# An Investigation Into The Factors Affecting Performance Of Fuzzy Logic Systems

Naisan R. Benatar

Computer Science

University of Nottingham

A thesis submitted for the degree of

*Philosophy Doctor (PhD)*

July 2014

# Abstract

Fuzzy logic is a frequently used solution to control problems, especially when there are elements of human knowledge that may be incorporated into the system. Fuzzy logic comes in several varieties with the most common being based on either type-1 or type-2 fuzzy logic. Modifications to these standard varieties, termed Non-Stationary (NS) and Dual Surface (DS) are also investigated. Each variety allows a certain amount of flexibility in its expression. However, with this increased flexibility (and potentially performance) comes additional resource requirements: either during run time with higher processing and memory requirements; or at design time, with additional parameters requiring selection and optimisation.

There have been several comparisons into the performance obtained from type-1 and type-2 investigating such factors as their internal configuration (such as membership functions as defined by their Footprint of Uncertainty), task difficulty and the environment in which the experiments are performed. However, no studies have been performed incorporating each of these factors with the goal of determining how they impact upon performance. The end goal of this work is the development of a methodology to understand which combination of conditions will cause type-2 control to consistently outperform type-1 based systems. This would enable the rationalisation of moving from a type-1 to a type-2 system, which is currently done without understanding if and how performance will increase with such a move.

This thesis introduces a novel scheme by which several methods of comparing performance are employed to observe how the output and resulting performance levels change as factors including: controller configuration, task difficulty and environmental variability are varied. These methods

are performed over three applications which gradually increase in complexity: a simple tipping example, a more developed simulation based on an autonomous sailing robots application and subsequent real-world experiments, which also involve the autonomous sailing problem. The first method of comparison studies how the rules which fire for a given input set change as the configuration of the fuzzy logic controller is increased. The second comparative technique investigates the control surfaces produced by a selection of fuzzy logic controllers to observe how they change as the internal configuration is changed. Observations such as the smoothing of the transitions between surfaces suggest that controllers with a larger FOU may give a better response. The third method for comparison is developed in which outputs from a controller operating in a simulated environment are compared to an ideal value, giving a single numeric output with which comparisons can be made.

It was found that there are situations in which type-2 based fuzzy control outperforms type-1. However, these are found to be less common than expected. It is determined that this may be due to the simplicity of some of our case studies environments (especially the tipping example), where there may not be enough scope for large improvements to become apparent. These findings lay ground for future work in which (i) more developed and complex applications and (ii) a more tuned fuzzy system should be investigated to find if this will result in more obvious differences between configurations.

# Publications

This section presents a list of publications formed as part of the research work for this thesis.

1. Naisan Benatar, Uwe Aickelin and Jonathan M. Garibaldi. **Performance Measurement Under Increasing Environmental Uncertainty In The Context of Interval Type-2 Fuzzy Logic Based Robotic Sailing**. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2013.

2. Naisan Benatar , Uwe Aickelin and Jonathan M. Garibaldi. **An investigation into the relationship between type-2 FOU size and environmental uncertainty in robotic control**. In IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2012. Pg 1-8

3. Naisan Benatar, Uwe Aickelin and Jonathan M. Garibaldi. **A Comparison of Non-stationary , Type-2 and Dual Surface Fuzzy Control**. In IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2011. Pg 1193-1200

4. Naisan Benatar, Omer Qadir, Jenny Owen, Paul Baxter, Mark Neal. **P-Controller as an Expert System for Manoeuvring Rudderless Sail Boats**. The 9th Annual Workshop on Computational Intelligence (UKCI 2009), 2009.

# Acknowledgements

A special thanks to my family: Yasmin and Peter Jones, and my brothers André and Dominic for their help and understanding throughout the years.

I would like to sincerely thank my supervisors, Professors Uwe Aickelin and Jonathan M. Garibaldi for their advice, patience and support during the process.

I would also like to express my appreciation for Dr Grazziela Figueredo, who spent many nights proof reading this thesis and provided many hours of interesting conversations.

Additionally I would like to thank my examiners Professor Robert John and Dr Mark Neal whose comments and suggestions dramatically helped the quality of this work.

# Contents

# 1

# Introduction

Fuzzy logic (FL) , introduced by Zadeh [107] is a common technique that is used as a solution to many types of task, including decision making and control problems. There are several reasons for this popularity including: the way it can be used to mirror the human decision making process [109]; the simplicity of implementation with a well-defined mathematical background; and finally, it has potential to provide better performance than techniques such as Proportional Integral Derivative (PID) control as described in articles such as those by Li *et al.* [53].

Fuzzy logic controllers can be broken down into several varieties as described in detail by Celikyilmaz and T$\ddot{u}$rksen [21]. The two most commonly used varieties are termed 'type-1' and 'type-2' with the former being the simpler of the two and therefore the one which has been more widely adopted, due its lower resource requirements — an important consideration for robotic systems. The use of the latter is increasing as the resources required fall more easily within the envelope provided by modern embedded hardware systems. However, being able to determine when each type is most suitable is still a difficult task as evidenced by works performed by Cao *et al.* [11].

Defining a systematic way of selecting between type-1 and type-2 fuzzy logic is the main goal of this thesis. While early work by Braae and Rutherford [16] investigates initial parameter selection in early fuzzy systems, many works utilise Genetic Algorithms (GAs) for selection of optimal parameters such as in the work of Martinez-Soto [63]. The use of GAs means that a general design process for type-2 systems is not important— it is simply evolved for each application, with the downside being the significant off-line processing requirement. The selection of the type of fuzzy system requires understanding the effect of multiple factors. Firstly, the different types of fuzzy logic control available in conjunction with their internal configurations such as the Footprint of Uncertainty (FOU) of type-2 controllers. Secondly, the environment in which experiments are performed can be varied by increasing both the task difficulty and the sources of variation present. Comparisons between different combinations of the above factors can be performed using different approaches. However, which of these methods gives the best picture of performance is still an unanswered question within the field.

There are several deviations from the standard model of type-1 and type-2 fuzzy logic inference systems. In this thesis two are selected for study in addition to the standard types. The first of these is termed *Non-Stationary* fuzzy logic (NS) in which small deviations in the membership functions are introduced for each iteration of the controller. The second variety is termed *Dual surface* (DS) fuzzy logic, which is based on interval type-2 control with the addition of a simple algorithm for selecting between three possible output values — the upper and lower values obtained from the output interval and the mean of the two values. These types of controller have been selected firstly for their simplicity to implement and secondly because it is believed that they may be able to give a level of performance improvement over the standard models for a minimal increase in complexity.

## 1.1 Motivation

As discussed in Chapter 2 (Page 9), there does not currently exist any method to determine which fuzzy logic controller type is the most suitable for a given situation. The first step towards developing a framework that can decide on the suitability of each fuzzy logic controller depending on the problem characteristics would be to show that there exists situations in which type-1 and the more sophisticated varieties of fuzzy logic (such as type-2, NS and DS) give significantly different levels of performance, therefore showing that one would be preferable than the other in these circumstances. This leads to an investigation that focusses on the differences found between each fuzzy controller output, under what circumstances they are most apparent, and how this can be generalised. This is therefore, the main focus of our research in this work.

## 1.2 Aims, Objectives and Research Questions

The main research question of this thesis is to determine 'what combination of factors are necessary for more sophisticated fuzzy controller types such as interval type-2 to consistently out perform type-1 fuzzy control'. In order to attempt to answer this question the following are therefore presented as the aims of this thesis:

1. To show that variations on standard type-1 control, specifically interval type-2, DS, and NS fuzzy control can provide significantly different levels of performance over type-1 fuzzy control.

2. To study how performance changes as the environment is made more or less sophisticated, by altering aspects such as variation in the environment, and the difficulty of the attempted task.

3. To investigate how the internal configuration of a given controller (referred to as the FOU size for interval type-2 controllers) changes the level of performance of the more sophisticated fuzzy systems in comparison with type-1 based configuration.

4. To identify the point at which type-2 fuzzy logic consistently outperforms type-1 and try to define as many aspects as possible which cause this to occur. For example it may be observed that high levels of environmental variation will always lead to a type-2 controller outperforming type-1.

These aims leads directly to the setting the following research objectives:

To develop a means by which meaningful comparisons between different experimental scenarios can be performed.

To determine the effect of variation and task difficulty upon the performance of fuzzy logic controllers.

> To determine when and indeed if type-2, NS or DS fuzzy logic outperform type-1 and under what experiment scenarios make this is likely to occur.

If these objectives are met then it should be possible to make predictions about what circumstances are most appropriate for the implementation of more sophisticated (and therefore more computationally expensive to implement) types of fuzzy logic. However, the overall end goal can be defined as a sliding scale with a more successful outcome being a more precise set of circumstances for when type-2 fuzzy logic outperforms type-1. Even if no differences are found, as long as the methodology covers a sufficiently wide scope, the research objectives can still be met, e.g. it may be that there are no circumstances where type-2 control outperform type-1, and this in itself would be considered an acceptable, though unsatisfactory result.

In order to fulfil our objectives, two experimental applications are used for the comparative work in this thesis. The first is a very simple artificial problem, in which the goal is to determine the appropriate tip for a given set of food and service levels in a restaurant. From the results found it is felt that this application is too simple to be able to fully realise the research objectives, although it provides a starting point on which additional work can be based. The results obtained and the process of developing comparative methodologies lead to the introduction of a robotic sailing problem which has significantly more scope for variation. In addition, it allows both simulation and real-world experiments to be explored. In general, this second case study was found to give more interesting results.

## 1.3 Thesis Findings and Contributions

This thesis develops several approaches to comparing the performance of type-1, type-2, DS and NS varieties of fuzzy logic using three case studies: the tipping problem, the simulated boat environment and the real-world boat environment. Two of the approaches for performing comparisons look directly at a selection of fuzzy logic controllers and how their outputs change across the entire space of possible input combinations. These approaches by themselves give a broad view of the potential performance a controller may be expected to achieve in relation to each other. However, without putting the controller into a complete environment, it is difficult to provide an absolute measure of performance. This issue is therefore addressed by the use of Root Mean Square Error (RMSE) as a measure of performance applied in a large variety of experimental scenarios, across a range of application areas, both simulated and real world.

Using these three approaches: the comparison of which rules fire, the inspection of the control surfaces and the comparison of RMSE values are the main means by which controllers and their environments are compared, allowing the different factors including: the environmental variation, the internal configuration of the controller and the task difficulty to be investigated. Both the internal configuration of the controllers under test and the environment in which they operate were varied to observe how each factor alters the overall performance as defined by the RMSE value obtained. It was found that when the task is simple (such as the Tipping application) with no variations being introduced, differences between all of the controllers under test is seen to be minimal, with none of the approaches being able to consistently differentiate between the controllers under test. Once the application and the controllers under test are made more sophisticated, however, changes start to become more evident and the type-2 fuzzy logic based controller, in particular, starts to show improved performance over its type-1 based counterparts.

The total number of differences observed were not as frequent it may have been expected. Several reasons for this have been identified: Firstly the design of the controllers under test were not optimal and indeed were not tuned, which may have led to all controllers being similarly limited in the performance they could achieve. Secondly, the effect of environmental set-up, including both the task difficulty and introduced variation upon the system was perhaps not considered carefully enough, potentially not containing the scope required to show the desired changes in performance. Finally, the use of RMSE as the sole means of performance comparison may have limited the ability to contrast run where the environmental variation changed such as in the simulated sailing application. For example, a RMSE value of 5 in a difficult experimental scenario may be considered good, but bad in an easier scenario, and this is not captured by the RMSE value alone.

Overall the following contributions can be drawn from the research made in this thesis:

- In the most simple applications and environments, the majority of fuzzy logic configurations do not show significant differences in performance, even when several different methods of comparison are used.
- As the environment and experimental design is developed and made more complex, such as those described in Chapter 5 (Page 95), the ease with which different fuzzy logic controller can be differentiated increases.
- Type-2 fuzzy logic can provide an improvement over the less sophisticated type-1 variety for the case studies performed. However, time must be spent determining the correct parameter values, such as the FOU size. The use of type-2 cannot be recommended in simple applications and even in more developed areas a suitably tuned type-1 may well give equal performance.

## 1.4 Thesis Organisation

Chapter 2 (Page 9) provides the necessary background knowledge and a review of the current relevant literature. In addition, it presents a critical discussion of fuzzy logic in variable and complex environments, their interactions, current studies into the area, and alternative methods for controlling robots in such environments. Finally, the specific gap in the literature is established along with reasoning as to why it is a suitable topic for study.

The first experimental work is described in Chapter 3 (Page 44), in which a simple fuzzy controller is introduced and is used to develop the ideas of this thesis. Firstly an investigation is presented in which the rule base, which rules fire and how this changes with the internal configuration is studied. This tipping controller is then used in a simple control-like experiment to observe if this controller shows differences in performance as measured by the RMSE value, and if so how these differences manifest themselves as the FOU size is changed. This method of comparison is then further developed to include comparing the output of the fuzzy logic controllers to 'ideal' values and finally by injecting a source of variation into the experiment to observe how this affects the RMSE values obtained. Overall the results in this chapter are not as good as expected, with few differences found in the majority of experiments. This is believed to be due to the simplicity of the problem and is addressed by the selection and development of the sailing boat controller application in later chapters.

This is followed by Chapter 4 (Page 73), in which the methodology used in this thesis is described. The hardware and software used, as well as the design and implementation of the fuzzy logic systems under investigation are described. Some common experimental design is also introduced, including the justification as to why certain design decisions have been made.

Chapter 5 (Page 95) studies identified shortcomings of previous work by increasing the sophistication and complexity of both the controllers under test, and the environments in which they operate. The chapter starts with two investigations in which the aim is to study how the different controllers vary their outputs, including the rules which fire and the output of the controller as given by the shape of the control surface. This is followed by Section 5.4 (Page 107), in which a simulation environment is used to study a wide variety of controller types and configurations, and to investigate the association between FOU and performance. This is followed by Section 5.5 (Page 115) in which the methodology used to make the comparisons is further developed and the subject of study is more focused more upon type-1 and type-2 comparison. The chapter is concluded with a discussion regarding the findings, strengths and weaknesses observed from the results obtained. The results found overall are much more encouraging, with several scenarios identified in which the more sophisticated controllers exhibit improved behaviours, such as smoother control surfaces, which in turn lead to improved performance, as shown by the experiments in which RMSE is used to characterise the performance.

Chapter 6 (Page 139) introduces real-world data, as opposed to simulated data, for the investigations, with the intention of introducing significantly more environmental variation and task difficulty into the experiments. The collected data is analysed in a similar manner as the simulation data. The reasons for these observations made is discussed. In addition, changes to the methodology are proposed to overcome the problems found in this experimental set-up. The use of a real world application has the desired effect of introducing much more variation, resulting in much wider intervals in the results, however other potential issues such as the lack of tuning of the fuzzy controllers start to become apparent.

Chapter 7 (Page 158) presents a summary and discussion of the research conducted in this thesis. An analysis of strengths and weaknesses with the methodology is conducted. Moreover, the contributions and their utility in assisting the field advance are presented. Finally, potential improvements that may be pursued in future work are discussed.

# 2

# Literature Review

## 2.1 Introduction

The focus of this thesis, as discussed in the previous chapter, is fuzzy logic and what factors can change performance. Different combinations of factors including controller configuration, environmental variation and task difficulty are put under test with the aim of trying to find answer to questions such as 'What combination of factors are necessary for more sophisticated fuzzy controller types (such as interval type-2) to consistently out perform type-1 fuzzy control?'. Changes in the parameters used to define these systems are investigated to see how they can be used to change the levels of performance found. In this chapter fuzzy logic as a research topic is introduced, together with a discussion of its composition, including the operation and applications of fuzzy logic based systems. The literature discussing the effects of variable environments upon fuzzy logic-based systems is also analysed, and the specific aims of this thesis are then discussed.

This chapter is organised as follows: Section 2.2 introduces the basics of fuzzy logic theory including the mathematical background of fuzzy logic. Section 2.3 discusses fuzzy inference systems, and is followed by Section 2.6 which presents fuzzy logic as a solution to control problems as well as various aspects of control theory. A discussion of dynamic environments and how they can affect these applications is presented in Section 2.7 and is followed by a statement of this thesis' research focus, possible research question in Section 2.8 as well as the gaps in the literature which make these questions interesting. Finally in Section 2.9 the chapter is summarised.

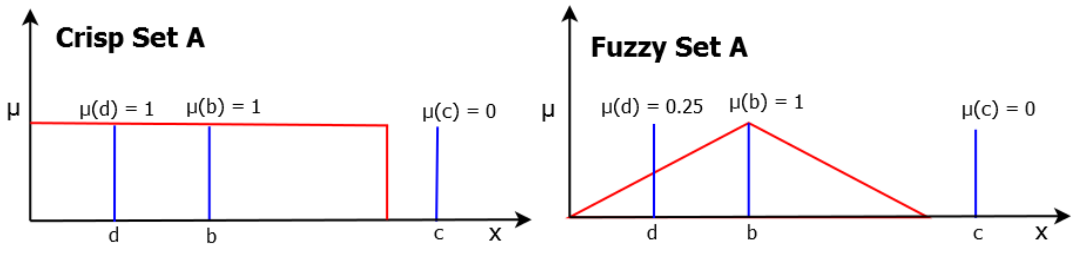
## 2.2 Fuzzy Logic Theory

Fuzzy set theory was initially described by Zadeh in 1965 [107]. It is, from a mathematical standpoint, a generalisation of crisp set theory. In this paper, the mathematical definition of a fuzzy set is given along with the fuzzy operations required to manipulate these sets, such as the union, intersection and complement. Such work is required for fuzzy logic to be used effectively in real-world applications.

In a standard crisp set, a given input $x$, has the option of being either member or a non-member of a given set — it has therefore, a binary membership. A complete description of crisp logic, its operations and so forth can be found in works such as those by Thomas [45] or Leondes [52]. Figure 2.1(a) shows a crisp set $A$ and two inputs $b$ and $c$. These inputs represent member and non-member values respectively, with no other values possible. This is the standard method for classification using such sets. However, this approach has some shortcomings. For example, when trying to classify heights of people — and trying to find those who are tall — using crisp sets necessitates a cut-off point of, for example 200cm. This would lead to someone of height 199cm not being classified as tall, even though their height is very close to the cut-off value.

When a crisp set is turned into a type-1 fuzzy set, as shown in Figure 2.1(b), the two inputs $b$ and $c$ are still valid. However, their membership now represents two ends of an axis — giving them membership values 1 and 0. These values indicate complete and no membership to the set respectively. Additionally, input $d$ is also valid and equates to the membership value 0.25, a value not possible using crisp sets. Using the above height example, a gradual transition can be established, allowing a better specification to be defined. If the same height example is used as described above, a height of 199cm would be given a very large membership value such as 0.99, showing that it is still considered a degree of tall.

The line in red in Figure 2.1(b) describes a mathematical function, termed the **membership function** (MF). The MF maps an input value (a crisp number) into a




(a) Crisp set

(b) Fuzzy set

**Figure 2.1:** Crisp and Fuzzy sets. The red line in Fig 2.1(b) is known as the **membership function**

membership value. The most common type of fuzzy set is the type-1 set, in which this function is a single line in the x-y plane. Zadeh [108] describes fuzzy types greater than 1 in which the membership functions are of higher dimensions, giving rise to type-2 fuzzy logic. Type-2 fuzzy logic is the most common in the literature (aside from type-1), and although type-n systems are also described in the publication, there are no significant applications of such types at present. Both type-1 and type-2 fuzzy logic are active research areas, and as such, both therefore be described and discussed in this chapter. Fuzzy logic can be broken down into the following active research areas:

**Theory** The mathematical theory is the research area which formally defines what constitutes a fuzzy set including both type-1, type-2 and more non-standard varieties. In addition, it establishes how operations on these sets can be represented using formal mathematical notation. Works such as the one by Sadeghian *et al.* [77] provide insights in to recent development in this area.

**Optimisation and new methods** Current methods for calculations using fuzzy sets can often be optimised to improve real-world performance. Furthermore, they are also employed in the development of entire new methods for performing the same operation. Defuzzification and type-reduction are two areas where this is prevalent. Luhandjula [60] and Torshizi *et al.* [94] present reviews which show the recent activity in these fields.

**Applications** Fuzzy logic has already been applied to a great number of problems and application domains, such as control of robotic vehicles as by Doitsidis *et al.* [29], filtering and classification (Mendel [68]) as well as numerous medical decision making systems such those described by Schuha [84]. With new methods and hardware advances becoming available, this number continues to expand rapidly.

**Comparative works** The many different permutations and varieties of fuzzy systems often lead to studies into which performs better in a given situation. A common example of this kind of work is type-1 *versus* type-2 control studies, such as those conducted by Czarez *et al.* [20], in which type-1 and type-2 controllers are evaluated and the type-2 is found to work operate better under uncertain conditions. Farooq *et al.* [31] present a similar study in which mobile robots are controlled by using both type-1 and type-2 fuzzy types in which it is found that type-2 control once again gives superior performance over type-1 based control.

## 2.3 Background of Fuzzy Logic

### 2.3.1 Crisp Sets

Fuzzy logic, the subject of this thesis, is based on standard set theory — a common mathematical concept, briefly outlined above, that is frequently used in fields within computer science such as databases and compiler technologies. Set theory uses simple binary concepts — an object $x$ is either 'in' or 'not in' a given set $A$. Given the object $x$, it can be formally specified that it is, or is not, a member of set $A$, as shown in equations 2.1 and 2.2 respectively. The contents of set $A$ is stated using the format shown in Equation 2.3. For a more complete examination of set theory, works such as Thomas [45] or Fraenkel *et al.* [35] should be consulted.

$$x \in A \tag{2.1}$$

$$x \notin A \tag{2.2}$$

$$A = a1, a2, ...a_x \tag{2.3}$$

.

Given two sets, $A$ and $B$, if all of the elements of $A$ are also in $B$ then the sets are *equal*. If set $B$ contains all of the elements of $A$ with one or more additional items, then set $A$ is a *proper subset* of set $B$. These concepts are formally specified in equations 2.4 to 2.5:

$$A = B \tag{2.4}$$

$$B \subset A \tag{2.5}$$

Sets require a means by which elements can be judged to be a member or a non-member. This can be defined mathematically using a binary function, that is, a function which returns one of two possible values for a given input. Generally in set theory, this is function is denoted by the letter $\mu$ and can be stated mathematically as in equations 2.6 and 2.7 and is termed a *membership function*.

$$\mu_A(x) = 1 \; if \; only \; if x \in A.0 \; if \; only \; if \; x \notin A. \tag{2.6}$$

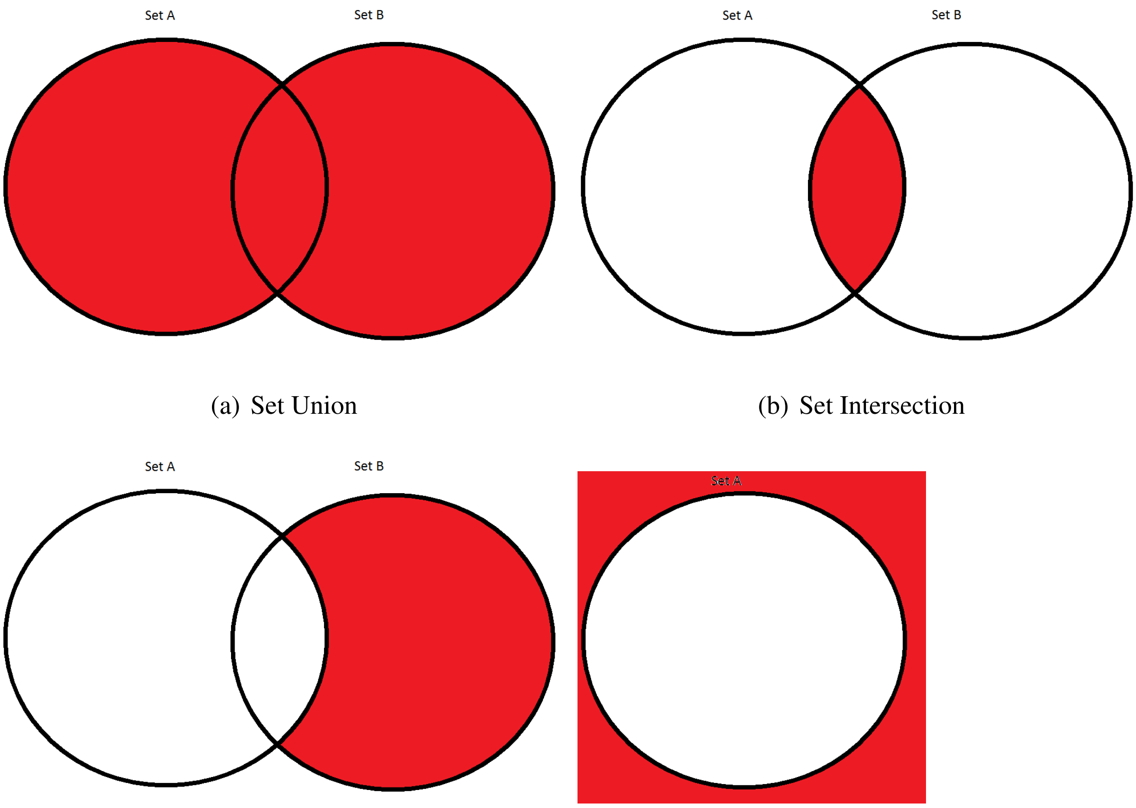
$$\mu_A : U- > 0, 1 \tag{2.7}$$

### 2.3.2 Crisp Set Operations

Given the basic definitions in the previous section, four operations between sets can be defined termed: Union, Intersection, Difference and Complement. These are the basic set operations in much the same way that addition, subtraction, multiplication and division are the basis of operations between natural numbers.

Figure 2.2 gives a graphical representation of each of the operations. $A$ and $B$ in figures 2.2(a) 2.2(b) and 2.2(c) are crisp sets. The shaded area indicates the output of the indicated operation upon the inputs — that is what members would be the output set, a mathematical backgrounds of these operations can be found in Stoll [93].

These crisp set operations can be adapted for use with fuzzy sets of all varieties. There are three main varieties of fuzzy set in common usage today: type-1, interval type-2 and general type-2, with only the first two being used in the experimental works of this thesis and therefore, only the operations of these types of set are described here. General type-2 fuzzy logic was excluded due to the processing requirements being too great for CPU on-board the robot used in real-world experiments in Chapter 6. Section 2.4 describes the operations and type specific parts related to type-1 fuzzy logic, while interval type-2 theory is set out in Section 2.5.




(a) Set Union

(b) Set Intersection

(c) Set Difference

(d) Set Complement

**Figure 2.2:** Venn Diagrams of Results of Basic Crisp Set Operations

| Input 1 | Input 2 | Result |
|---|---|---|
| | | |
| **AND** | | |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| | | |
| **OR** | | |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |
| | | |
| **NOT** | | |
| | | |
| | 1 | 0 |
| | 0 | 1 |

**Table 2.1:** Truth tables for AND, OR and NOT boolean operators

### 2.3.3 Boolean Logic

Boolean logic defines a collection of operators that work with binary valued items, that is, values which have one of two possible values. These operators take one or two binary values as inputs and give a single binary value as output. Because of the nature of the input and output values used for each operator, a clear and easy way to express the results of each operator is the use of *truth tables*. Truth tables for AND, OR, NOT and EQUIVALENCE operators are shown in Table **??**. Textbooks such as Enderton [30] or Arnold [4] provide a more complete overview of boolean logic.

These operators are often used as connectives when multiple comparisons are required in a given statement such as 'IF A is True AND B is NOT True THEN Perform Action C' and 'IF A is True OR B is True THEN Perform action D'. These sorts of constructs are commonly used throughout control applications, such as when there are multiple inputs to a system and each needs evaluating before a decision can be made.

These types of construct are adapted and used within the field of fuzzy logic where they are used to connect multiple antecedents together to form a single rule within an inference systems rule base. However as the inputs are no longer binary values but fuzzy sets, the operators themselves must also be adapted. In order to maintain clarity, they are also given distinct terms: AND (Union), OR (Intersection), and NOT (Complement).

## 2.4 Type-1 Fuzzy Maths

As discussed in the main body of this thesis, the fundamental element of fuzzy set theory is the *fuzzy set*. This is a modification to standard crisp set theory and can be defined by Equation 2.4. Type-1 fuzzy sets can be either continuous or discrete, as defined by equations 2.9 and 2.11. This leads to being able to define a fuzzy set as the tuple shown in Equation 2.9. Zimmerman [111] and Klir and Yan [48] both provide a more complete overview of this and surrounding mathematical background for further reference.

> Let X be equal to a non-empty set. Within the set X, a fuzzy set A can be characterised by a membership function of the form $\mu_A$.

$$\mu_A : X \rightarrow [0, 1] \tag{2.8}$$

$$A = \{(u, \mu_A(u)) | u \in X\} \tag{2.9}$$

$$\int \mu_{\tilde{A}}(x)/x \tag{2.10}$$

$$\sum_{i=1..n} \mu_i/x_i \tag{2.11}$$

### 2.4.1 Type-1 Fuzzy Inference Systems

A standard type-1 fuzzy inference systems is a system that uses fuzzy sets in a more developed setting instead of just using fuzzy sets alone. Cox [28] provides a fundamental look at fuzzy inference systems and gives methodologies for their design. Its general structure is shown in figure 2.3:

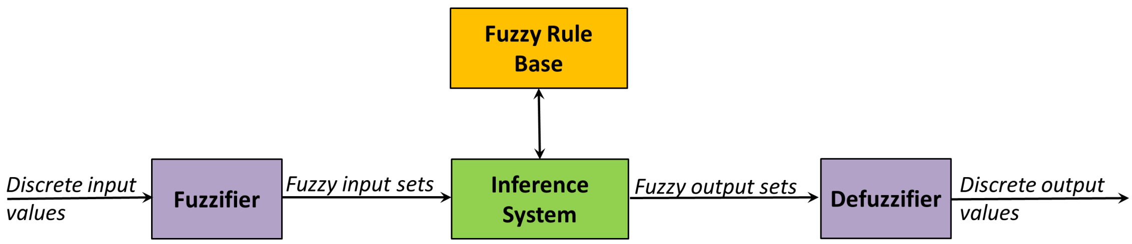



**Figure 2.3:** Type 1 fuzzy inference system.

**Fuzzifier** - This component maps each input value to a membership value of one or more fuzzy sets, based on the defined membership functions.

**Inference system and rule base** - Output sets are generated based on the inputs sets and the rule base. The rule base is a set of linguistic rules and are used to specify which output sets should be triggered with a given set of inputs.

