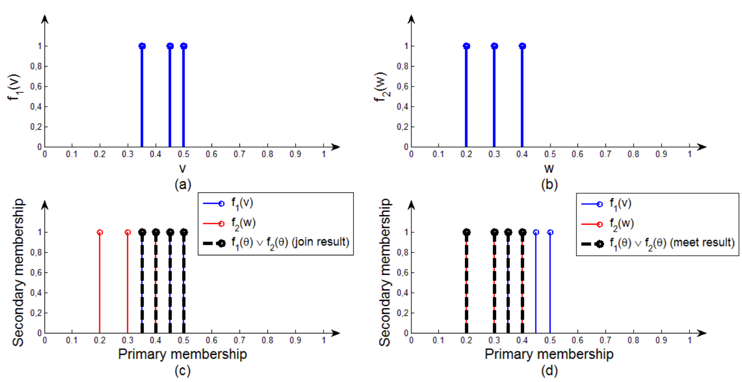

Figure 4.5.31
(a) $f_1(v)$. (b) $f_2(w)$. (c) Sets and join result. (d) Sets and meet result.

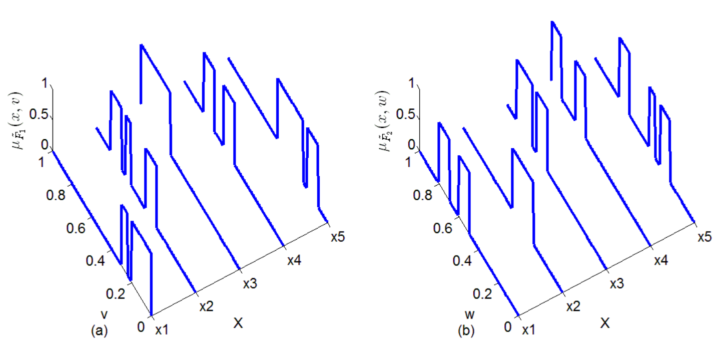

Figure 4.5.32
Multi-IVFSs IT2FSs to perform the join and meet operations.

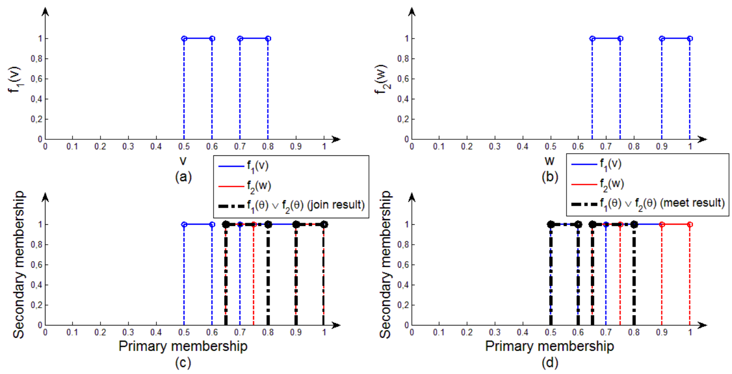

Figure 4.5.33
(a) $f_1(v)$. (b) $f_2(w)$. (c) Sets and join result. (d) Sets and meet result.

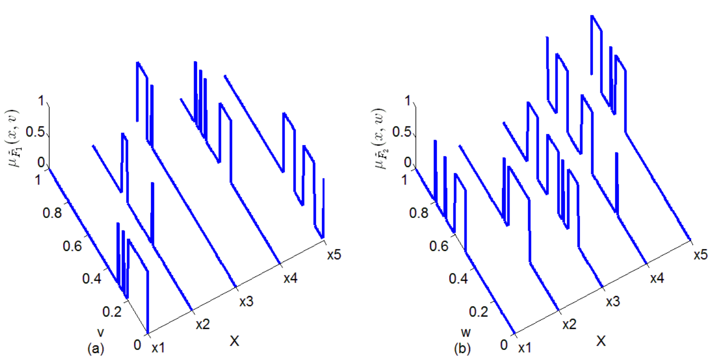

Figure 4.5.34
Sets to perform the join and meet operations.

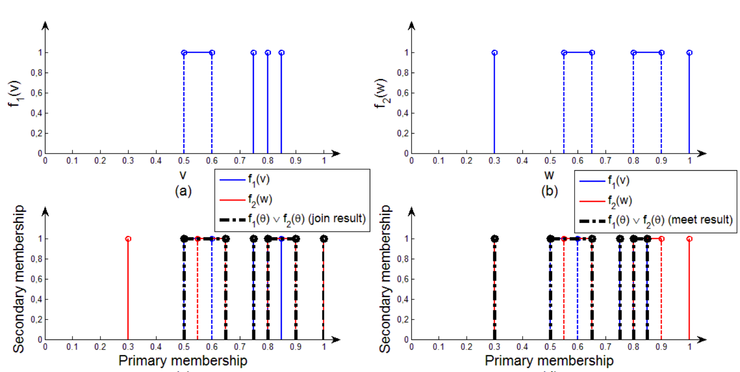

Figure 4.5.35
(a) $f_1(v)$. (b) $f_2(w)$. (c) Sets and join result. (d) Sets and meet result.

4.6 CONCLUSIONS AND FUTURE WORKS

In this Chapter, we have presented two theorems to perform the join and meet operations on any GT2FSs having arbitrary secondary grades, where the restrictions about the normality or convexity on the secondary grades are no longer required. These results have allowed us to deal with both GT2FSs and the general forms of IT2FSs as presented in [89]. Hence, the Chapter will help to explore the potential of gfIT2FLSs that use gfIT2FSs, which are not equivalent to IVFSs.

To complete all the framework related to FLSs using these general forms of IT2FSs, the next chapter will focus on the definition of such FLSs, paying special attention to the type reduction operation, when the sets involved are IT2FSs that are not equivalent to type-1 sets or IVFSs. We will also explore possible applications that will benefit from using more general forms of IT2FSs.

5 TOWARDS A FUZZY LOGIC SYSTEM BASED ON GENERAL FORMS OF INTERVAL TYPE-2 FUZZY SETS

Probability is the only satisfactory measure of one's personal uncertainty about the world and I put forward the challenge that anything that can be done by alternatives to probability can be better done by probability.

— D. V. Lindley

The recent years have witnessed a widespread in the use of interval type-2 fuzzy logic systems (IT2FLSs) in real world applications. It has been shown recently that interval type-2 fuzzy sets (IT2FSs) are more general than interval-valued fuzzy sets (IVFSs) in [89] (and introduced in Chapter 4). Hence, there is a need to explore the capabilities of the more general forms of IT2FSs (beyond IVFSs) and the applications areas they will be more suitable for. In addition, there is a need to develop the theory of the general forms of IT2FLSs (gfIT2FLSs), which employ IT2FSs which are not equivalent to IVFSs and can have non-convex secondary membership functions. Although these systems could be considered within the scope of General Type-2 Fuzzy Logic Systems (GT2FLSs), the practical implementation of GT2FLSs has traditionally required the secondary membership functions to be convex type-1 fuzzy sets (T1FSs); to overcome this drawback, the theory of the join and meet operations on type-2 fuzzy sets having arbitrary secondary grades was developed in Chapter 4. In addition, the type-reduction operation still presents a challenge for GT2FLSs because of its computational complexity. In this Chapter, we present a complete framework for a type-2 FLS (gfIT2FSs) that uses general forms of interval type-2 fuzzy sets, which can be non-convex type-1 fuzzy sets. We will introduce the various operations employed within the gfIT2FLSs, from fuzzification (including singleton and non-singleton) to inference to type-reduction and defuzzification. We will also present some application areas where the use of gfIT2FSs can be beneficial.

5.1 INTRODUCTION TO GFIT2FLSS

Type-1 (T1) and type-2 (T2) FLSs have been widely and successfully used in many real world applications, such as robotics [38], [96], control [12], [83], image processing [123], network traffic control [44], function approximation [84], pattern recognition [42] and many others.

Although GT2FLSs were defined as soon as 1999, their practical application has been limited due to their higher computational complexity, favouring the simpler version of IT2FLSs. Although traditionally IT2FSs have been considered to be equivalent to IVFSs [78], it has been recently shown in [89] and extensively discussed in Chapter 4 that IT2FSs are more general than IVFSs. In addition, some of these IT2FSs have secondary grades which are non-convex T1FSs. Hence, although there is a big literature in both IT2 and GT2FLSs, most of the existing work focuses either on IVFSs or GT2FSs with convex secondary grades. In this Chapter, we consider the gfIT2FSs where the secondary grades are not equivalent to IVFSs and can be non-convex.

GfIT2FSs can easily capture the faced uncertainty without introducing unneeded and unrealistic uncertainty to the IT2FS. For example, a typical method to obtain IVFSs [76] for the antecedents or consequents involve a survey from different people (who know about fuzzy logic) to provide the person type-1 fuzzy set MF which represents a given concept from the person's point of view, as shown in Figure 5.1.1(a). These type-1 fuzzy sets are then aggregated to develop upper and lower MFs, which can embed the various type-1 fuzzy sets obtained within the resulting footprint of uncertainty (FOU) as shown in Figure 5.1.1(b).

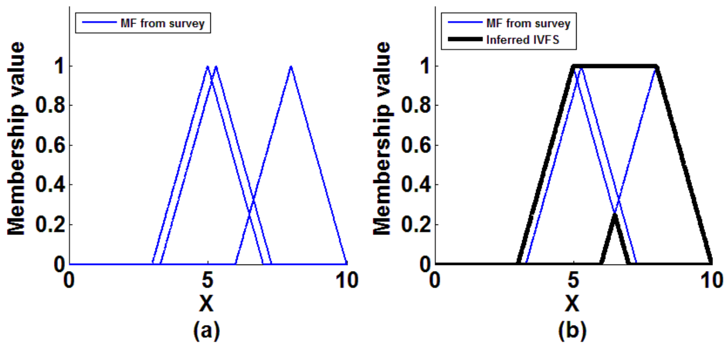

Figure 5.1.1
(a) Type-1 MFs from a given survey about a certain concept. (b) IVFS obtained via embedding the various obtained type-1 fuzzy sets.

Nevertheless, if the different T1 MFs provided are sparse (which might happen in most cases, as shown in Figure 5.1.1(a)), a very wide FOU can be obtained, which implies that a higher level of uncertainty is modelled than what might be present in real world situations. In addition, the obtained FOU in Figure 5.1.1(b) might include huge number of emerging non triangular embedded sets which might not represent the surveyed population opinion.

This problem can be naturally solved by using a specific shape of the general forms of IT2FSs, which we called *multi-singleton IT2FSs*, whose secondary membership is comprised of several singletons at each point within the X-domain (as shown in Figure 5.1.2). These sets can easily represent all sets gathered in the survey to model the faced uncertainty without adding extra unneeded and/or unrealistic uncertainty to the final type-2 fuzzy set.

In Chapter 4 the theory for the join and meet operations on GT2FSs with non-convex/arbitrary secondary grades was presented; in addition, special attention was drawn to the case of non-convex IT2FSs, which we have been referring to as *general forms of IT2FSs* (gfIT2FSs). Once these set theoretic operations are available, the fuzzy inference engine for the gfIT2FLSs can be defined. This work aims to present the whole structure of gfIT2FLSs as many of the elements are analogous to those in other kinds of FLS (i.e. IVFLSs and GT2FLSs).

The rest of the Chapter is organised as follows: In Section 5.2 a new way of representing the gfIT2FSs is introduced, which will be very useful in the notations during this Chapter. Section 5.3 presents two new theorems related with the join and meet operations on gfIT2FSs. Section 5.4 presents the structure of the singleton and non-singleton gfIT2FLSs. Section 5.5 presents experi-

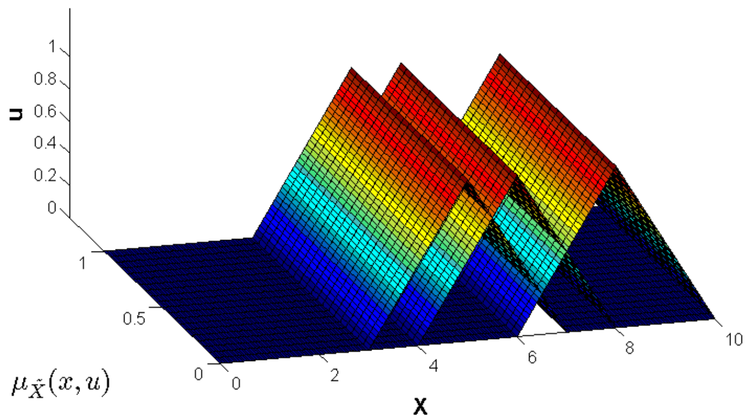

Figure 5.1.2
A multi-singleton IT2FS to model the uncertainty amongst the type-1 fuzzy sets in Figure 5.1.1(a).

mental analysis with worked examples explaining the operation of the gfIT2FLSs, and a summary of the operation of a gfIT2FLS, whereas Section 5.6 introduces conclusions and future works.

5.2 REPRESENTING GFIT2FSS

In [89] and also n Chapter 4 (Section 4.5) the different types of gfIT2FSs were introduced; in this Section a new way of representing such sets is introduced, which will be useful for the notations used in this dissertation.

If we consider that a singleton can be represented as an interval having the same extreme points, then all cases of IT2FSs, from A to E in Section 4.5, can be represented as the union of a finite number of closed intervals, say $N_{\tilde{X}}(x)$. It is worthwhile to highlight that $N_{\tilde{X}}$ depends on the primary variable x, as not all the secondary grades may have the same number of disjointed singletons/intervals. Hence, a gfIT2FS $\tilde{X}$ whose memebership function is $\mu_{\tilde{X}}(x)$, which is defined over a universe of discourse X (with $x \in X$), can be written as follows:

$$\mu_{\tilde{X}}(x) = \bigcup_{i=1}^{N_{\tilde{X}}(x)} \mu^i_{\tilde{X}}(x) = \bigcup_{i=1}^{N_{\tilde{X}}(x)} 1/\left[l^i_{\tilde{X}}(x), r^i_{\tilde{X}}(x)\right] \tag{5.2.1}$$

Where each of the closed intervals $\mu^i_{\tilde{X}}(x) = \left[l^i_{\tilde{X}}(x), r^i_{\tilde{X}}(x)\right]$ comprising $\mu_{\tilde{X}}(x)$ will be referred to as *the subintervals of* $\mu_{\tilde{X}}(x)$. As we are dealing with gfIT2FSs, all the non-zero secondary membership values will be equal to one, hence the 1 in Equation 5.2.1 is redundant and sometimes we will drop it from the notation for the sake of simplicity. Moreover, in Equation 5.2.1 we have explicitly indicated the dependency of $N_{\tilde{X}}$, $l^i_{\tilde{X}}$ and $r^i_{\tilde{X}}$ with x, also for the sake of simplicity, we may not always indicate so. Hence, Equation 5.2.1 can be simplified as:

$$\mu_{\tilde{X}}(x) = \bigcup_{i=1}^{N_{\tilde{X}}} \mu^i_{\tilde{X}}(x) = \bigcup_{i=1}^{N_{\tilde{X}}} \left[l^i_{\tilde{X}}, r^i_{\tilde{X}}\right] \tag{5.2.2}$$

At this point it would be a good idea to revise the concepts of lower membership function (LMF) and upper membership function (UMF). In [58] Liang and Mendel defined the LMF and UMF of an IVFS as the bounds for the footprint of uncertainty (FOU); however, in their definition the FOU was a closed and connected region. When dealing with gfIT2FSs, the FOU may have several boundaries; thus, to avoid any kind of ambiguity, we will redefine the LMF and UMF of a gfIT2FS:

Definition 5.2.1. Let $\tilde{X}$ be a gfIT2FS and let $\mu_{\tilde{X}}(x)$ be a vertical slice placed at each x, $\mu_{\tilde{a}}(x) = f(x, \theta))$, $\theta \in [0, 1]$. For each $f(x, \theta)$, let $\nu_1 = \nu_1(x)$ be the infimum of the support of $f(x, \theta)$. Hence, the *lower membership function (LMF)* of $\tilde{X}$ is:

$$\text{LMF}(\tilde{X}) = \text{LMF}(\mu_{\tilde{X}}(x)) = \underline{\mu}_{\tilde{X}}(x) = \{\nu_1(x) \mid x \in X\} \tag{5.2.3}$$

Definition 5.2.2. Let $\tilde{X}$ be a gfIT2FS and let $\mu_{\tilde{X}}(x)$ be a vertical slice placed at each x, $\mu_{\tilde{a}}(x) = f(x, \theta))$, $\theta \in [0, 1]$. For each $f(x, \theta)$, let $\nu_{end} = \nu_{end}(x)$ be the supremum of the support of $f(x, \theta)$. Hence, the *upper membership function (UMF)* of $\tilde{X}$ is:

$$\text{UMF}(\tilde{X}) = \text{UMF}(\mu_{\tilde{X}}(x)) = \overline{\mu}_{\tilde{X}}(x) = \{\nu_{end}(x) \mid x \in X\} \tag{5.2.4}$$

Definitions 5.2.1 and 5.2.2 can also be applied to every single subinterval $\mu^i_{\tilde{X}}(x)$ comprising a gfIT2FS. Hence, we will distinguish between the LMF/UMF of a gfIT2FS $\mu_{\tilde{X}}(x)$ and the LMF/UMF of a subinterval $\mu^i_{\tilde{X}}(x)$ within a gfIT2FS, which will be denoted as $\text{LMF}\left(\mu^i_{\tilde{X}}(x)\right) = \underline{\mu}^i_{\tilde{X}}(x)$ and $\text{UMF}\left(\mu^i_{\tilde{X}}(x)\right) = \overline{\mu}^i_{\tilde{X}}(x)$, respectively, with $i = 1, ..., N_{\tilde{X}}$. Once the LMF and UMF for subintervals have been defined, Equation 5.2.2 could be rewritten as follows:

$$\mu_{\tilde{X}}(x) = \bigcup_{i=1}^{N_{\tilde{X}}} \mu^i_{\tilde{X}}(x) = \bigcup_{i=1}^{N_{\tilde{X}}} \left[\underline{\mu}^i_{\tilde{X}}(x), \overline{\mu}^i_{\tilde{X}}(x)\right] \tag{5.2.5}$$

It is worthwhile to highlight that for the sake of simplicity, we will drop (x) from the notation of $\underline{\mu}^i_{\tilde{X}}(x)$ and $\overline{\mu}^i_{\tilde{X}}(x)$ and they will be written as $\underline{\mu}^i_{\tilde{X}}$ and $\overline{\mu}^i_{\tilde{X}}$.

5.3 NUMBER OF SUBINTERVALS PRESENT IN THE JOIN AND MEET RESULTS FROM TWO GFIT2FSS

The join and meet operations on the gfIT2FSs were extensively discussed in Chapter 4; firstly, these operations were introduced for GT2FSs having arbitrary secondary grades (Section 4.3) obtaining the results given by Equations (4.3.1) and (4.3.25); later, in Section 4.4, specific versions of those equations were found when the involved sets were all kinds of gfIT2FSs, from A to E, as described in Section 4.5.3. These Equations, (4.4.7) and (4.4.14), describe the join and meet operations on the gfIT2FSs, which will be useful when defining the inference engine of the gfIT2FLSs. As this topic was extensively discussed in Chapter 4, it will not be repeated here. For further information, please refer to that Chapter.

Nonetheless, in this section two new theorems new results are introduced by two new theorems, regarding the number of subintervals the join and meet operation result will have.

5.3.1 *Number of subintervals in the final meet result*

Theorem 5.3.1. *Let* $\tilde{X}_1$, $\tilde{X}_2$ *be two IT2FSs, defined in a universe of discourse* X*, and characterised by their respective membership functions*

$\mu_{\tilde{X}_1}(x)$ *and* $\mu_{\tilde{X}_2}(x)$*, with* $x \in X$*. Let* $\mu_{\tilde{X}_1}(x)$ *and* $\mu_{\tilde{X}_2}(x)$ *be represented by the union of their respective finite number of subintervals, i.e.:*

$$\begin{aligned} \mu_{\tilde{X}_1}(x) &= \bigcup_{i=1}^{N_{\tilde{X}_1}} \left[l^i_{\tilde{X}_1}, r^i_{\tilde{X}_1}\right] = \bigcup_{i=1}^{N_{\tilde{X}_1}} \left[\underline{\mu}^i_{\tilde{X}_1}, \overline{\mu}^i_{\tilde{X}_1}\right] \\ \mu_{\tilde{X}_2}(x) &= \bigcup_{j=1}^{N_{\tilde{X}_2}} \left[l^j_{\tilde{X}_2}, r^j_{\tilde{X}_2}\right] = \bigcup_{j=1}^{N_{\tilde{X}_2}} \left[\underline{\mu}^j_{\tilde{X}_2}, \overline{\mu}^j_{\tilde{X}_2}\right] \end{aligned} \tag{5.3.1}$$

Then, at a given $x \in X$*, the meet operation* $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$ *on the two vertical slices* $\mu_{\tilde{X}_1}(x)$ *and* $\mu_{\tilde{X}_2}(x)$ *will be a T1FS comprised by a finite number of closed, connected and disjointed intervals between* 1 *and* $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$*.*

Proof. Before starting the proof, it is worthwhile to highlight that both quantities $N_{\tilde{X}_1}$ and $N_{\tilde{X}_2}$ are really dependent with x, it is, $N_{\tilde{X}_1} = N_{\tilde{X}_1}(x)$ and $N_{\tilde{X}_2} = N_{\tilde{X}_2}(x)$, as every single vertical slice within $\tilde{X}_1$ and $\tilde{X}_2$ may have different number of subintervals. However, for the sake of simplicity, we will dropthe dependecy with x of these quantities during the proof.