**Defuzzifier** - The defuzzifier module calculates an output value from the output set that was calculated in the previous element.

The background and operation of each of these functional areas in the context of type-1 fuzzy inference systems is the subject of this section.

### 2.4.2 Fuzzification

In order to make use of fuzzy sets, the first step is the fuzzification of a crisp input into a membership grades of the set, that determines how much the given input is part of the set in question. This is achieved by the evaluation of the associated membership function for the fuzzy set in the system. This means that the membership function given by $\mu_A(x)$, can be thought of as a mapping of the input $x$ to a degree of membership to the fuzzy set A, as shown in Figure 2.3. Sinha and Dougherty [89] discuss details of fuzzification and how membership functions can be generated to provide fuzzifiers that provide desired outputs.

### 2.4.3 Rules and Inferencing

In each fuzzy inference system there is a set of rules, known as a *rulebase*. Each rule is of the form 2.4.3. There are several methods for performing the inference process with Takagi-Sugeno and Mamdani being the most common. In this thesis, as in most control applications, the Mamdani method is used. This provides a mapping of inputs to outputs using Equation 2.12.

IF A is XXX and B is YYY then OUTPUT is ZZZ

$$\phi[\mu_A(x), \mu_B(y)] \equiv \mu_A(x) \wedge \mu_B(y) \tag{2.12}$$

Where $\mu_A(x)$ is the input membership function and $\mu_B(y)$ is the output function). The result is the output fuzzy set which is the product of the input and output fuzzy variables. These are then passed into the defuzzifier in order to generate crisp outputs as outlined in the next section.

Fuzzy rules can be generated from input data as shown by Nozaki [75], Wang and Mendel [100], and Hong and Lee [43], each of who generate fuzzy rules from training data or other input data. They can also be designed using standard software development and optimisation techniques or derived from expert knowledge.

### 2.4.4 Fuzzy Operators

The Union and Intersection operators are used to connect multiple antecedents in a single rule. They are equivalent to the boolean operators AND and OR used in classical logic. In the inference system in this thesis only the union operator for connecting the two inputs as described in Section 2.5.2. Cordn *et al.* [26] discuss the use of different fuzzy operators in the design of fuzzy controllers.

The Union operator is most commonly implemented using min() of the sets. This is formally defined in Equation 2.13:

$$A \bigcup B = \{x : x \in or\ x \in B\} \tag{2.13}$$

The intersection operator is most commonly implemented using max() and is shown mathematically in equation 2.14:

$$A \bigcap B = \{x : x \in and\ x \in B\} \tag{2.14}$$

#### Defuzzification

Once the inference system has calculated output fuzzy sets, the process of defuzzification is required in order to calculate inputs suitable for use in the specific applications. While there is a vast number of methods for defuzzification of type-1 fuzzy sets, only the Centre of Gravity (COG), one of the most common methods, will be considered in this section. For a more complete look at methods for defuzzification authors such as Hellendoorn and Thomas [42] or Leekwijck and Kerre [51] who present overviews and comparisons of differing methods of defuzzification. While COG does not always perform the best, its simplicity to implement often makes it the technique of choice for the majority of situations.

One of the most common means of defuzzification in robotic applications, as used in this thesis, is COG defuzzification known for its accuracy and speed. This process is defined in Equation 2.15:

$$x^{\star} = \frac{\int \mu_i(x) x dx}{\int \mu_i(x) dx} \tag{2.15}$$

Where $x^{\star}$ is the output value that will be used by the system. $\mu_i(x)$ are the aggregated membership functions and x is the output variable.

## 2.5 Interval Type-2 Fuzzy Sets

Interval type-2 fuzzy logic, as introduced by Karnik and Mendel [47] or alternatively in Liang and Mendel [54] is a restricted form of general type-2 fuzzy logic. Its basis is the interval type-2 fuzzy set. An interval type-2 fuzzy set, $A$, is fully defined by a type-2 membership function, given by Equation 2.16 and is graphically represented in Figure 2.4. Nieminen [74] discusses the algebraic structure of type-2 fuzzy sets in more detail.

$$A = \{((x,u), \mu_A(x,u))|\forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{2.16}$$

$$0 \leq \mu_A(x,u) \leq 1 \tag{2.17}$$

The difference between general and interval type-2 fuzzy logic is the nature of the *secondary membership function*. In a general type-2 system, the secondary membership function is a type-1 fuzzy set. In contrast, interval type-2 uses a binary membership, meaning this secondary membership function is an interval. This reduction of the secondary set to a binary relation reduces the resource required for processing such sets significantly. In this case, the 3D nature of the general type-2 set can be reduced into two membership functions within a 2D space, termed the *upper membership function* and the *lower membership function* and is shown in Figure 2.4, with the area bounded by these two functions termed the *Footprint of Uncertainty* (FOU). Unless otherwise specified in this thesis, all of the type-2 based work refers to interval type-2 fuzzy logic and not the general variety.

### 2.5.1 Fuzzification and Membership functions

Fuzzification of interval type-2 sets uses the upper and lower membership functions ($\overline{\mu}(x)$ and $\underline{\mu}(x)$) and calculates an interval, which is then used in the inference procedure described in the section below. Membership functions themselves are functions

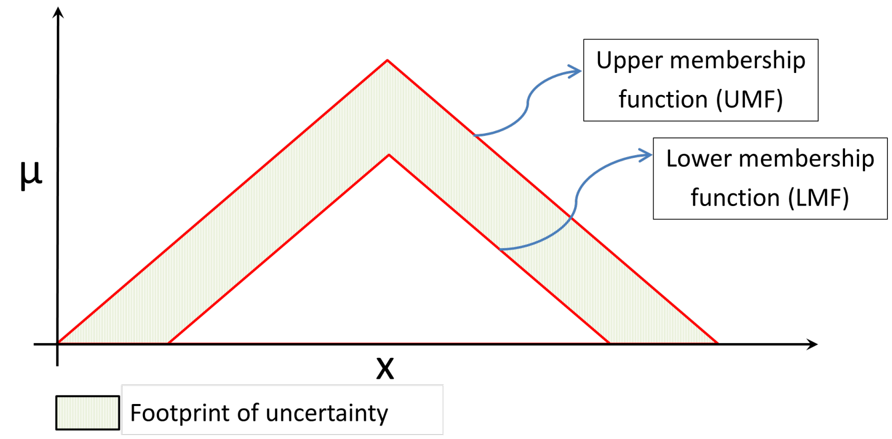



**Figure 2.4:** An Interval type-2 fuzzy set, showing the upper and lower membership functions the define the Footprint of Uncertainty

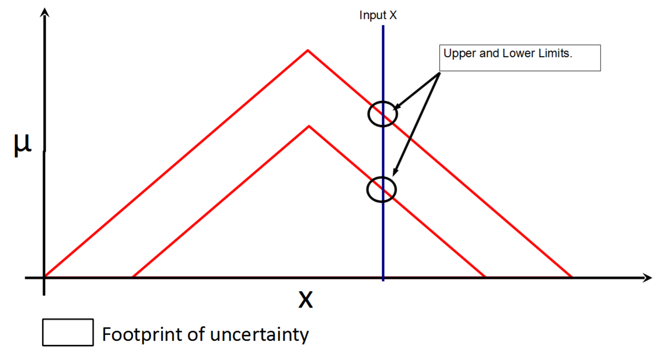



**Figure 2.5:** Interval Type-2 fuzzification — The result is an interval

bounded within the universe of discourse of the given input variable generally represented as in Equation 2.18. Common function shapes used are triangular, trapezoidal and Gaussian. Commonly membership functions are determined by picking a shape of function and then tuning it until performance requirements are satisfied but many other methods exist as described by Bouchon-Meunier *et al.* [10] and Chiu [23].

$$X_A = \mu_A(x) \tag{2.18}$$

Evaluating the upper and lower membership functions for each fuzzy set results in an interval for each fuzzy set, which is then passed into the inference module as described below.

### 2.5.2 Rules and Operators

Once the membership intervals for each of the fuzzy sets has been calculated during the fuzzification stage as described above, the rule base must be evaluated to produce a *firing interval* for each rule defined in the system.

In a standard interval type-2 fuzzy inference system, the rule base consists of a set of rules $R$, each of the form:

$$\text{IF } (i_1 \text{ IS } y)[\text{ CONNECTIVE }](i_2 \text{ IS } z) \text{ THEN } (j_n) \text{ IS } z \tag{2.19}$$

Where $i$ is an input variable, $y$ an associated fuzzy set, CONNECTIVE is a fuzzy operator such as Union or Intersection. $j_1$ is the output variable with its associated fuzzy output set, $z$.

For each rule defined in the system (i.e. $\forall rinR$), the firing strength (again, represented by an interval) is calculated using Equation 2.20, resulting in a set of tuples, $R_i$, each containing an upper and lower value.

$$R_i = [\underline{\mu_A}(X) \star \underline{\mu_B}(Y), \overline{\mu_A}(X) \star \overline{\mu_B}(Y)] \tag{2.20}$$

$R_i$ represents a firing strength for a specific rule. The firing strengths for all rules in the system are stored in a set $F$.

$\star$ indicates the operator used, specified in the design on the specific fuzzy inference system. This may include such operators including *max*, *min* or *product*. These operators are described in more detail below.

### Union

The union fuzzy operator is the equivalent of the OR operation in boolean algebra and is defined by Equation 2.21. Union is generally implemented using the max() function, which can be observed graphically in Figure 2.6.

$$A \cup B = 1/[\underline{\mu}_A(x) \vee \underline{\mu}_B(x), \overline{\mu}_A(x) \vee \overline{\mu}_B(x)] \quad \forall x \in X \tag{2.21}$$

As above, $\underline{\mu}_A(x)$ indicates the lower interval value, $\overline{\mu}_A(x)$ the upper interval value, while $\vee$ is the max function.

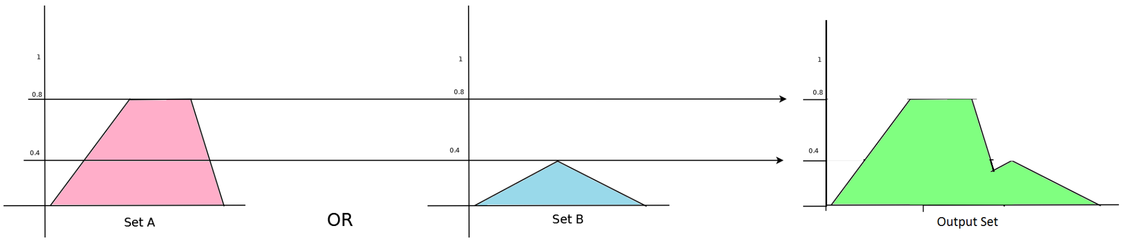



**Figure 2.6:** Union of 2 interval type-2 sets using OR operator. Output set shown in green.

### Intersection

The union fuzzy operator is the equivalent of the AND operation in boolean algebra. It is defined mathematically by Equation 2.22. Intersection is generally implemented using the min() function — as it can be observed graphically in Figure 2.7.

$$A \cap B = 1/[\underline{\mu}_A(x) \wedge \underline{\mu}_B(x), \overline{\mu}_A(x) \wedge \overline{\mu}_B(x)] \quad \forall x \in X \tag{2.22}$$

As above, $\underline{\mu}_A(x)$ indicates the lower interval value, $\overline{\mu}_A(x)$ the upper interval value, while $\vee$ is the min operator.

## 2.5.3 Type-Reduction and Defuzzification

In the case of interval type-2, by far the most common means of obtaining an output from the inference system is the Karnik and Mendel iterative procedure (K&M). As the name implies it was jointly developed by Karnik and Mendel [103] and Liu [58].

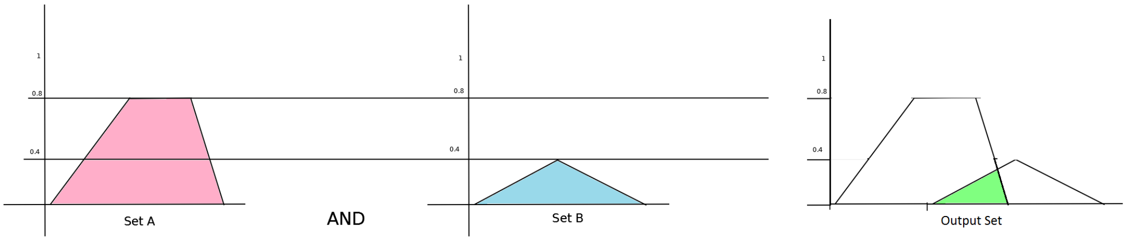



**Figure 2.7:** Intersection of 2 interval type-2 sets using AND operator. Output set shown in Green.

This is combines type reduction and defuzzification into a single process and results in the calculation of upper ($y_r$) and lower ($y_l$) values, the process is described in detail here:

1. Sort the set of output values ($y$). For calculating $y_r$ the upper values $\overline{y}$ are used and for calculating $y_l$ the lower values $\underline{y}$. The set of firing intervals, $f$, should be sorted to maintain the same indices as the set sorted.
2. Initialise the list $f$ to contain the mean of each firing interval strength, i.e Equation 2.23 and calculate the result of Equation 2.24 with the new values.
3. Iterate through the set of output values $y$, located the index in which the following condition is valid $\overline{y}^k \leq y \leq \overline{y}^{k+1}$
4. Reassign $f$ so that all elements before or equal to the index use the lower interval and those greater use the upper value i.e. if$n \leq k$then$\underline{f}^n$else$\overline{f}^n$.
5. Recalculate $y'$ using Equation 2.24.
6. If $y' = y$ then the result is y and $R$ is k, ELSE repeat from point 3.

$$f = \frac{(\underline{f}) + \overline{f}}{2} \tag{2.23}$$

$$y = \frac{\sum_{n=1}^{n} \overline{y}^n f^n}{\sum_{n=1}^{n} f^n} \tag{2.24}$$

Once $y_l$ and $y_r$ are calculated, it is most common to take the mean to obtain a single output from the system, as in Equation 2.25:

$$y = \frac{y_l + y_r}{2} \tag{2.25}$$

Generally speaking, the Karnik-Mendel iterative procedure is a fast and effective method for the type-reduction and defuzzification of interval type-2. It is used in this thesis for all interval type-2 systems under consideration — including those modified type-2 fuzzy controllers, such as the dual-surface control system.

Alternatives to this method include the methods described by Greenfield *et al.* [39] and by Melgarejo [66]. The Melgarejo method is compared to K& M and is found to be faster without lacking in precision, however it has not gained significant traction. Wu and Tan [102] give an overview of this, and several other type-reduction techniques, ten of which are found to be faster than the standard K& M method, with the fastest being the Wu-Tan and Nie-Tan methods.

### 2.5.4 Interval Type-2 Inference Systems

Interval type-2 fuzzy sets are commonly used within the confines of a fuzzy inference system, which uses the components described above to create a coherent system with fixed sets of input and output variables. Figure 2.8 shows how these components are arranged and the data flows between them:

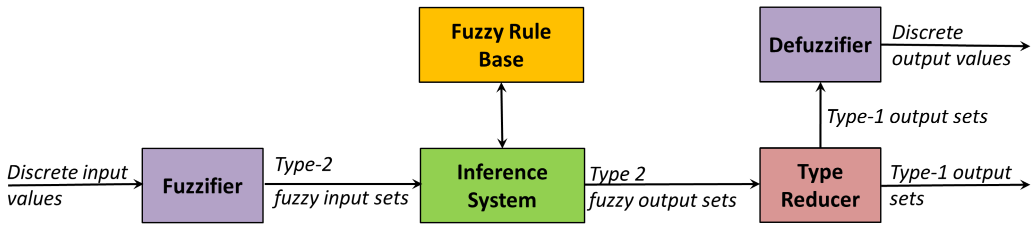



**Figure 2.8:** Interval Type-2 Inference System. The major difference between type-1 based system is the addition of a type reducer.

- Fuzzifier. As outlined in Section 2.5.1, the fuzzifier constructs fuzzy sets for each input variable using the membership functions defined within the system.

- The rule base and inference system as discussed in Section 2.5.2 applies the rule base to the input fuzzy sets to calculate rule firing strengths for each rule. This results in a set of type-1 and type-2 fuzzy sets that are passed into the type reduction section.
- Type reducer and defuzzifier. By far the most common method, the K&M iterative procedure outlined in Section 2.5.3, combines these two elements into a single process which takes as its input the fuzzy sets and the rule firing strengths to give an interval consisting of an upper and lower output values which is generally averaged to give the final crisp value from the system.

This generic form is obviously based on the type-1 inference system with the main difference being the addition of the type reduction block and the flow of type-2 sets as well as type-1 sets between the fuzzifier, inference system and type-reducing sections.

Mendel and Jon [69] show that in order to work with interval type-2 inference systems it is not necessary to work with general type-2 fuzzy sets and that all interval mathematics can be based upon type-1 fuzzy set theory, giving a significantly lower barrier to entry. This can be thought of by thinking of the upper and lower membership functions of the interval as individual type-1 sets.

There are many methods and approaches to the design of fuzzy inference systems including Choi *et al.* [25] and Guillaume [40] which both present techniques for their design and selection of parameters such as rule-base, input variables, and so forth.

### 2.5.5 Deviations from Standard Fuzzy Control

What has so far been described are termed *standard* fuzzy logic systems. There exist several modifications to these systems with different aims. For example, to optimise a particular component seen as a bottleneck, to make use of additional information, or to adjust to changes in the system on-the-fly. Two modifications to standard fuzzy logic control are described here along with reasoning as to why they may provide improved performance over standard controllers and why they are specifically of interest.

### Non-Stationary Fuzzy Logic

While there has been significant work into the development of type-2 based controllers, alternatives based on type-1 control have also been described. These studies aim to address some of its shortcomings — specifically its lack of flexibility in more complex environments. They were first mentioned by Garibaldi *et al.* [38], where the variations in the results of a decision making process were studied in order to understand why these variations happen and how they can be modelled and better utilised.

This lead to the development of 'Non-stationary fuzzy logic' (NS) described by Garibaldi and Jaroszewski [37]. The intention with non-stationary fuzzy logic is that the variation introduced by this method can be matched to the behaviour of the environment and therefore provide the flexibility that is not present in standard type-1 fuzzy logic based controllers. A Non Stationary fuzzy set is defined by Equation 2.26 where $t$ is a free variable (time) (Equation taken from the work by Garibaldi *et al.* [38]).

$$\dot{A} = \mu_{\dot{A}}(x,t).x, \mu_{\dot{A}} \in [0,1] \tag{2.26}$$

This type of fuzzy logic differs from standard type-1 by the addition of multiple fuzzifiers, as shown in Figure 2.9. Each fuzzifier has a slightly different membership function created using a transformation of a base function — one of the most common transformations is a horizontal movement of the base function. The intention of this transformation is to model situations in which there could be multiple correct answers. For example, in many medical systems a doctor is required to make a prognosis, but different doctors may give different answers based on their own experience. In a non-stationary system, each doctor is allocated their own membership function, which is then fed into the same rule base and defuzzification procedure, and produces a collection of outputs. Methods such as majority voting or the mean can be used to obtain a single output from such a system.

There are some advantages to the this approach: Firstly, it is simple to move from a standard type-1 to this kind of system, giving an easy method to potentially improve performance. Further, if it is assumed that a model of the environment can be calculated, it is may be possible to generate a perturbation function that matches this model. In addition, it provides a simple method to combine multiple opinions on a given decision problem into a single system such as may happen when several doctors provide

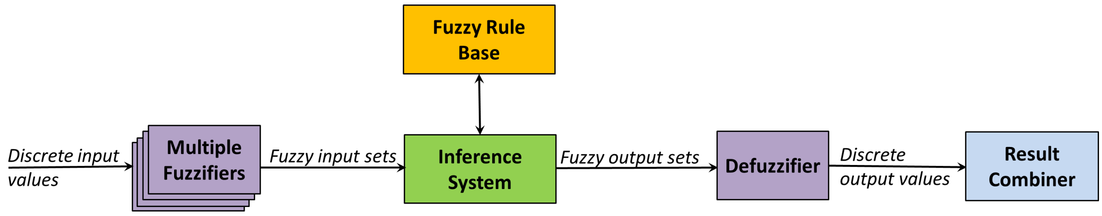



**Figure 2.9:** A non-stationary fuzzy inference system

several different diagnoses from the same input data i.e. the patients test results and symptoms.

When trying to decide what fuzzy system to move to from a standard type-1, the choice between non-stationary and type-2 is potentially tricky. Garibaldi *et al.* [36] present an investigation into the differences between non-stationary and type-2 fuzzy sets. The authors show that a non-stationary fuzzy set could potentially be used to approximate a general type-2 inference system. Although this technique needs further investigation, it would give an obvious pathway to follow — starting from type-1, moving to non-stationary and then to general type-2, based on system requirements such as desired performance levels.

NS fuzzy logic is not restricted to using type-1 membership functions as its basis. Zhao [110] presents the first demonstration of non-stationary system based on type-2 fuzzy logic. The starting mathematical operators including intersection and union are defined and explained. Additional work is required to concretely show the advantages of this system over, for example, a general type-2 system, however this work acts as a starting point for such investigations.

### Dual Surface Fuzzy Logic

Although the Enhanced Karnik-Mendel procedure is a fast and effective method of type-reducing and defuzzifying a type-2 output set into a single crisp output, it does so at the expense of discarding information. Once the crossing points of the upper and lower values of the output set are found, the resulting values are simply averaged and the average used. This means that the relative values of each output are lost.

It has been discussed by Birkin and Garibaldi [13] that with a simple modification to the K & M algorithm, it might be possible to increase performance over standard type-2 control. Further, it is suggested that a selection criteria may be used to chose from either the upper, lower, or mean values, based on the input error value. This selection algorithm is shown in Algorithm 1. This type of controller has been termed **dual-surface** (DS) because it uses both of the outputs from the KM type-reduction algorithm.

```
error = control_var - set_point;
diff = abs(error);
if diff < THRESHOLD then
|   control_action ← (LS + US) / 2;
end
else
|   if error > 0 then
|   |   control_action ← LS;
|   end
|   else
|   |   control_action ← US;
|   end
end
```

**Algorithm 1:** The dual-surface control algorithm. Obtained from [13]

Aside from the work performed by Birkin and Garibaldi, DS control has not gained a much attention in the exiting literature. This may be in part due to standard type-2 control itself still not being fully understood and researchers prioritising this over new varieties. There is, however, a great deal of additional work that could be done in this field, as it was done with standard fuzzy control. Both the DS algorithm and the inputs selected provide interesting subjects for future investigation.

### 2.5.6 Comparisons of Type-1 and Type-2 Fuzzy Logic Systems

As type-2 fuzzy logic has become better understood and cheaper to implement, many researchers have begun to adapt their existing type-1 systems to use type-2 fuzzy logic, such as the work performed by Cazarez *et al.* [20]. The reasoning behind this move is not yet fully justified. It is based on the reasoning that if there is 'considerable' or 'large amounts' of uncertainty within the system in which the controller operates, then type-2 fuzzy logic will likely outperform type-1 based systems, as discussed by Aliasghary *et al.* [2].

There have been many comparisons of the ways in which type-2 fuzzy logic systems outperform type-1 systems under the same experimental set-ups. An example is the comparison made by Sepulveda *et al.* [86]. One of the potential reasons for the superiority of type-2 is discussed by Wu [104], in which the continuity of input-output mappings for type-1 and type-2 systems are studied. This study uses control surfaces generated by different controllers to show the discontinuities present in type-1, in contrast to the continuous character of type-2. Wu concludes that type-1 control surfaces can be discontinuous at certain points, while the interval type-2 controller is continuous in the same situation. This means that, at given points within the input-output mapping, there are points for which the type-1 is not able to calculate an output — making the type-2 systems more suitable in these cases. There is considerably more work required on this subject in order determine the reason and implications this find has upon type-2 controller design.

Cara *et al.* [19] use a servo system as the application basis for their comparison between singleton type-1, non-singleton type-1 and type-2 controller varieties. The authors argue that type-2 controllers — even singleton varieties — employ considerably more variables. These controllers are therefore, described as having more flexibility than the equivalent type-1 control systems. In addition, by employing non-singleton fuzzification, the authors hypothesize that the flexibility of type-1 can be increased further, reaching a level closer to the type-2 controller. This hypothesis is subsequently tested by the development of each of the varieties — singleton type-1, non-singleton type-1 and singleton type-2 controllers and applying them to a non-linear servo problem. Each controller is run under three different levels of uncertainty — termed "none/small", "medium" and "high" — with the terms based on the noise present in the servo model. The results show that under small amounts of uncertainty the singleton type-1 system exhibits the best performance. However, as uncertainty is increased both the non-singleton type-1 and type-2 based controllers improve in comparison. The authors conclude that in general, the singleton type-2 system outperforms the type-1 based controllers, both singleton and non-singleton. The authors hypothesize that this is due to the uncertainty and variation in the type-2 system flowing through the whole inference system as permitted by the FOU, contrasted with the non-singleton type-1 controller, in which only the fuzzifier handles the variation via the standard membership function. One of the shortcomings of this work is the application used — a servo system. This is a simple one-input one-output system, although it is described as non-linear in nature. While this type of applications has advantages, such as the ability to tightly control the system, allowing sources of variation to be controlled, it has the disadvantage that such a simple system does not provide enough scope for different controllers to differentiate themselves in performance. The ideal of the complexity and sophistication of the test application selected for experimentation is developed further in this thesis.

Figueroa *et al.* [34] use a much more complex application problem of robotic football for their study of the performance between standard type-1 and interval type-2 fuzzy control systems. In this work, multiple robots move around a playing field with the objective of pushing or kicking a ball into a goal. The experimental set-up is complex, with a significant number of software and hardware modules required to coordinate each of the robots around the playing field. These modules include image capture and processing, high-level control system, wireless communication, and embedded controllers on-board of each robot. This application introduces a potential for large amounts of uncertainty including latency introduced by radio frequency (RF) links, image processing time, and different motor movement response for each robot. In this situation, it is found that the type-2 system outperforms the type-1 system from which it is derived, showing that a complex environment does seem to favour type-2 fuzzy logic control.

Another comparison of type-1 and interval type-2 systems is presented by Méndez *et al.* [70]. In this work, a large industrial process is used as the test application. The interval type-2 controller outperforms the type-1 controller, as in the previous comparative study above however a significant difference from the previous study is that the type-1 controller is already in use and operating at an acceptable level. This means that although type-2 controller is superior, there is a requirement to show considerably better performance to be worth the expense of upgrading the system. In this case, the performance gains are significant but a true cost/benefit analysis is not performed.

One possible reason given for not moving from type-1 to type-2 fuzzy logic is the increased computational requirements demanded by type-2 systems. This extra loading has led to the study of methods for reducing the computational cost of type-2 systems. The work from Wu and Mendel [105], for instance, present a method that eliminates type reduction when using interval type-2 fuzzy logic. This method however has not seemed to attract much attention in the literature. This may have occurred because techniques such as the Karnik-Mendel iterative procedure for type-reduction are straightforward to implement and, in general perform at acceptable levels. Coupland and John [27] present another method of increasing computational speed. They focus on faster methods for **join** and **meet** operators, which are some of the most commonly used operations performed upon fuzzy sets and therefore give a considerable boost to any system in which they are used. It is probable that these sorts of techniques are used more frequently when the advantage of type-2 systems is shown.

That type-2 fuzzy logic systems can outperform type-1 systems, as discussed above and further in studies such as Phokharatkul and Phaiboon [76]. However, the exact reason as to why this occurs is still unclear. Under what conditions these effects will reliably occur is a subject of much discussion by numerous authors, including Liang and Mendel [55] and Wagner and Hagras [99]. Recommending type-2 logic under all circumstances is therefore a somewhat difficult proposition. Ideally there should exist a methodology by which one could characterise the system environment, including the amount of variation present, how “dynamic” or variable the environment is, and how difficult the task is to complete, which in turn would allow design decisions such as the type of fuzzy logic to be determined.

## 2.6 Applications of Fuzzy Logic

### 2.6.1 Control Applications

One of the most significant applications of fuzzy logic is that of control, in which a specific system requires controlling to give a desired output with an excellent overview provided by Jager [44]. Aircraft [61], boats [87] and cars [97] have all shown that fuzzy logic can be applied to their control systems. Many large industrial companies make use of it for a multitude of applications. These include General Electric, as described by Bonissone *et al.* [15], in which a single FLC framework is discussed. The framework is then used in many different applications of increasing complexity — from domestic dishwashers, steam turbines, locomotive wheel slip controllers and aircraft engine control systems. Sanchez-Solano [79] discuss FPGA implementation of type-1 fuzzy control making it possible to add it to many embedded applications, though with an additional cost of one microprocessor.

Initial studies with robots and fuzzy systems have generally made use of type-1 fuzzy logic such as those performed by Seng [85] and Yakzan*et al.* [106]. There are two significant factors why this is has occurred, the first and most obvious is the generally limited resources on board mobile robotic systems. Depending on size, weight and energy constraints, the exact configuration will vary, but in general, an embedded system will have a fraction of the capability of a contemporary desktop or server system. Secondly, the mathematics and background behind type-1 systems is much better understood. In addition, several methods exist for the generations of type-1 systems from example or training data. Type-2 until recently has been considerably more complex and harder to implement, with the sheer number of variables and parameters making many approaches unworkable.

For research purposes, one of the most common experimental set-ups for fuzzy logic is in the use of wheeled robots. Phokharatkul and Phaiboon [76], Hagras [41] and Saffiotti [78] present various experiments for the demonstration of the applicability of fuzzy logic to wheeled robot control. While they vary in the exact objective that they are focussed on, in general they show that fuzzy logic is appropriate for these types of applications. The variation present in these sorts of applications is generally considered low, as there are few physical processes (such as wind) that are present in indoor environments. This leaves the problem with only sensor and motor uncertainties, which in many robot systems, are small in the scale of the experiments presented.

Control theory is a field of engineering concerned with the behaviour of dynamic systems, in which the system changes over time [59]. A system in this context is a mathematical formalisation for the description of the relationship between a given point in time and its location in space. There are three main elements to a control system: inputs, control process and outputs, as shown in Figure 2.10. It can be observed in the figure that there exists a cycle that links the output and inputs — known as a feedback loop, which is an important concept in control theory. The field of control theory provides a large number of techniques and tools for analysing, specifying and formulating the desired behaviour of dynamic systems. These techniques in turn, allow controllers such as PID to be formally defined, implemented, tuned and have their expected performance calculated without access to hardware. Song & Tai [90] give a simple application concerning navigation of mobile robots in which fuzzy logic is employed and found to be well performing.

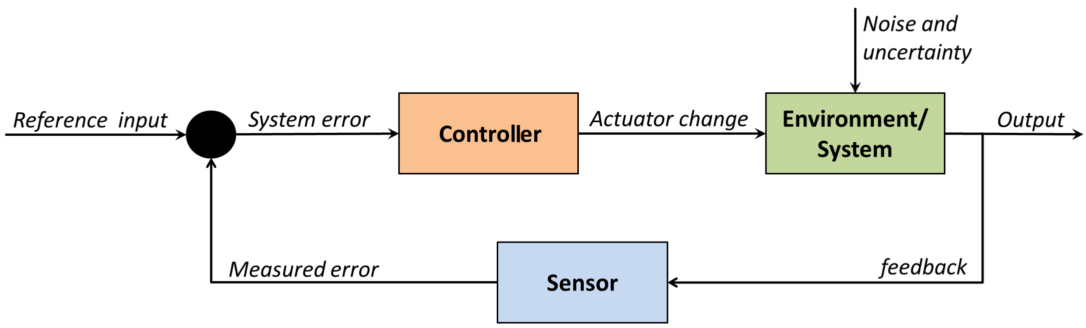



**Figure 2.10: A Control theory system loop**

While control theory is a powerful and well understood tool that has been successfully applied to a great number of fields and many different applications, it does however has some shortcomings. These limitations, along with the additional resources available with modern computer systems, have led systems engineers to develop solutions based upon fuzzy logic. Using control theory as described in this section is not an intuitive procedure as gain values are very much an abstract concept. Fuzzy logic, on the contrary, is much closer to the human decision making process and makes the incorporation of expert knowledge a straight-forward task.

### 2.6.2 Sailing and Sailing Robots

Robotic sailing is one of the studied applications in this thesis. The following section presents a brief overview of sailing theory and associated sailing hardware. This guide is not intended to be comprehensive and tries to remain generic with regards to the specific boat hardware. The principals introduced in this section should therefore, apply equally to any-sized sail boat. For a more complete guide encompassing particularities of different systems, refer to Bond [14], which introduces a beginners guide to sailing traditional sailing boats.

#### Sailing Theory

A generic sailing boat has two main controls: the direction of the rudder and the direction of the boom onto which the sail is connected. Figure 2.11 shows a generic sailing boat with each of these elements labelled and described below:

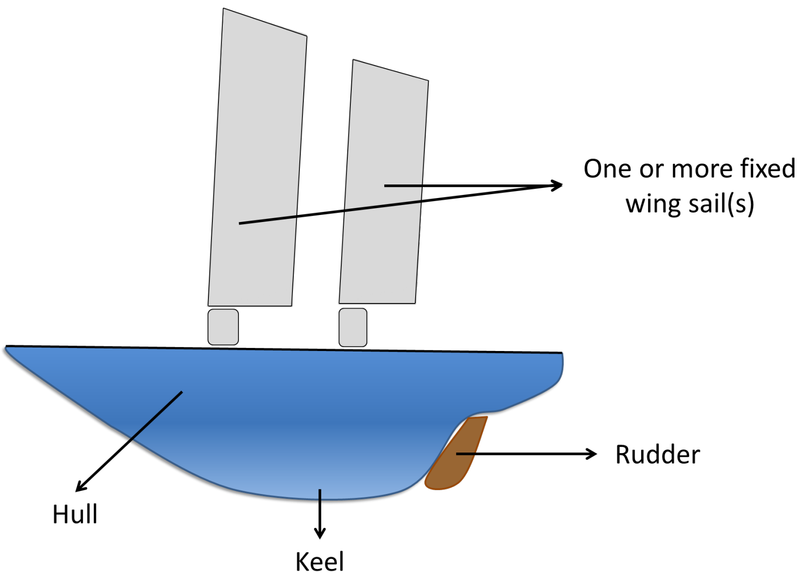



**Figure 2.11: A generic fixed wing sailing boat component layout**

**Fixed wings** This sail type are more robust than traditional cloth sails as well as being easier to control using a single motor. The sails provide the forward motion and can give limited steering control.

**Rudder** The rudder has the greatest effect on the direction of the boat.

**Keel** This prevents sideways movement of the boat when winds come from directions that are not straight on.

**Hull** The shell of the boat. This gives a hydrodynamic shape and allows easy movement through water.

During sailing, there are two major inputs that a human sailor needs to pay attention to: (1) the wind direction and (2) the direction in which travel is desired. An experienced sailor can determine from these inputs — along with data such as the optimal attack angle [1] — what changes are required to the outputs of the system (the rudder and sail positions) in order to perform the desired course change.

One problem that often arises while sailing occurs when the desired direction is equal to the wind direction, i.e. the sailor wants to move directly into the wind. This can not be achieved directly, so a process known as **tacking** must be used. This process involves altering the desired angle of travel by a given angle ($A$)(based on how the boat is affected by wind) for a given time, followed by the same period of time when the boat is directed by the inverse of the first angle ($A$‘)in the opposite direction. This is best illustrated in Figure 2.12, in which it can be observed that the average course over the duration of the tacking gives a straight line directly into the wind.

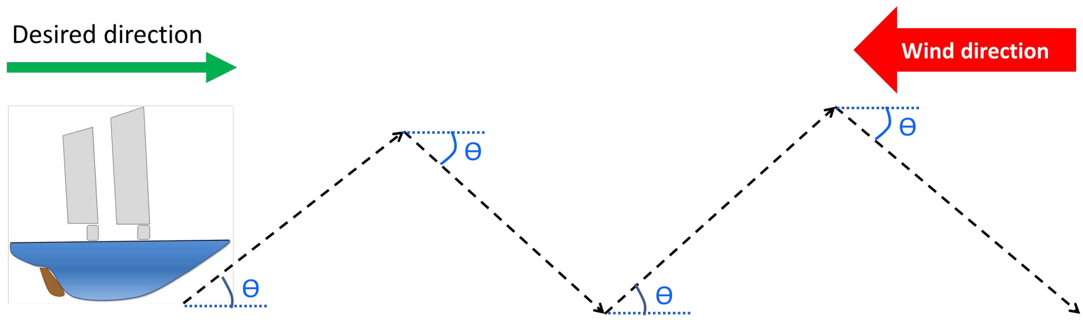



**Figure 2.12: Tacking behaviour of sailing boat**

[1]Attack angle is that angle by which a specific boat obtains the best forward momentum and is a constant value.

### Automating the Sailing Theory

There have been several investigations into the use of autonomous sailing boats, both as an engineering problem and also as a test bed for various autonomous control systems. A selection of the most relevant studies to our research are presented here, along with a discussion of the strengths and weaknesses in their chosen control technique.

The first autonomous sailing boats appeared in 1997 and used fuzzy logic as a basis for its control system. Abril *et al.* [1] demonstrate a very simple fuzzy-based controller for adjusting the rudder and sails of a small boat. Limitations with the hardware are an issue — with only 32KBit of memory available and with an 8-bit 2Mhz processor. This imposes a limit on the number of membership functions and rule base, as well as the overall complexity of the system. Vaneck [96] on the contrary, presents a much more capable hardware platform, providing 256Kb of memory and a 25Mhz processor. This advance in hardware allows a more detailed and sophisticated controller to be implemented. In addition it also allows membership functions to be held in a more developer friendly manner. This is due to the fact that memory is not so restricted, which allows high level languages and data structures to be used. This makes modifications a considerably easier matter, while at the same time offering increased performance.

The specific hardware basis of the robot used in this thesis is described in details in Chapter 4. It was originally developed in the University of Aberystwyth. Sauze [80] initially discusses some possible techniques for the control of this robotic sailing boat. This was followed up by the works [82] and [83], in which bio-inspired techniques applied to control sailing robots are more fully realised. Specifically, the authors introduce artificial endocrine systems, which use concepts from the human endocrine system as a basis of a fault tolerance measure for sailing boats. The main focus of their work is to develop a long running and robust platform suitable for long periods of autonomous operation, rather than optimal sailing such as moving in the straightest line possible. Due to the objectives of their work, the authors therefore sacrifice performance in terms of course accuracy and fast sailing for robustness in their controller designs. The characteristics embedded in their sailing boat hardware, therefore, help the studies performed in this thesis, as the hardware has been thoroughly tested and potential shortcomings have been detected and addressed. Furthermore, their work provide knowledge regarding the battery lifespan and motor characteristics of the hardware. This facilitates the conduction of the real-world experiments later in this thesis, as the potential variability in the outcomes of sailing runs can then be mostly attributed to the volatility of the environment, rather than hardware faults.

The software controllers in this thesis use the work performed by Stelzer [92] [91]. The intention of Stelzer in this work is to develop a boat control system that can sail autonomously. Specifically, it should be able to perform turns, tacks and other manoeuvres of a sailing boat in a similar manner to how a human would perform them and in similar a time frame. In contrast to the work of Sauze, the boats here are not generally intended for a long term usage autonomy. Instead, they are intended to operate on a much shorter time period, hours, rather than days or weeks — making the time it takes to perform a turn, for example, much more important than conserving battery energy.

The hardware selected was chosen for various reasons. The work of Sauze shows that the hardware platform is robust, reliable and lasts many hours. These features are important for field work, where access to mains power is limited. Additionally, it uses fixed wing sails, which tend to be more reliable and less prone to breakage than cloth sails. The hardware on-board, specifically the gumstix PC, provides adequate computational resources to execute both type-1 and type-2 fuzzy systems while still maintaining a high update rate of sensor readings (further information about the hardware will be presented in the next chapter). The boat used by Stelzer is less well described in the literature and is therefore not as easy to determine how closely it matches the requirements of a robot for testing fuzzy logic controllers.

The general approach of all works above is to use sophisticated controllers along with a a significant tuning and debugging cycle of the boat itself to give a well-performing controller. Briere [17] uses a simpler state machine controller. However, he spends significant time and resources performing detailed investigations into physical aspects of the boat, such as drag and hydro-mechanical properties. While this means a simple controller was used and gave good performance, any changes to the boat would demand a significant investment into their effects upon the controller. This sort of approach is also very prone to failure issues, as slight changes in the responsiveness of hardware will necessitate changes in the controller.

The work performed so far by the above authors show that autonomous sailing provides a difficult but not insurmountable challenge for several different varieties of fuzzy controller. Fuzzy logic as a control method for autonomous sailing has been attempted by several different authors with varying degrees of complexity and success as discussed above [1], [96]. It is shown here that fuzzy logic is a feasible approach for autonomous boat control. The effect of environmental variability is also easier to study in this type of application, as sources of uncertainty can have such a large effect on overall performance. This makes trends and patterns potentially easier to spot, when compared with applications such as wheeled robots where, spotting the effect of, for example, surface material can be difficult and require careful and precise measurements potentially requiring a significant investment in time or equipment.

## 2.7 Variable Environments

### 2.7.1 Defining Variable Environments

As has been described in the previous section, robotics is a common application area used in the investigation of fuzzy logic based control methods. In this field, one of the most common issues that must be considered when designing and using any type of automated controllers, such as those based on fuzzy logic, is the environment in which it is designed to operate. Robotic experiments generally operate in an given *Environment* as defined in Definition 2.7.1. One of the aspects of such an environment is how much it is liable to change over time. An environment which changes over time is termed *variable* environment and is given a formal definition in Definition 2.7.1.

> An environment is a defined as fixed set of physical processes within a fixed area.
>
> An environment is considered variable if sensors within the system are subject to changes by one or more external sources, termed a *source of variation*. The greater the number of sources and the size and frequency of their variations will determine how dynamic a given environment is.

> This can be characterised by the term *variability* with more variability indicating a more dynamic environment.

There are many possible reasons for the occurrence of these sorts of variations within environments — some environments naturally change over time, while others are generally static but have changes triggered by external agents, such as people moving through them. The wind is the the main cause of changes in the environments used in the sailing based experiments of this thesis. However, other examples include changing light levels, sound or heat levels. Each of these causes of change are referred to as *sources of variation*. For a source of variation to be considered relevant to a given experiment, its effect upon the system must be measurable in some way by the robot. For example, sensors must be able to measure the variation induced by the entropy — in the case of the experiments in this thesis, the changes in wind parameters can be measured by the wind sensor aboard the sailing robot being used for the experiments. Additionally, the wind will cause the boat to move position in the environment.

The level of variation within a given experiment is difficult to accurately quantify. It is not a simple binary concept, (e.g. the environment is not either dynamic or not) but instead it is a complex continuous function of each source of variability and their relative degrees of change over time. Changes and comparisons in variability are easier to obtain than absolute values, i.e. it is generally easier to determine environment 'A' has a higher level of variability than environment 'B'. Due to this difficulty, dynamic environments are an interesting area for the investigation of autonomous controllers, such as those based on fuzzy logic. However, this research must be pursued with caution to ensure that experiments are well controlled and controllers are not overwhelmed by the variation present. For example, trying to sail a small boat in gale force winds in which no matter what the controller tried to do, the observed behaviour would be the same — the boat would move essentially randomly. McBratney and Odeh [65] discuss the use of fuzzy logic within the field of soil sciences in which imprecise and variable data is a common hurdle to be overcome.

### 2.7.2 The Effect of Variable Environments on Autonomous Navigation Systems

The effects of operating robots in variable environments can be considerable — the changes in the environment cause changes in the performance characteristics of the robot under test. In general, the more sophisticated forms of controller, the more likely it is to be able to cope with variations in the environment, as discussed in Section 2.5.6. While it is known that dynamic environments have generally detrimental effects, structured investigations into how this occurs are not common in the literature. Due to this shortcoming, our work aims to use structured increases of variability in the experimental environments to determine relationships between performance and environmental conditions.

Some examples of studies into the effect of variable environments include the work of Antoun and Mckerrow [3], who define their environment as a village in which an autonomous robotic agent represents a lost tourist within the town who is trying to find their way to a given destination. The dynamic elements introduced in these experiments includes people — both singular and collectively to form crowds, along with other vehicles, such as cars and buses. The eventual goals of the work are ambitious however, this specific publication is an introduction to their platform, methodology and goals — the experimental data presented is minimal. However, the concepts used including how they define variable environments rather than actual experimental data is the interesting point.

In this thesis the wind is considered the main source of variation of the experimental environment used in the sailing experiments performed in Chapters 5 and 6. The more frequently and the bigger the size of the changes in wind speed and direction, the more dynamic the environment (i.e. it has higher variability), and it is hypothesized that the more sophisticated fuzzy logic control types such as type-2 fuzzy logic will be better able to handle these more dynamic environments than type-1 fuzzy logic. And it is this hypothesis that is further investigated as one of the main subjects of the research conducted in this thesis.

It has been hypothesized that the use of type-2 fuzzy logic as an evolution of type-1 fuzzy logic, specifically its more sophisticated membership functions which utilise a

Footprint of Uncertainty, can be utilised to counter the problems with highly dynamic environments.

## 2.8 Research Focus

This thesis will initially use a simple 'Tipper' application in order to introduce the concepts and methodologies for comparing fuzzy logic controllers. Several different approaches are used to investigate how the factors being considered change the behaviour of the systems. Based on the results obtained a new application which uses autonomous sailing is introduced for the remainder of the experimental work. This sailing application provides the basis for a more developed investigation into the effects of dynamic and variable environments on the behaviour of fuzzy systems including, but not limited to their performance as defined by an RMSE value.

Parameters related to the internal configuration of the fuzzy controllers, most importantly the FOU size are predicted to be important in defining the behaviour of the controllers, especially when the environment in which they operate contains sources of variation. Work is performed in which the FOU size is varied across a number of environments that contain increasing amounts of variation, allowing the effect of the FOU to be observed.

In addition to the configuration of the controller and the environmental variation, the difficulty of the task is the final factor of interest, where it is hypothesized that more sophisticated controllers will perform better. Scenarios in which all three factors are varied one by one will make up the bulk of the experimental work attempted. The overall intention would be the ability to be able to support statements such as 'Type-2 controllers with a large FOU size generally outperform type-1 fuzzy systems in environments with large amounts of variation' in the most general manner possible.

As part of this work comparisons of controllers with different internal configurations, operating under increasing levels of variation and task difficulty are performed. How the results change between these experiments lead to being able to draw conclusions about how each of these alter performance, and therefore determine which conditions are better for a given controller configuration.

## 2.9 Summary

This chapter outlines the background material that is used and referred to in the rest of this thesis. Fuzzy logic as a concept is introduced including its background mathematical theory. The components of basic fuzzy logic and the make up of generic fuzzy inference systems are discussed. Furthermore, modifications to the standard type-1 and type-2 are explored including non-stationary and dual-surface varieties. In addition, their relationship to the more standard varieties and why they have the potential to be a useful avenue of study is presented.

Applications which utilise fuzzy controllers are discussed together with an analysis of the advantages and disadvantages of applying fuzzy logic to the explored domains. The basics of control theory are introduced and the reasoning why it is important subject of this thesis is discussed.

Several studies which compare type-1 and type-2 fuzzy logic are presented, with particular reference to shortcomings of the techniques used to make the comparisons and how these can be addressed. One of the most common limitations is that the amount of environmental variability is often not controlled or varied in a structured manner. Vague and imprecise values such as “low” and “high” are commonly used, making answering the question “At what level of variability in a dynamic environment does type-2 begin to present considerable performance increases over type-1 control?” a difficult task.

The use of sailing boats and its suitability for the experiments in this thesis is discussed. Sailing boat based applications are found to present an interesting control problem and a challenge that should allow a wide gap between different performing controllers to occur. This is much preferable to controllers in which performance differs only marginally and therefore require very specialised measuring equipment to observe. Large differences therefore make determining the best or worst performer an easier task, which is important when so many different combinations are simultaneously considered.

# 3

# Tipper Application and Initial Methodology Evaluation

## 3.1 Introduction

In this chapter a simple application is introduced with the intention of demonstrating the comparative methodologies used in the rest of this thesis. Several sets of experiments are performed and their results discussed along with any conclusions that can be drawn from the work. Both type-1 and type-2 varieties of fuzzy logic controller are utilised — with the type-2 controllers utilising four different sizes of FOU. The first experiment looks at the rules inside the fuzzy logic controller and how they fire under different conditions. These experiments use the entire input space of the fuzzy controller and is focussed on looking at its internals working. The following experiments use the same application but work to develop a methodology for comparing controllers, including introducing such elements as variation and the concept of comparison with an ideal output value.

The main aim of the experiments in this chapter is to show that type-2 fuzzy logic can provide a significant difference in performance to type-1 based controller in one or more experimental scenarios — ideally by showing an improvement overall.

This chapter is organised as follows: The test fuzzy logic controller is described in Section 3.2. Section 3.3 presents experiments investigating how the rules which fire change across the different FOU sizes. A comparative methodology using RMSE is introduced in section 3.4 and further developed in Sections 3.5 and 3.6 where the concepts of ideal values and randomness are added respectively. Finally the findings of this chapter are discussed in Section 3.7 and the chapter is summarised in Section 3.8.

## 3.2 Tipper Fuzzy Logic System Description

All of the experiments performed in this chapter use the fuzzy logic controller described in this section. This includes the inputs variables, outputs variables with their associated fuzzy sets, the rules base and which operators are employed. The example application used in this section is a fuzzy decision support system originally described in the Matlab documentation [64] which, in this thesis is referred to as the tipping problem. This application has been chosen because of its simplicity, making it ideal to show the operation of the comparative techniques used in this chapter as well as being fast and straight forward to implement due to the high quality documentation provided. There are many approaches to designing fuzzy systems as described by Berkan and Trubatch [9] and Feng *et al.* [32], both of which present straight forward approaches to the design of standard type-1 fuzzy controllers.

This application is concerned with determining what percentage tip should be given to a restaurant server based on the quality of the service and food provided. This type of system is generally termed a 'fuzzy decision support system'. The exact working of each of the elements of a fuzzy inference system is given in Section 2.3 (Page 12).

### 3.2.1 Input Variables

This system uses two input variables termed 'service' and 'food quality'. Chiu [24] and Lin *et al.* [57] discusses two different approaches of selection of inputs variables for fuzzy systems with the results of the technique being compared to a benchmark problem with improved results being found.

The 'service' input variable has three associated membership functions labelled 'Poor', 'Average' and 'Excellent'. From Figure 3.1(b) it can be seen that service is rated between zero and ten, with 0 indicating the worst possible ("Poor") and 10 the best possible service ("Excellent"). The original designers of this system use Gaussian shaped membership functions for all of the fuzzy sets, however due to shortcomings in the software library used in this thesis, Gaussian membership functions are not supported and therefore these have been changed to triangular shaped function, this change addresses the software shortcoming and secondly also means that both input variables have the same shaped functions which may be beneficial for later analysis.

Food quality is the second input variable in this system and is described using the two adjectives (fuzzy sets) 'Rancid' and 'Delicious', both of these are represented using trapezoidal membership functions as shown in Figure 3.1(a) with limits between 0 and 10. The sets for this variable are kept constant throughout each experiment so that only one aspect of the experiment is changed at a time.

The membership functions for the service input variable is changed to generate several different system configurations used for comparison. Specifically, the food membership functions is formed into type-2 footprints of uncertainties (FOU) as defined in Table 3.1, making the whole system a type-2 fuzzy logic based system. The different sizes used in these experiments are shown in Figures 3.2(a), 3.2(b) 3.2(c) and 3.2(d). The original membership functions defined by the original designers (but including the changes stated above) of this system will be termed the 'standard' configuration in the discussion later in the thesis.

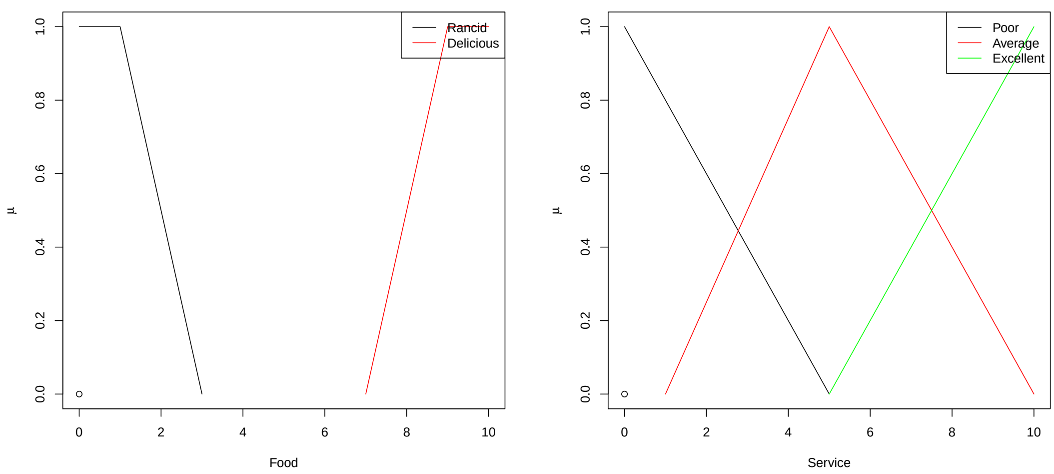
One point of note is the fact that the membership functions for the food quality input variable do not overlap. Some designers of fuzzy logic systems believe that this should be avoided in such systems as it will, depending on factors such as rule base, can lead to decreased performance. However in order to minimise changes from the original system this is maintained as is except when the FOU size of type-2 systems causes overlap.

### 3.2.2 Output Variables

The output variable calculated by this system is the percentage tip that should be given and therefore it is labelled 'tip'. This output is represented by three fuzzy sets la-

| FOU Size | Poor Upper | Excellent Lower |
|---|---|---|
| 1 | 1 | 9 |
| 2 | 2 | 8 |
| 3 | 3 | 7 |
| 4 | 4 | 6 |

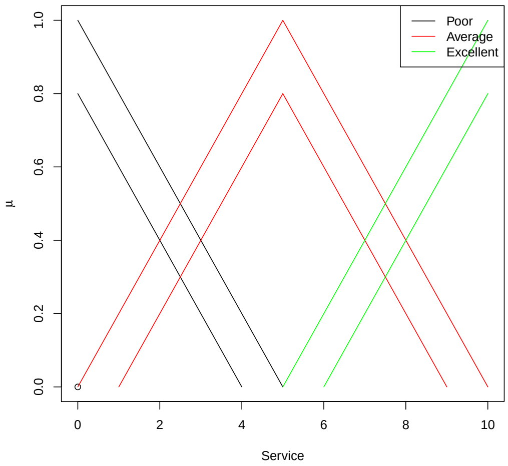
**Table 3.1:** Upper and lower limits of the Poor and Excellent fuzzy sets for each FOU size



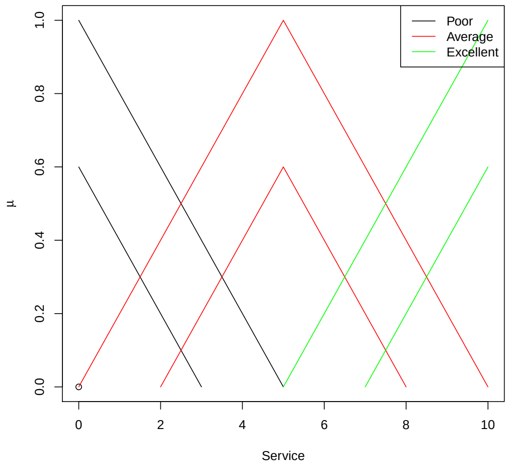

(a) Food input variable type-1 membership functions (b) Service input variable type-1 membership function

**Figure 3.1:** Input Membership functions for Tipping Problem

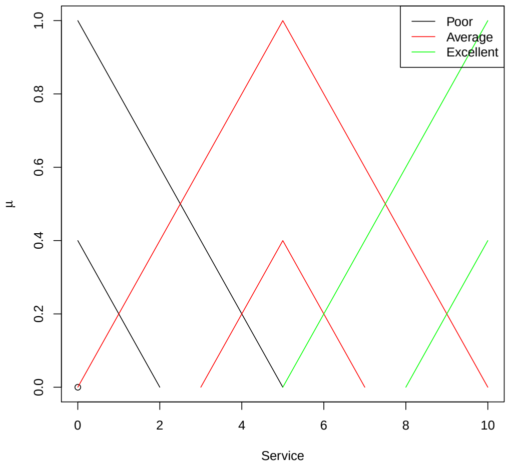
belled ‘cheap’, ‘average’ and ‘generous’ which use triangular membership functions, as shown in Figure 3.3. The limits for this variable are between 5% (a cheap tip) and 35% (a generous tip). It must be noted that the output sets selected by the original designers would not make an effective real-world solution as it is possible for input sets with low food and service scores to obtain an equally generous output as those with high scores. However, as with the input variables they are maintained for the purposes of this work.



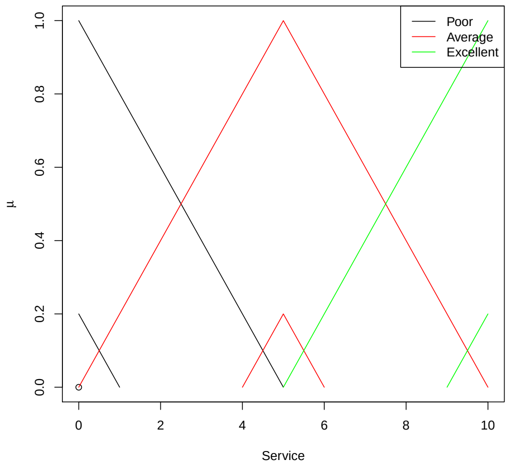

(a) FOU Size 1 Service Membership function




(b) FOU Size 2 Service Membership function




(c) FOU Size 3 Service Membership function




(d) FOU Size 4 Service Membership function

**Figure 3.2:** Type-2 input membership functions for ‘service’ input variable

### 3.2.3 Rules

The rules used in this system are outlined in Table 3.2 below. These rules are evaluated during the inference process when the input fuzzy sets have been calculated from the

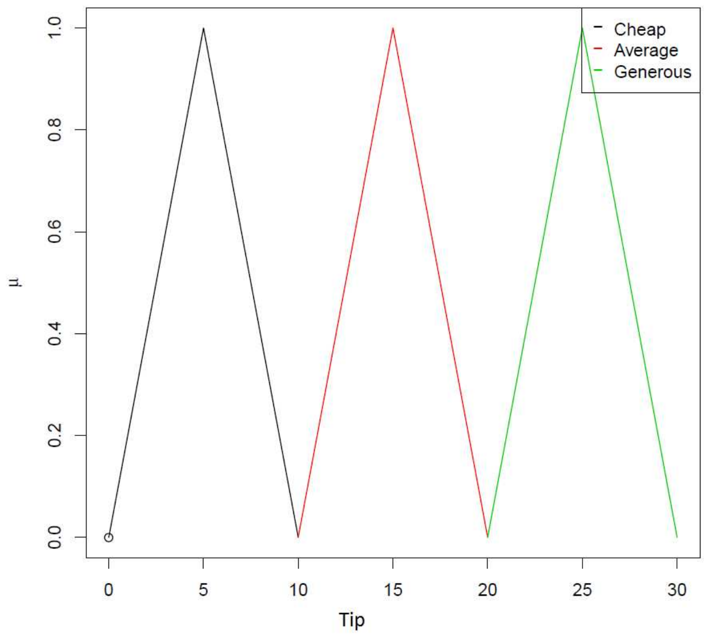



**Figure 3.3:** Tip output membership function

crisp input values.

| Rule Number | Service | Food Quality | Tip |
|---|---|---|---|
| 1 | Poor | Rancid | Cheap |
| 2 | Good | N/A | Average |
| 3 | Excellent | Delicious | Generous |

**Table 3.2:** Rules for the Tipping Problem in table form

Rule 2 only uses one input antecedent and therefore the service input is defined as N/A.

These rules can also be stated in a more English form as follows:

If service is poor or the food is rancid, then tip is cheap
If service is good, then tip is average
If service is excellent or food is delicious, then tip is generous

Rules can be generated in numerous ways including collection from experts; such as the work done by Chen and Linkens [22] in which rules are generated using input data for several fuzzy models; and generating them from Rule base such as in the work by Wang and Mendel [100].

### 3.2.4 Operators

The AND/OR functions use the min and max operators respectively. The rules specified above use the OR conjunction.

The centroid method is used for defuzzification of the output set to obtain a crisp value.

These operators were selected by the original designers and have been maintained as-is. The fuzzy set mathematics and operators and how they are calculated is given in Section 2.5.2 22.

## 3.3 Tipper Rule Experiments

### 3.3.1 Experimental Purpose

These experiments aim to look at how the rules which fire change as the controller configuration is varied. Specifically, as the FOU of a type-2 controller is increased, does the number of rules that fire increase in number, or is there otherwise a recognisable pattern to the changes that occur. It is hoped that these results assist developing the reasoning that may help explain why type-2 fuzzy logic system do (or do not) provide better performance than type-1 systems under given conditions, which is the main focus of this thesis. An example ideal result would be to find that under hypothetical scenario ‘A’ more rules fire overall and then for this to be correlated to a peak RMSE value in a similar scenario in later experiments.