Let us represent both $\mu_{\tilde{X}_1}(x)$ and $\mu_{\tilde{X}_2}(x)$ as in Equation (5.3.1). It is important to note that the quantities referred to as v_1, v_{end}, w_1 and w_{end} correspond to $l^1_{\tilde{X}_1}$, $r^{N_{\tilde{X}_1}}_{\tilde{X}_1}$, $l^1_{\tilde{X}_2}$ and $r^{N_{\tilde{X}_2}}_{\tilde{X}_2}$, respectively. By the definition of subinterval (i.e., each of the closed, connected and disjointed intervals within a gfIT2FS), at a given $x \in X$, the following is trivial:

$$\begin{aligned} \mu^{i_1}_{\tilde{X}_1}(x) \cap \mu^{i_2}_{\tilde{X}_1}(x) &= \emptyset \qquad \forall i_1, i_2 = 1, ..., N_{\tilde{X}_1} \qquad i_1 \neq i_2 \\ \mu^{j_1}_{\tilde{X}_2}(x) \cap \mu^{j_2}_{\tilde{X}_2}(x) &= \emptyset \qquad \forall j_1, j_2 = 1, ..., N_{\tilde{X}_2} \qquad j_1 \neq j_2 \end{aligned} \tag{5.3.2}$$

Let us also remember the equation for the meet operation on two gfIT2FSs, which is as in Equation (4.4.14) and is repeated here for the convenience of the reader:

$$\mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta) = \begin{cases} f_1(\theta) \vee f_2(\theta) & \forall \theta \in [\min(v_1, w_1), \min(v_{end}, w_{end})] \\ 0 & elsewhere \end{cases} \tag{5.3.3}$$

Where v_1 and v_{end} are the infimum and supremum of the support of f_1, and w_1 and w_{end} are the infimum and the supremum of the support of f_2, as introduced in Section 4.4).

By definition of v_1 and w_1, the lower bound $\min(v_1, w_1) \leqslant \theta$ does not truncate the term $f_1(\theta) \vee f_2(\theta)$ as $f_1(\theta)$ and $f_2(\theta)$ are equal to 0 $\forall\theta < \min(v_1, w_1)$. Hence, we can rewrite Equation (5.3.3) as Equation (5.3.4):

$$\mu_{\tilde{F}_1 \cap \tilde{F}_2}(x, \theta) = \begin{cases} f_1(\theta) \vee f_2(\theta) & \forall\theta \leqslant \min(v_{end}, w_{end}) \\ 0 & elsewhere \end{cases} \tag{5.3.4}$$

Part 1: let us assume that all the subintervals from $\mu_{\tilde{X}_1}(x)$ and $\mu_{\tilde{X}_2}(x)$ are disjointed except the last one from each of them, it is:

$$\mu^i_{\tilde{X}_1}(x) \cap \mu^j_{\tilde{X}_2}(x) = \emptyset \qquad \begin{aligned} &\forall i = 1, ..., N_{\tilde{X}_1} \\ &\forall j = 1, ..., N_{\tilde{X}_2} \end{aligned} \tag{5.3.5}$$

Considering that $v_{end} \in \mu^{N_{\tilde{X}_1}}_{\tilde{X}_1}(x)$ and $w_{end} \in \mu^{N_{\tilde{X}_2}}_{\tilde{X}_2}(x)$, then we can assure that $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$ will have, at least, $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 2$ subintervals (it is, all subintervals except the last one from each vertical slice). Now let us see what happens depending on the relative position of $\mu^{N_{\tilde{X}_1}}_{\tilde{X}_1}(x)$ and $\mu^{N_{\tilde{X}_2}}_{\tilde{X}_2}(x)$.

Part 1.1: $\mu^{N_{\tilde{X}_1}}_{\tilde{X}_1}(x)$ and $\mu^{N_{\tilde{X}_2}}_{\tilde{X}_2}(x)$ do not overlap (I).

In this case, as in Figure 5.3.1, $v_{end} < w_{end} \rightarrow \min(v_{end}, w_{end}) = v_{end}$, hence, the term $\theta \leqslant \min(v_{end}, w_{end})$ in Equation (5.3.4) truncates $\mu^{N_{\tilde{X}_2}}_{\tilde{X}_2}(x)$ completely, and just $\mu^{N_{\tilde{X}_1}}_{\tilde{X}_1}(x)$ remains in the resulting $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$. That situation leads to $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$ subintervals in $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$.

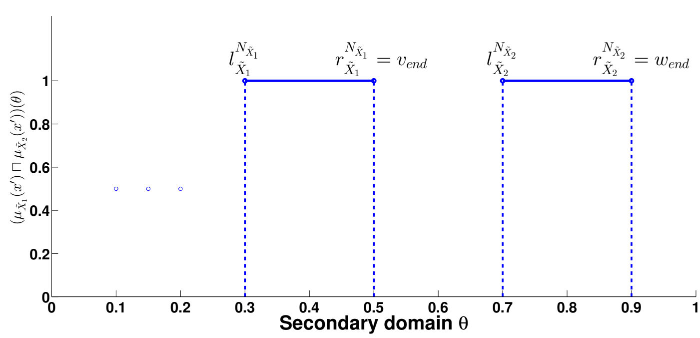

Figure 5.3.1
Final subintervals do not overlap (I).

Part 1.2: $\mu^{N_{\tilde{X}_1}}_{\tilde{X}_1}(x)$ and $\mu^{N_{\tilde{X}_2}}_{\tilde{X}_2}(x)$ do not overlap (II).

This situation (depicted in Figure 5.3.2) is completely analogous to the one described in Part 1.1, but interchanging the order of $\mu_{\tilde{X}_1}^{N_{\tilde{X}_1}}(x)$ and $\mu_{\tilde{X}_2}^{N_{\tilde{X}_2}}(x)$. Now, $w_{end} < v_{end} \rightarrow \min(v_{end}, w_{end}) = W_{end}$, hence, the term $\theta \leqslant \min(v_{end}, w_{end})$ in Equation (5.3.4) truncates $\mu_{\tilde{X}_1}^{N_{\tilde{X}_1}}(x)$, and just $\mu_{\tilde{X}_2}^{N_{\tilde{X}_2}}(x)$ remains in $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$, thus leading to $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$ subintervals.

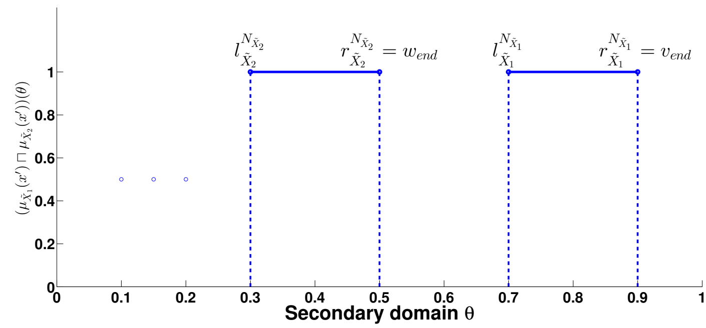

Figure 5.3.2
Final subintervals do not overlap (II).

Part 1.3: $\mu_{\tilde{X}_1}^{N_{\tilde{X}_1}}(x)$ and $\mu_{\tilde{X}_2}^{N_{\tilde{X}_2}}(x)$ do overlap (I).

In this case, illustrated in Figure 5.3.3, the term $\theta \leqslant \min(v_{end}, w_{end}) = v_{end}$ just truncates the section $\left(r_{\tilde{X}_1}^{N_{\tilde{X}_1}}, r_{\tilde{X}_2}^{N_{\tilde{X}_2}}\right] = (v_{end}, w_{end}]$. The result is one single interval that will remain in $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$, again leading to $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$ subintervals.

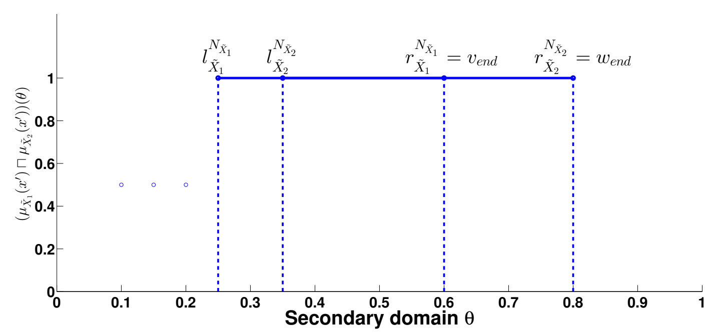

Figure 5.3.3
Final subintervals do overlap (I).

Part 1.4: $\mu_{\tilde{X}_1}^{N_{\tilde{X}_1}}(x)$ and $\mu_{\tilde{X}_2}^{N_{\tilde{X}_2}}(x)$ do overlap (II).

Analogous to Part 1.3 (as in Figure 5.3.4), now the term $\theta \leqslant \min(v_{end}, w_{end}) = w_{end}$ just truncates the section $\left(r_{\tilde{X}_2}^{N_{\tilde{X}_2}}, r_{\tilde{X}_1}^{N_{\tilde{X}_1}}\right] = (w_{end}, v_{end}]$. The result is again one single interval, and thus $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$ will have $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$ subintervals.

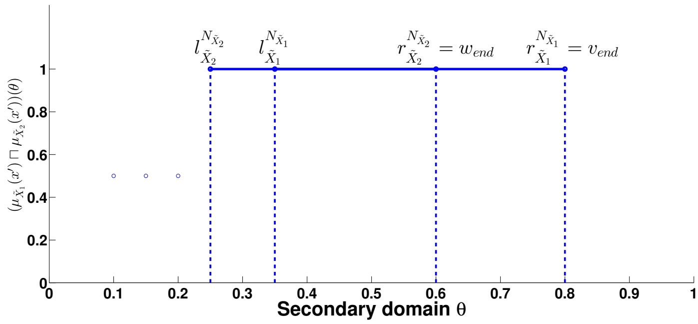

Figure 5.3.4
Final subintervals do overlap (II).

Part 1.5: $\mu_{\tilde{X}_1}^{N_{\tilde{X}_1}}(x)$ and $\mu_{\tilde{X}_2}^{N_{\tilde{X}_2}}(x)$ are contained one within the other.

As it can be seen from Figures 5.3.5 and 5.3.6, no matter if $\mu_{\tilde{X}_2}^{N_{\tilde{X}_2}}(x) \subseteq \mu_{\tilde{X}_1}^{N_{\tilde{X}_1}}(x)$ or $\mu_{\tilde{X}_1}^{N_{\tilde{X}_1}}(x) \subseteq \mu_{\tilde{X}_2}^{N_{\tilde{X}_2}}(x)$, the result is one interval and $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$ will have $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$ subintervals.

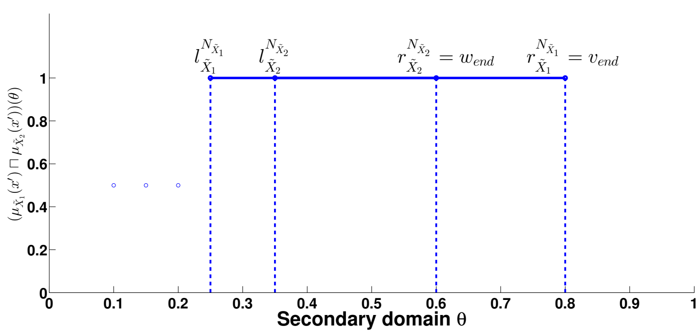

Figure 5.3.5
Subintervals are contained one within another (I).

Hence, to summarise, Part 1 proves that $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$ can have, ***at most***, $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$.

Part 2: now let us assume a slightly different version of Equation (5.3.5), as in Equation (5.3.6):

$$\mu_{\tilde{X}_1}^{i}(x) \cap \mu_{\tilde{X}_2}^{j}(x) = \emptyset \qquad \begin{matrix} \forall i = 1, ..., N_{\tilde{X}_1} & \forall i \neq i_1 \\ \forall j = 1, ..., N_{\tilde{X}_2} & \forall j \neq j_1 \end{matrix} \tag{5.3.6}$$

It is, there is a given i_1 and a given j_1 for which $\mu_{\tilde{X}_1}^{i_1}(x) \cap \mu_{\tilde{X}_2}^{j_1}(x) \neq \emptyset$. Hence, repeating the reasoning presented in Part 1 but considering Equation (5.3.6) instead of (5.3.5) would lead us to $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$ having $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 2$ subintervals.

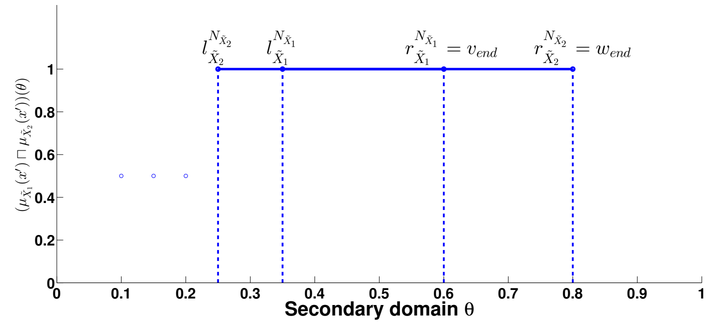

Figure 5.3.6
Subintervals are contained one within another (II).

Part 3: the reasoning presented in Part 2 can be generalised, and it is easy to think of any distribution of $\mu^i_{\tilde{X}_1}(x)$ and $\mu^j_{\tilde{X}_2}(x)$, some of them overlapping and some other not, that would lead to a $\mu_{\tilde{X}_1}(x) \sqcap \mu_{\tilde{X}_2}(x)$ having any number of subintervals between 1 and $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$.

□

5.3.2 *Number of subintervals in the final join result*

Theorem 5.3.2. *Let $\tilde{X}_1$, $\tilde{X}_2$ be two IT2FSs, defined in a universe of discourse X, and characterised by their respective membership functions $\mu_{\tilde{X}_1}(x)$ and $\mu_{\tilde{X}_2}(x)$, with $x \in X$. Let $\mu_{\tilde{X}_1}(x)$ and $\mu_{\tilde{X}_2}(x)$ be represented by the union of their respective finite number of subintervals, as in Equation (5.3.1). Then, at a given $x \in X$, the join operation $\mu_{\tilde{X}_1}(x) \sqcup \mu_{\tilde{X}_2}(x)$ on the two vertical slices $\mu_{\tilde{X}_1}(x)$ and $\mu_{\tilde{X}_2}(x)$ will be a T1FS comprised by a finite number of closed, connected and disjointed intervals between 1 and $N_{\tilde{X}_1} + N_{\tilde{X}_2} - 1$.*

Proof. The proof ot Theorem 5.3.2 is completely analogous to that of Theorem 5.3.1, but in this case the term truncating $f_1(\theta) \vee f_2(\theta)$ is $\theta \geqslant \max(v_1, w_1)$, instead of $\theta \leqslant \min(v_{end}, w_{end})$, and the involved subintervals in the reasoning presented in Parts 1 to 3 are $\mu^1_{\tilde{X}_1}(x)$ and $\mu^1_{\tilde{X}_2}(x)$ instead of $\mu^{N_{\tilde{X}_1}}_{\tilde{X}_1}(x)$ and $\mu^{N_{\tilde{X}_2}}_{\tilde{X}_2}(x)$, respectively. □

5.4 STRUCTURE OF THE GFIT2FLSS

In this section we will review the structure of a gfIT2FLS, which is depicted in Figure 5.4.1

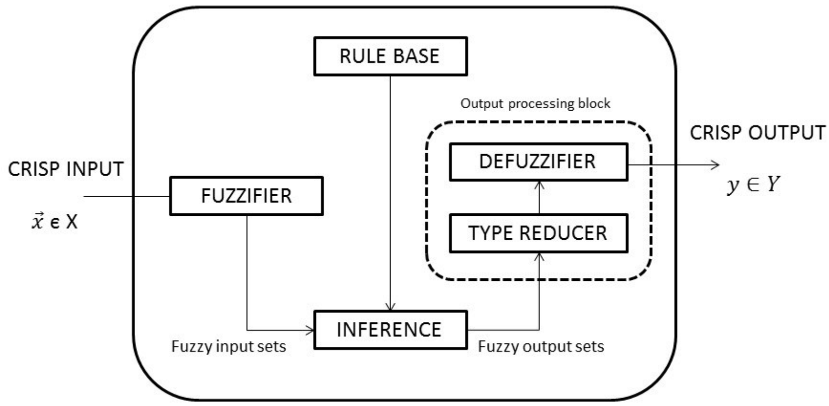

Figure 5.4.1
Structure of an gfIT2FLS.

5.4.1 *The fuzzifier: singleton and non-singleton gfIT2FLSs*

The fuzzifier block maps each component within the input vector $\vec{x} = (x_1, ..., x_p)$ into a gfIT2FS, which will be the input fuzzy sets for the inference engine, denoted as $\mu_{\tilde{X}_i}(x_i)$, $i = 1, ..., p$.

5.4.1.1 *Shapes of the input FSs for singleton and non-singleton fuzzification*

When using singleton fuzzification, each input x_i is mapped into a fuzzy singleton, i.e., a fuzzy set having a single point of non-zero membership, placed at x_i; both the primary and the secondary memberships are equal to unity. An example of this kind of sets is depicted in Figure 5.4.2(a). On the other hand, when using non-singleton fuzzification, each input x_i will be mapped into a gfIT2FS; an example is plotted in Figure 5.4.2(b). It is important to highlight that the secondary grade/third dimension of Figure 5.4.2(b) has not been plotted in order to make the gfIT2FS easier to visualise.

5.4.1.2 *Modelling inputs as gfIT2FSs*

In this subsection we introduce a general method to fuzzify crisp input values into gfIT2FSs. We assume we have a given input, say

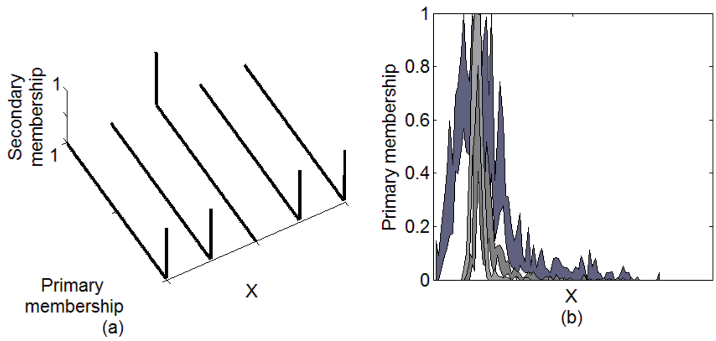

Figure 5.4.2
Examples of input fuzzy sets for: (a) Singleton fuzzification. (b) Non-singleton fuzzification.

X_i, which is measured in the presence of several sources of uncertainty, say S_j, providing an input value of x_i. This modelling has to be done ahead of time and will lead to a multi-interval IT2FS, which could be summarised as:

1. For a given source of uncertainty, S_j, $j = 1, ..., J$, which will provide some input uncertainty σ_{S_j}, *force* the input variable to have a certain value, say X_i, $i = 1, ..., V$, where V is the number of *real* values selected for the input variable. Perform M measurements for X_i to have X_i^m, $m = 1, ..., M$, in the presence of noise σ_{S_j}.

2. Obtain the histogram H_1 of the M measured values X_i^m for the real input value X_i.

3. Normalise the previous histogram to its maximum value, in order to scale it within the range $[0, 1]$, obtaining the normalised histogram HN_1.

4. Repeat steps 1-3 K times, in order to obtain K different normalised histograms HN_k, $k = 1, ..., K$, for the real input value X_i and uncertainty source S_j.

5. For each measured value within the histogram x_i, assign the minimum of the histograms at that x_i as the lower membership value, and the maximum of the histograms as the upper membership value. This will lead to an IVFS representing uncertainty from source S_j.

6. Repeat steps 1-5 for every source of uncertainty considered. This will lead to a total of J IVFSs available.

7. Create a multi-interval IT2FS by overlapping the J IVFSs obtained; the resulting set will be a gfIT2FS representing uncertainty *from all sources considered*.

This process (summarised as a diagram in Figure 5.4.3) has to be done for each possible *real* input value X_i, which can actually require a huge amount of work. This drawback can be solved by doing the modelling just for a finite number of input values, say X_1, X_2, X_3, etc. Hence, the gfIT2FS corresponding to a value verifying $X_i < X - X_{i+1}$ can be obtained by interpolating the closest sets from both left and right, i.e., interpolating X_i and X_{i+1}. A detailed example about how to apply this method is presented in Section 5.5.

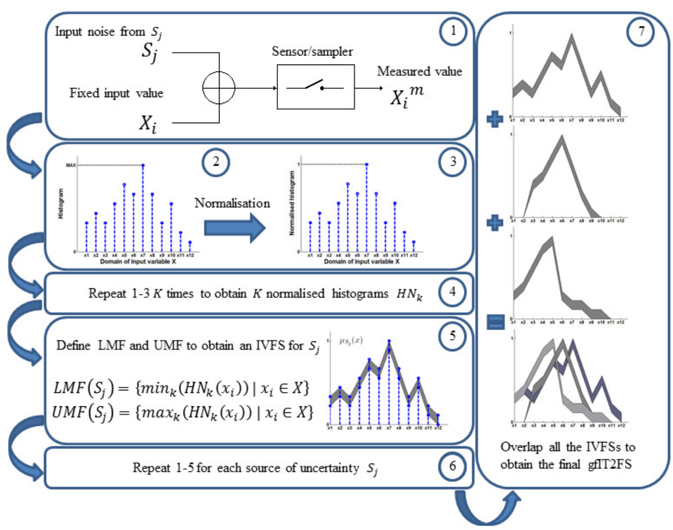

Figure 5.4.3
Diagram of the method to model input noise from different sources as gfIT2FSs.

5.4.2 *The rule base*

As in the T1 and GT2 versions, the rule base of a multiple-input single-output (MISO) gfIT2FLS is comprised by a set of M IF-THEN rules, where the l-th rule, $l = 1, ..., M$, is denoted as R^l and is as follows:

$$R^l : \text{ IF } x_1 \text{ is } \tilde{F}_1^l \text{ AND } x_2 \text{ is } \tilde{F}_2^l \text{ AND ... AND } x_p \text{ is } \tilde{F}_p^l \text{ THEN } y \text{ is } \tilde{Y}^l \tag{5.4.1}$$

Where x_is are the inputs to the system, $\tilde{F}_i^l$ is the antecedent of the i-th input ($i = 1, ..., p$) of the l-th rule ($l = 1, ..., M$), y is the output of the system and $\tilde{Y}^l$ is the consequent set for that rule. As we are considering gfIT2FLSs, all the fuzzy sets involved in each rule (both antecedents and consequents) are assumed to be gfIT2FSs.

5.4.3 *The inference engine*

Once the inputs have been fuzzified, the fuzzifier block provides the input gfIT2FSs to the inference engine, which is described by a fuzzy rule-base. The inference engine, which is comprised of M IF-THEN rules, is activated and each fired rule provides an output fuzzy set, resulting from combining all the antecedents in the rule for the given inputs with their corresponding consequent fuzzy sets. The output fuzzy set for that rule $\tilde{B}^l$, $l = 1, ..., M$, which is characterised by its membership function $mu_{\tilde{B}^l}(y)$, is given by [78]:

$$\mu_{\tilde{B}^l}(y) = \left[\prod_{i=1}^{p} \left(\bigsqcup_{x_i \in X_i} \mu_{\tilde{X}_i}(x_i) \sqcap \mu_{\tilde{F}_i^l}(x_i) \right) \right] \sqcap \mu_{\tilde{Y}^l}(y) \quad (5.4.2)$$

Where $\tilde{X}_i$ is the gfIT2FS associated with the input x_i, $\tilde{F}_i^l$ is the antecedent of the i-th input of the l-th rule and $\tilde{Y}^l$ is the consequent set. Equation (5.4.2) is valid in the general non-singleton input fuzzification for GT2FLSs; when using the simpler singleton fuzzification, the input fuzzy $\tilde{X}_i$ is a singleton type-2 fuzzy set, the term $\mu_{\tilde{X}_i}(x_i) \sqcap \mu_{\tilde{F}_i^l}(x_i)$ reduces to $\mu_{\tilde{F}_i^l}(x_i)$, and Equation (5.4.2) is significantly simplified and becomes Equation (5.4.3):

$$\mu_{\tilde{B}^l}(y) = \left[\prod_{i=1}^{p} \mu_{\tilde{F}_i^l}(x_i) \right] \sqcap \mu_{\tilde{Y}^l}(y) \quad (5.4.3)$$

The term $\left[\prod_{i=1}^{p} \left(\bigsqcup_{x_i \in X_i} \mu_{\tilde{X}_i}(x_i) \sqcap \mu_{\tilde{F}_i^l}(x_i) \right) \right]$ in Equation (5.4.2) and $\left[\prod_{i=1}^{p} \mu_{\tilde{F}_i^l}(x_i) \right]$ in Equation (5.4.3) are called the *rule firing strength*, a term which is usually denoted as $F^l(\vec{x})$:

$$F^l(\vec{x}) = \left[\prod_{i=1}^{p} \left(\bigsqcup_{x_i \in X_i} \mu_{\tilde{X}_i}(x_i) \sqcap \mu_{\tilde{F}_i^l}(x_i) \right) \right] \quad (5.4.4)$$

$$F^l(\vec{x}) = \left[\prod_{i=1}^{p} \mu_{\tilde{F}_i^l}(x_i) \right] \qquad (5.4.5)$$

When the IT2FSs involved have just one interval (which have been previously called IT2FSs, but are referred as IVFSs in this Chapter), Mendel provides a graphical interpretation to obtain the firing strength [78]. According to Equation (5.4.2), when dealing with IVFSs, after meeting the fuzzy input $\mu_{\tilde{X}_i}(x_i)$ with its corresponding antecedent $\mu_{\tilde{F}_i^l}(x_i)$, we will obtain an interval, which we will denote as $\tilde{A}_i^l = [\underline{a}_i^l, \overline{a}_i^l]$. The extremes of this interval can be obtained from the graphical intersection of the plotted sets as depicted in Figure 5.4.4:

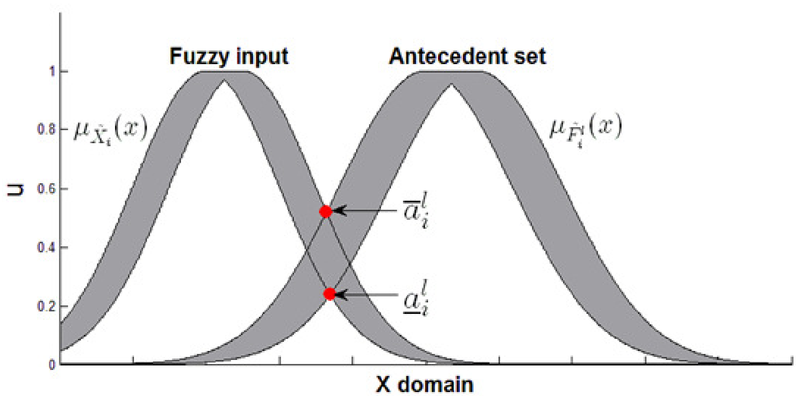

Figure 5.4.4
Graphical interpretation of the antecedent activation when using IV non-singleton IVFLSs.

The lower firing strength extreme is the supremum of the intersection of the LMFs, whereas the upper firing strength extreme is the supremum of the intersection of the UMFs.

Using both Definition 5.2.1 and 5.2.2 in Section 5.2, we can reinterpret the antecedent activation when the involved sets are gfIT2FSs. An example is provided in Figure 5.4.5. The result of this operation, also described in Equation (5.4.4), will be a T1FS comprised of one or more closed and connected intervals. Similar to the interpretation of Figure 5.4.4, the infimum ν_1 and the supremum ν_{end} of the support of the resulting set will be the supremum of the intersection of the LMF and UMF of both $\mu_{\tilde{X}_i}(x_i)$ and $\mu_{\tilde{F}_i^l}(x_i)$, respectively; however, some gaps are expected to appear between these two values as the delimited region is not continuous. Hence, all the present intervals between ν_1 and ν_{end} will arise from the several intersections occurring in Figure 5.4.5 from the UMF and LMF of each of the subintervals comprising $\mu_{\tilde{X}_i}(x_i)$ and $\mu_{\tilde{F}_i^l}(x_i)$, i.e., from all the intersections of $\underline{\mu}^q_{\tilde{X}_i}(x_i)$, $\overline{\mu}^q_{\tilde{X}_i}(x_i)$, $\underline{\mu}^j_{\tilde{F}_i^l}(x_i)$ and $\overline{\mu}^j_{\tilde{F}_i^l}(x_i)$, which are highlighted with red markers in Figure 5.4.5. Figure 5.4.6 is the same as Figure 5.4.5 but zooming in the intersection region.

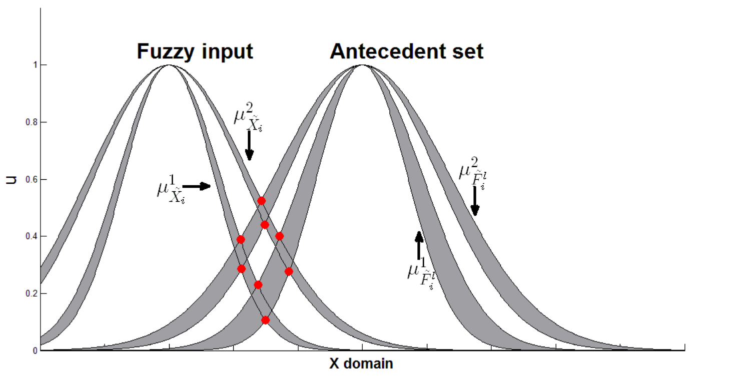

Figure 5.4.5
Graphical interpretation of the antecedent activation for gfIT2FLSs.

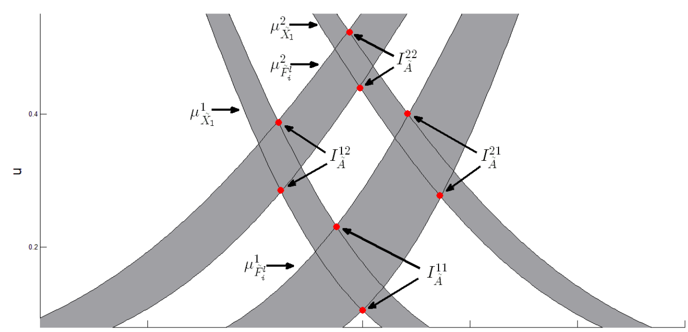

Figure 5.4.6
Region with the intersections of each of the involved subintervals that will comprise the antecedent activation degree.

From Figure 5.4.6 it can be easily deduced that the antecedent activation degree for this example, which we will denote as $\mu_{\tilde{A}}(x)$, will be as:

$$\mu_{\tilde{A}}(x) = \bigcup_{q=1}^{N_{\tilde{X}_i}} \bigcup_{j=1}^{N_{\tilde{F}_i^l}} \sqcup_{x \in X} \left(\left[\underline{\mu}^q_{\tilde{X}_i}(x), \overline{\mu}^q_{\tilde{X}_i}(x) \right] \sqcap \left[\underline{\mu}^j_{\tilde{F}_i^l}(x), \overline{\mu}^j_{\tilde{F}_i^l}(x) \right] \right) =$$

$$= \bigcup_{i=1}^{N_{\tilde{A}}} \left[\underline{\mu}^i_{\tilde{A}}(x), \overline{\mu}^i_{\tilde{A}}(x) \right]$$

(5.4.6)

It is, for every combination of subintervals $\left[\underline{\mu}^{q}_{\tilde{X}_i}(x), \overline{\mu}^{q}_{\tilde{X}_i}(x)\right]$ and $\left[\underline{\mu}^{j}_{\tilde{F}^l_i}(x), \overline{\mu}^{j}_{\tilde{F}^l_i}(x)\right]$, $q = 1, ..., N_{\tilde{X}_i}$ and $j = 1, ..., N_{\tilde{F}^l_i}$, an interval will be obtained, whose left-most point will be the supremum of the minimum of the LMFs of those subintervals, and whose right-most point will be the supremum of the UMFs of those subintervals. The final set will be the union of all the intervals obtained for $q = 1, ..., N_{\tilde{X}_i}$ and $j = 1, ..., N_{\tilde{F}^l_i}$.

Equation (5.4.6) can be generalised to any antecedent belonging to a general rule, as those specified in Equation (5.4.1). Hence, for a given non-singleton input $\mu_{\tilde{X}_k}(x_k)$ will be as in Equation (5.4.7):

$$\mu_{\tilde{A}^l_k}(x_k) = \bigcup_{i=1}^{N_{\tilde{X}_k}} \bigcup_{j=1}^{N_{\tilde{F}^l_k}} \sqcup_{x_k \in X_k} \left(\left[\underline{\mu}^{i}_{\tilde{X}_k}(x_k), \overline{\mu}^{i}_{\tilde{X}_k}(x_k)\right] \sqcap \left[\underline{\mu}^{j}_{\tilde{F}^l_k}(x_k), \overline{\mu}^{j}_{\tilde{F}^l_k}(x_k)\right]\right) = \\ = \bigcup_{i=1}^{N_{\tilde{A}^l_k}} \left[\underline{\mu}^{i}_{\tilde{A}^l_k}(x_k), \overline{\mu}^{i}_{\tilde{A}^l_k}(x_k)\right] \tag{5.4.7}$$

Back to the example depicted in Figure 5.4.6, the subintervals arising from the intersection between $\mu^{i}_{\tilde{X}_k}(x_k)$ and $\mu^{j}_{\tilde{F}^l_k}(x_k)$, $i = 1, ..., N_{\tilde{X}_k}$, $j = 1, ..., N_{\tilde{F}^l_k}$, will be denoted as:

$$I^{ij}_{\tilde{A}} = \sqcup_{x_k \in X_k} \left(\left[\underline{\mu}^{i}_{\tilde{X}_k}(x_k), \overline{\mu}^{i}_{\tilde{X}_k}(x_k)\right] \sqcap \left[\underline{\mu}^{j}_{\tilde{F}^l_k}(x_k), \overline{\mu}^{j}_{\tilde{F}^l_k}(x_k)\right]\right) = \left[l^{ij}_{\tilde{A}}, r^{ij}_{\tilde{A}}\right] \tag{5.4.8}$$

Hence, from Figure 5.4.6 it can be easily deduced that $N_{\tilde{X}_k} = N_{\tilde{F}^l_k} = 2 \, \forall x_k \in X_k$, and Equation (5.4.6) can be written for that specific example as follows:

$$\mu_{\tilde{A}}(x_k) = \bigcup_{i=1}^{2} \bigcup_{j=1}^{2} \sqcup_{x_k \in X_k} \left(\left[\underline{\mu}^{i}_{\tilde{X}_k}(x_k), \overline{\mu}^{i}_{\tilde{X}_k}(x_k)\right] \sqcap \left[\underline{\mu}^{j}_{\tilde{F}^l_k}(x_k), \overline{\mu}^{j}_{\tilde{F}^l_k}(x_k)\right]\right) = \\ = \bigcup_{i=1}^{2} \bigcup_{j=1}^{2} I^{ij}_{\tilde{A}} \tag{5.4.9}$$

Those four intersections have been highlighted with red dots in Figure 5.4.6, and each $I^{ij}_{\tilde{A}}$ is properly indicated within the same figure.

Let us explain these computations via a worked example. Set $\tilde{X}_1$ is described by the UMF and LMF of each subinterval comprising it, and are gathered in Equation (5.4.10), whereas set $\tilde{F}_1^l$ is described analogously in Equation (5.4.11). For simplicity, we write x instead of x_1 in the subsequent equations.

$$\mu_{\tilde{X}_1}(x) = \bigcup_{i=1}^{2} \left[\underline{\mu}^i_{\tilde{X}_1}, \overline{\mu}^i_{\tilde{X}_1}\right]$$
$$\underline{\mu}^1_{\tilde{X}_1}(x) = \exp\left(-\frac{(x-2)^2}{2 \cdot 0.5}\right); \qquad \overline{\mu}^1_{\tilde{X}_1}(x) = \exp\left(-\frac{(x-2)^2}{2 \cdot 0.65}\right)$$
$$\underline{\mu}^2_{\tilde{X}_1}(x) = \exp\left(-\frac{(x-2)^2}{2 \cdot 1.35}\right); \qquad \overline{\mu}^2_{\tilde{X}_1}(x) = \exp\left(-\frac{(x-2)^2}{2 \cdot 1.6}\right) \tag{5.4.10}$$

$$\mu_{\tilde{F}_1^l}(x) = \bigcup_{j=1}^{2} \left[\underline{\mu}^j_{\tilde{F}_1^l}, \overline{\mu}^j_{\tilde{F}_1^l}\right]$$
$$\underline{\mu}^1_{\tilde{F}_1^l}(x) = \exp\left(-\frac{(x-5)^2}{2 \cdot 0.5}\right); \qquad \overline{\mu}^1_{\tilde{F}_1^l}(x) = \exp\left(-\frac{(x-5)^2}{2 \cdot 0.9}\right)$$
$$\underline{\mu}^2_{\tilde{F}_1^l}(x) = \exp\left(-\frac{(x-5)^2}{2 \cdot 1.4}\right); \qquad \overline{\mu}^2_{\tilde{F}_1^l}(x) = \exp\left(-\frac{(x-5)^2}{2 \cdot 1.9}\right) \tag{5.4.11}$$

Hence, the intersection intervals will be:

$$I^{11}_{\tilde{A}} = \sqcup_{x \in X}\left(\left[\underline{\mu}^1_{\tilde{X}_1}(x), \overline{\mu}^1_{\tilde{X}_1}(x)\right] \sqcap \left[\underline{\mu}^1_{\tilde{F}_1^l}(x), \overline{\mu}^1_{\tilde{F}_1^l}(x)\right]\right) = [0.1054, 0.2320]$$
$$I^{12}_{\tilde{A}} = \sqcup_{x \in X}\left(\left[\underline{\mu}^1_{\tilde{X}_1}(x), \overline{\mu}^1_{\tilde{X}_1}(x)\right] \sqcap \left[\underline{\mu}^2_{\tilde{F}_1^l}(x), \overline{\mu}^2_{\tilde{F}_1^l}(x)\right]\right) = [0.2838, 0.3895]$$
$$I^{21}_{\tilde{A}} = \sqcup_{x \in X}\left(\left[\underline{\mu}^2_{\tilde{X}_1}(x), \overline{\mu}^2_{\tilde{X}_1}(x)\right] \sqcap \left[\underline{\mu}^1_{\tilde{F}_1^l}(x), \overline{\mu}^1_{\tilde{F}_1^l}(x)\right]\right) = [0.2758, 0.3992]$$
$$I^{22}_{\tilde{A}} = \sqcup_{x \in X}\left(\left[\underline{\mu}^2_{\tilde{X}_1}(x), \overline{\mu}^2_{\tilde{X}_1}(x)\right] \sqcap \left[\underline{\mu}^2_{\tilde{F}_1^l}(x), \overline{\mu}^2_{\tilde{F}_1^l}(x)\right]\right) = [0.4412, 0.5252] \tag{5.4.12}$$

These are the Y-coordinate values of the points marked in red in Figure 5.4.6. Hence, the final result, according to Equation (5.4.9), will be:

$$\mu_{\tilde{A}}(x) = \bigcup_{i=1}^{2}\bigcup_{j=1}^{2} I^{ij}_{\tilde{A}} =$$
$$= [0.1054, 0.2320] \cup [0.2838, 0.3895] \cup [0.2758, 0.3992] \cup [0.4412, 0.5252]$$

(5.4.13)

It is important to highlight that intervals $I^{12}_{\tilde{A}}$ and $I^{21}_{\tilde{A}}$ do overlap, as $[0.2838, 0.3895] \cup [0.2758, 0.3992] = [0.2758, 0.3992]$; hence, the final result will be:

$$\mu_{\tilde{A}}(x) = [0.1054, 0.2320] \cup [0.2758, 0.3992] \cup [0.4412, 0.5252] = \\ = \bigcup_{i=1}^{N_{\tilde{A}}} \left[\underline{\mu}^{i}_{\tilde{A}}(x), \overline{\mu}^{i}_{\tilde{A}}(x)\right] \quad (5.4.14)$$

Finally, we plot the antecedent activation degree described in Equation (5.4.14) in Figure 5.4.7.

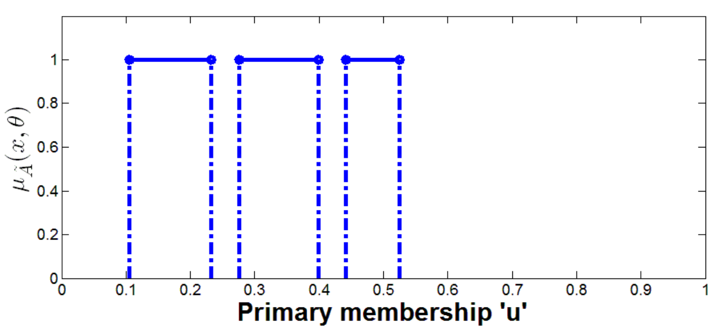

Figure 5.4.7
Antecedent activation degree for the example in Figure 5.4.6 as described in Equation (5.4.14).

As stated in Equation (5.4.4), for a given rule l, each antecedent activation has to be obtained, each of them leading to a T1FS comprised of several closed, connected and disjointed intervals, similar to the one depicted in Figure 5.4.7. To obtain the firing strength for that rule, the meet operation between these antecedents activation degrees has to be performed according to Equation (5.4.4).

$$F^{l}(\vec{x}) = \left[\prod_{i=1}^{p} \left(\sqcup_{x_i \in X_i} \mu_{\tilde{X}_i}(x_i) \sqcap \mu_{\tilde{F}^{l}_{i}}(x_i)\right)\right] = \prod_{i=1}^{p} \left[\mu_{\tilde{A}^{l}_{i}}(x_i)\right] \quad (5.4.15)$$

Hence, the final rule firing strength $F^{l}(\vec{x})$ will be a T1FS, according to Theorem 5.3.1. The number of disjointed intervals within $F^{l}(\vec{x})$ will depend on the number of inputs of the system, the number of disjointed intervals each antecedent has and whether these intervals overlap.

5.4.4 *Type reduction*

In this dissertation a new type-reduction procedure is proposed, named *modified centre-of-sets type reducer*, which can be applied to any gfIT2FLS.