### 3.3.2 Experimental Design

In this experiment, each combination of inputs for each controller configuration is generated and passed into the system with its output recorded. In addition to the crisp output value, whether each rule in the system fired or not is recorded and stored as a binary number. A '0' will indicate that a rule did not fire and a '1' indicating the rule did fire. The system used here uses 3 rules in the rule base and therefore each output is 3 digits long, giving 8 possible combinations.

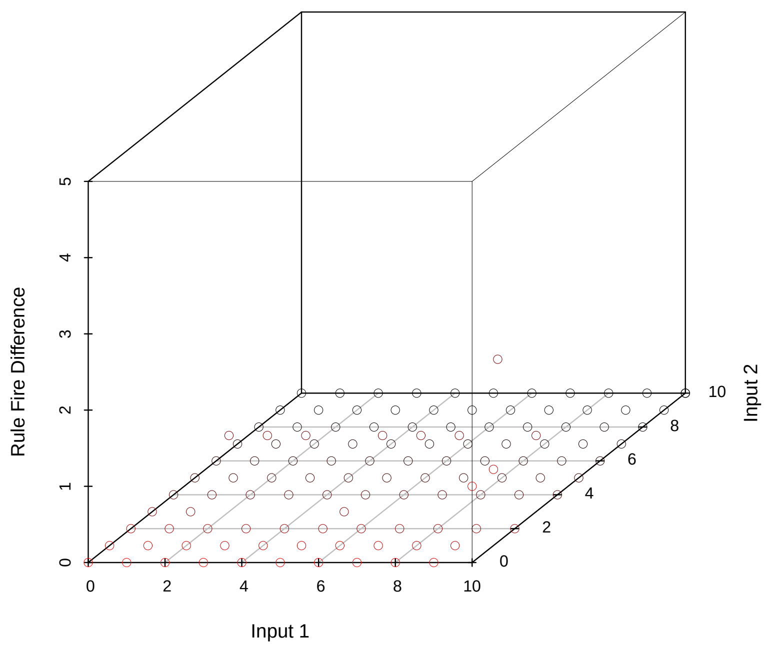
The difference between which rules fire under the type-1 configuration and each different type-2 controller is calculated. This is performed for each set of input combinations, by performing a bitwise comparison of the binary representation of which rules fire and summing the total that are different. Because there are 3 rules the maximum difference is therefore 3 — all the rules in one set fired while all the rules in the other set did not fire. This allows graphs to be generated and therefore the trend, and how the rules change between controller configurations can be be observed. An example of a single 'rule difference' would be if one controller output a rule fire pattern of 001 while the configuration to compared to gave the pattern 010 the difference between them would be two — the result of the bitwise comparison.

One shortcoming of these experiments is that they do not give any idea about the relative performance levels of a given controller — there is no real method for determining which configuration of rules firing for a given set of inputs is the best. This is further addressed in Section 3.4 in which the controller is put into a more realistic experiment where performance is a more relevant issue and it becomes easier to judge whether a given controller configuration is better or worse than another.

### 3.3.3 Hypothesis

It is hypothesized that as the FOU size is increased the number of rule differences between the type-1 and the type-2 will correspondingly increase. This is because as a the FOU size is increased, each membership function will increase in size, covering more and more of the universe of discourse, triggering more rules to be fired, though this can depend upon the design of the rule base of the system under test.

### 3.3.4 Results




(a) Type 1 vs FOU 1

**Figure 3.4:** *x*-axis and *y*-axis show the combinations of input variables. $z$ axis shows the difference in the number of rules which fire. Large $z$ axis values indicate large differences occur between the two controllers. Colouring used to improve 3D visibility.

Table 3.3 shows the total number of differences in the rules that fire across all the possible inputs in the universe of discourse. It can be seen that there are very few differences present, out of the total of 363 ($11 \times 11 \times 3$ rules) possible firings which could fire, in the largest case only 9 are different. This can be observed in 3.4 in which there are very few points where $z$ is not 0.

In all cases the maximum number of differences between any two controllers is one, it is believed that this occurs because of the nature of the rules used in this system.

| FOU | Differences |
|---|---|
| 1 | 11 |
| 2 | 9 |
| 3 | 8 |
| 4 | 8 |

**Table 3.3:** Rule fire differences between Type-1 and Type-2 Controllers with increasing FOU sizes. The maximum possible differences for a row is 363.

It is believed that this is because the rules have no overlap in them, i.e. there are no rules in which there is a repeated fuzzy set for a given input variable. This can lead to a discontinuity in the control surface because if the input variable is within this range no rules will fire, and unless this is a specifically desired outcome, may lead to degraded performance.

As the FOU configuration moves from standard (FOU size 0) to FOU size 4 the amount of overlap between the middle fuzzy set ('Average') and the two end sets ('Poor' and 'Excellent' respectively) increases, means that there are fewer times when more than one fuzzy set for this input variable will fire.

### 3.3.5 Discussion

While the results shown here have shown some issues — they do not generally show any differences between the difference configurations as was anticipated in Section 3.3.3, some conclusions can still be inferred. It is believed that the results are partly due to the fact that the fuzzy system used here is simple and the rule base is "incomplete" — that is not every possible combination of input sets is explicitly defined within the rule set as is commonly done.

The differences in rules that fire does not give an indication of performance but may help with understanding why certain performance levels are obtained. For example, if a hypothetical configurations 'A' gave 33% performance improvement over another hypothetical configuration 'B' and always fired rule 1 (out of 3 possible rules), it may be possible that these two facts are related. However it is not possible to draw such as solid conclusion from the limited experiments in this section. It is hoped that these experiments in conjunction with those in the next chapter will allow further development of these ideas.

Overall, this methodology for investigating the inner workings of fuzzy systems seems to show surprising outcomes as discussed above. While some limitations have been observed, this seems to be mainly based on the controller configuration used rather than the methodology itself and if the problems with the controller is fixed then the results obtained may improve and allow stronger conclusions to be made. This is tested by reapplying this methodology with a more sophisticated application in a later chapter.

## 3.4 Tipper Controller Application Example

### Experimental Purpose

In this section, the tipping problem is further developed and adapted into an environment that may be considered more traditional for fuzzy logic systems. The intention of this experiment is to show how the controller change its performance profile as the internal controller configuration (such as FOU size) and the external environment is varied. Finally, observations and reasoning and conclusions are presented based on the results obtained. From this it is hoped that the effect of the controller configuration and environmental set-up upon performance can be determined.

This experiment also acts as a preliminary and introductory study into this comparative methodology and so different strengths and weaknesses are sure to be revealed. In later chapters this method of comparison is used in more complex application areas. The application under consideration here is somewhat contrived and simplistic as it is acting as a introduction to the method. However, this means that the entire system can easily be understood, and each input and output can easily be controlled — limiting unexpected and unintended side effects.

**Experimental Design**

The tip controller used is as described in Section 3.2, with the same values selected for the type-2 FOU sizes. Specifically FOU sizes 1,2,3 and 4 are used, these have been selected because between them they cover between 0 and 75% of the universe of discourse which is felt gives a good range of values to start evaluating the methodology.

In order to compare fuzzy logic controllers, the difference in output between type-1 and type-2 based fuzzy will be determined. The output of each controller will be compared with the type-1 and this difference will be used to calculate the RMSE value using Equation 3.1. The results will include graphs showing how the difference between the type-1 and various type-2 RMSE values changes across the input space. This data will also be summed up in tables which will show the total numeric differences, average differences and other supporting data.

$$\sqrt{\frac{(ControllerOutput - IdealOutput)^2}{n}} \tag{3.1}$$

**Hypothesis**

It is hypothesised that as the FOU size of the type-2 fuzzy logic controllers is increased the relative performance will change in response. Specifically, that there should exist an FOU size which obtains better performance than the rest. Logically this should be either at one end of the range under test, (i.e. FOU sizes 0 or 4).

It is also possible that this may not occur, as it may be that too large an FOU size will cause the membership functions of the input variables to all cover the entire universe of discourse. If this occurs, it is likely (depending on the exact method used to derive the FOU) every rule will fire for every input value, generally an undesirable result for control systems.

### 3.4.1 Results

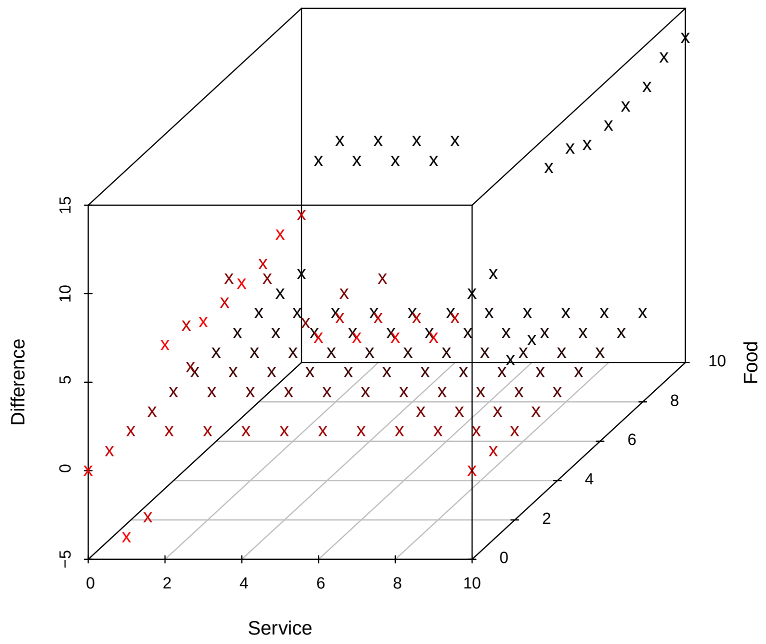
Figure 3.5 show the difference between the type-1 and FOU size 1 type-2 fuzzy logic controllers in a graphical format. Few differences and patterns between the different configurations can be observed directly from this graph. This has led to the use of a

| FOU | RMSE | P-Value |
|---|---|---|
| 0 | 0 | NA |
| 1 | 4.85 | 1.23E-007 |
| 2 | 4.74 | 5.50E-006 |
| 3 | 4.28 | 1.21E-005 |
| 4 | 3.71 | 0 |

**Table 3.4:** RMSE and P-Values obtained when comparing type-1 and type-2 fuzzy controllers. The P-Value is the result Mann Whiteney test with a a smaller P-Value indicating a less significant difference.

| FOU | Number of Differences | Mean Difference |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 44 | 6.8 |
| 2 | 46 | 6.29 |
| 3 | 42 | 5.04 |
| 4 | 26 | 4.87 |

**Table 3.5:** A Difference is counted when the output from the type-1 controller does not match the output of the type-2. The average magnitude of these differences is shown by the 'Mean Difference' column. In both cases a larger value indicates a bigger difference between the two controllers in question.




(a) Type-1 Fuzzy System Compared with FOU Size 1 Interval type-2 system

**Figure 3.5:** $x$-axis and $y$-axis show the combinations of input variables. $z$ axis shows the difference in the output of fuzzy system under test. Large $z$ axis values indicate large differences occur between the two controllers. Colouring used to improve 3D visibility.

statistical test in order to systemically test whether or not these differences are in any way significant.

$$\sqrt{\frac{((Type-1_{O}utput_{V}alue)\ \ --(Type-2_{O}utput_{V}alue))^2}{n}} \tag{3.2}$$

Table 3.5 shows the raw number and average magnitude of the differences between the different FOU sizes and the type-1 configuration (i.e. FOU size 0) which, in this case is considered the 'ideal' value. It can be observed that the number of differences stays relatively constant across all of the FOU sizes, constituting approximately 30% of the total number of input variable combinations. The magnitude of the differences tends to decrease as the FOU size is increased, which is somewhat counter intuitive, as it would be thought that with a more significant difference between two controllers, such as larger FOU sizes would give larger differences however this does not seem to be the case, the reasoning for this is discussed in more detail below.

Table 3.4 shows the results of statistical tests performed on the results obtained from the experiments. Single sided Mann Whitney tests were used to test the significance of the differences between the type-1 (i.e. FOU size 0) and the rest (type-2 based with FOU sizes between 1 and 4). It can be seen that the P-Value obtained in each case is very small, indicating there is no significant difference between the different configurations which supports what can be observed from the graphs and other supporting data.

### 3.4.2 Discussion

Based on the results obtained from this experiments, one of several conjectures can be proposed, firstly it may be that type-2 fuzzy logic does not make a significant difference in performance under any conditions. Secondly, it may be that the methodology used for the comparisons is unfit for performing these comparisons. The ideal value used in calculation of the RMSE comes from a fuzzy logic controller under test — it would be better if this value could be obtained from a separate source, and this is what is addressed in the next section.

## 3.5 Tipper with Ideal Value Comparison

### 3.5.1 Experimental Purpose

The application considered so far in this chapter has been identified as somewhat contrived and simplistic, this is justified as it is acting as a introduction to the comparison methodologies being developed. This means that the entire system can easily be understood, and each input and output can easily be controlled — limiting unintended side effects. However it does currently lack some concepts that are commonly used in control applications, one of which is addressed here that of an ideal value.

In most control applications, an ‘error’ value is generally used as one of the main inputs into the system [62]. In order to calculate this error value, there must also be an ideal value which is specific to the application the controller operates in [50]. In this chapter so far, this has been omitted for simplicities sake, however it is felt that this is to the detriment of the results obtained and therefore what conclusions can be drawn from the experiment in question. This is therefore the focus of this section, where a simple ‘ideal’ value introduced to observe how this affects the results obtained in the comparisons made.

### 3.5.2 Experimental Design

| | Service | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | None | a | a | b | b | c | c | d | d | Av | Av |
| 1 | a | a | b | b | c | c | d | d | Av | Av | e |
| 2 | a | b | b | c | c | d | d | Av | Av | e | e |
| 3 | b | b | c | c | d | d | Av | Av | e | e | f |
| 4 | b | c | c | d | d | Av | Av | e | e | f | f |
| Food 5 | c | c | d | d | Av | Average | e | e | f | f | g |
| 6 | c | d | d | Av | Av | e | e | f | f | g | g |
| 7 | d | d | Av | Av | e | e | f | f | g | g | h |
| 8 | d | Av | Av | e | e | f | f | g | g | h | h |
| 9 | Av | Av | e | e | f | f | g | g | h | h | Max |
| 10 | Av | e | e | f | f | g | g | h | h | Max | Max |

**Table 3.6:** Ideal values for tipper output. Values for each character given in Table 3.7

The tip calculation controller used is the same as the one described in Section 3.2, with the same values selected for the type-2 FOU sizes. This is done in order

| | | | |
|---|---|---|---|
| 1 | | None | 0 |
| 2 | a | Low Cheap | 3.33 |
| 3 | b | Cheap | 6.67 |
| 4 | c | Hi Cheap | 10 |
| 5 | d | Average | 13.33 |
| 6 | e | Av | 16.67 |
| 7 | f | Low Gen | 20 |
| 8 | g | Gen | 23.33 |
| 9 | h | Hi Gen | 26.67 |
| 10 | | Max | 30 |

**Table 3.7:** Exact Tips table lookup. The numeric values have been selected to maintain a constant difference between each row. Values have been selected to maintain constant difference.

to minimise what changes between each set of experiments and introducing a single aspect at a time into the experimental set-up.

Every integer combination of inputs will be used for each run. This means there will be a total of 122 runs of each controller type — a large enough sample size to allow the comparisons made to have a good level of confidence behind them. For each run, values will be obtained from a specific fuzzy controller and from the result of a look up of the inputs in Table 3.6, and based on these two values the level of performance will be determined.

The difference between the two values will be the main focus of interest in terms of performance. This will be quantified by calculating a new RMSE value, using the formulae shown in Figure 3.3. In this formula, $n$ is the number of runs performed from the controller totalling 121 as discussed above. This RMSE value will provide the basis for the comparisons between the difference controller configurations. As is usual a large RMSE value indicates that there is more of a difference between the two values under comparisons, and assuming the look up table is considered optimal, then the performance of the fuzzy controller is therefore worse.

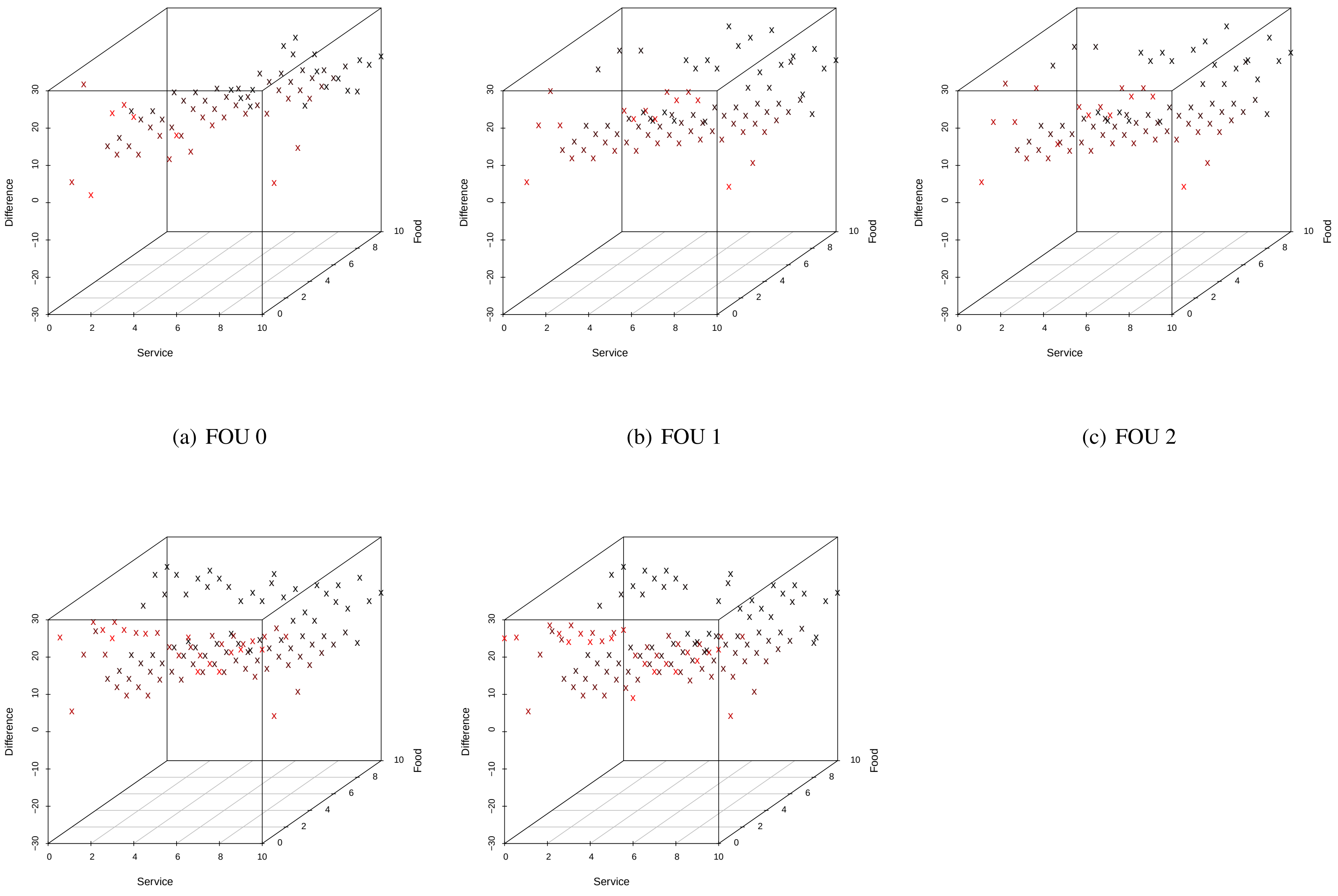
$$RMSE = \sqrt{\frac{(ControllerOutput - IdealOutput)^2}{n}} \tag{3.3}$$

Several assumptions and assertions are made in the process of performing these experiments in order to keep them consistent and simple. These assumptions are:

- The restaurant idea is simplified. There is only considered to be one food dish and a single member of front of house staff. This minimises some sources of variability that should be considered in more sophisticated and complete experimental set-up.
- There is only a single customer therefore the same standards and so forth are required to give the same output. Theoretically this means that the same customer will repeatedly visit the same restaurant multiple times in the same day, and in some cases the same customer may be in the same restaurant multiple times simultaneously. Once again this is a simplification of normal situations.
- The result of the static lookup table is considered to be correct, ideal and optimal for the purposes of this experimental set-up.

### Hypothesis

It is hypothesised that as the FOU size of the type-2 fuzzy logic controllers is increased the spread of results will increase, which should be able to be seen visually. Specifically, while the direction of the change is not know, the RMSE is hypothesized to gradually change in a systematic and relatively linear manner up a fixed point. It is anticipated that in a complex enough scenario and over enough different FOU sizes, the performance will reach a peak and then begin to degrade, whether this happens in this simplistic set-up however is not known.




(a) FOU 0 (b) FOU 1 (c) FOU 2

(d) FOU 3 (e) FOU 4

**Figure 3.6:** *x*-axis and *y*-axis show the combinations of input variables. *z* axis shows the difference in the output of fuzzy system under test and the ideal output. Large *z* axis values indicate large differences occur between the two. Colouring used to improve 3D visibility.

| FOU | RMSE | Mean | Standard Deviation |
|---|---|---|---|
| 0 | 7.92 | 10.2 | 6.9 |
| 1 | 11.11 | 7.73 | 8.07 |
| 2 | 11.13 | 7.81 | 8.03 |
| 3 | 11.44 | 8.45 | 7.81 |
| 4 | 11.83 | 9.15 | 7.6 |

**Table 3.8:** RMSE Values in comparison with the static lookup table

### 3.5.3 Results

Figure 3.6 shows 5 figures in which the 5 different FOU sizes under test are compared to the static lookup table results. The difference between these two values is plotted along the $z$ axis with the $x$ and $y$ indicating the set of inputs used. It can be observed that overall there are a few differences between the different figures, though they are not identical. This is once again showing that this application does not give sufficient ability for better or worse performing controllers to show their abilities.

Table 3.8 shows the RMSE values obtained when comparing the different FOU sizes under test to the look up values found in Table 3.6. One obvious point of interest is the increase of RMSE value when moving from type-1 to type-2 fuzzy control, observable when the FOU size increases from 0 to 1. After this point, there is a very slight increase for each FOU size increase. While it is not thought that these differences are significant in themselves, its is believed that in a more developed situation these differences would increase in magnitude, which would closer match the hypothesis made in section 3.5.2.

The mean difference, as shown in Table 3.8 is largest for the FOU size 0, and smallest for FOU size 1. The variance shows a different picture however, with the smallest value, indicating better performance, being for the FOU size 0, confusing the picture about which controller is the best performance, even considering the restrictions present with the application used.

### 3.5.4 Discussion

The introduction of an 'ideal' value, here represented by the values obtained from what is being termed the 'static lookup table', presents a small amount of additional evidence that this methodology has potential for making the the differences between different fuzzy logic controller configurations observable. While the results obtained in this section have once again not been entirely convincing, certain artefacts such as the way in which the RMSE value changes as the FOU sizes increases from 0 towards to the max value of 4 indicates that under different conditions, discussed in the next section, this methodology could well provide a useful means of comparing fuzzy controller configurations.

An additional reason for the lack of difference between the different configurations may be that the controller and environment in which it was used are not sufficiently complex for type-2 to show significant improvement — it has often been hypothesized that type-2 control only show benefits when the environment in which they operate has significant sources of uncertainty, randomness or variation. The experiments performed here do not introduce anything in the way of randomness into the environment, and hence do not present a situation in which type-2 control can show better performance therefore this is the point that will be addressed in this next and final section.

## 3.6 Tipper with Sources of Variation

### 3.6.1 Experimental Design

The inputs to the fuzzy system under test will use the same integer progression as used in the previous set of experiments going from 0 to 10. However in contrast to the previous experiments, in this batch of experiences a source of variation will be introduced to generate small random values which will be added to the normal inputs into the system. These random numbers will follow a Gaussian distribution with a mean and standard variation of $1$.

The RMSE will again be calculated between the ideal values, calculated using the results of the look up table and the output of the fuzzy logic controller under test. The mean and standard deviation of the differences will also be calculated in order to give a second view of the data.

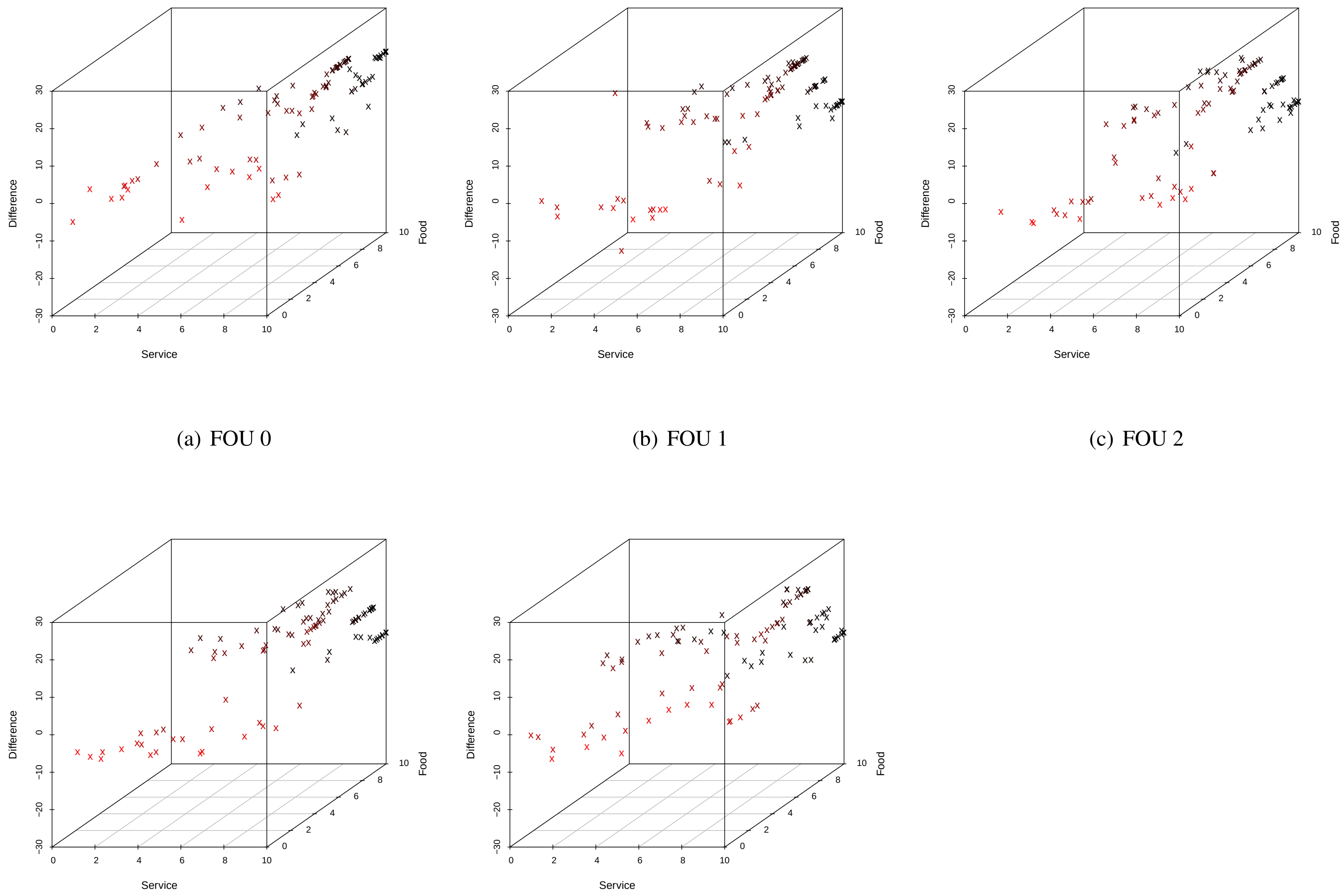
### 3.6.2 Results

| FOU | RMSE | Mean Difference | Standard Deviation |
|---|---|---|---|
| 0 | 15.53 | 13.07 | 8.43 |
| 1 | 14.25 | 10.28 | 9.92 |
| 2 | 13.9 | 10.65 | 8.98 |
| 3 | 14.01 | 10.56 | 9.26 |
| 4 | 13.28 | 10.34 | 8.31 |

**Table 3.9:** RMSE indicates the mean difference between Ideal Output and Type-1. Standard deviation indicates the range of the difference obtained for the given FOU size

Figure 3.7 shows the raw difference between the outputs of the fuzzy logic controller (one figure per FOU size) and the table look-up. The $x$ and $y$ axes indicate the two inputs into the system once the random value has been added, as has occurred with the previous experiments, absolute levels of performance are hard to determine from these plots. In order to address this, box plots have been plotted (Figure 3.8). From these plots it can be observed that the differences are not large from a statistical point of view — the medians show are all within 4 units of each other.

Table 3.9 enumerates the results found and shows the RMSE and how it changes as the FOU size is increased. It can observed that these results are different from the previous couple of sections as the RMSE decreases as the FOU size is increased. At first glance this implies that under variable conditions type-2 fuzzy control, especially those with large FOU sizes will tend to perform better. However is is felt that the results from this single, simple study are not sufficient to make this with any certainty. However this work gives a good starting point to make further investigations of this type, as performed in later chapters. The mean and standard deviations of the differences between the look up table and the output of the fuzzy logic controller are shown




(a) FOU 0

(b) FOU 1

(c) FOU 2

(d) FOU 3

(e) FOU 4

**Figure 3.7:** $x$-axis and $y$-axis show the combinations of input variables. $z$ axis shows the difference in the output of fuzzy system under test and the ideal output once variation has been added to the system. Large $z$ axis values indicate large differences occur between the two. Colouring used to improve 3D visibility.

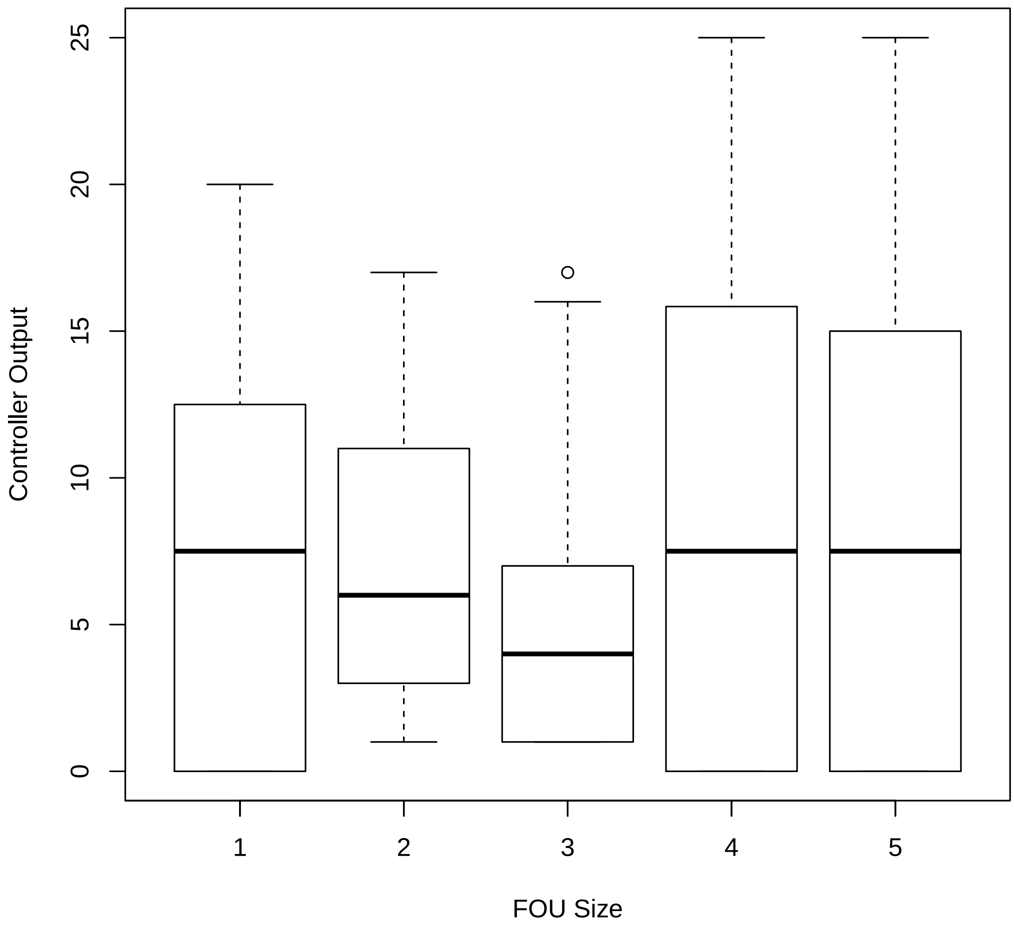



**Figure 3.8:** Box plots of the distribution of the outputs of each controller under test. Each plot represents the distribution of outputs from a different FOU size

in the third and forth columns of the table. These values present the same picture as the RMSE value, in that the largest change in values is located between the FOU 0 and FOU 1 points, i.e. between the type-1 and type-2 controller, further indicating potential for type-2 performance improvement over type-1.

### 3.6.3 Discussion

This set of experiments show the most significant differences between the controllers under test so far. However it is difficult to draw firm conclusions beyond this statement. This once again implies that this application is too simple to be able to make firm conclusions about the performance of type-1 and type-2 fuzzy logic controllers and when or if a type-2 would out perform a type-1 based controller.

The application used here had a particularly narrow universe of discourse, so the ability to choose larger intervals was limited. Results such as the mean difference show very small differences between FOU sizes 1,2,3 and 4, which certainly suggests larger spreads of FOU size would be worth investigating in future work.

Overall the introduction of a source variation causes the experimental output to more closely match the hypothesis made in Section 3.4, specifically type-2 fuzzy logic control does appear to out perform type-1 as outlined by the differing RMSE values obtained. This supports the hypothesis that this comparative methodology is capable of finding differences between type-1 and type-2 controller types. However with such a limited experimental set-up care must be taken not to make the case to strongly — additional experimental work is required before any general statements can be made with any confidence.

## 3.7 Chapter Discussion

The goal of this chapter was to assess how well the comparative methodology performs in differentiating different types and configurations of fuzzy logic controller within the context of a simple application. The ideal goal of this work would have been to be able to show how the methodology was able to demonstrate that each FOU size and fuzzy logic controller gave a very specific performance profile across the different experiments. An example of this would be that as the FOU size increased from 0 to 4 the performance would increase by 10% at each FOU size. This sort of result was not found however, and in general the number differences observed were minimal.

From the rule based experiments shown in Section 3.3 it can be seen that the degree by which the FOU changes the control surface obtained, and therefore the rule differences are not large. One anticipated reason for this is the application used and the general simplicity of its design, this is discussed below as it this problem also seems to manifest in the results of the second set of experiments and therefore will be discussed as an overall shortcoming with the work performed in this chapter.

A second point of note when looking at the results obtained is that, even though there are very few differences, the experiment in which there are the most is the one which has the smallest difference in FOU size, i.e. FOU size 1 in comparison with the type-1 system. This is the inverse to what would was hypothesized, as it was expected that a larger FOU size would cause a larger difference as FOU sizes cause the different fuzzy sets to be more disparate.

When looking at the results in Section 3.4, shortcomings in both the application selected and the methodology used can be observed. The application based shortcomings are less of an issue as it has been stated above that the application is a non realistic and is used merely as an experimental problem for the testing of the methodology — it has no practical application and is easily addressed by changing application. However the issues identified with the methodology must be addressed in some way before it can be used in further work and this is discussed further below.

The experimental design in Section 3.4, did not include an ”ideal value” to define performance levels, such as used in many control applications, where an “optimal” output is known. Without such a value, direct comparisons between the various configurations are somewhat difficult to perform. This has been somewhat countered by comparing each FOU size with the FOU size 0 (that is equivalent to type-1), which is being considered as the “base” performance level, and does help towards the stated objective of comparing type-1 and type-2 fuzzy logic systems, however it is hypothesized that an external ideal control value would improve the comparisons made, and so this was attempted in the following section — with limited success.

It is found that the introduction of this ideal value as done in Section 3.5, and the resulting ability to calculate a more realistic RMSE value did not give vastly different results, and the reason for this was narrowed down to the simplicity of the experiment — there was no source of randomness or variation within the environment.

The final set of experiments in Section 3.6, added a source of variation into the application and this was found to help a difference between type-1 and type-2 fuzzy logic control become more apparent. This result was successful and shown in the increased magnitude of the differences between the RMSE and mean differences values. This suggests that this is an important direction for future work — and the introduction of more sources of variation and randomness in a systematic way would be interesting to analyse. Doing this within the context of the current application would be somewhat difficult due to its simplistic nature, and so a new application would be desirable as discussed below.

In this chapter the FOU size of the controllers used do not in general cause significant differences in the results obtained in each of the experiments. This may be due to the relatively small differences between each of them with only a change of 4 between the largest and smallest FOU sizes. From this it can be determined that in the situations here, small changes to FOU size do not cause major differences in the results obtained. In future work this FOU difference will be increased in an attempt to try and observe what sorts of values are required to cause significant changes.

Overall from the work performed here it can be concluded that moving from type-1 to type-2 based fuzzy logic control will not necessarily give a performance increase. In several of the experiments explored here no differences in performance at all were found, especially when there is no variation present within the experimental scenario. To investigate this further, in the following chapters a significantly more complex, applications and environments will be introduced as a new test application to see if they will present more promising results.

The first and most obvious change to be made is to the application that is being used to test the fuzzy logic controllers. The results obtained here generally show minimal differences, and as this application is so simple, with only the final sets of experiments showing easily observable differences. A more sophisticated application may involve using more complex inputs (potentially with relationships between the inputs), a greater number of membership functions with more complex shapes. This increase would be matched by a corresponding increase in the task difficulty that controllers must perform with the combination of these two factors allowing "better" controllers to show themselves.

Type-1 and type-2 based fuzzy logic systems are the most prolific in the literature, however these are not the only varieties that are feasible to run in real time on current generation hardware. Dual surface and Non-stationary are two types of fuzzy logic control that have been studied and have shown to give (in general) performance increases over standard type-1 and their use in the new application will be considered, in addition to type-1 and type-2 based controllers to observe how these less standard varieties perform in comparison. Alongside the additional types of fuzzy logic control, the changes of the size of the FOU used in this chapter are small, this is mostly due to the universe of discourse having a relatively small total area. In future chapters, larger ranges will be used for the input variables, which in turn will allow a larger number of different FOU sizes to be investigated without having to resort to measures such as floating point.

In the next chapter, a significantly more sophisticated application is introduced, that of robotic sailing, in which the task in its most simple form is for the controller under test to steer a robot around a defined course. The lessons learnt from this chapter are applied to the new application area, with the environment made more sophisticated. The controllers tested are also more considerably more complex with a larger rule base along with more input and output sets and a larger universe of discourse as discussed here.

Based on the shortcomings discussed above, the stated objectives in Section 3.1 have not been fully achieved in this chapter. Changes in performance between type-1 and type-2 have been shown in several scenarios, however in general very differences are present in this chapter. With the changes discussed in this section, it is hoped that these problems can be addressed in later chapters and it that they will give better results using the more sophisticated applications tested later than those in this chapter.

## 3.8 Summary

In this chapter two approaches for comparing different configurations of fuzzy logic controllers are developed. The results obtained when they are applied to a simple tipping controller application do not generally show significant differences except in the final set of experiments in which a source of variation was added to the environment. The reasoning for this lack of change is discussed in Section 3.7 but are mostly intertwined with the fact that the application is very simple with few opportunities for better applications to become apparent.

The lack of differences in this application are not of major concern as the main objective of this chapter is to show that the methodology used for performing the comparisons with differing FOU sizes was effective and it felt to have been achieved. In the next chapter significantly more complex experimental set-ups are described using a robotic sailing application in both real world and simulated environments. It is believed that this application provides much more scope for performance differences and so should allow the comparison methodology described in this chapter to demonstrate more differences between the various fuzzy controllers.

# 4

# Robotic Sailing Background

## 4.1 Introduction

In this chapter the FLOATS (Fuzzy Logic Operated AuTonomous Sailboat) platform is introduced. FLOATS is a robotic sailing platform that provides interfaces for simulation and real-world robots.[1].

In Section 4.2 background information is discussed, including a reiteration of the research question. In addition the varieties of fuzzy logic controller used are introduced. This is followed by Section 4.4, in which the hardware used in the experiments, both real world and simulated, is described. Next, Section 4.4.3 provides information about the software used for performing the experiments including the design of the fuzzy controllers used in the majority of the sailing boat experiments. Section 4.3 describes the methodology that is employed for performing experiments and analysing the results obtained. Section 5.3 presents a small discussion on control surfaces and how they change with different controllers and why this is relevant to the work.

[1] AUTHORS Note — The methodology is based off work previously submitted and accepted at FuzzIEEE [7] [8]

## 4.2 Background

### 4.2.1 Robotic Sailing

As discussed in the previous chapter, autonomous sailing using small, unmanned craft in restricted environments such as ponds, small lakes or swimming pools, is used as the one of the two experimental domains in which experiments are performed. Previous work discussed in Section 2.6.2 shows that this application presents a challenge to standard type-1 based controllers because of the potential of rapid changes to the inputs which are used to control the speed and direction of the boat. It is believed that with a good design, fuzzy logic can provide a satisfactory solution to this problem as the use of membership functions and rule base allow and good mirroring of the human sailor decision making process.

It must be stated that the intention of these experiments is not to develop the most effective solution for a sailing robot. Rather, the application is used to compare the relative effectiveness of various fuzzy-based controllers. The effect of different parameter settings and the reasons for the occurrence of relative performance levels is also a specific point of interest.

### 4.2.2 Reiterating the Research Question

The main research question previously stated in 1.2 on Page 3 addressed in this thesis is the formally stated as 'What combination of factors are necessary for more sophisticated fuzzy controller types such as interval type-2 to consistently out perform type-1 fuzzy control'. In order to investigate this, several fuzzy logic controllers are compared using a variety of different approaches. Firstly, direct comparisons, using their observed outputs, such as which rules fire for a given set of inputs, and control surface shapes are investigated; Secondly by using these controllers in-situ within the context of control based experiments is studied. Both of these approaches are executed using several different sets of conditions, where parameters such as environmental variation, task difficulty and the internal configuration of the fuzzy logic controllers are gradually changed. This is done in a systematic manner to observe how of each factor affects the performance of the controllers in relation to each other.

Overall, the intention of this work is to observe the effect of the environment and parameters of the fuzzy controllers upon performance. This will work towards the aim of being able to develop methods by which optimal controllers can be determined for a given situation.

In order for our research aims to be achieved, several objectives must be met:

- Perform an in-depth and thorough comparison of different FOU sizes and observe the effects they have upon the performance of standard type-2 fuzzy logic controllers. It is hoped that this will result in being able to pick an optimal FOU size for given circumstance, or at the very least provide a better understanding of what effect FOU size will have on a controller in different conditions.
- Perform an investigation and analysis of increasing levels of environmental variation on the performance of type-2 fuzzy logic-based sailing systems. This in turn will allow for both quantification of the variation and use of this quantification to support the controller development process.
- There are several modifications to standard type-1 and type-2 fuzzy logic systems but their exact performance profile and their benefits are unknown and in question. An investigation to observe their performance in comparison to the standard type-1 and type-2 varieties is required to develop an understanding of their effects and when or indeed if they provide better performance in the studied application areas.

### 4.2.3 Overview of Controllers Used in this Thesis

Four main controller types derived from standard fuzzy logic are put under test: type-1, interval type-2, NS and DS. The previous chapter has described the theory behind these controllers, how they differ from each other and why they present an interesting selection of controllers for use in experiments of this type. In this section more concrete details about each of the implementations of the controllers is given.

Type-1 fuzzy logic is used as a baseline controller for all of the experiments performed in this thesis. This controller is considered the most simple type of fuzzy logic, according to Melin and Castillo[67] and has long been the most used fuzzy system in real-world scenarios of control. Furthermore, it is the most applied method in embedded systems where resources such as memory and processing power are scarce. While type-1 fuzzy logic does not have the flexibility of more sophisticated types, the simplicity of implementation and the way it mirrors the human decision making processes still make it a powerful tool. Therefore, it is an obvious inclusion to be considered in a comparative work.

Non Stationary fuzzy logic was chosen to be included in this work as it is hypothesized that the random perturbation of the MFs could be used to model the variation in the readings of sensors in the boat. NS fuzzy logic has so far not been widely used in the field of fuzzy robotics. It is hoped however, that this work may highlight under what circumstances if there is any, NS fuzzy logic is preferable to other types. NS has an advantage over the more sophisticated controllers, such as interval type-2 in that it is very easy to implement from a given type-1 controller, though the efficiency of such an implementation is questionable.

Usually considered more sophisticated and flexible, general type-2 fuzzy logic are often presented as a preferable solution to situations in which there is a great deal of variation within the environment in which it will operate. General type-2 fuzzy logic, as its name suggests, is the most general variety of fuzzy logic that is widely used. Its potentially high computational cost has led many parties to investigate interval type-2 control as an alternative. The robot used in later chapters has somewhat restricted resources available and therefore interval type-2 fuzzy logic is used in this thesis.

One of the most commonly used methods for obtaining an output value from a type-2 set is the KM iterative procedure [46], as mentioned in the previous chapter. However, it seems unintuitive to always take the mean of the upper and lower values of the output set, as this discards some of the information obtained from a potentially computationally expensive operation. Dual Surface control is an approach developed to overcome this issue, as it introduces an algorithm to select between upper, lower and mean values, based upon information present in a standard type-2 controller. The intention the inclusion of this derivation of standard type-2 fuzzy logic is to further improve performance over type-2 with minimal changes to its overall structure.

## 4.3 Simulation Methodology and Analysis of Results

For each combination of controller, controller parameter and course layout, 30 repetitions are performed under simulation. This allows statistical comparisons to be made with some degree of certainty as it reduces the variability to a tolerable level.

### 4.3.1 Common Experimental Design for Experiments III and IV

In the Sections 5.4 and 5.5 of Chapter 5 the FLOATS platform is leveraged using simulation in order to observe correlations between the various aspects under investigation as well as to address some of the shortcomings identified in the previous chapter related to the simplicity of the environment and experimental design. The aspects of interest are: (1) the FOU size of the type-2 controllers and associated parameter vales for Dual surface and Non stationary varieties; (2) the levels of variation and randomness present in the environment; (3) the different controller types under test; and (6) the differences in performance that occur between these controllers.

Section 5.4 on Page 107, investigates how performance changes when varying the type of controller and the associated configuration values. Type-1, interval type-2, NS and DS controllers with their associated parameter values are described and used in experiments. The objective is to determine an association between parameter values and performance. Experiments are performed using different combinations of parameter value and controller type. During these experiments, the level of difficulty is gradually increased to observe how this affects the RMSE value obtained. In addition to the course difficulty, three different levels of environmental variation are defined: 'low', 'medium' and 'high'. This variation is aimed at causing further differentiation in the performance of the various controllers, allowing differences and trends to be more easily observed.

This is followed by Section 5.5 on Page 115 which is limited to a comparison between type-1 and interval type-2 fuzzy logic-based controllers under varying conditions. The size of the FOU is varied and its effect upon performance under increasing levels of difficulty is observed and discussed. The difficulty of the task is defined by two main characteristics specifically: the amount of variation within the environment and the difficulty of the programmed course. The level of variation is set at different levels — denoted by letters ‘A’ to ‘I’, with ‘A’ representing no variation and ‘I’ denoting a large amount of variation.

The main differences between the two experiments is in the breadth of their coverage. The first set sacrifices the number of different environment variables, including course difficulty and environmental variation combinations to cover more types of controllers. This contrasts with the second set where only interval type-2 is considered. This allows significantly more levels of uncertainty to be used over more course layouts. It is anticipated that, by performing these two different sets of experiments the desired correlations can be observed. Furthermore, comparative techniques used so far can be validated and evaluated. With this validation process it is hoped that any shortcomings that may be present in the methodology or controller design can be identified before the real world experiments are performed.

### 4.3.2 Case Studies Examined

Several case studies were used in this thesis in an attempt to answer the research question. This was an appropriate approach for this kind of work because each case study can be used to directly address a specific aim as specified in Section 1.1.

Initial work involved a very simple application set-up in which a Tipping controller is introduced and used to study various methods of comparing different configurations of type-1 and type-2 fuzzy controllers. The results found are disappointing but the reason is hypothesized to be due to the simplicity of the environment and controller rather than major flaws in the methodology employed.

Sections 5.4 and 5.5 present two studies, described above which tries to observe how parameters, including size of the FOU, affects the performance of fuzzy logic systems such as type-1, type-2, DS and NS in a simulated sailing boat application. This was done using very simple courses for the boat to be steered around while varying the internal configuration of each controller type. RMSE (Root mean squared error) was used to compare the performance level and it was found that type-2 based fuzzy logic systems most often performed the best out of those under test (signified by a lower RMSE), however there were several data points which refute this, indicating extra work is required.

Finally, in Chapter 6, real world experiments are used to observe how the inherent variability affects the results obtained under simulation. A pilot experiment is used to try and eliminate as many problems from the experimental technique as possible, However the final results obtained in a main study, while giving some points of interest are overall disappointing.

### 4.3.3 Course Description

In each of the simulation based experiments performed there is a scheme to the design of the courses that the controller is designed to complete. Different numbers of turns, each with differing angles of turn required will used to vary the difficulty of a given experiment, the different courses that will be used is shown in Figure 4.1.

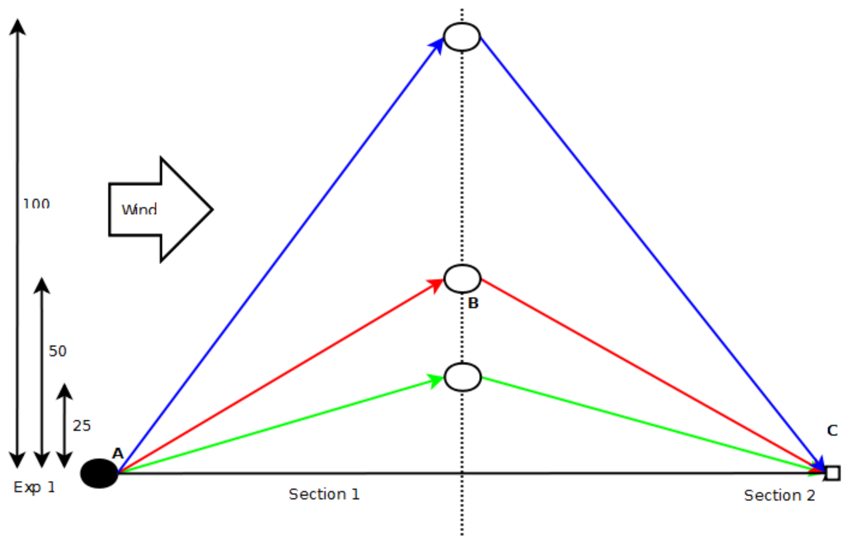



**Figure 4.1:** Each coloured line represents a single experimental course layout. The white circles represent possible end points and the black circle the start point. The angles required for the first turn are 5.71° (green line courses), 11.42° (red line course) and 21.84° (blue line courses) for 25, 50 and 100 meters vertical movements respectively. Not to scale.

| Variability | Variability Score | Lower Limit | Upper Limit |
|---|---|---|---|
| None | 0 | 180 | 180 |
| Low | 1 | 160 | 200 |
| High | 2 | 140 | 220 |

**Table 4.1:** Levels of variability

### 4.3.4 Statistical Analysis Methods

In the experiments performed in Sections 5.4 and 5.5, one-sided Mann Whitney tests are used to determine if RMSE values of different experiments show statistically significant differences. This test was selected because the data generated was independent and non-parametric in nature, which was tested by comparing the mean with the median of several data sets. Unless otherwise specified in the text a p-value of 0.05 was deemed sufficient to reject the null hypothesis in each of these tests.

### 4.3.5 TrackSail

The simulator used, TrackSail [49], is a Java based sailing game that has been modified in order to allow the control of the boat to decided by an external source — which in this case is the fuzzy logic controller under test. To define the environment in a given experiment, a configuration file is passed into the simulator. This file defines the following parameters:

| Wind Speed $\rightarrow$ | | **None** | **Low** | **High** |
|---|---|---|---|---|
| Direction Change $\downarrow$ | **None** | A | D | G |
| | **Low** | B | E | H |
| | **High** | C | F | I |

**Table 4.2:** Wind change configuration definitions

| Direction Change | Variability Score | Lower Limit | Upper Limit |
|---|---|---|---|
| None | 0 | 180 | 180 |
| Low | 1 | 160 | 200 |
| High | 2 | 140 | 220 |

| Wind Speed | Variability Score | Lower Limit | Upper Limit |
|---|---|---|---|
| None | 0 | 7 | 7 |
| Low | 1 | 4 | 10 |
| High | 2 | 1 | 13 |

**Table 4.3:** Upper and lower values of wind speed (m/s) and direction (°) alongside variability score

- Course name
- Laps
- Max elapsed time
- Wind speed upper limit
- Wind speed lower limit
- Wind direction upper limit
- Wind direction lower limit
- Wind update rate
- List of way points

Of these items numbers 4 to 7 are set to the values defined in Table 4.3. The way points are defined to describe the course attempted in the experiment such as those shown in Figure 4.1.

| Command Id | Command | Description |
|---|---|---|
| 1 | set sail x y | Sets sail x to y degrees |
| 2 | set rudder x | Sets rudder to x degrees |
| 3 | set waypoint x | Sets waypoint to navigate to |
| 4 | get sail | Gets sail position |
| 5 | get rudder | Gets rudder position |
| 6 | get windspeed | Gets wind speed |
| 7 | get winddir | Gets wind direction |
| 8 | get compass | Gets current direction of boat |
| 9 | get waypointdir | Gets direction of current waypoint |
| 10 | get waypointdist | Gets distance to current waypoint |
| 11 | get waypointnum | Gets id of current waypoint |
| 12 | get easting | Gets current latitude |
| 13 | get northing | Gets current longitude |

**Table 4.4:** Commands that can be sent to the simulator. All commands trigger the simulator to respond with a single byte indicating the value requested or success of the command.

While running Tracksail provides an interface that can be accessed by any programming language which supports POSIX sockets via a local loopback interface (defined as 127.0.0.1) on port 6667. The fuzzy controller rig used in Chapter 5 connects directly to this socket in order to exchange information bwith the simulator. Once connected, a ASCII based request/response mechanism is used to send and receive data. Table 4.4 lists the commands implemented by TrackSail.

Minimal changes were made to the Tracksail application code-base. The changes made consist of bug fixes to fix corner conditions which often resulted in a simulation crash which in turn would be detected by the rest of the software rig, causing the data to be discarded and a replacement experiment to be performed. In addition hooks were made in the initialisation phase in order to make the automation easier to execute and easier to detect when failures occurred within the simulation.

### 4.3.6 Simulation *versus* Real World

Simulation by its very nature is a simplification of the simulated real world environment. This simplification has positive and negative effects on experimental work performed using simulation. The ability to control how inputs are passed into the controller under test is a strong benefit for this type of work, allowing the rate and distribution of the input variables to be tightly controlled and repeated across different experimental scenarios, making it easier to draw conclusions from the results obtained. This more controlled manner of input however can have some downsides — if all the inputs are drawn from a particular distribution or follow fixed patterns then it may be that results obtained contain bias, which in turn will not allow the results to be generalised effectively. Simulation also lends itself well to automation, increasing the rate at which experiments can be run and therefore increasing the sample size of the results obtained meaning, for example, that statistical tests can be made with increased confidence.

Even those sources of variation which are held in common by both approaches are simplified to a greater or lesser degree in simulation. The simulator does not implement concepts such as tides, currents or other hydrodynamic properties of large bodies of water. Real-world objects such as wildlife and fixed obstructions (buoys, rocks or debris) are also not present in the simulator, both of which would indirectly cause deviations in the course taken by the robot. Wind in a real world environment is a complex process and measuring its properties accurately is difficult, with several issues present that do not occur during simulation such as inaccuracies, update rate problems and potential imprecision. Even if the inaccuracies of wind sensors are ignored, many things may alter the wind reading real-world sensors obtain — even the height of the sensor can cause deviations in the value obtained.

The differences between the two approaches discussed here demonstrate the need for both techniques. However, in order to be able to effectively compare results between the two environments, care must be taken to keep as many factors constant as possible. This includes the internal configuration of the fuzzy controllers under test (e.g. the FOU size in the case of type-2 controller) in addition to the task that is being attempted.

## 4.4 Hardware Set-up

In this section the hardware that is used for the sailing boat based experiments in this thesis is described. The experiments presented in later sections are separated into simulation based and real world based environments, each with differing aspects. Simulation was used in the initial work (Section 4.4.2) due to its ability to rapidly produce many experiments and to give precise control over sources of variation. Real world experiments are used to verify the work done in simulation, in an environment without the rigid control (Section 4.4.1).

### 4.4.1 Real-World Hardware

The robotic hardware for the real world sailing experiments is shown in Figure 4.2. The processing hardware is based around a gumstix-based micro PC containing an ARM-7 CPU running at 600Mhz with 128Mb of RAM and running a version 2.6.21 of $\mu$ linux. This PC connects via an RS-232 serial line to a PIC microprocessor, which provides access to all the on-board sensors — including wind direction, GPS and digital compass. The PIC also provides access to the actuators on the system — two sails and a magnetically coupled rudder. The interface between the PIC and the gumstix is an custom ASCII-based protocol.

The sails are controlled by electric motors. This gives them approximately 290° of rotational freedom for the front sail and 310° for the rear sail. Software present in the motor controller module of the PIC limits this rotation down to approximately 280° for the front motor. This prevents the the two sails from hitting each other, which could potentially cause motor burn out. The fine grain control of the motors allow the sails to be moved in approximately 2° increments.

The rudder is built using a magnetically coupled servo. This means that there is no physical linkage between the motor and the rudder itself. Therefore, no hole in the hull


**Figure 4.2:** The robotic boat platform

is required, as with some methods of rudder control such as a direct drive shaft. The rudder has approximately 90° of movement freedom — 45° in each direction, with an accuracy of approximately 3°. The main potential issue with the rudder mechanism is the possibility of detritus becoming caught in it. However with appropriate care this can be avoided, especially when the test water body is carefully selected.

GPS information is provided by a Telit GE863 GPS module attached via a serial line to the PIC processor. For the purposes of these experiments, it is used to provide location and current system time for the gumstix. As this module does not provide A-GPS capability, it may be inaccurate and take significant time in order to gain an accurate fix. Due to this limitation, an additional external GPS source is used to check the values received by the on-board transceiver. In this case, this source is an Apple iPhone 4S — which includes an A-GPS (Assisted GPS) module. Assisted GPS uses additional data sources, including Wifi and GPRS, to obtain a more accurate fix than that obtained using a standard GPS module.

The boats current bearing is obtained from a GY-26 digital compass, which interfaces with the PIC using an I2C line. This module also has the capability to read its current yaw and pitch [1]. However, for the purposes of these experiments this functionality is not used. This sensor provides bearing accuracy to within 1°.

The wind direction is read from a digital motor encoder which is rotated by a wind vane attached to the top of the taller sail. It has a full 360° of rotational freedom and is accurate to within 3°, which is based upon the smallest electrical pulse the encoder can capture.

Data and commands are relayed to and from the gumstix unit by way of 802.11b wifi via the gumstix Ethernet port and an on-board wifi base station. Standard unix tools including ssh and sftp are used to control what the gumstix does, including starting experiments, setting way-points and retrieving logs. The base station is placed on the boat because the connection to a wifi hotspot by the gumstix can cause the CPU to block while waiting on wifi initialisation responses. This causes the fuzzy software to stop or slow down its operations — an undesirable effect.

The boat is powered by 18 AA NiMH batteries, which make up some of the weighting of the keel and provide power for approximately 6 hours of experimentation. Charging is performed by a mains powered charging unit and takes 2 hours for a fast charge and 8 hours for a trickle charge.

Due to limitations in the speed of interconnects between different modules — specifically the baud rate of the serial line between PIC and gumstix, there is an upper limit on the speed that data can be collected from the sensors. If every sensor is read on each iteration this limit is approximately 2Hz.

The majority of the code for the controllers was implemented using C and Python programming languages. The high level object orientation provided by Python, allows the fuzzy inference system to be constructed quickly and easily to incorporate type-2 and non-stationary fuzzy logic systems. C, being lower level, is optimised for speed, this ensures that critical sections of the code were run as optimally as possible which is important for a constant update rate.

[1] tilt on x-y plane

### 4.4.2 Supplementary Hardware and Software

As previously stated, simulations are run using a linux-based laptop with a 1.8Ghz Intel Atom with 1GB of RAM running Ubuntu 9.10 [88]. The Java RE [5] version used is $1.6.0_20$ and Python is version 2.5.52.5.5 for Linux.