5.4.4.1 *Preliminaries: approximating fuzzy quantities*

Defuzzification is a procedure used to choose a crisp number as representative of the whole fuzzy set, and is provided as an output of the FLS. However, it could be argued that this process causes a significant loss of information. Hence, some authors defend that some less radical solutions should be adopted and have proposed other approaches, such as interval, triangular or trapezoidal approximations. This topic has attracted significant attention from many researchers [6], [36], [64] relating fuzzy numbers (which are normal and convex type-1 fuzzy sets). This approximation is done by defining a *distance* between a given fuzzy number and an interval, triangle or trapezoid, and then *finding* the interval, triangle or trapezoid minimising that distance. In [2], a similar procedure is proposed for not necessarily convex and normal T1FSs (which are called "fuzzy quantities"), where the authors proved that, for fuzzy quantities, each α-cut is a finite set of closed, connected and disjointed intervals with height equal to α, and each set of intervals is approximated by another interval minimising a given distance. Here, we will use the same approximation of the union of disjointed intervals and apply it to the type-reduction operation.

5.4.4.2 *Approximating a union of disjointed intervals*

Let N be a positive integer, and let A be the union of N disjointed intervals, each interval named A_i and characterised by its two endpoints, $A_i = \left[a_L^i, a_R^i\right]$, such that $a_L^i \leqslant a_R^i$, $i = 1, ..., N$. It is important to note that if $a_L^i = a_R^i$, that interval reduces to a singleton. Hence, A could be expressed as:

$$A = \bigcup_{i=1}^{N} A_i = \bigcup_{i=1}^{N} \left[a_L^i, a_R^i\right] \tag{5.4.16}$$

In order to obtain the interval $C = [c_L, c_R]$ that best approximates the set A, we first have to define a *distance* between two closed intervals $B = [b_L, b_R]$ and $D = [d_L, d_R]$. If we denote $\mathrm{mid}(b) = (b_L + b_R)/2$ the midpoint of an interval, and $\mathrm{spr}(B) = (b_R - b_L)$ its spread, then the distance between two intervals B and C (introduced in [94]) could be defined as follows according to [94]:

$$d_{\overline{\theta}}\left(B, D\right) = \sqrt{\left(mid(B) - mid(D)\right)^2 + \overline{\theta}\left(spr(B) - spr(D)\right)^2} \tag{5.4.17}$$

Where $\theta \in (0, 1]$ is a parameter to weigh the relative importance of the spreads against the midpoints. Now let us consider a set of N weights $p = \{p_i\}_{i=1,\ldots,N}$, such that $p_i > 0$ and $\sum_{i=1}^{N} p_i = 1$, which allow us to weigh the different intervals comprising the set A. Hence, given the distance between two intervals as in Equation (5.4.17), the set of weights p and the parameter $\overline{\theta}$, we could define the *distance function* between a given set of intervals A and a given closed interval C as [2]:

$$J\left(C; A, p, \overline{\theta}\right) = \sum_{i=1}^{N} d^2_{\overline{\theta}}\left(C, A_i\right) p_i \tag{5.4.18}$$

Thus, the interval $C^* = \left[c^*_L, c^*_R\right]$ that best approximates the union of a set of disjointed intervals A is defined as the one that minimises the function in Equation (5.4.18). In the same work [2], it is proven that the interval C^* is given by:

$$C^*(A) = [c^*_L, c^*_R] = \left[\sum_{i=1}^{N} a^i_L p_i, \sum_{i=1}^{N} a^i_R p_i\right] \tag{5.4.19}$$

It is important to note that the result given in Equation (5.4.19) is independent of the parameter $\overline{\theta}$. In this work we will consider all the weights p_i to be equals; hence, $p_i = 1/N$ and Equation (5.4.19) becomes Equation (5.4.20).

$$C^*(A) = [c^*_L, c^*_R] = \left[\frac{1}{N}\sum_{i=1}^{N} a^i_L, \frac{1}{N}\sum_{i=1}^{N} a^i_R\right] \tag{5.4.20}$$

This closed interval C^* is the one that best approximates the set A, which is the union of a finite number N of closed, connected and disjointed intervals.

5.4.4.3 *Approximating a gfIT2FS by an IVFS*

Once a method to approximate the union of a finite number of closed, connected and disjointed intervals is defined, we propose a new type-reduction method based on it. It is important to note that all different kinds of gfIT2FSs presented in [89] and Section 4.5 can be considered as a particular case of D or E, which are the most general: a combination of intervals and singletons (which can, in fact, be considered intervals with the same extreme points). Hence, we will assume that our FLS has gfIT2FSs as consequents, and T1FSs comprised of several subintervals (as shown in Equation (5.4.7)) as the rule firing strengths. Thus, the proposed type-reduction is as follows:

1. For each consequent set $\tilde{Y}^l$, which is a gfIT2FS, perform the approximation described in Section 5.4.4.2 for *every single vertical slice*. This operation will transform the gfIT2FS into an IVFS where every crisp value y will have an interval approximating the primary membership value. We will denote the resulting set as $\tilde{Y}^l_{approx}$.

2. Once we have $\tilde{Y}^l_{approx}$, we can apply the well-known KM algorithm [48] to obtain the centroid of $\tilde{Y}^l_{approx}$, which we will denote as $C\left(\tilde{Y}^l_{approx}\right) = \left[y^L_{\tilde{Y}^l}, y^R_{\tilde{Y}^l}\right]$.

3. For each rule firing strength $\tilde{F}^l$, obtain the interval that best approximates it, which we will denote $C_{\tilde{F}^l} = \left[c^L_{\tilde{F}^l}, c^R_{\tilde{F}^l}\right]$.

4. Once we have an interval per rule firing strength and per rule consequent, the *centre-of-sets* type-reduction for IVFLSs [48] is applied.

The result will be an interval T1FS, which we will denote by $Y = [y_L, y_R]$. This output from the type-reduction block will be the input of the defuzzification block.

5.4.5 *Defuzzification*

As in other works dealing with IVFLSs [58], [78], [74], the output of the type-reduction block is a T1FS comprised of a single interval; thus, the most used defuzzification method consists in taking the midpoint of the interval as the output of the system:

$$y = \frac{y_L + y_R}{2} \tag{5.4.21}$$

5.5 EXPERIMENTAL ANALYSIS AND WORKED EXAMPLES FOR THE OPERATION OF THE GFIT2FLSS

In this Section, we will present two different methods that can be applied in real world applications to obtain gfIT2FSs. As an example, across this section we will consider the following *illustrative scenario*: a mobile robot controlled by a FLS. The inputs will be the distances measured by two sonar sensors: one placed at the front, the other at the rear; the outputs will be the speed to be applied to both wheels, so the robot can navigate and avoid obstacles. This speed is specified to the robot as an integer number between 0 and 400. The first method will be used to model sensor input noise coming from different sources, and the resulting sets can be used as non-singleton inputs; the second method will be used to obtain gfIT2FSs from the survey method, using knowledge from different expert, to model both antecedents and consequents of the FLS.

5.5.1 *Modelling sensor input noise from different sources: non-singleton fuzzification*

The main difference between the singleton and non-singleton fuzzification is the process of obtaining a fuzzy set from an input value (which is usually a single number) in order to represent the uncertainty associated with it. When using linguistic labels, the uncertainty is associated with imprecision of human language and perception while when using numerical values or signals, the uncertainty is usually related to input noise or input uncertainties coming from sensors. In this work, we will associate each input value with a multi-interval gfIT2FS, which will be modelled from data gathered from the sensors.

The proposed method is similar to the one presented in [86], but in our case we will be eliminating some of the restrictions imposed in that work, and the resulting set will not be an IVFS, but a multi-interval gfIT2FS. This method will consist of gathering the data provided by the sonar sensor under different uncertainty sources. For each of these conditions we will obtain an IVFS; hence, when overlapping all IVFSs, we will obtain a multi-interval gfIT2FS, which will represent the input when a certain combination of conditions is present during the sensor measurement process. We will apply the algorithm presented in Section 5.4.1 to obtain such sets, which could be used as non-singleton inputs to a gfIT2FLS.

Three different conditions will be considered in our example to separately generate uncertainty in the sonar sensor measurement. These conditions will be heat (provided by a hand held electric heat gun), wind (provided by two fans) and sound noise (generated from a hoover being moved around the sensor).

The sonar sensor assumes the speed of sound to be a constant; however, it is well known that temperature is a variable affecting the speed of sound; more precisely, a widely used approximation to express that dependency is represented in Equation (5.5.1):

$$v = 20.05 \cdot \sqrt{T + \frac{e}{p}} \tag{5.5.1}$$

Where v is the speed of sound, T is the absolute temperature in Kelvin, e is the partial pressure of water vapour, and p is the barometric pressure [86].

Secondly, wind is another factor affecting the propagation of the sonar ping signal, changing the physical media the sound wave uses to propagate which produce a refraction phenomenon and thus changes the direction of the sound wave. This prevents the sonar signal from following a completely straight trajectory, thus measuring a different distance value or even not receiving a ping signal back [86].

Last but not least, sonic noise is also to be considered as a source of disturbance for the sonar sensor. Relatively high frequency sound noise sources are more probable to create ultrasonic waves, which may in fact interfere with the ones emitted by the sonar sensor and affect the obtained measurement [55].

In the following section we will explain how data was collected to model such uncertainty and how gfIT2FSs can be inferred from these data to model faced uncertainties.

5.5.1.1 *Experimental setup*

The experimental setup to model multi-interval gfIT2FSs is as follows: a mobile robot having one front sonar sensor is used to measure the distance to an obstacle; such obstacle will consist of two joint wooden boxes offering a flat surface. The sonar sensor measures the distance in millimetres, thus providing an unsigned integer from 50 mm to 5000 mm (5 metres).

To precisely obtain the positions of both the sonar sensor and the obstacle, and hence the *real* distance between them, the VICON Tracking System (VTS) will be used. This system consists of a set of eight high-resolution cameras which are able to locate a set of markers (small reflective spheres) fixed all around both the robot and the obstacle. By detecting those markers, the VTS can create a 3D model of the robot and the obstacle, and measure distances between them with millimetre accuracy. An example of these 3D models created using several markers is depicted in Figure 5.5.1.

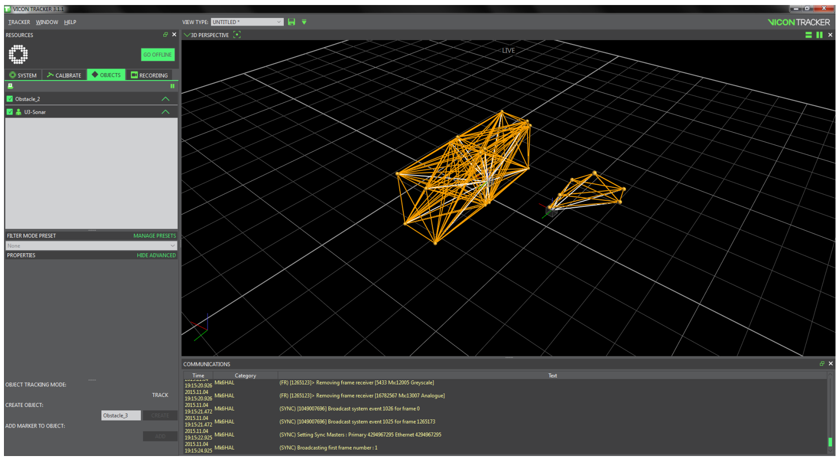

Figure 5.5.1
A screenshot of the VICON Tracking System, showing the 3D models for the robot and the obstacle. The 3D model to the left represents the obstacle, whereas the one to the right represents the robot.

The robot and the sonar sensor are then roughly placed at a given set of distances: using measuring tape, they are placed at the following approximated distances: 500 mm, 1000 mm, 1500 mm, 2000 mm, 2500 mm, 3000 mm, 3500 mm, 4000 mm and 4500 mm. For those distances, the VTS provided the following exact distances from the sonar sensor to the obstacle: 504 mm, 1009 mm, 1511 mm, 2013 mm, 2514 mm, 3014 mm, 3517 mm, 4014 mm and 4516 mm, respectively.

For each of those distances, the three disturbances mentioned in the previous subsection are introduced separately: the heat is provided by a heat gun properly attached to the robot (Figure 5.5.2(Left)); the wind was created by two fans along the path of the sonar ping (Figure 5.5.2(Right)), whereas the noise was created using a hoover and moving it around the robot.

For *each distance value* and *each of the conditions mentioned above to introduce noise and uncertainty*, the algorithm introduced in Section 5.4.1 was used to obtain each histogram from 10000 measurements.

5.5.1.2 *Results: obtained sets and interpolation*

Figure 5.5.3 shows an example of sets obtained for 2514mm. Let us focus on the four sets depicted in Figure 5.5.3. Figures (a)-(c)

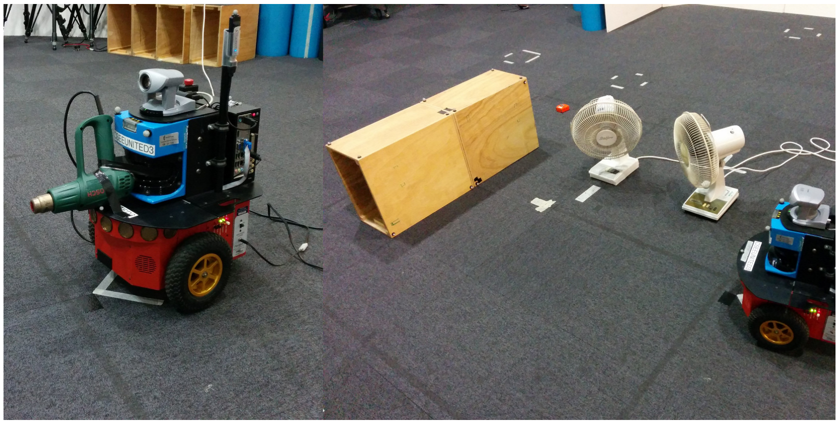

Figure 5.5.2
(Left) A heat gun attached to the robot to provide heat through the ping signal path. (Right) Two fans create wind which will affect the sonar measurements.

are the classic representation of an IVFS; however, the set represented in Figure (d) can be trickier as it shows a gfIT2FS. It can be seen that three different shades of grey (not fifty) have been used, but this is only to emphasize that the set was created by overlapping three different IVFSs. The three overlapped FOUs combined create one gfIT2FS; more precisely, one belonging to Case D as explained in Section 4.5.3 and in [89]. That means that, for every value within the domain (0-5000 mm, in our case), every primary membership is a multi-interval set; i.e., their secondary grades are T1FSs consisting of one or several subintervals in [0, 1] (remember a singleton can be represented as an interval having the same left and right end points).

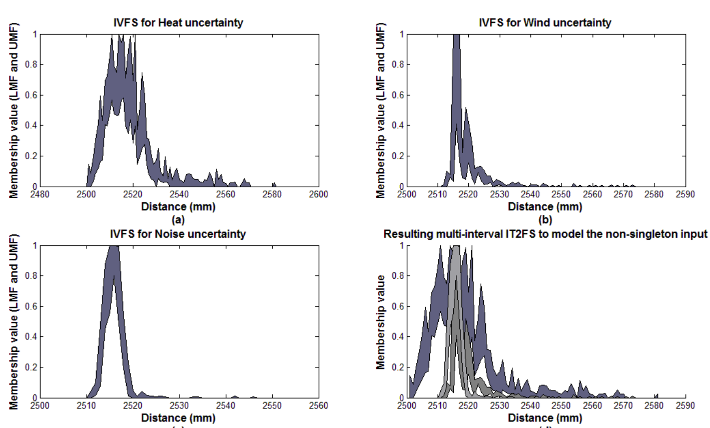

Figure 5.5.3
Example of a multi-interval gfIT2FS by overlapping three IVFSs. (a) IVFS obtained from heat. (b) IVFS obtained from wind. (c) IVFS obtained from noise. (d) Multi-Interval gfIT2FS modelling all uncertainties.

As an example, we will plot several secondary grades from the set depicted in Figure 5.5.3(d). As it can be seen in that figure, the values below 2510 mm (as shown for the case of measurement of at 2505 mm, which is plotted in Figure 5.5.4(a)) just have one single interval as primary membership, as in that domain's region only the uncertainty related to temperature is present; hence, its secondary grade will be a single interval. The second example will be some value in the domain between 2520 and 2525 mm (as shown for the measurement of 2524 mm depicted in Figure 5.5.4(b)); the vertical slices placed at these points have the uncertainty related to the three sources (heat, wind and noise) in such a way that the sets do not overlap; thus, the vertical slice will be comprised of three different disjointed intervals. A third case is shown for measurements between 2515 and 2520 as shown for the measurement of 2518 mm depicted in Figure 5.5.4(c)); for those points, all three IVFSs modelling the three uncertainties perfectly overlap, having no gaps between them, thus creating again one single interval as its secondary grade.

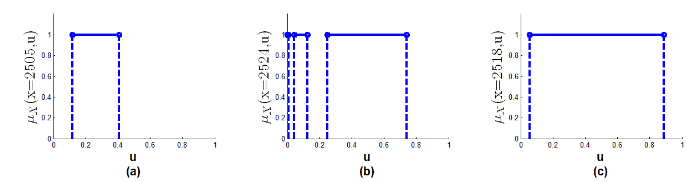

Figure 5.5.4
Several examples of vertical slices within the set in Figure 5.5.3(d) for: (a) 2505 mm. (b) 2524 mm. (c) 2518mm.

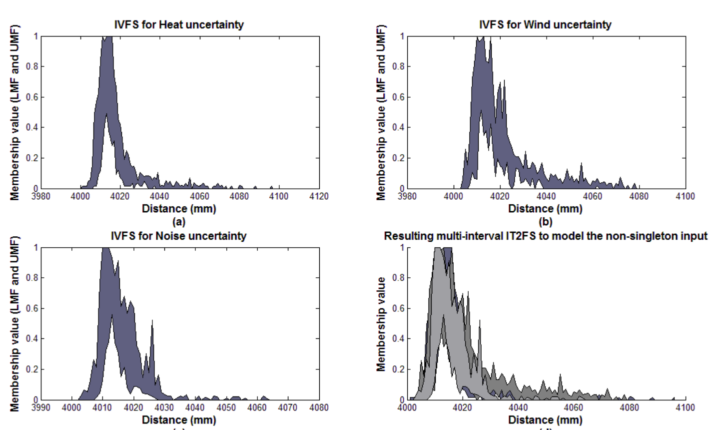

Figure 5.5.5
Multi-interval IT2FS obtained for 4014 mm. Observe how all sets in (a), (b) and (c) naturally merge in (d), forming almost an IVFS.

However, overlapping can happen not only for a given set of values, but also along all the fuzzy set. This fact is more pronounced in the set obtained for 4014 mm, which is depicted in Figure 5.5.5. It can be seen from Figure 5.5.5(d) that nearly every value in the X-domain has a secondary membership of a single interval as the three IVFSs overlap almost perfectly. This fact shows the flexibility of the proposed non-singleton fuzzification method: even when a multi-interval scheme is supposed to be obtained, a classic IVFS can be obtained if the uncertainty to be modelled requires it.

To properly model the sensor behaviour within all its possible measuring range, it would be required to perform the algorithm described in the previous section for every single possible value the sensor can measure. However, this would require an enormous amount of work, and besides, most of the sets would be somehow redundant, as it is reasonable to expect the sensor would behave in a very similar way when measuring 3122 and 3123 mm. For this reason, it is reasonable to obtain measurements just for distinct values (in our case: 504 mm, 1009 mm, 1511 mm, 2013 mm, 2514 mm, 3014 mm, 3517 mm, 4014 mm and 4516 mm); the multi-interval gfIT2FSs for the rest of values will be obtained by interpolation using the two closest sets.

The equation to obtain an interpolated set for a given measurement d, from the two closest measurements neighbours placed at d_1 (left) and d_2 (right) (thus verifying $d_1 \leqslant d \leqslant d_2$) is as follows:

$$\mu_d(i) = \frac{d_2 - d}{d_2 - d_1}\mu_{d_1}(i - (d - d_1)) + \frac{d - d_1}{d_2 - d_1}\mu_{d_2}(i + (d_2 - d)) \tag{5.5.2}$$

Where $\mu_d(i)$ is the interpolated membership value at sample i and $\mu_{d_1}(i)$, $\mu_{d_2}(i)$ are the membership functions of the left and right neighbours at sample i, respectively. It is important to note that $\mu_{d_1}(i)$ and $\mu_{d_2}(i)$ are multi-interval gfIT2FSs; $\mu_{d_1}(i - (d - d_1))$ is a version of $\mu_{d_1}(i)$ shifted $(d - d_1)$ to the right; and $\mu_{d_2}(i + (d_2 - d))$ is a version of $\mu_{d_2}(i)$ shifted $(d_2 - d)$ to the left. It is worthwhile mentioning that, when $d = d_1$, Equation (5.5.2) reduces to $\mu_d(i) = \mu_{d_1}(i)$, and when $d = d_2$, then it reduces to $\mu_d(i) = \mu_{d_2}(i)$. An example of such interpolation process is depicted in Figure 5.5.6.