The majority of the code base is written in Python, which includes the majority of the fuzzy logic controller code, is therefore platform-independent. Some C code is also used which uses either GCC 3.3.6 or gcc-arm cross compiler 4.1.1, depending on the target hardware (simulator or real world).

Based on the exact experimental set-up (such as course difficulty, parameter value and controller type) a single simulated run takes approximately 3 minutes. This includes the java virtual machine (JVM) start-up time and all post processing such as log creation and formatting.

The fuzzy logic controllers are based on an implementation using PyFuzzy, an open-source fuzzy logic library written in Python. The most update version is 0.1.0 obtained from the source forge page [56]. The library provides only type-1 fuzzy logic functionality though it does offer many types of defuzzifers, fuzzifiers and operators for the combination of input variables and fuzzy sets. Due to the object oriented nature of python, it is a simple matter to inherit the fuzzifiers and modify the code to return type-2 sets and add logic to handle the inference and type-reduction of such sets. KM was the only type-reduction method implemented as it is the most common method used and is the only one used in the work here.

PyFuzzy was selected from the numerous open source fuzzy logic libraries available for various reasons. Juzzy [98], a java based library, written in Java was considered too slow, while the Free fuzzy library(FFLL) [33] is too complex and difficult to extend to add type-2 fuzzy logic required.

### 4.4.3 Controller Design

As a basis for the design decisions made, the fuzzy controllers used the work done by Stelzer [92] and [91] as a starting point. In this work, it is shown that the design used by lead to a well performing type-1 fuzzy controller, which is able to make turns and sail along given routes with minor deviations from a straight line. For the work in this thesis some changes have been made due to the lack of data required to generate some constants used in Stelzers' controllers.

A running rate of 1Hz was fixed in the controller code for all controller configurations. This value was chosen in order to ensure that the more sophisticated controllers could run a complete cycle, as there were initial concerns that for type-2 based especially would run too slowly. While this low running rate leads to overall lower performance, it is believed that the consistency between controllers is more important in this work than optimal performance.

The sails are controlled using simple lookup table in which each of their positions are changed in response to the wind direction. The rudder is the main source of changes of direction of the boat and this therefore is where the fuzzy design effort is focussed.

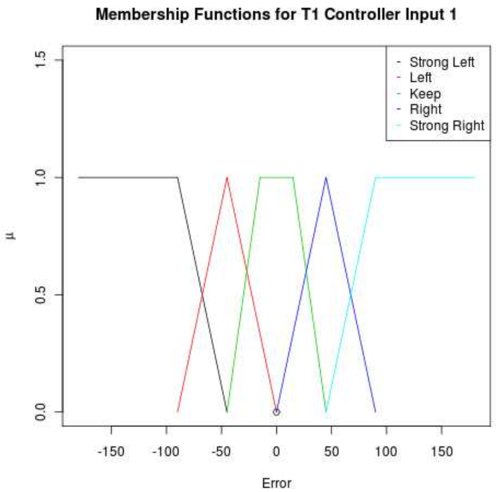
### Inputs Variables

The most often used inputs for control applications are Error and Delta E [53]. In the context of the application under discussion here, these values are calculated using Equations 4.1 and 4.2, respectively. In the design of our fuzzy controller, both of these inputs use the same base shapes for their MFs.

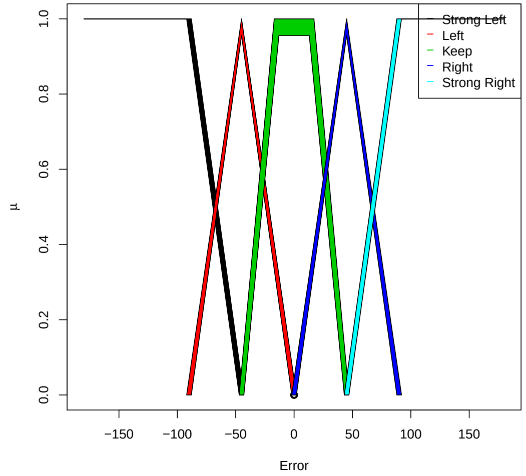
$$error = Desired\ Direction - Current\ Direction \tag{4.1}$$

$$\Delta error = Current\ Error - Previous\ Error \tag{4.2}$$

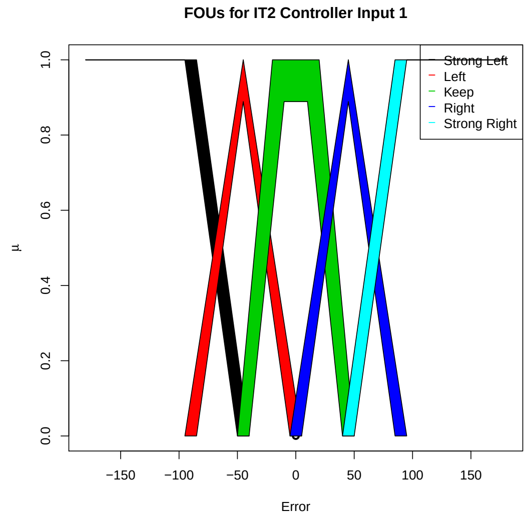
In interval type-2 fuzzy systems systems, secondary MFs are binary instead of continuous and in general can be visualised as a two-dimensional area, termed a Footprint of Uncertainty (FOU). This makes interval type-2 systems considerably more manageable than the general type-2 variety. The FOUs have been derived by starting with the simple type-1 and moving a uniform distance along the $x$-axis in both directions generating lower and upper bounds. The size of the FOU is varied throughout the different experiments performed in this thesis. A selection of membership functions using common FOU sizes are shown in Figure 5.1



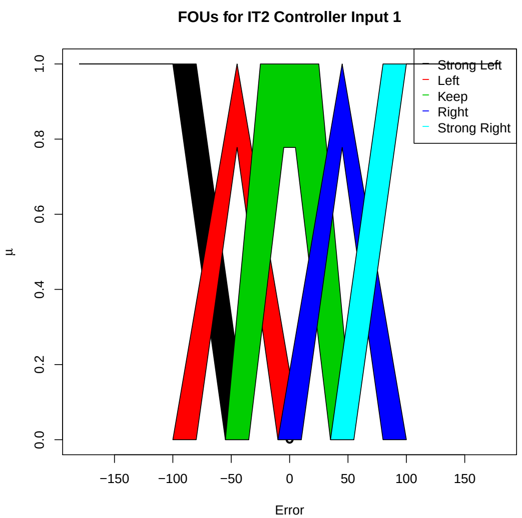

(a) Type-1 membership function



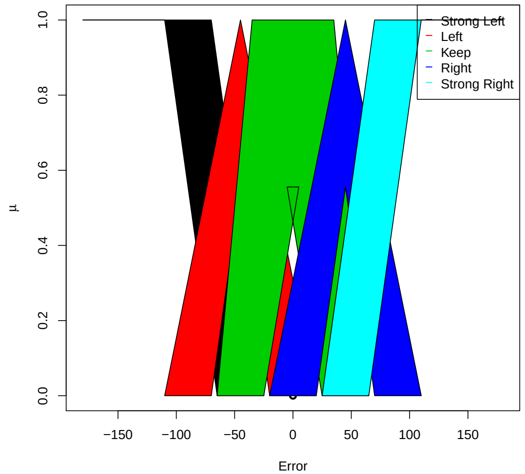

(b) Type-2 FOU Size 2 Membership function




(c) Type-2 FOU Size 5 Membership function




(d) Type-2 FOU Size 10 Membership function




(e) Type-2 FOU Size 10 Membership function

**Figure 4.3:** Membership functions of fuzzy controllers.

### Output Variables

The output value of all of the controllers is termed *RudderOutput* and represents a percentage change of the current rudder position. The output variable is represented by 5 singleton sets, as shown in Figure 4.4.

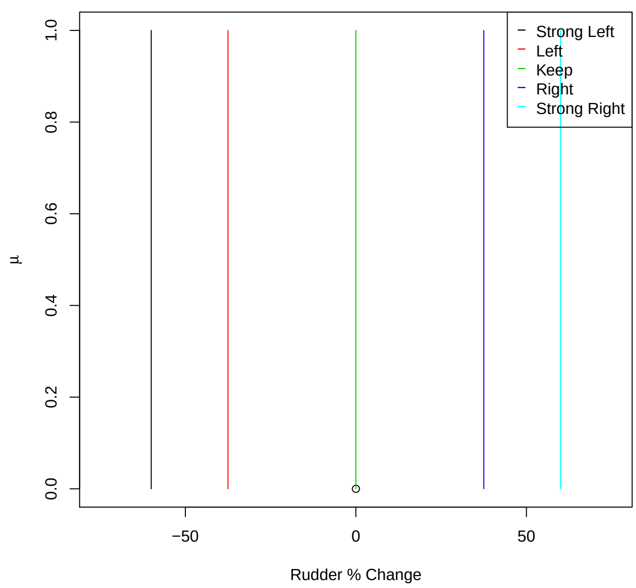



**Figure 4.4:** Fuzzy output singletons

### Rule Base

Each of the input variable has five associated fuzzy sets. This gives a combined rule-base set of 25 rules as shown in Table 4.5. The fuzzy inference system calculates which sets have a non-zero firing strength for each input. This gives a collection of membership values for some of the output sets based on this table depending on the specific inputs.

The rules have been derived from the work of Stelzer [92]. Alternatively they can be looked at from a logical point of view in that when sailing a boat and a turn in a given direction is required, the rudder must be changed in the opposite direction. The

| | Large Positve | Positive | None | Negative | Large Negative |
|---|---|---|---|---|---|
| **Strong Left** | Strong Right | Strong Right | Right | Right | Right |
| **Left** | Keep | Left | Right | Keep | Keep |
| **None** | Keep | Left | Keep | Right | Keep |
| **Right** | Keep | Keep | Left | Right | Keep |
| **Strong Right** | Left | Left | Left | Strong Left | Strong Left |

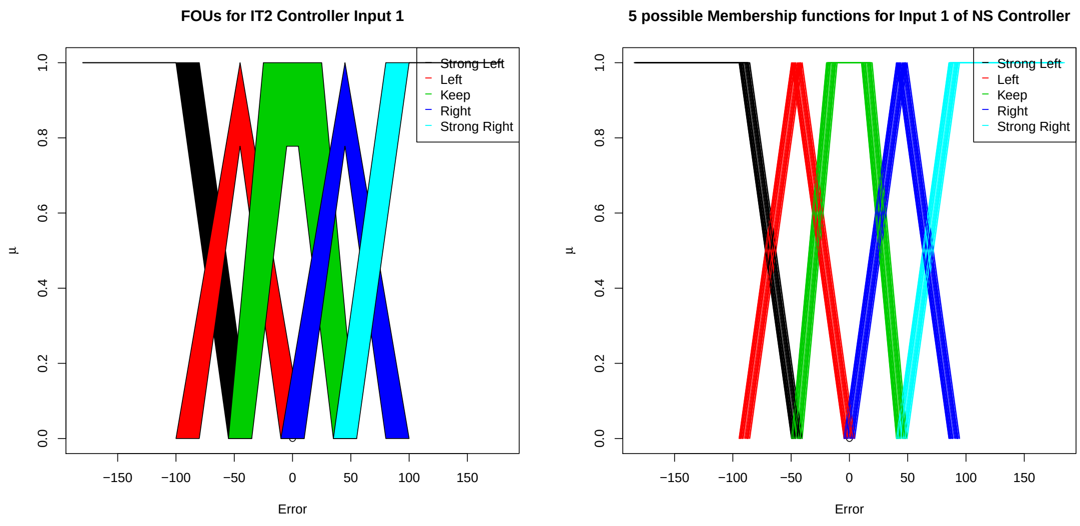
**Table 4.5:** Rule base. The table shows 25 rules generated by multiplying the number of fuzzy sets for each input (5 of each)

size of this change depends on a number of physical factors such as the size of the rudder but the overall directions can be reasoned out as done here.

### Dual Surface and Non-Stationary Controller Modifications

The NS controllers as previously discussed, are a modification to standard type-1 fuzzy logic. Standard type-1 MFs are perturbed to generate new functions, such as those shown in Figure 4.5(b) (on page 92). The perturbation function is defined as a horizontal movement obtained from a Gaussian distribution with a mean of zero and an standard deviation that acts as this controller types parameter. During execution, the overall controller selects 30 values from the random distribution to create 30 sub-controllers, each including MFs which deviate slightly from the standard type-1 sets. Each of these fuzzy sets are then processed as a standard type-1 system and the mean of the outputs from each of the 30 controllers is taken to give a final output.

A DS type-2 controller is implemented to determine if improved results can be achieved through incorporating extra information, such as the upper and lower outputs as outlined in Birkin and Garibaldi [12]. This employs the algorithm described in Algorithm 1 (page 29) for selection of control surfaces and determination of the value returned. This algorithm compares a user chosen metric, in this case the magnitude of the input error, with a threshold value. On this basis the final output of the system is selected from either the lower, upper or mean value. For this comparison, the original




(a) Movement 10 type-2

(b) Movement 10 non-stationary

**Figure 4.5:** 10 sized FOU for NS and Type-2

method of using the magnitude of the error in the system is retained. As with other controllers, several different threshold values are used to determine any observable effect on the system and its performance.

During each experiment, each variety of controller excluding the type-1, has an associated parameter value which determines its internal configuration. For the interval type-2 and DS controllers the parameter refers to the area size of the FOU. For the NS, the parameter represents the maximum horizontal movement which is defined by the standard deviation of the associated Gaussian random number generated — which gives an area similar to a FOU. This parameter value is altered several times for each course layout to observe how the change affects performance.

Due to the nature of the sailing boat problem, even proven optimal controllers may have difficulties operating in all wind conditions. This is because it is impossible for any sailing boat to move directly into the wind (i.e. $DesiredDirection == WindDirection - 180$). The solution to this problem used by human sailors is to use a procedure known as *tacking*. This involves altering the desired direction so it does not move directly into the wind. As with many parameters such as this, the exact angle required varies per boat but is typically between 10° and 30° [14]. This behaviour is mimicked by the software in the fuzzy controller by implementing a simple check to work out if tacking is required if so which direction is optimal.

### 4.4.4 Supplementary Software

FLOATS also include a set of software tools designed to help automation for the testing of robotic sailing boats, in both simulation and real-world environments. Figure 4.6 shows the modular layout of this system, with details of each being discussed below.

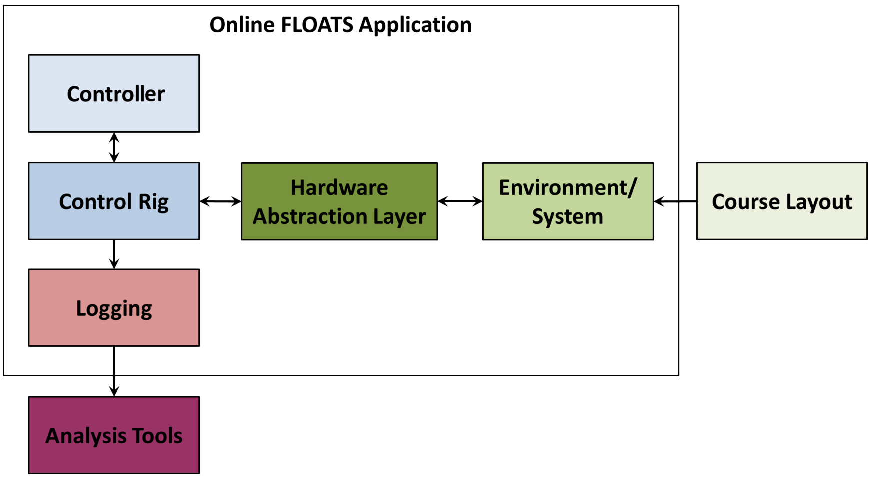



**Figure 4.6:** Module layout for the FLOATS system

**Controller** Calculates the new sail and rudder positions from the data provided by the controller rig. All controllers under test in these experiments are implemented in the Python programming language, although virtually any other programming language could have been used.

**Central Control System** A C based program that interfaces with a single given controller and hardware abstraction layer, providing data for the logging subsystem.

**Logging** The Logging module generates comma-separated values (CSV) formatted log files from the data provided by the controller rig.

**Analysis tools** This module reads the data stored in the CSV files created by the Control rig. It generates the performance metrics and any graphs that may be required. Unlike the rest of the system, this operates off-line after each experiment has been completed.

**Hardware abstraction layer** This module provides the correct hardware/software interface for the given environment (simulation or real world).

**Environment** FLOATS is designed to support both real-world and simulation-based approaches for conducting experiments. This module represents which approach is in use for a given experiment.

**Course layout** The course is preprogrammed before each experiment. It is either defined within the simulator configuration or by using GPS coordinates when operating in the real world.

# 5

# Simulation Experiments

## 5.1 Introduction

This chapter presents the second set of experimental work of this thesis. Several investigations based on the FLOATs simulation architecture are conducted, which are intended to address the shortcomings of the work performed in the previous chapter. One of the main conclusions of the previous chapter was that the tipping application selected was too simple to give interesting results. This simplicity can be looked at in several ways: First it did not include any sources of variation, randomness or uncertainty. Second, the universe of discourse, width of membership functions and rule base were all too small to allow sufficient change to be easily introduced. Thirdly, due to the experimental design, there was no real concept of feedback in which outputs from the system will change the environment and in turn affect the inputs into the next iteration of the system.

This chapter applies three different approaches of performing comparisons between fuzzy logic controllers. Two of these were applied in the previous chapter, with minimal differences being found between the controllers under test. This previous work used a simple controller in a simple environmental set-up and it is believed that this is one of the main reasons for the lack of differences observed. These findings therefore have lead to the decision to reapply the methods used in conjunction with a more complicated fuzzy logic controller.

In addition to reusing the two comparative methods, a new approach looking into the control surfaces of the controllers under investigations is introduced. This was not done in the previous set of work as it was thought that the simplicity of the controller would not allow any interesting results to present themselves. The rule fire and control surface approaches use the controllers without any environmental context, relying more on the analysis of outputs from the inference system using input values for every combination present in the whole universe of discourse. This contrasts to the third and final approach to comparison, this is intended to be more a realistic setting in which a subset of inputs which progresses across the universe of discourse is used.

The chapter is organised as follows: Sections 5.2 and 5.3 present the two context free techniques of comparison. This is followed by two sets of experiments in Sections Section 5.4 and Section 5.5 in which the RMSE value obtained from experiments in which the environment, and variations within it is studied. The chapter is closed with a discussion of the results found in Section 5.6. Finally Section 5.7 provides a summary of the chapter.

## 5.2 Rule Fire Experiments

### 5.2.1 Experimental Purpose

Section 3.3 in the previous chapter presents an investigation into the effects differing the internal configuration of type-2 fuzzy logic systems upon which rules fire in a given system is performed. A simple 3 rule fuzzy system was used to evaluate the concept, and while there were issues with the design of the fuzzy system it was found that generally there are very few differences using the set-up described.

The present section utilises this technique in more complex application with a more sophisticated fuzzy controller design, specifically the FLOATS platform. As previously discussed, it has a larger rule base, a greater number of output fuzzy sets with each rule having a larger range. Additionally the number of different configurations of fuzzy controller that will be investigated will be increased from five to six.

### 5.2.2 Methodology

The methodology used in this section is the same as that discussed in Section 3.3.2, with the exception of the fuzzy logic controller used, which is thoroughly described in Section 4.4.3. The only significant change that will be made is to the output logs to include an indication as to which rules fire at each iteration of the fuzzy logic controller. As in the previous experiments in Section 3.3 this will be achieved by outputting a bitmap, with a '1' in given location in the map indicating that rule has fired while a '0' indicating it did not. Table 5.1 indicates which bit represents which rule and also shows which combination input variables each rule requires to fire.

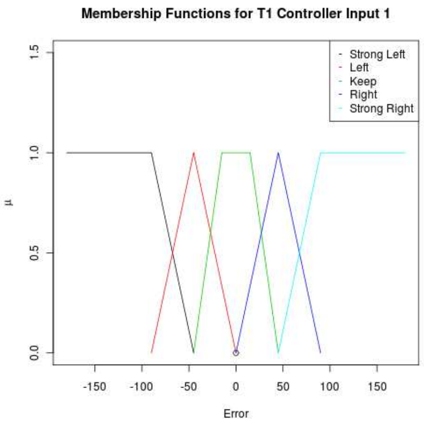
| | **Large Positve** | **Positive** | **None** | **Negative** | **Large Negative** |
|---|---:|---:|---:|---:|---:|
| **Strong Left** | 1 | 2 | 3 | 4 | 5 |
| **Left** | 6 | 7 | 8 | 9 | 10 |
| **None** | 11 | 12 | 13 | 14 | 15 |
| **Right** | 16 | 17 | 18 | 19 | 20 |
| **Strong Right** | 21 | 22 | 23 | 24 | 25 |

**Table 5.1:** Rule to Bit Map number mapping — the number indicates the bit number of the given rule

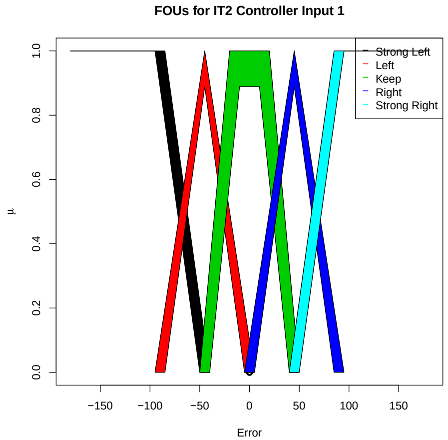
With 25 rules in total, a 32 bit integer provides enough space for each rule to be assigned. As an example the bitmap '0000011111000001111100000' would indicate the rules 5–10 and 16–20 have fired while the rest (i.e. rule numbers 0–5, 11-15 and 21-25) did not fire.

### 5.2.3 Hypothesis

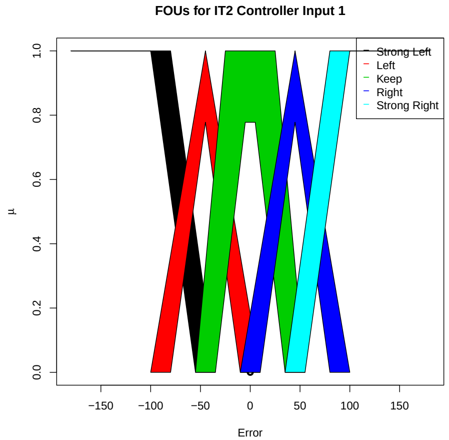
As stated in Section 3.3.3 (Page 51) that as the FOU sized is increased, more rules will fire for a given set of inputs. This is because as the size of the FOU increases, the membership functions of each input fuzzy set will be correspondingly larger, meaning



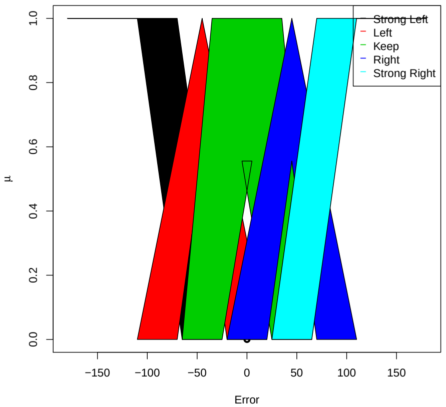

(a) Movement 0 Type-1



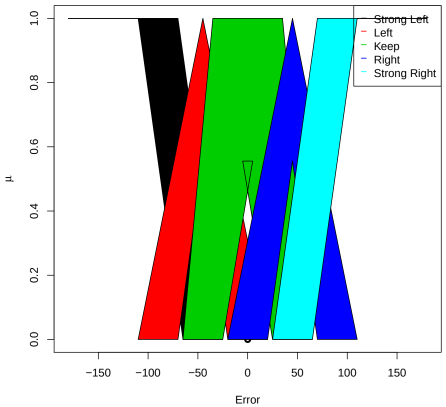

(b) Movement 5 type-2



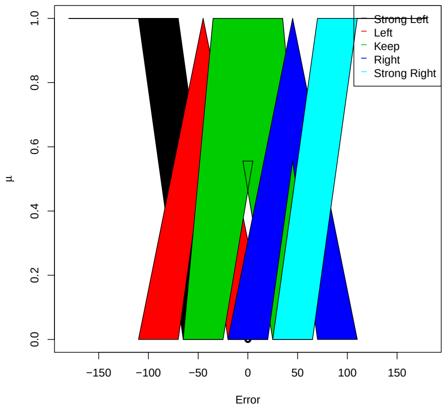

(c) Movement 10 type-2




(d) Movement 15 type-2




(e) Movement 20 type-2




(f) Movement 25 type-2

**Figure 5.1:** Membership functions of fuzzy controllers. The larger movement value directly corresponds to a large FOU size.

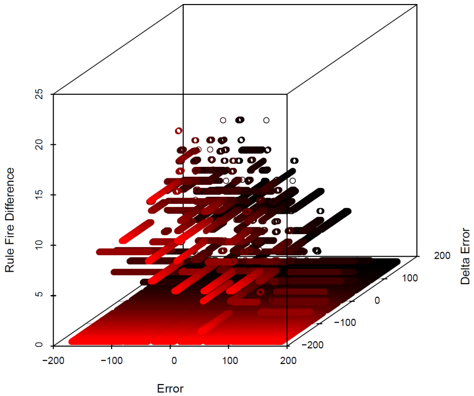
a single input value will trigger more fuzzy sets to have a non zero membership, which in turn will in turn trigger more rules to fire. It is not known if more rules firing will directly change the overall performance of that controller configuration, however this is not the direct focus of this experiment.

### 5.2.4 Results

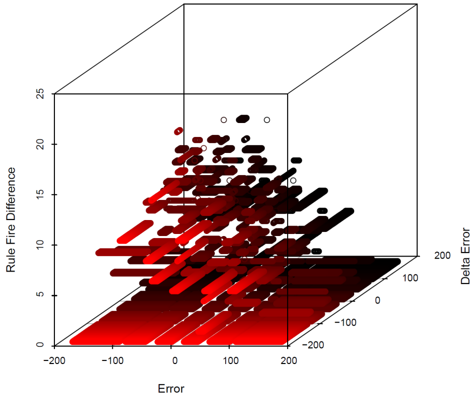
| FOU Size | Differences | Percentage |
|---|---|---|
| 5 | 161759 | 0.05 |
| 10 | 306246 | 0.09 |
| 15 | 436850 | 0.13 |
| 20 | 570095 | 0.18 |
| 25 | 681800 | 0.21 |

**Table 5.2:** Total number of differences and percentage of total possible difference when compared with FOU size 0 controller configuration

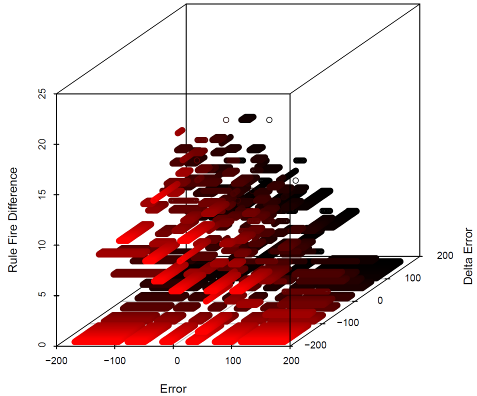
Figure 5.2 shows graphical representations of how different rules fire for each FOU configurations. The $x$ and $y$ axes represent the inputs to the fuzzy system, each of which have a range between -180 and 180 — therefore with 25 rules there can be a maximum difference of 3,240,000 firings — many more than in the previous set of rule firing experiments in the previous chapter. The final axis represents the difference between number of rules the specified FOU size and FOU size 0 which is being used as a baseline. If the output from the FOU 0 size controller was "0000000000000000000000001" and the output from FOU size 10 controller was "0000000000000000000011111" the difference would be 4, as there are 4 rules which fire in the latter and not in the former. In these experiments there is no difference between a rule firing in the first and not the second against a rule firing in the second and not the first — originally the concept of positive difference (the first case) and a negative difference (the second case) was considered but this was abandoned because



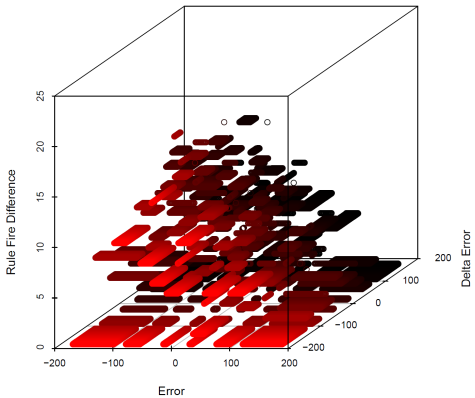

(a) Type-1 vs Interval Type-2 FOU 5



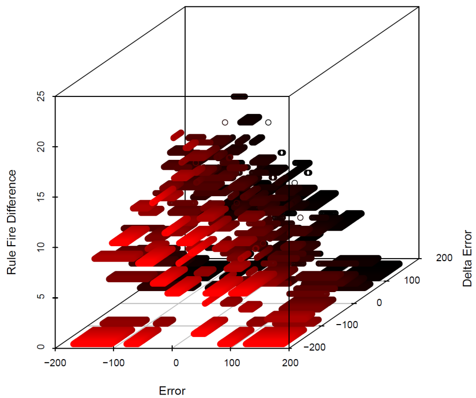

(b) Type-1 vs Interval Type-2 FOU 5




(c) Type-1 vs Interval Type-2 FOU 5




(d) Type-1 vs Interval Type-2 FOU 5




(e) Type-1 vs Interval Type-2 FOU 5

**Figure 5.2:** x-axis and y-axis show the combinations of input variables. z-axis shows the difference in the number of rules which fire between the type-1 and type-2 controllers. Large z axis values indicate large differences occur between the two controllers. Colouring used to improve 3D visibility.

a negative value could be cancel out a later positive giving a total difference of 0 where in fact there were potentially several differences. A larger difference indicates more rules fired in one than in the other, directly indicating difference but necessarily in better or worse direction.

More numerical results are shown in Table 5.2 where it can be seen that the percentage of differences increases by approximately %4 for each increase of FOU size. The smallest differences occurs between FOU sizes 20 and 25 while the largest differences lie between FOU sizes 15 and 15 and 20 where the difference is 5% and is discussed further below.

### 5.2.5 Discussion

The results obtained here show much more significant differences than in the previous set of experiments using this technique. Table 5.2 shows that the percentage of rules which are different between type-1 and the increasing FOU size is by approximately %4 for each increase of the FOU size. It can be observed that increasing the FOU size changes the output values across the large parts of the universe of discourse, and while the results shown are not large enough to draw strong conclusions from, it gives one the first observations of the potential changes in performance as the FOU size changes.