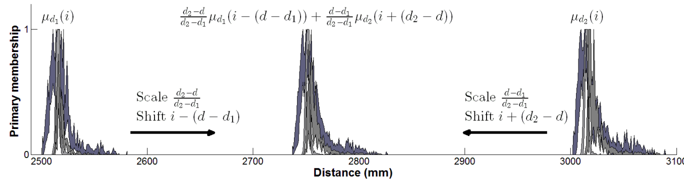

Figure 5.5.6
An interpolation example to obtain the multi-interval IT2FS for 2750 mm from 2514 mm and 3014 mm.

5.5.2 *Obtaining antecedents and consequents using the survey method*

In this section we present a method to infer gfIT2FSs from data, to model both the antecedents and the consequents of the FLS that could control the autonomous mobile robot. Table 5.5.1 shows a simple rule base to control the robot obstacle avoidance behaviour.

Table 5.5.1
Rule base used to control the right wheel speed.

Front sonar distance	Back sonar distance		
	CLOSE	**MEDIUM**	**FAR**
CLOSE	Low	Low	Low
MEDIUM	Low	Fair	High
FAR	High	High	High

Mendel introduced in [69] the term *type-2 fuzzistics* to refer to the process of collecting data from a group of subjects and then mapping that data into an FOU. We will use fuzzistics to obtain gfIT2FSs from a group of subjects. In [69], two different approaches are described to perform the fuzzistics process: the *person MF approach*, and the *interval end-points approach*. The former requires the surveyed people to know about fuzzy logic, as each of them provides a FOU for the given word/set to be modelled. We will follow a similar approach, where we have asked a group of twelve Msc students (who studied fuzzy logic) to draw a T1 MF for each of the previously stated labels. Figure 5.5.7 depicts all the sets proposed by the students for the labels in Table 5.5.1.

5.5.2.1 *Example 1: obtain multi-singleton gfIT2FSs*

The first option that naturally arises from the data collected is to use them with almost no further processing. As it was explained

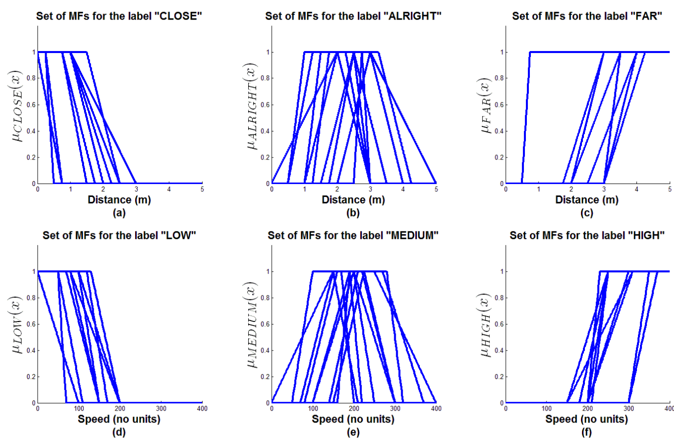

Figure 5.5.7
All T1FSs obtained by survey for the labels close, medium and far for distance, and low, fair and high for wheel speed.

in Section 4.5.3, a multi-singleton gfIT2FS is one of the specific versions of gfIT2FSs where the membership functions involved in the rule set described in Table 5.5.1 will look exactly like Figure 5.5.7.

5.5.2.2 *Example 2: obtain IVFSs*

This second proposal aims to obtain regular IVFSs antecedents from a group of T1FSs. This process can be done just by aggregating all the sets and define the UMF and LMF of the resulting set as in Equation (5.5.3) [62]:

$$\begin{aligned} \underline{\mu}_{LABEL}(x) &= \arg\min_{x \in X} \{\mu^i_{LABEL}(x) \mid i = 1, \ldots, N_{st}\} \\ \overline{\mu}_{LABEL}(x) &= \arg\max_{x \in X} \{\mu^i_{LABEL}(x) \mid i = 1, \ldots, N_{st}\} \end{aligned} \tag{5.5.3}$$

Where N_{st} is the number of students surveyed. This results in the MFs in Figure 5.5.8. This process consisting in inferring IVFSs from data is better known as *fuzzistics* and has been previously tackled in the literature, as in [62], [78], [76], [69] and [77].

5.5.2.3 *Example 3: obtain multi-interval gfIT2FSs*

In this section we will provide a brief example on how to obtain a multi-interval gfIT2FS from T1 MFs, based on the notion of uncertainty and its relation with the FOU; however, the example presented here just has illustrative purposes. Formally defining

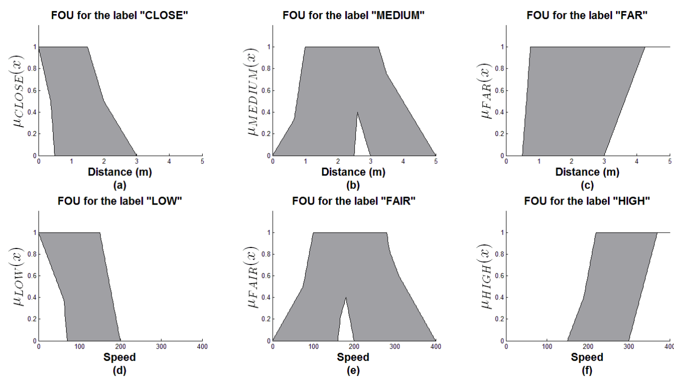

Figure 5.5.8
Antecedent IVFSs inferred from T1 MFs.

a complete algorithm to obtain multi-interval gfIT2FSs from T1 MFs is out of the scope of this dissertation.

When introducing the concepts of type-2 fuzzy sets, some authors [78] state that an IVFS can be obtained by *blurring* a T1 MF, hence obtaining an FOU. Hence, it seems intuitive to think that a greater FOU represents greater uncertainty associated to the concept represented by the set, as the T1 MF has been *blurred* a greater quantity.

When inferring an IVFS from several T1 MFs by taking their maximum and their minimum as UMF and LMF, respectively (as in the previous section and as represented in Figure 5.5.8), if the T1 MFs are sparsely distributed across the domain, a great area will be covered when obtaining the FOU's boundaries, and a lot of uncertainty will be unnecessarily and artificially introduced in the set. A very good example of this situation can be seen comparing Figures 5.5.7(c) and 5.5.8(c). This drawback can be naturally solved using multi-interval gfIT2FSs as MFs: the system designer can detect and decide which T1 MFs are close enough to be gathered in a subinterval, and which are separated enough to introduce a gap in the primary membership. Moreover, if a T1 MF is isolated and has no other T1MFs around to be gathered with, it can remain as a singleton in the inferred gfIT2FS. Examples of these situations are depicted in Figure 5.5.9, where (a), (b) and (c) depict the T1 MFs obtained from the survey for the labels *CLOSE*, *FAR* and *HIGH*; and (d), (e) and (f) show an example of inferred gfIT2FSs, where gaps are present in the primary membership. It should be noted that the sets in Figure 5.5.9 (d) and (f) are comprised by two subintervals, whereas the set in (e) has one interval and one singleton.

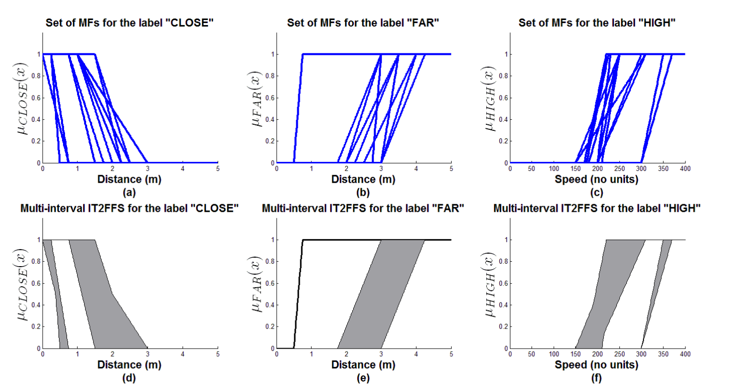

Figure 5.5.9
Sets of T1MFs for the labels *CLOSE*, *FAR* and *HIGH*, and their corresponding inferred multi-interval IT2FSs.

5.5.3 *A Numerical Example summarising the operation of the gfIT2FLSs*

In this section, we will provide an example to summarise the steps involved in a gfIT2FLS. We consider the illustrative example introduced in this section of a 2-input-1-output system controlling an autonomous mobile robot with obstacle avoiding behaviour. The inputs to the system are the noisy measurements (with uncertainty arising from temperature, wind and sound noise) coming from two sonar sensors (one placed at the front and another at the back), and the output is the wheel speed. We assume the sensors behaviour have been modelled as described in Section 5.5.1 ahead of time, and thus the gfIT2FSs for some given input measurements are available. Hence, the whole gfIT2FLS would work as follows:

1. The fuzzifier block receives the input vector $\vec{x} = (x_1, x_2)$, which are the noisy measures coming from the sonar sensors. Assume $x_1 = 580$ mm, which is a singleton as in Figure 5.5.10(a). The fuzzifier block will map this crisp input into a multi-singleton gfIT2FS, obtained from the interpolation of the two closest sets modelled ahead of time. In this example, the sets to perform the interpolation are placed at $d_1 = 504$ mm and $d_2 = 1009$ mm. The resulting set is obtained applying Equation (5.5.2), considering in that Equation d is the input, x_1. Thus, the resulting set would be as in Equation 5.5.4 and as in Figure 5.5.10(b).

$$\mu_{580mm}(i) = 0.8495\dot{\mu}_{504mm}(i-76) + 0.1505 \cdot \mu_{1009mm}(i+429) \tag{5.5.4}$$

2. For each rule within the system:
   A. The antecedent activation degree is obtained, for each input, as in Equation (5.4.7), by meeting the antecedent fuzzy set with the input gfIT2FS from Step 1. An illustrative antecedent is plotted in Figure 5.5.10(c), whereas the points in which the fuzzy input set meets the antecedent are depicted in Figure 5.5.10(d). The final antecedent activation is plotted in Figure 5.5.10(e).
   B. The rule firing strength is obtained, as in Equation (5.4.15), by meeting the antecedent activation degrees from both inputs. This rule firing strength will be a T1FS consisting of several subintervals, as in Figure 5.4.7 and similar to Figure 5.5.10(e).

3. For each consequent of each rule (an illustrative example is plotted in Figure 5.5.10(f), with two intervals and one singleton), which is a gfIT2FS, we apply the following steps *ahead of time*:
   A. The approximation method described in Section 5.4.4.3 is applied at each point within the output domain, in order to obtain the IVFS that best approximates the consequent. The resulting IVFS approximating the consequent in Figure 5.5.10(f) is depicted in Figure 5.5.11(a).
   B. The centroid operation is applied to the resulting IVFS, to obtain as the consequent a single interval defined in the output domain, as described also in Section 5.4.4.3. The resulting interval is shown in Figure 5.5.11(b).

4. The modified COS Type Reduction is applied as in Section 5.4.4, which consist of the following steps:
   A. For each rule firing strength, which is a T1FS comprised of several subintervals, the approximation method described in Section 5.4.4.2 is applied, in order to obtain a single interval. For instance, the interval that best approximates the set in Figure 5.5.10(e) is depicted in Figure 5.5.11(c).
   B. Once we have a single interval per rule firing strength (as in Figure 5.5.11(c)) and per rule consequent (as in Figure 5.5.11(b)), the classical COS type reduction is applied, as explained in Section 5.4.4.3.
   C. The result is a single interval, which is the output of the type reduction block, defined in the output domain.

5. The midpoint of the interval is obtained as in Equation (5.4.21) and provided as the output of the system.

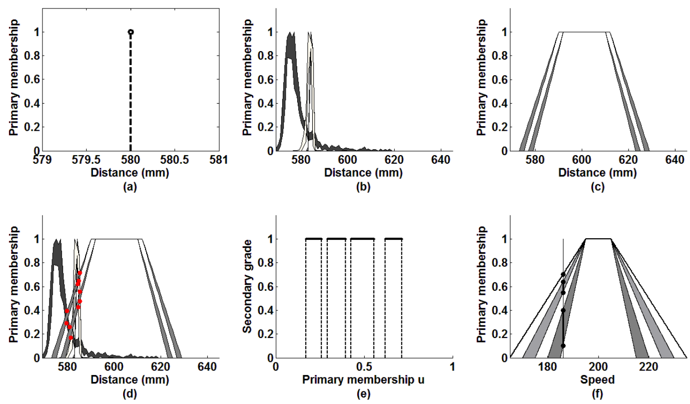

Figure 5.5.10
(a) Input singleton $x_1 = 580$ mm. (b) Non-singleton input for $x_1 = 580$ mm obtained by interpolation. (c) Antecedent for input 1 in a given rulen. (d) Meet on the non-singleton input and the antecedent. (e) Antecedent activation degree. (f) Consequent of a rule.

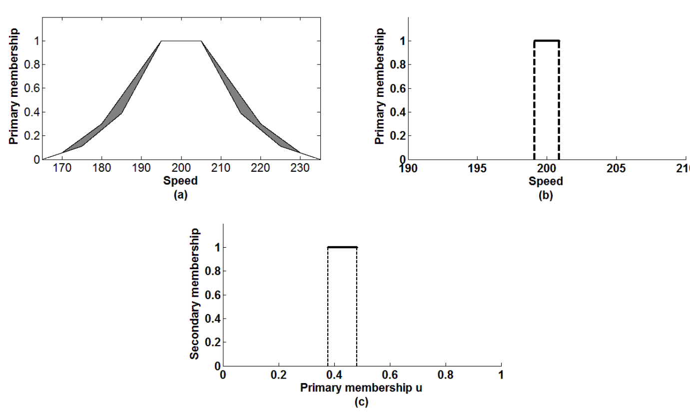

Figure 5.5.11
(a) IVFS approximating the consequent. (b) Centroid of the IVFS in (a) approximating the consequent in Figure 5.5.10(f). (c) Interval approximating the rule firing strength.

5.6 CONCLUSIONS AND FUTURE WORK

In this Chapter, the general forms of Interval-Type 2 Fuzzy Logic Systems (gfIT2FLSs) have been introduced, which use IT2FSs which are more general than IVFSs, and can actually have non convex secondary membership functions. These sets, which were introduced in [89], allow us to represent uncertainty in forms which cannot be represented by IVFSs. Hence, there was a need to develop the theoretical framework of such systems. All blocks within the FLS have been tackled across this chapter, including fuzzification, inference and type-reduction. The most general case of non-singleton fuzzification has been considered, and equations for the inference engine have been provided; in addition, a new method for type-reduction operation, which has been called *modified centre-of-sets type reducer*, has been proposed. Moreover, examples to use gfIT2FSs in real world applications have been presented, for both non-singleton inputs (using the histogram method) and antecedents modelling (using type-2 fuzzistics). Finally, a complete step by step example of how a gfIT2FLS operates was provided.

For our current and future work, we intend to apply the gfIT2FLSs for several real world applications. Besides, it is worthwhile to explore the relation between the gfIT2FSs/FLSs and non-stationary FSs/FLSs, formally introduced in [33]. These sets are T1FSs whose membership function is time dependant, aiming to model how uncertainty changes over time, which might be perceived as a way of *blurring* the T1MF. This notion motivated the authors to provide an initial insight into the relationship between non-stationary FSs and IVFSs; hence, as the gfIT2FSs are a natural generalisation of IVFSs, it seems reasonable to also explore the relationship between the gfIT2FSs/FLSs and non-stationary FSs/FLSs.

Part IV

CONCLUSIONS

6 CONCLUSIONS, MAIN CONTRIBUTIONS AND LIST OF PUBLICATIONS

What is useful in fuzzy logic theory is not particularly new and what is new is not particularly useful.

— Susan Haack

The work presented in this dissertation represents a contribution to the fields of fuzzy logic theory and type-2 fuzzy logic systems. The main contributions and future work were introduced at the end of Chapters 3, 4 and 5. These conclusions are summarised again here, linking them with the initial motivations and goals presented in Chapter 1. Moreover, a list of publications and work in progress is provided.

6.1 CONCLUSIONS AND CONTRIBUTIONS

6.1.1 *Discussion regarding type-1 and type-2 fuzzy logic systems*

In Chapter 1, and using a brief historical overview about fuzzy logic as a guiding thread, we introduced a controversial debate that has been discussed since the dawn of type-2 fuzzy logic. This controversy faced the opinions of different authors: it is well known that type-2 fuzzy logic systems, both general and interval, outperform their type-1 counterparts, in terms of their ability to handle, model and minimise uncertainty and its effects. Regarding this topic, some authors argued that this ability lied in the higher number of free parameters (or degrees of freedom) available in type-2 MF when compared with the type-1 case; those who make this claim hypothesize that if the type-1 fuzzy sets were allowed to have the same number of parameters (and hence the same number of design degrees of freedom) as the type-2 fuzzy sets, and if both the type-2 and type-1 fuzzy FLSs were given equal opportunities of optimisation, then the type-2 and type-1 FLSs should end up with equal performances. On the other hand, other authors stated that most of this ability is due to how those type-2 systems *use* those parameters to model such uncertainty.

Although previous work has been done relating this topic (as in [13], [31], [38], [59]), they usually compare different classes of type-1 and type-2 FLSs; nonetheless, we considered some more light could be shed on this regard.

In Chapter 3 the claim for the better performance of T2FLSs is their use of extra parameters to describe MFs was examined. If this controversy revolves around *the number of free parameters* as the key to obtain a better performance, then we proposed a novel approach in which we only compare the number of available parameters per MF in different types of FLSs. Using nine different function approximation problems, we compared two different kinds of type-1 FLSs using different types of MFs (triangular and trapezoidal), which are essentially the same (piecewise linear functions) but differ in the number of parameters per MF they require to be described. Such a framework aimed to verify if there exist statistically significant differences between the performances of these kinds of systems, in terms of their function approximation and uncertainty handling ability. We intended to find out whether the number of free parameters is the *keystone* to improve the performance of a FLS or not, because if the claim is not valid in this framework, it cannot be valid for the IT2 versus T1FLS framework. This would help us to prove of disprove if there are some other factors involved.

We showed in five differente experiments, using nine function approximation problems, that allowing type-1 FLSs to have one extra parameter per antecedent MF during the optimisation process does not enable them to deliver improved function approximation ability and uncertainty handling. It was also highlighted that these findings are consistent with those in [11]. In light of these results, it seems reasonable to conclude that the ability of T2FLSs to perform better than their T1 counterparts is due to the way the former make use of their free parameters, rather than the number of degrees of freedom available in the system. Hence, the use of type-2 fuzzy logic is completely justified.

6.1.2 *Extending the join and meet operations on type-2 fuzzy sets*

Conclusions from Part ii allowed us to completely justify the use of type-2 fuzzy logic systems and hence, we focused our efforts on further developing it. In order to do so, in Chapter 1 we introduced two important facts about GT2 and IT2 fuzzy logic, which motivated the rest of our research.

Firstly, IT2FSs were proved to be more general than IVFSs, as shown in [89] and introduced in Chapter 4. This new perception of IT2FSs (which we have been calling "general forms of IT2FSs") evinced that some of these sets might actually have non-convex secondary grades, which posed significant trouble when trying to utilise FLSs using these sets: on the one hand, the set theoretic operations on IVFSs, which are needed to define the inference engine, required the secondary grades to be closed and connected intervals, a condition that is no longer met. Secondly, although these operations could be approached from the GT2 framework, the previous existing literature and closed formulas for the join and meet operations required the secondary grades to be normal and convex type-1 fuzzy sets, which does not hold any more. Hence, a need to develop the set theoretic operations on GT2FSs with arbitrary secondary grade emerged, in order to later expand the theory of the gfIT2FLSs.

This need was tackled in Chapter 4, in which we presented and proved two new theorems providing closed formulas for the join (union) and meet (intersection) operations on GT2FSs having arbitrary secondary grades, i.e., T2FSs in which the restrictions regarding the normality and convexity of the secondary grades are no longer required. Those theorems were also particularised for the more specific case of gfIT2FSs as presented in

[89], and closed formulas were also obtained for these sets. Several examples were provided, including the cases of secondary grades being convex and normal, non-convex and non-normal and either 0 or 1.

Hence, the work presented in Chapter 4 allowed us to explore the potential of the gfIT2FLSs which use gfIT2FSs that are not equivalent to IVFSs, which was performed in Chapter 5. In addition, it could help to explore the potential of GT2FLSs using T2FSs with non-convex and/or non-normal secondary grades.

6.1.3 *Further developing the theory of the gfIT2FLSs*

The appearance of the new perception of the gfIT2FSs loomed up two new needs: on the one hand, to further develop the set theoretic operations on these sets (which was tackled in Chapter 4); on the other hand, to completely define the fuzzy logic systems using the gfIT2FSs, which we have called gfIT2FLSs. To do so, a brand new theoretical framework for these FLSs was required, as some of the involved fuzzy sets might actually have non-convex secondary grades. This second need was tackled and solved in Chapter 5.

The whole structure of the gfIT2FLSs was overviewed, and especial attention was drawn to those blocks presenting significant differences with other well-known FLSs. We focused our efforts in the following blocks:

- **Fuzzification stage:** For the block mapping the crisp input values into fuzzy sets, both singleton and non-singleton schemes were considered, and the mathematical equations to perform this operation were provided. In addition, an algorithm to create non-singleton fuzzy input sets to model noisy measurements coming from real world sensors was presented. To finalise, an example on how to use that algorithm was performed in order to obtain gfIT2FSs to model the noisy measurements from a sonar sensor affected by several sources of uncertainty, such as temperature, wind and sonic noise.

- **Inference engine:** The mathematical framework required to define the inference engine was revisited, where the results from Chapter 4 were used in order to allow the FLS to use the gfIT2FSs in both antecedent and consequents. In addition, an example to obtain gfIT2FSs as antecedents was provided, in which we made use of the survey method to model them.

- *Type-reduction:* As we are dealing with the gfIT2FSs, which can actually have disconnected regions in their secondary grades, a new type-reduction method was required. To do so, we used a partial result presented in [2], in order to approximate a finite set of closed, connected and disjointed intervals or singletons by one single interval. Using that result, both the rule firing degrees from the inference engine and the rule consequents can be approximated, so that the classical centre-of-sets type-reducer can be applied afterwards. We called this new method the *modified centre-of-sets type-reducer*.

In Chapter 5, the general forms of Interval-Type 2 Fuzzy Logic Systems (gfIT2FLSs) were introduced, which use IT2FSs that are more general than IVFSs, and can actually have non convex secondary membership functions, allowing us to represent and model uncertainty in a way the classical IVFSs cannot.

Hence, the need to develop the theoretical framework of FLSs using these gfIT2FSs has been covered, closing the initial motivations and covering all the goals proposed at the beginning of this dissertation.

6.2 LIST OF PUBLICATIONS

This section gathers a list of the publications derived from the research of the work presented in this dissertation, as well as other related research.

Journal publications

1. G. Ruiz, H. Hagras, H. Pomares, I. Rojas and H. Bustince, *Join and Meet Operations for Type-2 Fuzzy Sets With Nonconvex Secondary Memberships*, in IEEE Transactions on Fuzzy Systems, vol. 24, no. 4, pp. 1000-1008, Aug. 1 2016. doi: 10.1109/TFUZZ.2015.2489242

2. A. Olivares, G. Ruiz-Garcia, G. Olivares and J. M. Gorriz, *Automatic Determination of Validity of Input Data Used in Ellipsoid Fitting MARG Calibration Algorithms*, in Sensors, vol. 13, no. 9, pp.11797, 2013. doi: 10.3390/s130911797.

3. G. Ruiz-Garcia, H. Hagras, H. Pomares and I. Rojas, *Towards a Fuzzy Logic System Based on General Forms of Interval Type-2 Fuzzy Sets*, in IEEE Transactions on Fuzzy Systems (revision process).

4. G. Ruiz-Garcia, H. Hagras, J. Mendel, H. Pomares and I. Rojas, *Effects of Extra Type-1 Fuzzy Set Parameters on the Performance of a Fuzzy System*, in Information Sciences (revision process).

International conferences

1. G. Ruiz, H. Hagras, H. Pomares, I. Rojas, (2016, November). *Towards general forms of interval type-2 fuzzy logic systems*, in Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on (pp. 1216-1223).

2. G. Ruiz, H. Hagras, H. Pomares, I. Rojas, (2017, January). *The Non-singleton Fuzzification Operation for General Forms of Interval Type-2 Fuzzy Logic Systems*, in Fuzzy Systems (FUZZ-IEEE) (revision process).

Book chapters

1. G. Ruiz, H. Hagras, H. Pomares, I. Rojas, (2016, November), *Towards a Framework for Singleton General Forms of Interval Type-2 Fuzzy Systems*, in Lecture Notes in Computer Science, vol. 10147 (to be published).

6

CONCLUSIONES, PRINCIPALES CONTRIBUCIONES Y LISTA DE PUBLICACIONES

Lo que es útil en la lógica difusa no es especialmente nuevo y lo que es nuevo no es especialmente útil.

— Susan Haack

El trabajo presentado en esta tesis es una contribución a los campos de la teoría de la lógica difusa y los sistemas difusos tipo-2. Las principales contribuciones y los trabajos futuros se presentaron al final de los Capítulos 3, 4 y 5. Dichas conclusiones se resumen aquí de nuevo, enlazándolas a su vez con las motivaciones y objetivos iniciales expuestos en el Capítulo 1. Además, se incluye una lista de publicaciones y trabajos en proceso.

6.1 CONCLUSIONES Y CONTRIBUCIONES

6.1.1 *Discusión sobre los sistemas difusos tipo-1 y tipo-2*

En el Capítulo 1, y haciendo uso de una revisión histórica como hilo conductor, se introdujo un polémico debate que ha sido discutido desde la misma aparición de la lógica difusa tipo-2. Esta controversia enfrentó las opiniones de distintos autores: es bien sabido que los sistemas difusos tipo-2, tanto generales como intervalo, presentan mejor rendimiento que sus homólogos tipo-1, en términos de su capacidad para gestionar, modelar y minimizar la incertidumbre y sus efectos. Sobre este tema, algunos autores argumentan que dicha capacidad reside en el mayor número de parámetros (o grados de libertad) disponibles en una función de pertenencia de tipo-2 comparada con la tipo-1; aquellos que reivindican esto hipotetizan que si a los conjuntos difusos tipo-1 se les permitiera disponer del mismo número de parámetros (y por tanto el mismo número de grados de libertad en el diseño) que a los conjuntos tipo-2, y si a ambos sistemas difusos se les concedieran las mismas oportunidades de optimización, entonces tanto los sistemas tipo-1 como los tipo-2 deberían tener el mismo similar. Por otra parte, otros autores manifiestan que la mayor parte de esa capacidad se debe a cómo los sistemas tipo-2 utilizan esos parámetros para modelar dicha incertidumbre.

A pesar de que existen trabajos previos relacionados con este tema (como [13], [31], [38] y [59]), estos normalmente comparan distintas clases de sistemas difusos tipo-1 y tipo-2; no obstante, se ha considerado que puede arrojarse algo más de luz a este respecto.

En el Capítulo 3 dicha reivindicación sobre que el rendimiento de los sistemas difusos tipo-2 se debe al uso de parámetros extra en la descripción de sus funciones de pertenencia fue puesta a examen. Si la controversia gira en torno al *número de parámetros libres* como pieza clave para obtener un mejor rendimiento, entonces se propuso un nuevo enfoque en el que solamente se compara el número de parámetros disponibles por cada función de pertenencia en distintos tipos de sistemas difusos. Utilizando nueve problemas de aproximación funcional diferentes, se compararon dos tipos de sistemas difusos tipo-1, que utilizan distintos tipos de funciones de pertenencia (triangulares y trapezoidales), que son esencialmente iguales (funciones lineales a trozos) pero que difieren en el número de parámetros por función de pertenencia que necesitan para describirse. Dicho marco de referencia aspiraba a verificar si existen diferencias estadísticamente significativas entre los rendimientos de estos tipos de sistemas, en términos de su capacidad de aproximación funcional y gestión de la incertidumbre. Se pretendía averiguar si el número de parámetros libres es la *clave* para mejorar el rendimiento de un sistema difuso o no, porque si la reivindicación no es válida para este marco de referencia, entonces no puede serlo en el marco comparativo entre sistemas difusos tipo-1 y tipo-2 intervalo. Este enfoque nos ayudaría a probar o refutar si hay otros factores involucrados.

Se mostró en cinco experimentos diferentes, utilizando nueve problemas de aproximación funcional, que permitir a los sistemas difusos tipo-1 tener un parámetro extra por cada función de pertenencia en los antecedentes durante el proceso de optimización no les permite ofrecer mejor capacidad de aproximación funcional ni de gestión de la incertidumbre. También se señaló que estos hallazgos son consistentes con los presentados en [11]. A la luz de estos resultados, parece razonable concluir que la capacidad de los sistemas difusos tipo-2 de ofrecer mejor rendimiento que sus homólogos tipo-1 se debe a la forma en que los primeros hacen uso de sus parámetros libres, más que en el número de grados de libertad disponibles en el sistema. Por tanto, el uso de la lógica difusa tipo-2 está plenamente justificado.

6.1.2 *Extendiendo las operaciones join y meet sobre conjuntos difusos tipo-2*

Las conclusiones extraídas de la Parte ii nos permitieron justificar plenamente el uso de la lógica difusa tipo-2 y, por tanto, centrar nuestros esfuerzos en desarrollarla. Para ello, en el Capítulo 1 se introdujeron dos hechos importantes sobre la lógica difusa tipo-2, tanto general como intervalo, que motivaron el resto de nuestra investigación.

En primer lugar, se probó que los conjuntos difusos tipo-2 intervalo (IT2FSs) son más generales que los de valores intervalados (IVFSs), como se mostró en [89] y se introdujo en el Capítulo 4. Esta nueva percepción de estos conjuntos IT2FSs (a los que nos hemos referido como "formas generales de IT2FSs") puso en evidencia que algunos de estos conjuntos pueden tener grados secundarios no convexos, lo que planteaba dificultades significativas al utilizar sistemas difusos que hicieran uso de dichos conjuntos: por una parte, las operaciones entre conjuntos IVFSs, necesarias para definir el motor de inferencia, requerían que los grados secundarios fuesen intervalos cerrados y conexos, una condición que ahora no siempre se cumple. Por otra parte, pese a que estas operaciones podían enfocarse desde el marco de referencia de los conjuntos difusos tipo-2 generales, la literatura previa y las fórmulas existentes para las operaciones *join* y *meet* requerían que los grados secundarios fuesen conjuntos difusos tipo-1 normales y convexos, lo cual tampoco se cumple. Por tanto, emergió la necesidad de desarrollar las operaciones entre conjuntos tipo-2 generales con grados secundarios arbitrarios, para posteriormente expandir la teoría de las formas generales de sistemas difusos tipo-2 intervalo (gfIT2FLSs).

Tal necesidad se abordó en el Capítulo 4, en el que se presentaron y demostraron dos nuevos teoremas que proporcionan fórmulas cerradas para las operaciones *join* (unión) y *meet* (intersección) sobre conjuntos difusos tipo-2 generales con grados secundarios arbitrarios, es decir, conjutos tipo-2 en los que no son necesarias las restricciones relacionadas con su convexidad y su carácter normal. Estos teoremas también se particularizaron para los casos más específicos de gfIT2FSs tal y como se presentaron en [89], y también se obtuvieron fórmulas cerradas para dichos conjuntos. Se proporcionaron varios ejemplos, incluyendo los casos de grados secundarios normales y convexos, no normales y no convexos, y con valores iguales a únicamente 0 o 1.

Por tanto, el trabajo presentado en el Capítulo 4 nos permitió explorar el potencial de los sistemas gfIT2FLSs, que utilizan conjuntos gfIT2FSs que no son equivalentes a los IVFSs, lo que se llevó a cabo en el Capítulo 5. Adicionalmente, podría ayudarnos a explorar el potencial de los sistemas difusos tipo-2 generales que hagan uso de conjuntos difusos con grados secundarios no normales y/o no convexos.

6.1.3 *Desarrollando la teoría de los sistemas gfIT2FLSs*

La aparición de la nueva percepción sobre los gfIT2FSs hizo surgir dos nuevas necesidades: por una parte, desarrollar las operaciones entre estos conjuntos (lo cual se abordó en el Capítulo 4); por otra parte, definir de forma completa los sistemas difusos que utilizan dichos conjuntos, y a los cuales hemos llamado gfIT2FLSs. Para realizar esto último se requerÃa un marco de referencia completamente nuevo para estos sistemas difusos, puesto que algunos de los conjuntos involucrados podían tener grados secundarios no convexos. Esta segunda necesidad se acometió y resolvió en el Capítulo 5.

Se revisó la estructura completa de estos gfIT2FLSs, y se prestó especial atención a aquellos bloques que presentan diferencias significativas con otros sistemas difusos bien conocidos. Nuestros esfuerzos se centraron en los siguientes bloques:

- **Etapa de *fuzzificación*:** Para el bloque encargado de mapear los valores numéricos de entrada en conjuntos difusos, se consideraron tanto esquemas *singleton* como *no singleton*, y se proporcionaron las ecuaciones matemáticas para realizar dicha operación. Adicionalmente, se presentó un algoritmo para crear conjuntos difusos de entrada para modelar medidas ruidosas procedentes de un sensor real. Para finalizar, se realizó un ejemplo sobre cómo utilizar dicho algoritmo a fin de obtener conjuntos gfIT2FSs para modelar las medidas ruidosas de un sensor sónar afectado por diversas fuentes de incertidumbre, tales como alta temperatura, viento y ruido sónico.

- **Motor de inferencia:** Se revisó el marco de referencia necesario para definir el motor de inferencia, en el que se utilizaron los resultados procedentes del Capítulo 4, a fin de permitir que los sistemas difusos puedan utilizar los conjuntos gfIT2FSs tanto en los antecedentes como en los consecuentes. Además, se mostró un ejemplo para obtener dichos conjuntos en los antecedentes, en el que se utilizó el método de la encuesta para modelarlos.

- **Reducción de tipo:** Dado que estamos tratando con conjuntos gfIT2FSs, que pueden de hecho tener regiones inconexas en los grados secundarios, se requería un nuevo método de reducción de tipo. Para ello, se utilizaron resultados parciales del trabajo presentado en [2], a fin de aproximar un conjunto finito de intervalos cerrados, conectados y disjuntos o *singletons* mediante un único intervalo. Haciendo uso de dicho resultado, tanto los grados de activación de las reglas en el motor de inferencia como los consecuentes pueden ser aproximados, de tal manera que el método clásico *centro de los conjuntos* (*centre-of-sets*) pueda utilizarse a posteriori. Este nuevo método fue bautizado como *reductor de tipo centro de los conjuntos modificado* (*modified centre-of-sets type-reducer*).

En el Capítulo 5 se introdujeron los sistemas difusos gfIT2FLSs, que utilizan conjuntos IT2FSs que son más generales que los IVFSs, y pueden tener funciones de pertenencia secundarias no convexas, permitiéndonos representar y modelar la incertidumbre de nuevas formas que los conjuntos IVFSs no podrían.

Por lo tanto, la necesidad de desarrollar el marco de referencia teórico de los sitemas difusos que utilizan estos gfIT2FSs ha sido cubierto, cerrando así las motivaciones iniciales y cubriendo todos los objetivos propuestos al inicio de esta tesis.

6.2 LISTA DE PUBLICACIONES

Esta sección recoge una lista con las publicaciones que se han derivado del desarrollo del trabajo de investigación presentado en esta tesis, así como otros trabajos de investigación relacionados de forma tangencial.

Publicaciones en revistas

1. G. Ruiz, H. Hagras, H. Pomares, I. Rojas and H. Bustince, *Join and Meet Operations for Type-2 Fuzzy Sets With Nonconvex Secondary Memberships*, in IEEE Transactions on Fuzzy Systems, vol. 24, no. 4, pp. 1000-1008, Aug. 1 2016. doi: 10.1109/TFUZZ.2015.2489242
2. A. Olivares, G. Ruiz-Garcia, G. Olivares and J. M. Gorriz, *Automatic Determination of Validity of Input Data Used in Ellipsoid Fitting MARG Calibration Algorithms*, in Sensors, vol. 13, no. 9, pp.11797, 2013. doi: 10.3390/s130911797.
3. G. Ruiz-Garcia, H. Hagras, H. Pomares and I. Rojas, *Towards a Fuzzy Logic System Based on General Forms of Interval Type-2 Fuzzy Sets*, in IEEE Transactions on Fuzzy Systems (revision process).
4. G. Ruiz-Garcia, H. Hagras, J. Mendel, H. Pomares and I. Rojas, *Effects of Extra Type-1 Fuzzy Set Parameters on the Performance of a Fuzzy System*, in Information Sciences (revision process).

Congresos internacionales

1. G. Ruiz, H. Hagras, H. Pomares, I. Rojas, (2016, November). *Towards general forms of interval type-2 fuzzy logic systems*, in Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on (pp. 1216-1223).
2. G. Ruiz, H. Hagras, H. Pomares, I. Rojas, (2017, January). *The Non-singleton Fuzzification Operation for General Forms of Interval Type-2 Fuzzy Logic Systems*, in Fuzzy Systems (FUZZ-IEEE) (revision process).

Capítulos de libros

1. G. Ruiz, H. Hagras, H. Pomares, I. Rojas, (2016, November), *Towards a Framework for Singleton General Forms of Interval*

Type-2 Fuzzy Systems, in Lecture Notes in Computer Science, vol. 10147 (to be published).

Part V

APPENDIX

A

APPENDIX: DETAILS ABOUT THE EVOLUTIONARY ALGORITHM

Index

In this Appendix we extensively describe the details of the GA employed in all experiments through Chapter 3.

A.1 SOLUTION ENCODING

When using a scattered input space, each fuzzy system will be represented by a $(N_{MF} \cdot p) \times N_r$ matrix, where N_{MF} is the number of parameters per MF, p is the number of inputs to the system, and N_r is the number of rules in the system/chromosome. Thus, the MF associated with the i-th input in the j-th rule, $\mu_i^j(x_i)$, will be described by three parameters $[a_i^j, b_i^j, c_i^j]^T$ in the case of triangular MFs, and four $[a_i^j, b_i^j, c_i^j, d_i^j]$ in the case of trapezoid MFs. It is important to note that in both cases, these descriptions of the MFs consider the parameters are ordered in ascending order, such that $a_i^j \leqslant b_i^j \leqslant c_i^j$ for triangular MFs and $a_i^j \leqslant b_i^j \leqslant c_i^j \leqslant d_i^j$ for the trapezoidal case.

All parameters of all MFs of a given rule are grouped in the same column, i.e., the j-th rule of a given chromosome C_{ind}, noted as C_{ind}^j, will be presented as Equation (A.1.1) for triangular MF, and as Equation (A.1.2) for trapezoid MFs.

$$C_{ind}^j = [a_1^j, b_1^j, c_1^j, ..., a_p^j, b_p^j, c_p^j] \tag{A.1.1}$$

$$C_{ind}^j = [a_1^j, b_1^j, c_1^j, d_1^j, ..., a_p^j, b_p^j, c_p^j, d_p^j] \tag{A.1.2}$$

A given chromosome will be formed by N_r columns. For the case of single objective function, we chose $N_r = 5, 10, 15, 20, 25, 30$, and for the case of a multi-objective function, because the number of rules is one of the objectives to minimise, the number of rules is different for each chromosome, so each individual chromosome has a different number of columns N_r, always lying within a given range $[N_r^{min}, N_r^{max}] \equiv [5, 20]$.

When using a partitioned input space (experiments 4 and part of 5) the chosen solution encoding is quite different. In this case, each individual within the population is described by a $N_{MF} \times (N_{MAX} \cdot p)$ matrix, where N_{MAX} is the number of MFs per input. In that matrix, columns from 1 to p will be as shown in

Equation (A.1.1) and (A.1.2), and will describe the MFs associated with input 1; columns from $p+1$ to $2p$ will describe the MFs from input 2; in general, columns from $(i-1) \cdot p+1$ to $i \cdot p$ describe the MFs of input i, with $i = 1, ..., p$. In this work, we have chose $N_{MAX} = 7$, thus leading to a partitioned input space using $7^2 = 49$ rules.

A.2 FITNESS EVALUATION

As in [11], we will use nine function approximation problems in Table 3.3.1 from [19]. In all experiments, for each function approximation problem, a data set of 400 training data is used. These points are obtained by dividing the input space into a 20×20 grid, and choosing a random point from each zone delimited by that grid. For the training process, a two-fold cross validation method is adopted, in which the 400 data points are divided into two groups by means of the NNO-CFA algorithm [32]. This algorithm divides the data set into two roughly balanced and equally sized subsets that can be used for training and validation (by *balanced* we mean that both subsets are created in such a way that two very close points will belong to different subsets, so both of them will roughly cover the same input space). Because both subsets are balanced, there is no need for a five-fold or ten-fold cross-validation procedure, thus significantly reducint the computation time.

After dividing the 400 data into two subsets by means of the NNO-CFA algorithm, the NRMSE is computed as follows: for a given chromosome (it is, a given antecedents' MF parameters):

1. The first 200 data are used to obtain the optimal consequents for those antecedents, and the second 200 data are used for validation.

2. We use the second set of points to obtain the optimal consequents and the first one to validate.

3. We compute the mean between the two NRMSE as the final NRMSE value.

In the case of the multi-objective scenario, the NRMSE is one of the two objectives to minimise, and is calculated in the same exact way as in the single-objective case. The second objective to minimise is the number of rules (i.e. the number of columns in the matrix representing the system/chromosome). The cost function for the second objective is just N_r, which is the number of rules of the chromosome.

A.3 INITIAL POPULATION

The population is initialised differently in the single-objective (experiments 1, 3, 4 and 5) and the multi-objective (experiments 2 and 5) situations: in the former, all individuals are randomly initialised, ensuring that the parameters representing each MF and each rule are properly ordered. Besides, the columns defining a rule are ordered in groups, where each group represents a MF; i.e., if we consider a system with two inputs, then the j-rule will be as $\left(a_1^j, b_1^j, c_1^j, a_2^j, b_2^j, c_2^j\right)$ in the triangular case and $\left(a_1^j, b_1^j, c_1^j, d_1^j, a_2^j, b_2^j, c_2^j, d_2^j\right)$ in the trapezoidal one.