Overall these experiments show that more sophisticated environments, do seem to allow type-2 fuzzy controllers with a larger magnitude of difference in their outputs. How these differences result in altered performance profiles in a more realistic setting is the subject of the rest of this chapter.

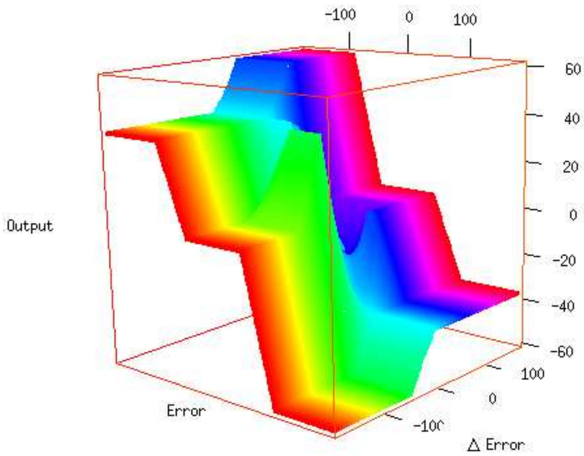
## 5.3 Investigation into Control Surfaces

To show that each controller under test creates a suitable mapping of input values to output, a test rig is used to generate control surfaces across the entire universe of discourse for several different controller configurations. Each fuzzy controller is executed in a minimal set-up without any of the tacking behaviours or sensor reading functionality present. Control surfaces result in a 3D graph, in which two axes represent the input values and the third axis represent the raw output obtained from the system. In order to plot a complete control surface, every integer value within the input variable ranges for each input is passed into the fuzzy logic system.

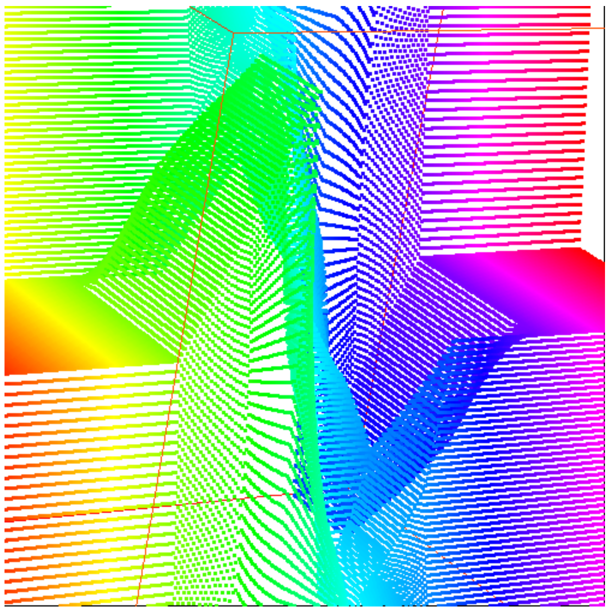
The type-1 control surface shown in Figure 5.3(a) (page 103) shows a typical control surface. It can be seen that all transitions are very crisp and sharp in nature. One reason for this is the resolution of the data used to plot the graphs, which as stated, uses an increment of one. If a smaller increment was adopted, then a more gradual transition would be observed. This is somewhat clearer in Figure 5.3(b), which is more closely zoomed. This figure also goes some way to showing why type-1 control is often said not to give as good performance as type-2 — very small changes in one input located at one of these transitions points could lead to a large output change.

Figures 5.4 (page 104) show the control surfaces of type-2 fuzzy controllers with FOU sizes 2, 10 and 20. There are two trends which can easily be seen from this progression. First, many of the corners between surface faces become increasingly smoothed, as the lines near a surface transition come closer together. This means that for the same input data (and ignoring such things as variation between them) a type-2 controller produces a smoother movement across a transition section in one of the input values, when compared to type-1. The smoother transition in may indicate better performance as supported by Wu [104] , especially if a given experiment has a great deal of inputs which appear in these areas.

The control surfaces shown in Figures 5.5 (page 105) are calculated from the non-stationary controllers. Each controller is differentiated by the standard deviation of the random number generator used for the perturbation functions in each specific configuration. These control surfaces show that the straight lines observed in the type-1 figures change and exhibit a speckling effect. This makes the edges less sharply defined as in the type-2 case, but with a more random element rather than the introduction of smoother curves as in the type-2 case, this can be observed more easily by using a cross sectional view of the control surface shown in Figure 5.6. In these plots the graphs of the smaller movement value (Figure 5.6 (a)) shows an almost straight line while a large value (Figure 5.6 (b)) exhibits a much more variable graph. It is theorised that this speckling smooths out the transitions in much the same way as the type-2 smoothing behaviour works as described above. In both cases, it may also equate to a more gradual transition in circumstances where the input triggers more than 1 fuzzy set to be triggered.

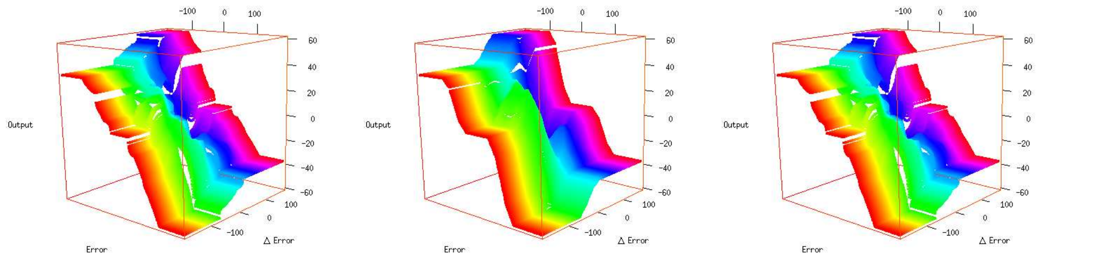



(a) Type-1 control surface

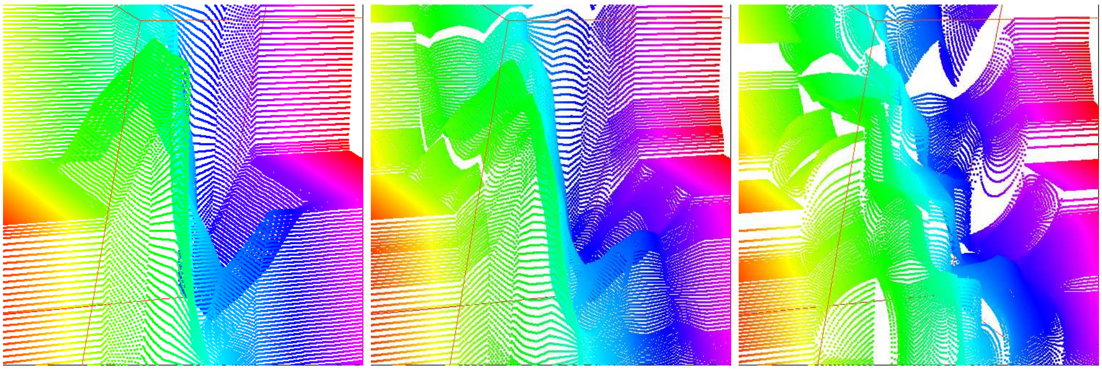


(b) Type-1 zoom surface

**Figure 5.3:** Type-1 control surfaces x-axis and y-axis show the combinations of input variables. z axis indicates the resulting output value. Colouring used to improve 3D visibility.




(a) FOU size 2 type-2 control surface
(b) FOU size 10 type-2 control surface
(c) FOU size 20 type-2 control Surface



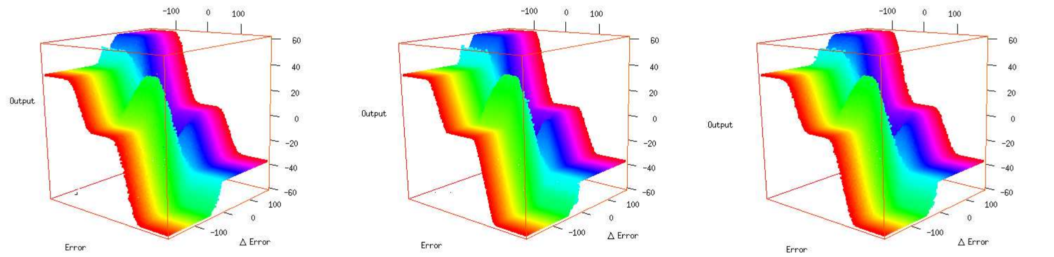
(d) FOU size 2 zoomed surface
(e) FOU size 10 type-2 zoom surface
(f) FOU size 20 type-2 zoom surface

**Figure 5.4:** Interval type-2 fuzzy controllers control surfaces with increasing FOU sizes. $x$ and $y$ axes show the input variable values. $z$ Axis indicates the resulting output.

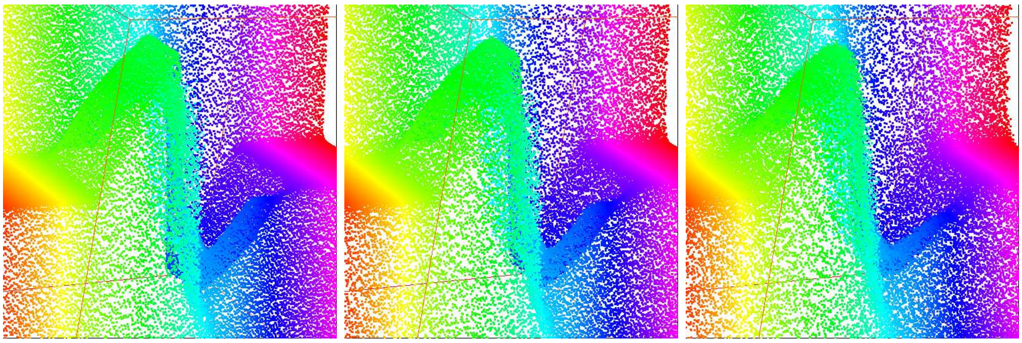
### 5.3.1 Discussion

This small study is focussed on looking how the control surfaces of different fuzzy logic controllers change as the fuzzy controller configuration (FOU size) is increased. In this set, only a single parameter is changed, for the type-2 and DS controllers this is the FOU size; the NS parameter is amount of variation introduced when generating the sub controllers. The type-1 controller in this context is used as the control surface against which the others can be compared.

Several interesting points can be observed in the results of this study, specifically,



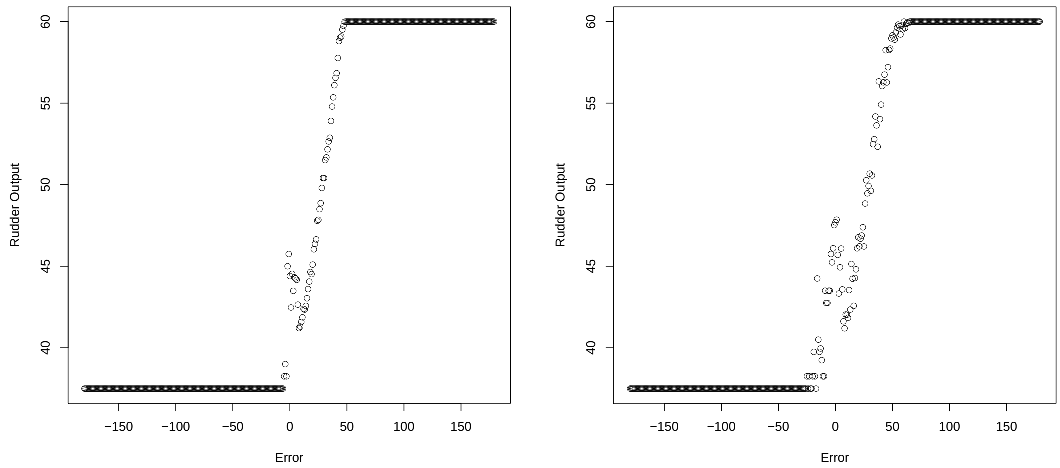

(a) Movement 2 NS control surface (b) Movement 10 NS surface (c) Movement 20 NS control surface



(d) Movement 2 NS zoomed surface (e) Movement 10 NS zoomed surface (f) Movement 20 NS zoomed surface

**Figure 5.5:** Non-stationary fuzzy controllers control surfaces. $x$ and $y$ axes show the input variable values. $z$ Axis indicates the resulting output.

that as the parameters of the type-2 controller is increased the transitions between different sections of the control surface become more smooth. This seems to imply that a larger parameter (essentially representing the FOU size of the set in question) should give a smoother response to sets of inputs which pass through these sorts of areas of the control surface. This is considered a positive result overall as it is evidence that type-2 controllers with larger FOU sizes have potential to give significantly different results — whether these results lead to better overall performance in real world conditions, it is hypothesized, will be application dependent, but is further investigated later in this chapter. Bastian [6] also discusses a similar topic with particular reference to the




(a) Movement 2 NS cross section

(b) Movement 20 NS cross section

**Figure 5.6:** Cross section of a subset of the control surfaces for movement sizes 2 and 20 non stationary controllers

transition areas where the rules which fire change.

The second point of observation in this selection of control surfaces is the appearance of artefacts such as warping as the FOU is increased. The exact reason behind these artefacts is not yet know. However, there is a theory presented by Wu [104] that there exist situations in which type-1 controller would produce a discontinuous control surface whereas type-2 would not. Wu [101] goes on to develop this idea and shows that, given an interval type-2 system, there does not always exist an equivalent type-1 system. Further work is required to determine if and when these discontinuities will occur and to test if they are not caused simply by the granularity of the data used to generate the control surfaces. It may be that using an increment of for example 0.01 instead of 1 may reduce the visibility of the discontinuities.

## 5.4 Factors Affecting Performance of Different Types of Fuzzy Logic Controller

### 5.4.1 Experiment Purpose

The main purpose of this set of experiments is to compare multiple varieties of fuzzy logic and evaluate how they perform with different levels of variability present in the environment. Additionally this, the first large set of results using FLOATs, ensures that the described experimental set-up is capable of generating data that assists in achieving the aims of this thesis.

The use of FLOATs in these experiments addresses several of the problems identified at the end of Chapter 3, where the simplicity of the experiments performed coupled with a lack of variation it is believed, led to the poor results obtained. FLOATS is considerably more sophisticated by most measures, so it is hoped that better (i.e. more distinct) results can be obtained.

### 5.4.2 Experimental Design

In this section, the type-1, interval type-2, NS and DS fuzzy controllers are defined and their usage explored. A PI (proportional integral) controller is also included to provide some benchmark levels of performance against the fuzzy types. This PI controller is derived from the work of Sauze *et al.* [81], in which the authors have spend significant work tuning the gain values used. This tuning, according to the authors, resulted in a well-performing controller under calm situations.

In order to address the secondary purpose of these experiments, the width of the FOU of the type-2 and DS controller is varied for each course, from a minimum of 0 (equivalent to a type-1 controller) up to a maximum of 20. This size (20) has been chosen as the maximum value after several large values were tested for viability and 20 was found to be the largest sensible value.

Three different levels of variability are defined and labelled as 'None', 'Low' and 'High'. These levels are created by altering the rate and magnitude of the change in wind direction within the simulator and are shown in Table 5.3.

| Variation | Variation Score | Lower Limit | Upper Limit |
|---|---|---|---|
| None | 0 | 180 | 180 |
| Low | 1 | 160 | 200 |
| High | 2 | 140 | 220 |

**Table 5.3:** Levels of Variation

As there are already many factors under consideration, including environmental variation, controller FOU size, the difficulty of the course will be kept simple so as to ease the task of determining a correlation between the various elements of study. A straight line of 550 metres in length has been selected will therefore be used.

The performance of a given experiment will be determined by the standard measure termed *RMSE*. The error term is defined as the difference between the desired direction and current direction the boat is facing. The best possible value for RMSE is therefore 0, indicating the boat was always facing the desired direction. Similarly, a large RMSE value indicates that there was significant deviation from the optimal course and therefore the controller performed poorly. Furthermore, the time taken and total distance travelled are also collected to support the RMSE in the comparative evaluation regarding the performance of given run.

### 5.4.3 Hypothesis

It is hypothesised that, assuming all other aspects are kept constant, as the environment is made more variable i.e. increased from 'None' to 'High', then the performance of all controllers will drop. The reasoning behind this is that the experiments with higher levels of variability will require more work from the controller under test. If the controller does not adapt to the environment variations, subsequently the overall performance should drop. Whether the high level of variability here is sufficiently high enough to cause controllers to fail or for performance to significantly deteriorate is as yet unknown.

It is also anticipated that if the FOU is increased, then the plot of FOU size against performance should result in a curve in which performance increases (shown by RMSE decreasing) until a certain point after which it will decrease (shown by increasing RMSE). This is because it is believed that there is a specific “optimal” FOU value for each controller configuration. With the increase of the levels of variation present after this point, performance is anticipated to drop.

As discussed in Section 5.4.1, the link between FOU and variation in the environment still remains to be investigated. However, it is hoped that these experiments can establish a base point for a discussion on how to derive the FOU size from the quantification of variation in the environment in which the controller is running.

### 5.4.4 Experimental Procedure

The experimental procedure for each experimental run is as follows:

- One of the five controller types (type-1, type-2, PI, NS and DS) under test is configured. Type-2 DS and non-stationary controllers have their parameter set to one of the following values (0,5,10,15,20).
- The simulator is configured to run at one of the three defined levels of variation: ‘low’, ‘medium’ or ‘high’ as specified in Table 5.3.
- The specified controller is connected to the simulator and configured to attempt to control the boat around the selected course.
- Data from the simulator and the controller is collected and analysed. Performance measures such as RMSE are calculated for each run. Images such as a course plot can also be generated from this data.

The experiment will be broken down by the level of variation used. Experiment set one uses no variation, experiment set two with ‘low’ variation and experiment set three with ‘high’ variation levels. Each combination of controller type, parameter value and variation level will be performed thirty times and the results collected. Thirty runs of each combination are performed in order to eliminate any erroneous runs and provide statistical significance, with the mean value of each taken and used for comparisons.

### 5.4.5 Results

Figure 5.7 (page 118) shows that as the variation increases (left to right in the sub figures), the courses taken by the boat exhibits increased deviations from a straight line course. This supports the hypothesis that increasing environmental variation results in routes that deviate more from the straight line "optimal" course. The amount of deviation from the optimal route is signified by a larger RMSE value.

The results of the low variation experiments are presented in Table 5.4. In the table, the performance at FOU size of 5 in the type-2 controller shows little difference compared to the baseline PI controller. When the FOU is increased to 10 however, type-2 performance becomes significantly worse. At its largest size of 20, the type-2 controller shows better performance than the type-1 and PI controllers. For the DS controller, both metrics, i.e. the RMSE and the time taken are worse with small FOU sizes, such as 5. As the FOU increases, however the performance correspondly increases, resulting in the best overall performance in this experiment.

Experiment two (medium variation) increases the amount of variation present in the environment and the results are shown in Table 5.5 (page 113). These outcomes clearly show the anticipated drop in performance with the average increase in RMSE being 5.06 and the mean increase in time being 40.2 seconds. A very similar pattern to the previous experiment can be observed in the performance values of the standard type-2 and DS results. A peak in performance is observed when FOU size is 10 for the interval type-2 controller. Similarly, a peak can be observed at FOU size of 20 for the DS controller. The hypothesis made in the previous paragraph is also expected to hold true here in this experiment. From these results it is believed that the experiments so far demonstrate that there certainty exist points where type-1 outperforms type-2. Conversely, there are also points in which type-2 performs better. These observations are further discussed in the next section.

Table 5.6 (on page 114) summarises the results of the experiments performed at high levels of variation. The average performance of the type-2-based controllers is somewhat lower than that obtained in the previous experiments. In addition, only one configuration of the DS controller obtained statistically significant improvements over the type-1 controller. None of the standard interval type-2 and NS approaches achieved this performance level regarding the *time taken* metric. However, the NS controller did produce two cases in which the RMSE was improved significantly. The mean RMSE increase between experiments one and three was 9.57 with an average time increase of 10.4s.

As discussed in Section 4.3.4 (page 80), statistical tests are used to determine any significant difference between type-1 and the NS, type-2 and DS controllers. Given the nature of the data collected, the one-sided Mann-Whiteney statistical test has been selected. This test is performed for both RMSE and *time taken* metrics with $5\%$ significance level being used to reject the null hypothesis.

This test is also performed for the PI and type-1 controllers for all three experiments. The type-1 RMSE proved significantly lower than the PI with low variation (experiment one). As variation increases however, the RMSE of the PI controller becomes significantly lower than the type-1 controller showing a worsening of relative performance.

### 5.4.6 Discussion

From the work performed here the following points have been concluded:

- At 'low' and 'high' levels of variation, the more sophisticated controllers such as type-2 generally do not show a significant improvement when compared to the type-1 controllers. These observations are supported by several Mann-Whitney test, which fails to reject the null hypothesis as shown in Tables 5.4 and 5.6 in which statistically significant differences are underlined. Specific controllers in each category do show this improvement, however.

- At 'medium' variation levels type-2, NS and DS controllers generally do exhibit statistically significant improvements on the type-1 method.

- The resulting difference between PI and type-1 controllers show that type-1 does improve upon the PI for the RMSE metric and improves under low variation conditions and low and highly variable conditions for the time metric. This is also supported by the Mann-Whitney tests performed specified in table 5.4

| Variety | Parameter Value | Mean RMSE | Std. Dev RMSE | Mean Time (secs) | Time Std.Dev (secs) |
|---|---|---|---|---|---|
| PI | N/A | 18.01 | 0.30 | 146.56 | 2.02 |
| Type 1 | N/A | 16.32 | 0.17 | 140.80 | 0.66 |
| Non Stationary | 2 | 17.03 | 0.64 | 139.96 | 1.16 |
| Non Stationary | 5 | *16.72* | *0.54* | ***139.28*** | ***0.63*** |
| Non Stationary | 10 | 16.99 | 1.14 | 139.59 | 1.41 |
| Non Stationary | 20 | 16.74 | 0.55 | 140.07 | 1.10 |
| Type 2 | 2 | 15.97 | 0.62 | *140.42* | *1.03* |
| Type 2 | 5 | *15.84* | *0.28* | 140.65 | 1.18 |
| Type 2 | 10 | 16.04 | 0.53 | 140.80 | 0.66 |
| Type 2 | 20 | 18.94 | 0.57 | 150.03 | 2.69 |
| Dual Surface | 2 | 19.13 | 0.61 | 153.80 | 1.86 |
| Dual Surface | 5 | 19.34 | 1.35 | 150.57 | 3.35 |
| Dual Surface | 10 | 16.73 | 0.59 | 145.43 | 1.33 |
| Dual Surface | 25 | ***15.80*** | ***0.24*** | 149.10 | 7.22 |
| Dual Surface | 50 | 15.99 | 0.25 | *142.38* | *3.59* |

**Table 5.4:** RMSE and total time taken for course completion at low variation levels. Mean and standard deviation of 30 runs with the best values per category shown in italic and the best overall controller shown in bold. The values that are statistically different from the type-1 controller are underlined. Parameter refers to movement in NS, FOU in IT2 cases and threshold in the DS case.

| Variety | Parameter Value | Mean RMSE | Std. Dev RMSE | Mean Time (secs) | Time Std.Dev (secs) |
|---|---|---|---|---|---|
| PI | N/A | 23.25 | 0.30 | 204.69 | 12.97 |
| Type 1 | N/A | 24.47 | 0.76 | 221.34 | 8.46 |
| Non Stationary | 2 | 22.86 | 1.99 | 160.50 | 9.17 |
| Non Stationary | 5 | 22.21 | 4.11 | 172.53 | 23.17 |
| Non Stationary | 10 | 20.27 | 3.18 | 158.53 | 3.61 |
| Non Stationary | 20 | ***21.09*** | ***2.80*** | ***161.09*** | ***9.23*** |
| Type 2 | 2 | 25.65 | 1.39 | 189.81 | 11.69 |
| Type 2 | 5 | 20.48 | 3.34 | 178.64 | 20.19 |
| Type 2 | 10 | *19.32* | *1.28* | *168.39* | *11.24* |
| Type 2 | 20 | 26.00 | 5.31 | 186.87 | 5.34 |
| Dual Surface | 2 | 20.59 | 0.96 | 168.62 | 7.85 |
| Dual Surface | 5 | 23.06 | 5.10 | 181.94 | 19.03 |
| Dual Surface | 10 | 22.02 | 0.92 | 173.54 | 12.54 |
| Dual Surface | 25 | 19.75 | 3.84 | *171.27* | *12.66* |
| Dual Surface | 50 | ***18.81*** | ***1.61*** | 174.35 | 18.10 |

**Table 5.5:** RMSE and total time taken for course completion at medium variation levels. Mean and standard deviation of 30 runs with the best per category shown in italic and the best overall controller shown in bold. The values that are statistically different from the type-1 controller are underlined. Parameter refers to movement in NS, FOU in IT2 cases and threshold in the DS case.

| Variety | Parameter Value | Mean RMSE | Std. Dev RMSE | Mean Time (secs) | Time Std.Dev (secs) |
|---|---|---|---|---|---|
| PI | N/A | 25.85 | 0.38 | 157.2 | 1.41 |
| Type 1 | N/A | 27.43 | 0.93 | 153.61 | 3.53 |
| Non Stationary | 2 | 31.22 | 4.55 | 153.83 | 7.37 |
| Non Stationary | 5 | 22.21 | 4.11 | 172.53 | 23.17 |
| Non Stationary | 10 | ***20.27*** | ***3.18*** | 158.53 | 3.61 |
| Non Stationary | 20 | 28.69 | 1.35 | *151.23* | *2.60* |
| Type 2 | 2 | 25.48 | 0.66 | *149.70* | *2.08* |
| Type 2 | 5 | *25.33* | *1.36* | 150.19 | 2.33 |
| Type 2 | 10 | 25.83 | 0.93 | 149.77 | 2.75 |
| Type 2 | 20 | 32.72 | 1.92 | 172.37 | 17.31 |
| Dual Surface | 2 | *24.11* | *1.15* | ***141.09*** | ***5.76*** |
| Dual Surface | 5 | 28.93 | 7.41 | 152.49 | 10.02 |
| Dual Surface | 10 | 29.12 | 8.46 | 151.91 | 12.63 |
| Dual Surface | 25 | 26.09 | 0.84 | 151.26 | 2.56 |
| Dual Surface | 50 | 25.95 | 2.66 | 149.81 | 2.86 |

**Table 5.6:** RMSE and total time taken for course completion at high variation levels. Mean and standard deviation of 30 runs with the best per category shown in italic and the best overall controller shown in bold. The values that are statistically different from the type-1 controller are underlined. Parameter refers to movement in NS, FOU in IT2 cases and threshold in the DS case.

- From the results it can be seen that this more developed application does seem to allow certain configurations to present significantly better performance over alternatives, matching our hypotheses.

The fact that the PI controller outperformed the type-1 controller in some experiments, however insignificantly, indicates that some aspects of the type-1 system were not tuned optimally in these experiments. Further work may therefore be required in this regard. However, any changes required would also affect the other controllers, which have been based on this type-1 set-up. For this reason, it is not anticipated that there would be much alteration in the general performance ordering of the various controllers if these modifications were performed. Additionally, as has been previously stated, the goal of this work is not to develop the best performing controller but to highlight the differences between those under test in the different scenarios used.

The use of FLOATS has shown that more sophisticated experimental set-ups do enable more obvious differences between the various experimental scenarios to be obtained, which in turn allows better discussion and conclusion to be drawn. However it is still felt that more can be done, as in several cases the differences were small.

## 5.5 An Investigation into the Effect of Environmental Variation upon the Root-Mean Square Error Performance for Type-2 Fuzzy Logic Systems

### 5.5.1 Experiment Purpose

Based on the results of previous experiments, it has been decided that greater granularity of variability levels would be helpful to better understand how it affects performance. This will also allow better evaluation of the comparative methodology, as additional data points may make trends in the data easier to spot.

The difficulty of the course used in the previous section contained no turns and difficulty was increased simply by increasing the environmental variability present. This has been deemed insufficient to allow different controllers to exhibit their changes in their relative performance and will be addressed by increasing the difficulty of the task being attempted as described below.

### 5.5.2 Experimental Design

In order to produce a series of courses in which the difficulty increases in a systematic manner, a straight line course with fixed wind blowing parallel to the boats starting direction is taken as a starting scenario. This is the simplest possible course in which the boat must simply move forward in order to reach the end point. In order to add difficulty to this course, deliberate turns are introduced by the addition of way points which are vertically offset from the straight line course. This offset is either 0, 25, 50 or 100 meters and creates turn angles of 5.71°, 11.42° and 21.84° respectively. Courses of this type will be termed 'single turn' courses for reference later on in these experiments. 'Double turn' courses will also be defined, in which the boat must return to a point on the original horizontal line therefore requiring a second turn. Angles of 11.4°, 22.84° and 43.68° for the 25, 50 and 100 meter vertical movements are used to achieve this. Figure 5.8 illustrates the different courses under test in this work.

Every combination of course and wind configuration, defined in Tables 4.2 and 4.3 (Pages 120 and 81), will be tested with each controller configuration. The first experiments will include no variation (configuration A) and move towards the most variable environment (configuration I). Every four seconds a wind change will be triggered by the simulator using a Gaussian random number generator to modify the values of the wind speed and direction within the range defined by the chosen wind configuration. Four seconds was chosen as the update rate due to several preliminary experiments showing its suitability. The variability score shown in Table 4.3 is used only for giving an arbitrary ordering for the configurations. This score is calculated by summing the direction and speed variability scores together. For example, the total variability score for 'Low' directional variability and 'low' speed variability (Wind Configuration 'E') would be $(1 + 1 = 2)$.