Random initialisation is frequently used in the literature [1], [37]; however, for the multi-objective case, we chose the following initialisation approach [11]:

1. For *each possible number of rules* within the range $[N_r^{min}, N_r^{max}]$, one chromosome is created; using a set of 400 data as explained previously, the centres of the antecedent MFs are located by means of the K-means method [63], whereas the spread is obtained using the K-nearest neighbours (KNN) algorithm [21]. It was chosen a value of $K = 2$ neighbourghs, at it seemed to be reasonable to avoid the MFs to have excessive overlapping. Calling Ce_i^j and S_i^j to the centre and spread, respectively, of the i-th input of the j-th rule, then the parameters for that given MF are initialised as follows:

   a Triangular case:

   $$\left(a_i^j, b_i^j, c_i^j\right) = \left(Ce_i^j - S_i^j, Ce_i^j, Ce_i^j + S_i^j\right)$$

   b Trapezoidal case:

   $$\left(a_i^j, b_i^j, c_i^j, d_i^j\right) = \left(Ce_i^j - S_i^j, Ce_i^j - \frac{S_i^j}{2}, Ce_i^j + \frac{S_i^j}{2}, Ce_i^j + S_i^j\right)$$

2. If necessary, more randomly generated individuals are introduced in the population to introduce diversity.

The reason why this approach has not been adopted in the single-objective case is very simple: because the number of rules is fixed, the K-means and KNN algorithms would always obtain the same solutions for a given set of training points, and thus the initial population would lack the proper diversity. Hence, random initialisation has been chosen in that case.

It is important to highlight that, apart from the random initialisation, in experiment 3 we also introduce the best triangular individual into the initial trapezoidal population.

A.4 GENETIC OPERATORS

For an evolutionary algorithm to perform properly and to be able to find an optimal solution, it is important to find a proper balance between the exploration of new zones in the solution space and the exploitation of the promising areas already found. The genetic operators chosen for this work aim to achieve this balance.

The GA implementation used in this work needs a parameter named "Crossover Fraction", which indicates the fraction of the offspring that are generated by crossover. The rest of the offspring are generated by the mutation operator. In this work, we set this parameter to 0.8, which means that the 80% of the offspring are generated by crossover, whereas the remaining 20% are created by the mutation operator.

A.4.1 *The cross-over operator in single-objective and scattered input space scenario*

In the crossover operation, the genetic material of two parents are combined to produce one child or offspring. In our T1FLSs scenario, the genetic material are combined at two different levels: interchanging information at MF level (i.e. combining the MFs' parameters) and interchanging complete MFs and/or rules. How these operations are performed is different for each experiment.

When using a scattered input space and single-objective optimisation (Experiments 1 and 3), the crossover operator combines genetic material from two parents to produce one child or offspring. We use a hybrid operator to combine the classical single-point crossover [40] with the BLX-α operator [40].

For each pair of parents, with probability P_{mod} (0.5 in our work), the single-point crossover is applied; for the rest of the parents, the BLX-α crossover is used.

The single-point crossover operator works as follows:

1. Select a random integer in the range $SP \in [1, N_r]$, where N_r is the number of rules of each chromosome.

2. Create a child by taking rules $1, ..., SP - 1$ from parent 1, and rules $SP, ..., N_r$ from parent 2, i.el:

$$\text{Child} = [P_1 (1 : (SP - 1)) \mid P_2 (SP : N_r)] \tag{A.4.1}$$

On the other hand, those pairs of parents that have not been affected by the single-point crossover operator will be combined using the well-known BLX-α [40] operator with $\alpha = 0.5$, in which each gene z_i of the child, which must lie within the range $[a_i, b_i]$, is a combination of the genes in the same position from the two parents, x_i and y_i. Hence, z_i is generated as a random value in $[l_i, r_i]$, where:

$$\begin{aligned} l_i &= \max(a_i, c_{min} - I \cdot \alpha) \\ r_i &= \min(b_i, c_{max} + I \cdot \alpha) \end{aligned} \tag{A.4.2}$$

In which $c_{min} = \min(x_i, y_i)$, $c_{max} = \max(x_i, y_i)$ and $I = c_{max} - c_{min}$. This operator may cause the resulting parameters within the same MF to be disordered; if this happens, the resulting parameters are reordered accordingly.

A.4.2 *The cross-over operator in multi-objective scenario*

In the multi-objective scenario (Experiments 2 and 5), which uses a scattered input space, the genetic material from the two parents are combined to produce two children. Let N_{r1} and N_{r2} be the number of rules of parent 1 and parent 2, respectively. The crossover function for the multi-objective scenario operates as follows:

1. Select a random integer in the range $SP \in [1, \min(N_{r1}, N_{r2})]$.

2. Create child 1 by taking rules $1, ..., SP-1$ from parent 1, and rules $SP, ..., N_{r2}$ from parent 2, i.e.:

$$Child_1 = [P_1 (1 : (SP-1)) \mid P_2 (SP : N_{r2})] \qquad (A.4.3)$$

3. Create child 2 by taking rules $1, ..., SP-1$ from parent 2, and rules $SP, ..., N_{r1}$ from parent 1, i.e.:

$$Child_2 = [P_2 (1 : (SP-1)) \mid P_1 (SP : N_{r1})] \qquad (A.4.4)$$

This operator creates two offspring from two parents (unlike the previous scenario), in such a way that child 1 has the same number of rules as parent 2, and child 2 has the same number of rules as parent 1.

Those pairs of parents which have not been affected by the single-point crossover operator are combined using the BLX-α operator, with $\alpha = 0.5$. Each gene z_i of the child, which must lie within the range $[a_i, b_i]$, is a combination of the genes in the same position from the two parents, x_i and y_i. Hence, z_i is generated as a random value in $[l_i, r_i]$, where:

$$\begin{aligned} l_i &= max(a_i, c_{min} - I \cdot \alpha) \\ r_i &= min(b_i, c_{max} + I \cdot \alpha) \end{aligned} \qquad (A.4.5)$$

In which $c_{min} = min(x_i, y_i)$, $c_{max} = max(x_i, y_i)$ and $I = c_{max} - c_{min}$. This operator may cause the resulting parameters within the same MF to be disordered; if this happens, the resulting parameters are reordered.

So, the whole BLX-α crossover is as follows: let N_{r1}, N_{r2} be the number of rules of parents 1 and 2, respectively. Let $G_1 = min(N_{r1}, N_{r2})$ and $G_2 = max(N_{r1}, N_{r2})$.

1. The first child will have G_1 rules that will be obtained by combining the parents' rules with the BLX-α.

2. The second child will have G_2 rules; the first G_1 are obtained with the BLX-α operation; the remaining rules are copied directly from the parent with the greater number of rules.

Although previous works [11] have only considered the second option as the procedure to apply the BLX-α operator, there is a reason to use this new two step method. Consider the following example: let $N_{r1} = 7$ and $N_{r2} = 14$; if only the second way to crossover the two parents is considered, an offspring with $N_{r3} =$

14 will always be created. Before the crossover was performed, the mean number of rules per individual was $(7+14)/2 = 10.5$. After the crossover, the mean is $(7+14+14)/3 = 11.67$. Hence, this combination contributes to increase the mean of rules per individual in the population, exploring more intensively the part of the Pareto Front corresponding to higher number of rules. By generating two offspring with both G_1 or G_2 rules, the mean of rules per individual remains unchanged, thus allowing the GA to explore the Pareto Front in an equitable way.

A.4.3 *The crossover operator in single-objective and partitioned input space scenario*

When using a partitioned input space (Experiments 3 and 5), solution encoding is quite different from the scattered input space scenario (Experiment 4) and was explained in Section A.1. Hence, in this case, the split point operator will work as follows:

1. **For each input** i, with $i = 1, ..., p$, a random integer SP is selected in the range $SP \in [1, N_{MAX}]$.
2. The child's MFs for input i are crafted by taking the rules from parent 1 ranging from 1 so $SP-1$, and the rules from parent 2 ranging from SP to N_{MAX}.

Hence, the whole crossover operator for Experiment 4 works as follows: for each pair of parents, with probability $P_{mod} = 0.5$, the single-point crossover is applied; and for the rest of the parents, the BLX-α crossover is applied.

A.4.4 *The mutation function in single-objective scenario*

The mutation operator creates new individuals or chromosomes by randomly changing their genetic material. In our approach, we use an operator that randomly changes some genes within the chromosome.

When using single-objective optimisation, the mutation will change a given gene z_i in such a way that it is within the range $[a_i, b_i]$. If the change causes the chromosome to lose the proper order, as explained previously, then the parameters are reordered. The probability of changing each gene is $P_m = 1/N_{params}$, where N_{params} is the number of parameters in the system. With this value, each individual affected by the mutation operator changes *on average* one single parameter.

A.4.5 *The mutation function in multi-objective scenario*

The mutation operator for using multi-objective optimisation creates new individuals or chromosomes by randomly changing their genetic material. In our approach, we use a combination of two mutation operators.

A.4.5.1 *Mutation by adding/substracting rules*

The first mutation operator modifies the number of rules in the chromosome, with a given probability P_mod (0.5 by default). If the chromosome is changed, a random integer in the range $[1, R_{max}]$ is chosen; then, addition or subtraction of rules is chosen in an equally probable manner. If addition is chosen, the rules are randomly generated and appended to the chromosome; on the other hand, if subtraction is chosen, some rules of the individual are randomly chosen and removed. The final number of rules must always lie within the range $[N_r^{min}, N_r^{max}] = [5, 20]$. In our work $R_{max} = 3$.

A.4.5.2 *Mutation by adding/substracting rules*

The second mutation operator, as in the single-objective case, randomly changes a given gene z_i in such a way that it lies within the range $[a_i, b_i]$. If the change causes the chromosome to lose the proper order in its parameters, then they are reordered. The probability of changing each gene is P_m. Previous works [11] have considered P_m to be a fixed value; however, we consider that this decision favours introducing diversity (changes) in individuals having greater number of rules/parameters, because if P_m is the probability to change a given gene, then $1 - P_m$ is the probability of not changing it. If the probability of changing two different genes is independent (which seems to be reasonable), then the probability of not changing any gene, and thus to let the chromosome unchanged, is:

$$P_{unchanged} = (1 - P_m)^N \qquad \text{(A.4.6)}$$

Here N is the number of parameters in the chromosome. Hence, the probability of actually changing the chromosome, with *at least* one modified gene, is:

$$P_{change} = 1 - (1 - P_m)^N \qquad \text{(A.4.7)}$$

As an example, for triangular MFs, $N = N_{MF} \cdot p \cdot N_r = 3 \cdot p \cdot N_r$, and assume $P_m = 0.02$; then, for $p = 2$ inputs, the function $P_{change}(N_r)$ is depicted in Figure A.4.1(a):

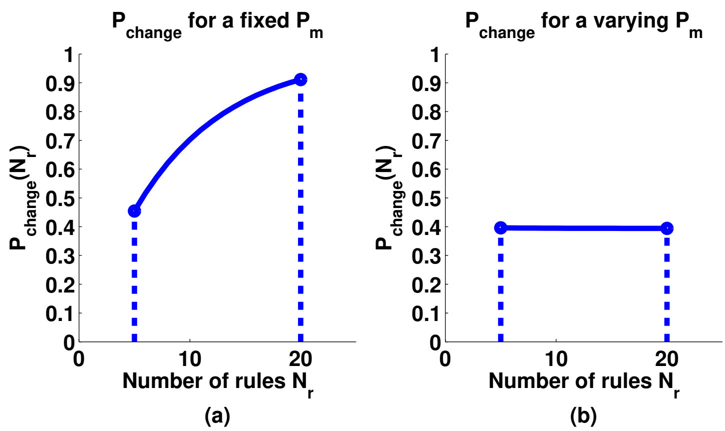

Figure A.4.1
Probability of changing a chromosome as a function of the number of rules, for a fixed P_m.

Figure A.4.1(a) clearly shows that it is far more probable to introduce changes in chromosomes with higher number of rules/-parameters; thus, the changes introduced by the mutation operator using a fixed P_m makes the exploration of the Pareto Front corresponding to greater number of rules more intensive than the part corresponding to lower number of rules. To resolve this problem, we use a value of P_m that varies with the number of rules, i.e.:

$$P_m = \frac{k}{N} \tag{A.4.8}$$

The probability to change an individual is then given by:

$$\begin{aligned} P_{change}(N_r) &= 1 - (1 - P_m)^N = 1 - \left(1 - \frac{k}{N}\right)^N = \\ &= 1 - \left(1 - \frac{k}{N_{MF} \cdot p \cdot N_r}\right)^{N_{MF} \cdot p \cdot N_r} \end{aligned} \tag{A.4.9}$$

As an example, for $k = 0.5$, then the function $P_{change}(N_r)$ is depicted in Figure A.4.1(b). Observe that, although the probability of changing a chromosome is not *strictly constant*, it remains almost unchanged in the range $[N_r^{min}, N_r^{max}] = [5, 20]$ and, thus, is a more equitable way of implementing the mutation operator.

BIBLIOGRAPHY

[1] Rafael Alcalá, Pietro Ducange, Francisco Herrera, Beatrice Lazzerini, and Francesco Marcelloni: *A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems*. IEEE Transactions on Fuzzy Systems, 17(5):1106–1122, 2009. (Cited on page 172.)

[2] Luca Anzilli, Gisella Facchinetti, and Giovanni Mastroleo: *A parametric approach to evaluate fuzzy quantities*. Fuzzy Sets and Systems, 250:110–133, 2014. (Cited on pages 135, 136, 157, and 163.)

[3] K. Atanassov and G. Gargov: *Interval valued intuitionistic fuzzy sets*. Fuzzy Sets and Systems, 31(3):343 – 349, 1989, ISSN 0165-0114. http://www.sciencedirect.com/science/article/pii/0165011489902054. (Cited on pages 5 and 12.)

[4] Krassimir T. Atanassov: *Intuitionistic fuzzy sets*. Fuzzy Sets and Systems, 20(1):87 – 96, 1986, ISSN 0165-0114. http://www.sciencedirect.com/science/article/pii/S0165011486800343. (Cited on pages 4 and 12.)

[5] Michał Baczyński and Balasubramaniam Jayaram: *Fuzzy Implications in Approximate Reasoning*, pages 243–277. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, ISBN 978-3-540-69082-5. http://dx.doi.org/10.1007/978-3-540-69082-5_8. (Cited on page 31.)

[6] Adrian Ban: *Approximation of fuzzy numbers by trapezoidal fuzzy numbers preserving the expected interval*. Fuzzy Sets and Systems, 159(11):1327–1344, 2008. (Cited on page 135.)

[7] Simon Blackburn: *The oxford dictionary of philosophy*, ISBN 9780199541430. //www.oxfordreference.com/10.1093/acref/9780199541430.001.0001/acref-9780199541430. (Cited on page 29.)

[8] James J Buckley: *Sugeno type controllers are universal controllers*. Fuzzy sets and systems, 53(3):299–303, 1993. (Cited on page 36.)

[9] J.J. Buckley: *Universal fuzzy controllers*. Automatica, 28(6):1245 – 1248, 1992, ISSN 0005-1098.

http://www.sciencedirect.com/science/article/pii/000510989290068Q. (Cited on page 36.)

[10] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Z. Xu, B. Bedregal, J. Montero, H. Hagras, F. Herrera, and B. De Baets: *A historical account of types of fuzzy sets and their relationships*. IEEE Transactions on Fuzzy Systems, 24(1):179–194, Feb 2016, ISSN 1063-6706. (Cited on pages 4, 5, 12, 46, and 55.)

[11] A. B. Cara, C. Wagner, H. Hagras, H. Pomares, and I. Rojas: *Multiobjective optimization and comparison of nonsingleton type-1 and singleton interval type-2 fuzzy logic systems*. IEEE Transactions on Fuzzy Systems, 21(3):459–476, 2013. (Cited on pages 6, 14, 67, 69, 73, 77, 80, 155, 161, 171, 172, 175, and 177.)

[12] Ana Belén Cara, Ignacio Rojas, Héctor Pomares, Christian Wagner, and Hani Hagras: *On comparing non-singleton type-1 and singleton type-2 fuzzy controllers for a nonlinear servo system*. In *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2011 IEEE Symposium on*, pages 126–133. IEEE, 2011. (Cited on page 116.)

[13] Oscar Castillo, Leticia Amador-Angulo, Juan R. Castro, and Mario Garcia-Valdez: *A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems*. Information Sciences, 354:257 – 274, 2016, ISSN 0020-0255. http://www.sciencedirect.com/science/article/pii/S002002551630189X. (Cited on pages 6, 13, 154, and 160.)

[14] Oscar Castillo, Juan R. Castro, Patricia Melin, and Antonio Rodriguez-Diaz: *Universal approximation of a class of interval type-2 fuzzy neural networks in nonlinear identification*. Adv. Fuzzy Sys., 2013:7:7–7:7, January 2013, ISSN 1687-7101. http://dx.doi.org/10.1155/2013/136214. (Cited on page 55.)

[15] Oscar Castillo and Patricia Melin: *3 type-2 fuzzy logic*. In *Type-2 Fuzzy Logic: Theory and Applications*, pages 29–43. Springer, 2007. (Cited on page 66.)

[16] Oscar Castillo and Patricia Melin: *Recent advances in interval Type-2 fuzzy systems*, volume 1. Springer Science & Business Media, 2012. (Cited on page 66.)

[17] J. L. Castro: *Fuzzy logic controllers are universal approximators*. IEEE Transactions on Systems, Man, and Cybernetics, 25(4):629–635, Apr 1995, ISSN 0018-9472. (Cited on page 36.)

[18] Sumantra Chakraborty, Amit Konar, Anca Ralescu, and Nikhil R Pal: *A fast algorithm to compute precise type-2 centroids for real-time control applications*. IEEE transactions on cybernetics, 45(2):340–353, 2015. (Cited on page 66.)

[19] Vladimir Cherkassky, Don Gehring, and Filip Mulier: *Comparison of adaptive methods for function estimation from samples*. IEEE Transactions on Neural Networks, 7(4):969–984, 1996. (Cited on page 171.)

[20] Simon Coupland and Robert John: *Geometric type-1 and type-2 fuzzy logic systems*. IEEE Transactions on Fuzzy Systems, 15(1):3–15, 2007. (Cited on page 85.)

[21] Thomas Cover and Peter Hart: *Nearest neighbor pattern classification*. IEEE transactions on information theory, 13(1):21–27, 1967. (Cited on page 172.)

[22] J. L. Deng: *Introduction to grey system theory*. J. Grey Syst., 1(1):1–24, November 1989, ISSN 0957-3720. `http://dl.acm.org/citation.cfm?id=90757.90758`. (Cited on pages 5 and 12.)

[23] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera: *A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms*. Swarm and Evolutionary Computation, 1(1):3–18, 2011. (Cited on page 72.)

[24] Yongsheng Ding, Hao Ying, and Shihuang Shao: *Necessary conditions on minimal system configuration for general miso mamdani fuzzy systems as universal approximators*. IEEE Trans. Systems, Man, and Cybernetics, Part B, 30:857–864, 2000. (Cited on page 37.)

[25] J Dombi: *A general class of fuzzy operators, the de morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators*. Fuzzy Sets and Systems, 8(2):149–163, 1982, ISSN 0165-0114. (Cited on pages 26 and 27.)

[26] D. Dubois and H. Prade: *Operations on fuzzy numbers*. International Journal of Systems Science, 9(6):613–626, 1978. (Cited on pages 5 and 13.)

[27] D. Dubois and H.M. Prade: *Fuzzy Sets and Systems: Theory and Applications*. Number v. 144 in *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, 1980, ISBN 9780122227509. https://books.google.es/books?id=BeBQAAAAMAAJ. (Cited on pages 26, 27, 51, and 52.)

[28] Didier Dubois: *Quelques outils méthodologiques pour la synthèse de réseaux de transport* . ThÃ¨se de doctorat, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (SupAéro), Toulouse, France, 1977. ks. (Cited on pages 4 and 12.)

[29] Didier Dubois and Henri Prade: *Operations in a fuzzy-valued logic*. Information and Control, 43(2):224 – 240, 1979, ISSN 0019-9958. http://www.sciencedirect.com/science/article/pii/S0019995879907307. (Cited on pages 4, 5, 12, and 13.)

[30] HM Fayek, I Elamvazuthi, N Perumal, and B Venkatesh: *A controller based on optimal type-2 fuzzy logic: systematic design, optimization and real-time implementation*. ISA transactions, 53(5):1583–1591, 2014. (Cited on page 66.)

[31] J. Figueroa, J. Posada, J. Soriano, M. Melgarejo, and S. Rojas: *A type-2 fuzzy controller for tracking mobile objects in the context of robotic soccer games*. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pages 359–364, May 2005. (Cited on pages 6, 13, 154, and 160.)

[32] JP Florido, Héctor Pomares, and Ignacio Rojas: *Generating balanced learning and test sets for function approximation problems*. International journal of neural systems, 21(03):247–263, 2011. (Cited on pages 72 and 171.)

[33] J. M. Garibaldi, M. Jaroszewski, and S. Musikasuwan: *Nonstationary fuzzy sets*. IEEE Transactions on Fuzzy Systems, 16(4):1072–1086, Aug 2008, ISSN 1063-6706. (Cited on pages 5, 12, and 150.)

[34] J.A Goguen: *L-fuzzy sets*. Journal of Mathematical Analysis and Applications, 18(1):145 – 174, 1967, ISSN 0022-247X. http://www.sciencedirect.com/science/article/pii/0022247X67901898. (Cited on pages 4 and 12.)

[35] I. Grattan-Guinness: *Fuzzy membership mapped onto intervals and many-valued quantities*. Mathematical Logic Quarterly, 22(1):149–160, 1976, ISSN 1521-3870. http://dx.doi.org/10.1002/malq.19760220120. (Cited on pages 4 and 12.)

[36] Przemysław Grzegorzewski and Edyta Mrówka: *Trapezoidal approximations of fuzzy numbers*. Fuzzy Sets and Systems, 153(1):115–135, 2005. (Cited on page 135.)

[37] Ouahib Guenounou, Ali Belmehdi, and Boutaieb Dahhou: *Multi-objective optimization of tsk fuzzy models*. Expert systems with applications, 36(4):7416–7423, 2009. (Cited on page 172.)

[38] H. A. Hagras: *A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots*. IEEE Transactions on Fuzzy Systems, 12(4):524–539, Aug 2004, ISSN 1063-6706. (Cited on pages 6, 13, 116, 154, and 160.)

[39] Pablo Hernández, Susana Cubillo, and Carmen Torres-Blanc: *On t-norms for type-2 fuzzy sets*. IEEE Transactions on Fuzzy Systems, 23(4):1155–1163, 2015. (Cited on page 87.)

[40] Francisco Herrera, Manuel Lozano, and Ana M Sánchez: *A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study*. International Journal of Intelligent Systems, 18(3):309–338, 2003. (Cited on page 174.)

[41] Denisse Hidalgo, Patricia Melin, and Oscar Castillo: *An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms*. Expert Systems with Applications, 39(4):4590–4598, 2012. (Cited on page 66.)

[42] Rahil Hosseini, Salah D Qanadli, Sarah Barman, Mahdi Mazinani, Tim Ellis, and Jamshid Dehmeshki: *An automatic approach for learning and tuning gaussian interval type-2 fuzzy membership functions applied to lung cad classification system*. IEEE Transactions on Fuzzy Systems, 20(2):224–234, 2012. (Cited on page 116.)

[43] Chia Hung Hsu and Chia Feng Juang: *Evolutionary robot wall-following control using type-2 fuzzy controller with species-de-activated continuous aco*. IEEE Transactions on Fuzzy Systems, 21(1):100–112, 2013. (Cited on page 66.)

[44] Emmanuel A Jammeh, Martin Fleury, Christian Wagner, Hani Hagras, and Mohammed Ghanbari: *Interval type-2 fuzzy logic congestion control for video streaming across ip networks*. IEEE Transactions on Fuzzy Systems, 17(5):1123–1142, 2009. (Cited on page 116.)

[45] C. F. Juang, R. B. Huang, and W. Y. Cheng: *An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems*. IEEE Transactions on Fuzzy Systems, 18(4):686–699, Aug 2010, ISSN 1063-6706. (Cited on pages 6 and 14.)

[46] N. N. Karnik and J. M. Mendel: *Introduction to type-2 fuzzy logic systems*. In *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, volume 2, pages 915–920 vol.2, May 1998. (Cited on pages 4, 5, 12, 13, 45, 55, and 59.)

[47] N. N. Karnik and J. M. Mendel: *Type-2 fuzzy logic systems: type-reduction*. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 2, pages 2046–2051 vol.2, Oct 1998. (Cited on page 59.)

[48] Nilesh N. Karnik and Jerry M. Mendel: *Centroid of a type-2 fuzzy set*. Information Sciences, 132(1–4):195 – 220, 2001, ISSN 0020-0255. http://www.sciencedirect.com/science/article/pii/S002002550100069X. (Cited on pages 5, 13, 59, 60, and 137.)

[49] Nilesh N. Karnik and Jerry M. Mendel: *Operations on type-2 fuzzy sets*. Fuzzy Sets and Systems, 122(2):327 – 348, 2001, ISSN 0165-0114. http://www.sciencedirect.com/science/article/pii/S0165011400000798. (Cited on pages 5, 7, 13, 15, 52, 53, 54, 84, 87, 93, 97, and 99.)

[50] Nilesh N Karnik, Jerry M Mendel, and Qilian Liang: *Type-2 fuzzy logic systems*. IEEE transactions on Fuzzy Systems, 7(6):643–658, 1999. (Cited on page 66.)

[51] Erich Peter Klement, Laszlo T. Koczy, and Bernhard Moser: *Are fuzzy systems universal approximators?* International Journal of General Systems, 28(2-3):259–282, 1999. (Cited on pages 36 and 37.)

[52] George J. Klir and Tina A. Folger: *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987, ISBN 0-13-345984-5. (Cited on page 31.)

[53] B. Kosko: *Fuzzy systems as universal approximators*. IEEE Transactions on Computers, 43(11):1329–1333, Nov 1994, ISSN 0018-9340. (Cited on page 36.)

[54] Vladik Kreinovich, Hung T Nguyen, and Yeung Yam: *Fuzzy systems are universal approximators for a smooth function and its derivatives*. 1999. (Cited on page 36.)

[55] J Lamancusa: *Outdoor sound propagation*. noise control, ME, 458:10–19, 2000. (Cited on page 139.)

[56] Werner Van Leekwijck and Etienne E. Kerre: *Defuzzification: criteria and classification*. Fuzzy Sets and Systems, 108(2):159 – 178, 1999, ISSN 0165-0114. http://www.sciencedirect.com/science/article/pii/S0165011497003370. (Cited on pages 41 and 42.)

[57] De chao Li, Zhong ke Shi, and Yong ming Li: *Sufficient and necessary conditions for boolean fuzzy systems as universal approximators*. Information Sciences, 178(2):414 – 424, 2008, ISSN 0020-0255. http://www.sciencedirect.com/science/article/pii/S0020025507003830. (Cited on page 37.)

[58] Qilian Liang and J. M. Mendel: *Interval type-2 fuzzy logic systems: theory and design*. IEEE Transactions on Fuzzy Systems, 8(5):535–550, Oct 2000, ISSN 1063-6706. (Cited on pages 85, 86, 119, and 137.)

[59] Tsung Chih Lin, Han Leih Liu, and Ming Jen Kuo: *Direct adaptive interval type-2 fuzzy control of multivariable nonlinear systems*. Engineering Applications of Artificial Intelligence, 22(3):420 – 430, 2009, ISSN 0952-1976. http://www.sciencedirect.com/science/article/pii/S0952197608001802. (Cited on pages 6, 13, 154, and 160.)

[60] Ondrej Linda and Milos Manic: *Importance sampling based defuzzification for general type-2 fuzzy sets*. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–7. IEEE, 2010. (Cited on page 66.)

[61] Feilong Liu: *An efficient centroid type-reduction strategy for general type-2 fuzzy logic system*. Information Sciences, 178(9):2224 – 2236, 2008, ISSN 0020-0255. http://www.sciencedirect.com/science/article/pii/S0020025507005385. (Cited on pages 49 and 51.)

[62] Feilong Liu and Jerry M Mendel: *An interval approach to fuzzistics for interval type-2 fuzzy sets*. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6. IEEE, 2007. (Cited on page 145.)

[63] Christopher Lynch, Hani Hagras, and Victor Callaghan: *Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines*. In

2006 IEEE International Conference on Fuzzy Systems, pages 1446–1453. IEEE, 2006. (Cited on pages 66 and 172.)

[64] Ming Ma, Abraham Kandel, and Menahem Friedman: *A new approach for defuzzification*. Fuzzy Sets and Systems, 111(3):351–356, 2000. (Cited on page 135.)

[65] James MacQueen *et al.*: *Some methods for classification and analysis of multivariate observations*. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967. (Cited on page 68.)

[66] E. H. Mamdani: *Application of fuzzy algorithms for control of simple dynamic plant*. Electrical Engineers, Proceedings of the Institution of, 121(12):1585–1588, December 1974, ISSN 0020-3270. (Cited on pages 4, 12, and 40.)

[67] Jerry M Mendel: *Fuzzy logic systems for engineering: a tutorial*. Proceedings of the IEEE, 83(3):345–377, 1995. (Cited on page 31.)

[68] Jerry M Mendel: *Advances in type-2 fuzzy sets and systems*. Information sciences, 177(1):84–110, 2007. (Cited on page 66.)

[69] Jerry M Mendel: *Computing with words and its relationships with fuzzistics*. Information Sciences, 177(4):988–1006, 2007. (Cited on pages 144 and 145.)

[70] Jerry M. Mendel: *Comments on alpha-plane representation for type-2 fuzzy sets: Theory and applications*. Trans. Fuz Sys., 18(1):229–230, February 2010, ISSN 1063-6706. http://dx.doi.org/10.1109/TFUZZ.2009.2039368. (Cited on page 49.)

[71] Jerry M Mendel: *On the geometry of join and meet calculations for general type-2 fuzzy sets*. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 2407–2413. IEEE, 2011. (Cited on page 85.)

[72] Jerry M Mendel: *On km algorithms for solving type-2 fuzzy set problems*. IEEE transactions on fuzzy systems, 21(3):426–446, 2013. (Cited on page 60.)

[73] Jerry M Mendel: *General type-2 fuzzy logic systems made simple: a tutorial*. IEEE Transactions on Fuzzy Systems, 22(5):1162–1182, 2014. (Cited on page 85.)

[74] Jerry M Mendel, Robert I John, and Feilong Liu: *Interval type-2 fuzzy logic systems made simple*. IEEE Transactions on Fuzzy Systems, 14(6):808–821, 2006. (Cited on page 137.)

[75] Jerry M Mendel, Feilong Liu, Daoyuan Zhai, *et al.*: *Alpha-plane representation for type-2 fuzzy sets: Theory and applications*. IEEE T. Fuzzy Systems, 17(5):1189–1207, 2009. (Cited on page 66.)

[76] Jerry M Mendel and Hongwei Wu: *Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems*. IEEE Transactions on Fuzzy Systems, 14(6):781–792, 2006. (Cited on pages 116 and 145.)

[77] Jerry M Mendel and Hongwei Wu: *Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 2, inverse problems*. IEEE Transactions on Fuzzy Systems, 15(2):301–308, 2007. (Cited on page 145.)

[78] J.M. Mendel: *Uncertain Rule-based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall PTR, 2001, ISBN 9780130409690. https://books.google.es/books?id=j3NQAAAAMAAJ. (Cited on pages 4, 7, 12, 14, 41, 43, 44, 45, 48, 55, 59, 66, 86, 116, 129, 130, 137, 145, and 146.)

[79] Gerardo M MéNdez and Maria De Los Angeles HernáNdez: *Hybrid learning mechanism for interval a2-c1 type-2 non-singleton type-2 takagi–sugeno–kang fuzzy logic systems*. Information Sciences, 220:149–169, 2013. (Cited on page 66.)

[80] Masaharu Mizumoto and Kokichi Tanaka: *Some properties of fuzzy sets of type 2*. Information and Control, 31(4):312 – 340, 1976, ISSN 0019-9958. http://www.sciencedirect.com/science/article/pii/S0019995876800113. (Cited on pages 4 and 12.)

[81] Taher Niknam, Mohammad Hassan Khooban, Abdollah Kavousifard, and Mohammad Reza Soltanpour: *An optimal type ii fuzzy sliding mode control design for a class of nonlinear systems*. Nonlinear Dynamics, 75(1-2):73–83, 2014. (Cited on page 66.)

[82] W. Pedrycz: *Shadowed sets: representing and processing fuzzy sets*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 28(1):103–109, Feb 1998, ISSN 1083-4419. (Cited on pages 5 and 12.)

[83] Hector Pomares, Ignacio Rojas, Jesús González, Miguel Damas, Begoña Pino, and Alberto Prieto: *Online global learning in direct fuzzy controllers*. IEEE Transactions on Fuzzy Systems, 12(2):218–229, 2004. (Cited on page 116.)

[84] Hector Pomares, Ignacio Rojas, Julio Ortega, Jesus Gonzalez, and Alberto Prieto: *A systematic approach to a self-generating fuzzy rule-table for function approximation*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 30(3):431–447, 2000. (Cited on pages 69 and 116.)

[85] Walter Rudin: *Principles of mathematical analysis*, volume 3. McGraw-Hill New York, 1964. (Cited on page 36.)

[86] Nazanin Sahab and Hani Hagras: *Adaptive non-singleton type-2 fuzzy logic systems: a way forward for handling numerical uncertainties in real world applications*. International Journal of Computers Communications & Control, 6(3):503–529, 2011. (Cited on pages 138 and 139.)

[87] Syed Moshfeq Salaken, Abbas Khosravi, and Saeid Nahavandi: *Modification on enhanced karnik–mendel algorithm*. Expert Systems with Applications, 65:283–291, 2016. (Cited on page 60.)

[88] Roberto Sepúlveda, Óscar Castillo, Patricia Melin, Antonio Rodríguez-Díaz, and Óscar Montiel: *Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic*. Information Sciences, 177(10):2023 – 2048, 2007, ISSN 0020-0255. http://www.sciencedirect.com/science/article/pii/S002002550600332X, Including Special Issue on Hybrid Intelligent Systems. (Cited on pages 6 and 14.)

[89] H. B. Sola, J. Fernandez, H. Hagras, F. Herrera, M. Pagola, and E. Barrenechea: *Interval type-2 fuzzy sets are generalization of interval-valued fuzzy sets: Toward a wider view on their relationship*. IEEE Transactions on Fuzzy Systems, 23(5):1876–1882, Oct 2015, ISSN 1063-6706. (Cited on pages 7, 14, 83, 84, 85, 93, 102, 104, 113, 115, 116, 118, 137, 141, 150, 155, 156, 161, and 162.)

[90] M. Sugeno: *Fuzzy measures and fuzzy integrals: a survey*. In M.M Gupta, G.N Saridis, and B.R Gains (editors): *Fuzzy automata and decision processes*, pages 89–102. North Holland, Amsterdam, 1977. (Cited on page 28.)

[91] Michio Sugeno: *Industrial Applications of Fuzzy Control*. Elsevier Science Inc., New York, NY, USA, 1985, ISBN 0444878297. (Cited on page 40.)

[92] Hooman Tahayori, Andrea GB Tettamanzi, and GD Antoni: *Approximated type-2 fuzzy set operations*. In *2006 IEEE International Conference on Fuzzy Systems*, pages 1910–1917. IEEE, 2006. (Cited on page 85.)

[93] Vicenç Torra: *Hesitant fuzzy sets*. International Journal of Intelligent Systems, 25(6):529–539, 2010, ISSN 1098-111X. http://dx.doi.org/10.1002/int.20418. (Cited on pages 5 and 12.)

[94] Wolfgang Trutschnig, Gil González-Rodríguez, Ana Colubi, and María Ángeles Gil: *A new family of metrics for compact, convex (fuzzy) sets based on a generalized concept of mid and spread*. Information Sciences, 179(23):3964–3972, 2009. (Cited on page 136.)

[95] C. Wagner and H. Hagras: *zslices - towards bridging the gap between interval and general type-2 fuzzy logic*. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pages 489–497, June 2008. (Cited on pages 51, 96, 97, 98, and 100.)

[96] Christian Wagner and Hani Hagras: *Evolving type-2 fuzzy logic controllers for autonomous mobile robots*. In *Analysis and design of intelligent systems using soft computing techniques*, pages 16–25. Springer, 2007. (Cited on page 116.)

[97] Christian Wagner and Hani Hagras: *Toward general type-2 fuzzy logic systems based on zslices*. IEEE transactions on fuzzy systems, 18(4):637–660, 2010. (Cited on page 51.)

[98] Carol L Walker and Elbert A Walker: *The algebra of fuzzy truth values*. Fuzzy Sets and Systems, 149(2):309–347, 2005. (Cited on pages 87, 88, and 92.)

[99] L. X. Wang: *Fuzzy systems are universal approximators*. In *[1992 Proceedings] IEEE International Conference on Fuzzy Systems*, pages 1163–1170, Mar 1992. (Cited on page 36.)

[100] L X Wang and Jerry M Mendel: *Fuzzy basis functions, universal approximation, and orthogonal least-squares learning*. IEEE transactions on Neural Networks, 3(5):807–814, 1992. (Cited on page 36.)

[101] Li Xin Wang: *A Course in Fuzzy Systems and Control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997, ISBN 0-13-540882-2. (Cited on pages 23, 25, 26, 27, 28, 32, 35, and 41.)

[102] Dongrui Wu and Jerry M Mendel: *Enhanced karnik-mendel algorithms for interval type-2 fuzzy sets and systems*. In *NAFIPS 2007-2007 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 184–189. IEEE, 2007. (Cited on page 60.)

[103] Ronald R. Yager: *On a general class of fuzzy connectives*. Fuzzy Sets and Systems, 4(3):235 – 242, 1980, ISSN 0165-0114. http://www.sciencedirect.com/science/article/pii/0165011480900135. (Cited on pages 26, 27, and 28.)

[104] John Yen and Reza Langari: *Fuzzy Logic: Intelligence, Control, and Information*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1999, ISBN 0-13-525817-0. (Cited on pages 3, 4, 11, and 12.)

[105] John Yen and Laing Wang: *An svd-based fuzzy model reduction strategy*. In *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, volume 2, pages 835–841. IEEE, 1996. (Cited on page 72.)

[106] H. Ying: *General interval type-2 mamdani fuzzy systems are universal approximators*. In *NAFIPS 2008 - 2008 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–6, May 2008. (Cited on page 55.)

[107] H. Ying: *Interval type-2 takagi-sugeno fuzzy systems with linear rule consequent are universal approximators*. In *NAFIPS 2009 - 2009 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–5, June 2009. (Cited on page 55.)

[108] Hao Ying: *Sufficient conditions on general fuzzy systems as function approximators*. Automatica, 30(3):521 – 525, 1994, ISSN 0005-1098. http://www.sciencedirect.com/science/article/pii/0005109894901309. (Cited on page 37.)

[109] Hao Ying: *Sufficient conditions on uniform approximation of multivariate functions by general takagi-sugeno fuzzy systems with linear rule consequent*. IEEE Transactions on Systems,

Man, and Cybernetics - Part A: Systems and Humans, 28(4):515–520, Jul 1998, ISSN 1083-4427. (Cited on page 37.)

[110] Hao Ying, Yongsheng Ding, Shaokuan Li, and Shihuang Shao: *Comparison of necessary conditions for typical takagi-sugeno and mamdani fuzzy systems as universal approximators*. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 29(5):508–514, Sep 1999, ISSN 1083-4427. (Cited on page 37.)

[111] F. You and H. Ying: *Interval type-2 boolean fuzzy systems are universal approximators*. In *2010 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–4, July 2010. (Cited on page 55.)

[112] L. A. Zadeh: *Outline of a new approach to the analysis of complex systems and decision processes*. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(1):28–44, Jan 1973, ISSN 0018-9472. (Cited on page 24.)

[113] L. A. Zadeh: *Concept of a linguistic variable and its application to approximate reasoning .1*. Information Sciences, 8(3):199–249, 1975, ISSN 0020-0255. (Cited on pages 3, 4, 7, 11, 12, 15, 20, 23, 45, 51, 66, 84, and 86.)

[114] L. A. Zadeh: *Concept of a linguistic variable and its application to approximate reasoning .2*. Information Sciences, 8(4):301–357, 1975, ISSN 0020-0255. (Cited on pages 3 and 11.)

[115] L. A. Zadeh: *Concept of a linguistic variable and its application to approximate reasoning .3*. Information Sciences, 9(1):43–80, 1975, ISSN 0020-0255. (Cited on pages 3 and 11.)

[116] L. A. Zadeh: *Fuzzy logic and approximate reasoning*. Synthese, 30(3):407–428, 1975, ISSN 1573-0964. http://dx.doi.org/10.1007/BF00485052. (Cited on pages 3 and 11.)

[117] L.A. Zadeh: *Fuzzy sets*. Information and Control, 8(3):338 – 353, 1965, ISSN 0019-9958. http://www.sciencedirect.com/science/article/pii/S001999586590241X. (Cited on pages 3, 11, 20, 21, and 51.)

[118] L.A. Zadeh: *Quantitative fuzzy semantics*. Information Sciences, 3(2):159 – 176, 1971, ISSN 0020-0255. http://www.sciencedirect.com/science/article/pii/S002002557180004X. (Cited on pages 4 and 12.)

[119] L.A. Zadeh: *Fuzzy sets and their applications to cognitive and decision processes*. Academic press rapid manuscript

reproduction. Academic Press, 1975, ISBN 9780127752600. https://books.google.es/books?id=ugPvAAAAMAAJ. (Cited on page 51.)

[120] Ke Zeng, Nai Yao Zhang, and Wen Li Xu: *A comparative study on sufficient conditions for takagi-sugeno fuzzy systems as universal approximators*. IEEE Transactions on Fuzzy Systems, 8(6):773–780, Dec 2000, ISSN 1063-6706. (Cited on page 37.)

[121] Xiao Jun Zeng and M. G. Singh: *Approximation theory of fuzzy systems-mimo case*. IEEE Transactions on Fuzzy Systems, 3(2):219–235, May 1995, ISSN 1063-6706. (Cited on page 37.)

[122] Xiao Jun Zeng and M. G. Singh: *Approximation accuracy analysis of fuzzy systems as function approximators*. IEEE Transactions on Fuzzy Systems, 4(1):44–63, Feb 1996, ISSN 1063-6706. (Cited on page 37.)

[123] Daoyuan Zhai, Minshen Hao, and Jerry M Mendel: *A non-singleton interval type-2 fuzzy logic system for universal image noise removal using quantum-behaved particle swarm optimization*. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 957–964. IEEE, 2011. (Cited on page 116.)

[124] H. J. Zimmermann: *The Extension Principle and Applications*. Springer Netherlands, Dordrecht, 2001, ISBN 978-94-010-0646-0. http://dx.doi.org/10.1007/978-94-010-0646-0_5. (Cited on page 51.)

[125] E. Zitzler: *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, ETH Zurich, Switzerland, 1999. (Cited on page 73.)