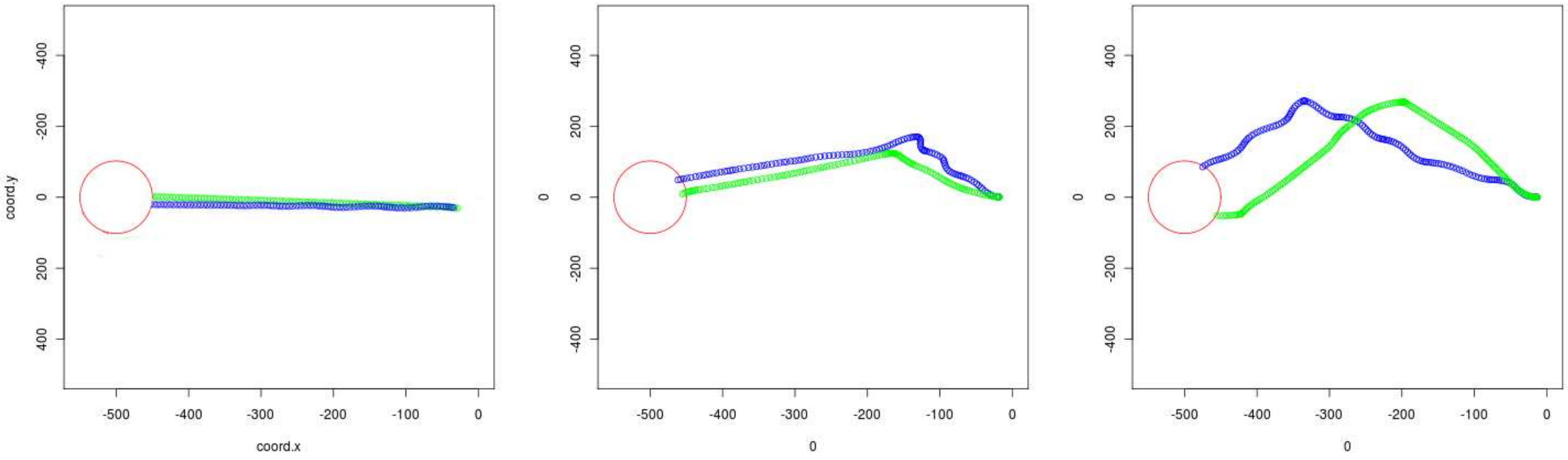
The main difference from the first experiments (Section 5.4 on page 107) is the increase in granularity of the experiments. The drawback is that this second sets of experiments cover fewer varieties of fuzzy logic, specifically the DS and NS fuzzy controller types. Overall, the difficulty in completing the course, even without variation, is greater than in the previous experiments caused by the inclusion of one or more turns required to reach the end goal.

### 5.5.3 Hypothesis

It is hypothesised that as the task difficulty and amount of variation increases, the FOU size of the best performing type-2 controller will increase. This is based on the idea that larger FOU sizes are able to handle greater levels of variation within the environment.

It is also anticipated that as the various wind configurations are tested, the calculated RMSE value will change in a predictable manner. This means that configurations ‘A’ and ‘B’ are likely to show a lower RMSE value than the configurations ‘H’ and ‘I’. A linear increase is not anticipated, as several configurations have the same variability score hinting that they are equal in difficulty. The exact ordering, however is still in question, as the relative effects of the two different sources of variation are unknown. It may be case that increasing the changes in wind direction may have a much higher effect on performance than changing wind speed or vice versa.

As the FOU size is increased, it is expected that the performance will start at type-1 levels (as a size 0 FOU is equivalent to a type-1), followed by an increase in performance followed by a drop, as the FOU increases to cover larger areas of the universe of discourse. It is anticipated that this will result in the worst performing controllers, and this, in the worst cases will prevent the course from being completed at all. For example a FOU size of 180 would mean every fuzzy set would cover the entire universe of discourse, which in turn would lead to a single input triggering every fuzzy set to have non zero membership, causing every rule to fire, which in turn would cause the output of the controller to remain constant.




(a) Experiment 1 course example (low variation) (b) Experiment 2 course example (medium variation) (c) Experiment 3 course example (high variation)

**Figure 5.7:** Plots of example courses performed by different controllers: PI (green) and type-2 (blue) at low, medium and high variation levels. The course end point is determined by a red circle.

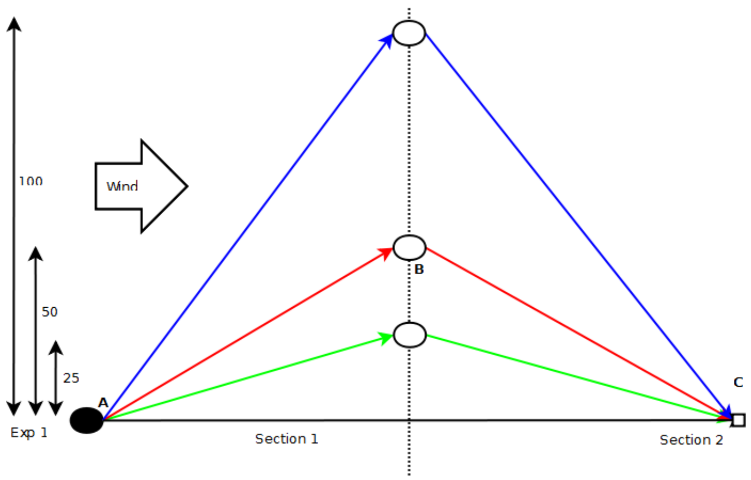



**Figure 5.8:** Each coloured line represents a single experimental course layout. The white circles represent possible end points and the black circle the start point. The angles required for the first turn are 5.71° (green line courses), 11.42° (red line course) and 21.84° (blue line courses) for 25, 50 and 100 meters vertical movements respectively. Not to scale.

### 5.5.4 Experimental Procedure

The experimental procedure is as follows:

- A type-2 fuzzy logic controller is initialised and the FOU size set to one of the following values: (0, 5, 10, 15, 20, 25).
- The simulator is configured to run at one of the configured wind set-ups as specified in Table 4.2 (page 80).
- The simulator is configured to use one of the defined courses. The course configurations are shown in Figure 4.1 (page 79).
- The specified controller is connected to the simulator and allowed to attempt to run the course.
- Data from the simulator and the controller is collected and analysed. Values including RMSE and total time taken for each run are collected. Images such as a course plot can also be generated from this data if required.

Each combination of controller type, parameter value and variability level will be performed and the results collected. Thirty runs of each combination will be executed in order to eliminate any erroneous runs, with the mean value of each taken and used in for the discussion of results.

| Wind Speed | None | Low | High |
|---|---|---|---|
| **None** | A | D | G |
| **Low** | B | E | H |
| **High** | C | F | I |

**Table 5.7:** Wind Configuration Definitions. Each character represents a the shown combination of changes in wind speed and direction

### 5.5.5 Results

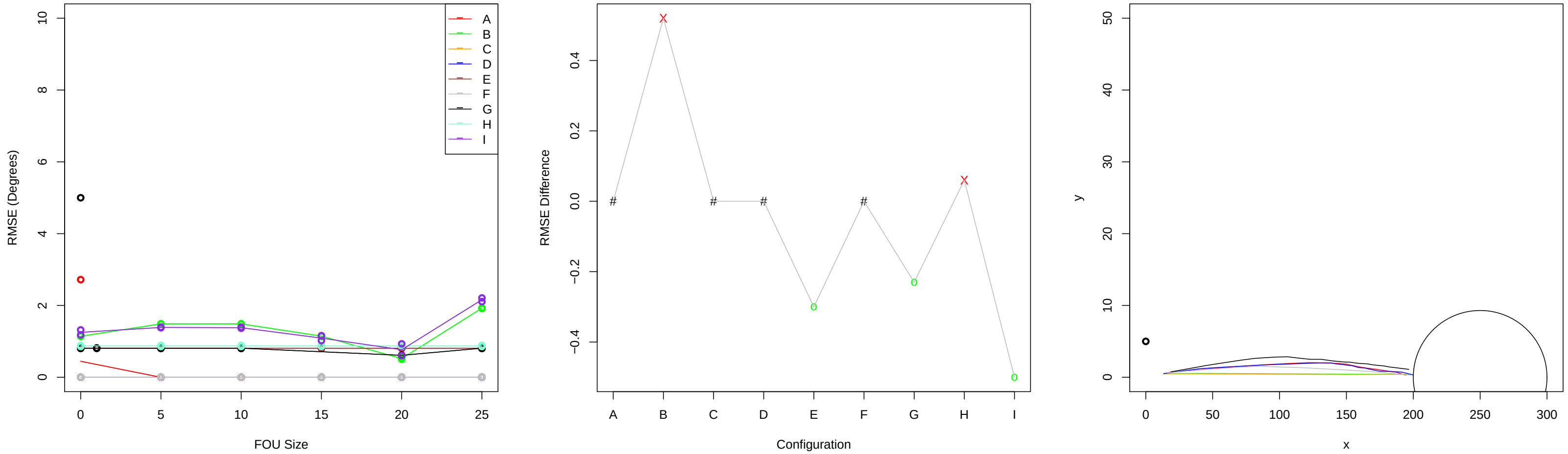
Similarly to the previous section, single-sided Wilcoxon tests are be used to evaluate the statistical difference between two individual batches of experiments. A 5% of significance level is be used to indicate difference between the two input runs. For clarity, course layouts are displayed throughout this section as the vertical distance hyphenated with the number of turns. For example, Single-25 would indicate a course requiring a single turn and 25m of vertical movement to complete.

The first set of experiments were designed as a simulation software verification of the entire software assembly. This involved a comparison of the type-1 controller metric values with the FOU size 0 type-2 controller values to ensure their outcome metrics were statistically similar.

Figure 5.9 below shows the results of a benchmark experiment in which the majority of controllers simply maintain a straight line course. The average RMSE obtained was the expected value of close to 0 with no statistical differences, except possibly at the largest FOU size. In this case, performance decreases significantly, as shown in Figure 5.9(a). It is believed that these results occur because the controllers under test do not need to execute any turns or course corrections in order to complete the task. This leads to the conclusion that any performance benefits or penalties a controller may exhibit do not have a chance to become apparent under such simple circumstances which has also been shown in several prior experiments in this thesis. Figure 5.9(b) shows the changes between the best performing type-2 and the baseline type-1 controller. Finally, Figure 5.9(c) shows several example course plots in which each coloured line represents a single experimental run. All runs indicate a straight line course was achieved as expected.

The next experiment to be considered is shown in Figures 5.10 (page 124) and 5.11 (page 125), which show how the RMSE value (on the $y$ axis) changes as the FOU size is increased from 0 to 25 (on the $x$ axis). Each wind configuration (as indicated by coloured lines) is represented in the figure. Across all the course configurations, the RMSE increases (signifying decreasing performance) as FOU size exceeds 20. Any improvements in performance that do occur, occur before the FOU reaches size 20. This is more obvious in Figure 5.10(b) (page 124), but can also be observed in Figure 5.10(a) (page 124) and Figure 5.11(b) (page 125).

Figures 5.12 and 5.13 (pages 126 and 127, respectively) show example course plots of both single and double turn courses with all the various wind configurations under test represented by coloured lines. The white circles in the figures indicate way points that must be reached to complete the course. The increase in course difficulty can be observed from both number of required turns and the required turn angle form left to right. This is also mirrored in the observed plots of controllers — plots of more difficult courses show many controllers having more turns when compared to less difficult configurations. The green line in Figure 5.13(c) is an example of this behaviour occurring.

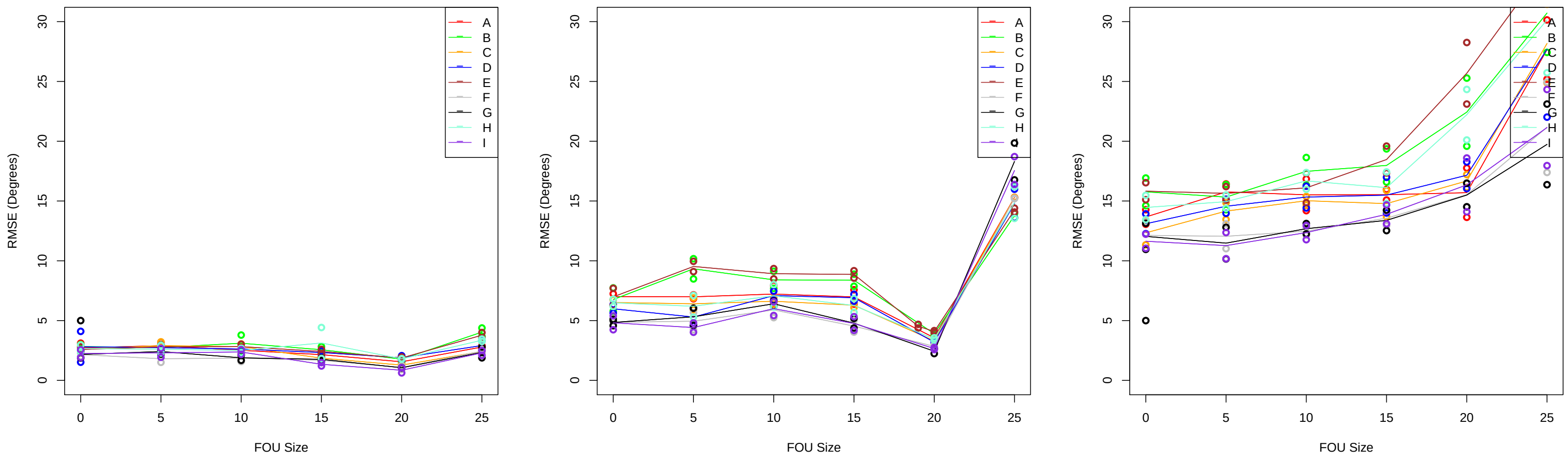



(a) Single-0 RMSE change

(b) Single-0 RMSE difference

(c) Single-0 course plot

**Figure 5.9:** Benchmark experiment results for simulation software verification. Figure 5.9(a) shows change in RMSE as FOU size is increased. Figure 5.9(b) shows the difference between type-2 and type-1 RMSE values — improvement shown by green, decrease shown by red, no change by black. Figure 5.9(c) are course plots for the validation course.

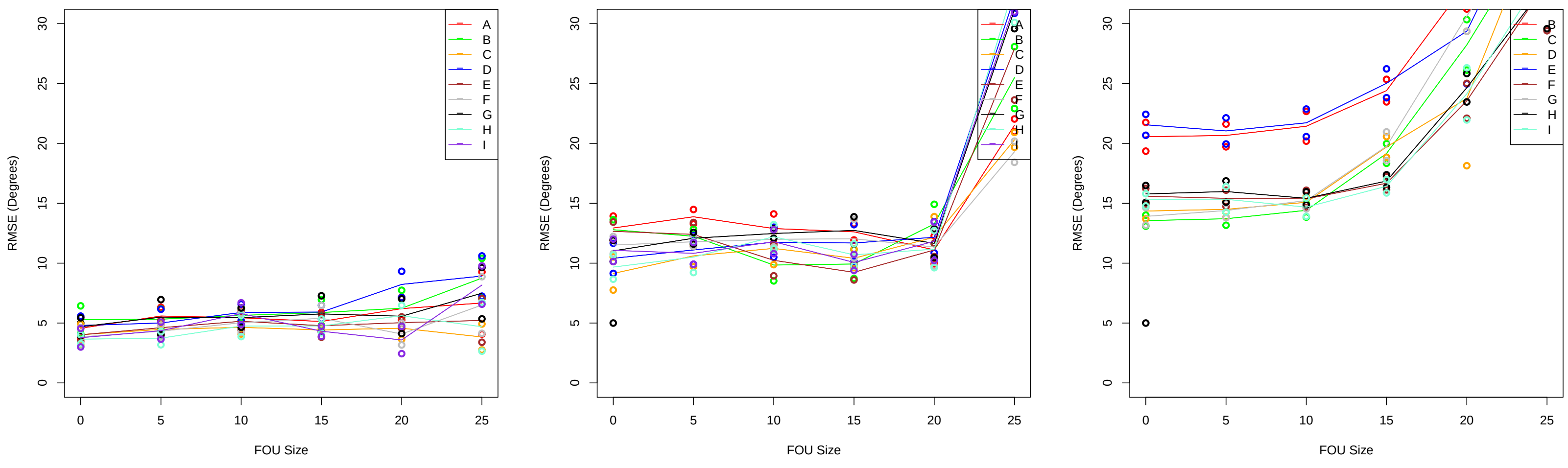



(a) 25m vertical movement

(b) 50m vertical movement

(c) 100m vertical movement

**Figure 5.10:** Single turn experiments showing how RMSE changes as vertical movement is increased (course difficulty increases from left to right). $x$-axis shows the change in FOU size, y-axis indiates the resulting RMSE value. Each line represents the increasing levels of environmental variability present in each experiment.

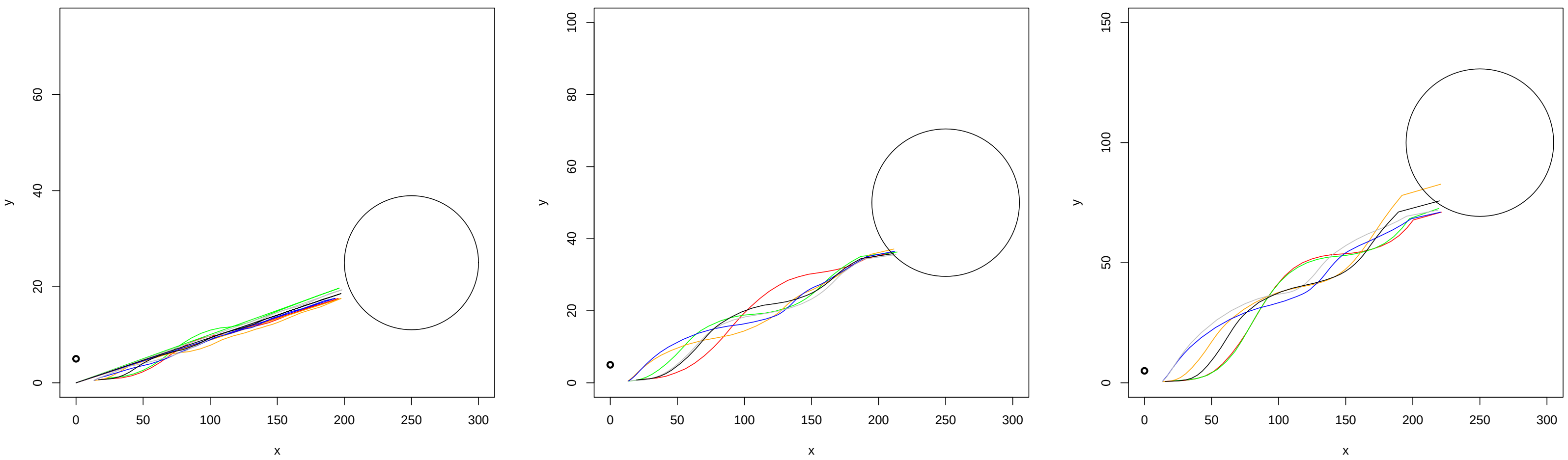



(a) 25m vertical movement

(b) 50m vertical movement

(c) 100m vertical movement

**Figure 5.11:** Double turn experiments showing how RMSE changes as vertical movement is increased (course difficulty increases from left to right). $x$-axis shows the change in FOU size, y-axis indiates the resulting RMSE value. Each line represents the increasing levels of environmental variability present in each experiment.

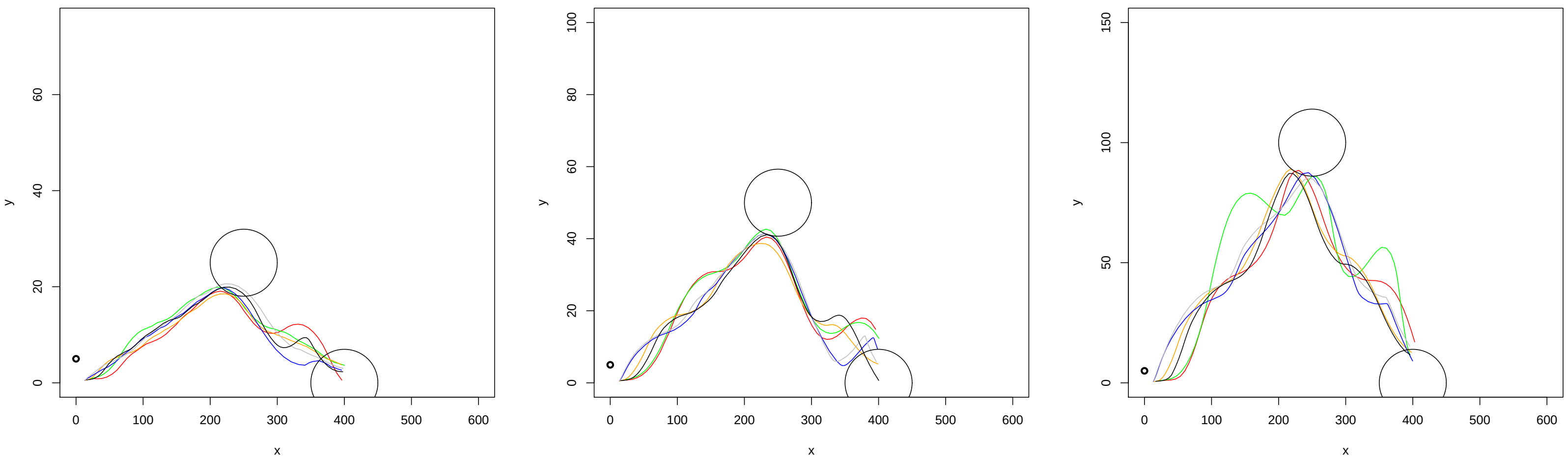


(a) 25m vertical movement course plot

(b) 50m vertical movement course plot

(c) 100m vertical movement course plot

**Figure 5.12:** Example course plots for single turn experiments. Each line represents a different wind configuration.




(a) 25m vertical movement course plot  (b) 50m vertical movement course plot  (c) 100m vertical movement course plot

**Figure 5.13:** Example course plots for double turn experiments. Each line represents a different wind configuration.

| Configuration | Type-1 RMSE | Type-2 RMSE | RMSE Difference |
|---|---|---|---|
| A | 5.93 | 3.56 | -2.37 |
| B | 8.35 | 3.91 | -4.44 |
| C | 6.34 | 3.31 | -3.03 |
| D | 5.90 | 3.20 | -2.70 |
| E | 7.41 | 4.08 | -3.33 |
| F | 4.83 | 2.84 | -1.99 |
| G | 6.32 | 3.46 | -2.86 |
| H | 5.10 | 2.44 | -2.66 |
| I | 4.72 | 2.66 | -2.06 |

**Table 5.8:** RMSE differences between Type-1 and a Type-2 controllers with a fixed FOU size (size 20) on single-50 course layout. This increase in performance can also be observed in Figure 5.10(b)

Tables 5.10 and 5.11 show the p-values obtained when the type-1 controller is compared with the best performing FOU size for every combination of wind configuration and vertical movement, broken down into tables based on the number of turns required to complete the course. If there is no FOU size in which better performance is observed, then this combination is omitted from the table. There are two obvious observations that can be made from these figures. First, there are no points in which the vertical movement is 100. In addition, double turn experiments have considerably fewer points than the single turn. These observations are further discussed in the next section. A p-value of less than 0.05 indicated that the null hypothesis should be rejected i.e. there is a significant difference between the inputs to the test.

| Configuration | Type-1 RMSE | Type-2 RMSE | RMSE Difference |
|---|---|---|---|
| A | 15.29 | 15.70 | 0.41 |
| B | 15.75 | 22.43 | 6.69 |
| C | 11.84 | 16.68 | 4.83 |
| D | 12.33 | 17.15 | 4.82 |
| E | 12.53 | 25.68 | 13.15 |
| F | 11.67 | 15.53 | 3.86 |
| G | 14.53 | 15.50 | 0.97 |
| H | 13.68 | 22.22 | 8.54 |
| I | 12.97 | 16.35 | 3.38 |

**Table 5.9:** RMSE difference between Type-1 and a Type-2 controllers with FOU size of 20 on a single-100 course layout

| Wind Config | Type-1 RMSE | Type-2 RMSE | Vertical Movement | FOU Size | P-Value |
|---|---|---|---|---|---|
| A | 12.94 | 11.11 | 50 | 20 | 1.11e-006 |
| B | 12.79 | 9.84 | 50 | 10 | 4.84e-013 |
| E | 12.66 | 9.23 | 50 | 15 | 3.02e-011 |
| I | 11.07 | 10.06 | 50 | 15 | 1.64e-005 |

**Table 5.10:** RMSEs and p-value of best performing FOU sizes in comparison with type-1 FOU size for double-turn course configurations.

| Wind | Type-1 | Type-2 | Vertical | P-Value |
|---|---|---|---|---|
| A | 2.72 | 1.55 | 25 | 2.40e-011 |
| B | 2.82 | 1.78 | 25 | 2.44e-011 |
| C | 2.60 | 1.28 | 25 | 7.66e-012 |
| D | 2.81 | 1.89 | 25 | 9.51e-010 |
| E | 2.58 | 1.87 | 25 | 2.29e-011 |
| F | 2.16 | 1.08 | 25 | 2.29e-011 |
| G | 2.17 | 1.06 | 25 | 2.48e-011 |
| H | 2.67 | 1.80 | 25 | 1.68e-011 |
| I | 2.24 | 0.85 | 25 | 2.73e-011 |
| A | 7.00 | 3.56 | 50 | 2.91e-011 |
| B | 6.76 | 3.91 | 50 | 2.78e-011 |
| C | 6.51 | 3.31 | 50 | 2.43e-011 |
| D | 5.99 | 3.20 | 50 | 2.73e-011 |
| E | 6.98 | 4.08 | 50 | 1.98e-011 |
| F | 4.86 | 2.84 | 50 | 2.58e-011 |
| G | 4.85 | 2.44 | 50 | 2.80e-011 |
| H | 6.49 | 3.46 | 50 | 2.84e-011 |
| I | 4.82 | 2.66 | 50 | 2.98e-011 |

**Table 5.11:** RMSEs and p-value of best performing FOU size (20 in all cases) in comparison with type-1 FOU size for single-turn course configurations. A smaller p-value indicates the type-1 and type-2 values are more similar hence all the values show there are no statistical differences

### 5.5.6 Discussion

From these results, there are several circumstances in which type-2 based controllers outperform type-1 controllers. However, this does not occur in the majority of cases. It is, in fact, more common for the performance to be similar to the type-1 value (statistically so in most cases) rather than outperform it as it would be expected.

In total there are 324 combinations of wind, controller and vertical movement configurations. From all these combinations, only 23 show statistical improvement when compared with type-1. This represents 8% of the total. This low percentage suggests that the expected increase of performance when switching from type-1 to type-2 does not occur. Its more likely that the outcome performance for both types remain the same. However, in most cases performance significantly worsens unless considerable design effort is undertaken. It can be noted that the RMSE values in these experiments are smaller than those in Section 5.4 (Page 107)., this is due to the nature of sailing in which sailing at a fixed angle to the wind is easier than sailing with the wind coming from directly behind the craft.

The results found in this section are supported by other works in which type-2 performance is compared with type-1 such. One example is the work by Musikasuwan *et al.* [73], where a type-1 controller outperforms, by a small margin, a type-2 based controller. The authors work was more focussed on the number of parameters of the model parameters in each controller. However, the essential result — that type-1 can outperform type-2 under the correct circumstances — agrees with the findings here. Birkin and Garibaldi [13] also demonstrate the improved performance of interval type-2 fuzzy over type-1 based controllers in a micro robot context, further supporting the work shown here.

The results obtained here to not match the the hypothesis made in Section 5.5.3 (page 117), specifically the higher variability levels do not always produce significantly higher RMSE values. This can be seen best in Figure 5.14, in which the RMSE for each wind configuration, vertical movement and turn count combination is plotted with the FOU size being held at 20. In the majority of cases, wind configuration 'B' (red crosses) tends to have one of the the highest RMSE over the entire range of FOU sizes. This contrasts with wind configuration 'I' (orange points), which seem to appear often at the bottom of the graph indicating the best performance. This seems contrary to what was expected, which would be for wind configuration A to have the lowest RMSE and configuration and I to have the highest (as common sense would seem to indicate that more noisy environments are more difficult to sail in). Whether this conclusion is a general result or an artefact of the nature of this specific control problem is not yet known.

Regarding the different wind configurations, (Figures 5.12(a) and 5.13(c)) (pages 126 and 127), as the courses become more difficult, the spread of the results also increases. When the vertical movement is 25 units of distance with a single turn, the results are much closer together, with a difference between highest and lowers RMSE value of 1.04. This contrasts significantly with the 100-double turn experiment, in which the difference is 9.98. This is an expected result, as with each increase in course difficulty the number of course corrections required by each controller also augments. This means that there is greater scope for a controller to demonstrate its improved performance (or lack thereof).

The correlation between the different wind configurations and the performance change that occurs between type-1 and type-2 controllers is difficult to determine as it is not consistent across all experimental scenarios. This could be due to the ordering of the configurations, as defined in Table 4.2 (page 80). In the tables, multiple configurations have been given an equal variability score based on the assumed equal weighting of the two variability sources. This may however be a faulty assumption. The results also contrasts with the findings made by Sepulveda *et al.* [86], in which type-1 and type-2 controllers are tested and the type-2 outperforms the type-1 in all cases. This occurs both with and without variation within the environment and the difference in performance seems to have an increasing correlation. This suggests either the difference is down to the different application or the tuning of the type-2 is considerably better than the type-1. Another point to consider is that Sepulveda *et al.* have not tried as many different levels of variation as have been presented here. Therefore, the differences found here have not been able to present themselves have not been considered in their set-up.

The addition of turns to increase the difficulty of the course has, as expected, a significant effect on the performance of all controllers. It can be observed between Figures 5.10 and 5.11 (pages 124 and 125) that every RMSE value is higher in the double turn situation when compared with the single turn. This can be explained by considering that in double turn courses the controller needed to turn the boat over twice as much than the single turn courses, making the track more difficult to complete.

These experiments have further increased the level of complexity of the experimental set-up and have resulted in somewhat more obvious differences between type-1 and type-2 fuzzy logic. This implies that the methodology used is capable of showing differences between type-1 and type-2 fuzzy control systems under a wide variety of situations. However as there were not as many cases where type-2 significantly outperformed type-1 it may still be the case that these experiments still do not have sufficient difficulty to enable the recommendation of type-2 control in this application.

## 5.6 Discussion

The rule based experiments in Section 5.2 show that, in comparison to the work performed in the previous chapter utilising the same techniques as in Section 3.3 that the more developed environment show better differentiation between the different configurations. With the largest difference between the type-1 and type-2 controllers equating to approximately 20% of the total number of input sets. This gives the first evidence that a more developed environment and controller set-up gives type-2 controllers a better opportunity to present differing performance levels compared with type-1 control, whether this equates to improved performance is the subject of study of the experiments that follow.

The control surface investigation in Section 5.3, presents an investigation into how control surfaces change as the controllers are varied by a fixed set of parameters. The overall results obtained show that type-2 and non-stationary controllers with larger FOU sizes tend towards smoother control surfaces which may imply improved performance, though this cannot be directly seen in these results. However it gives a good basis for the development and execution of the experiments that follow in the chapter.

Experiment four focuses only on type-1 and type-2 varieties of fuzzy control to the exclusion of the DS and NS varieties of fuzzy logic. It also introduces much more granularity into the levels of variation and course difficulty with nine different levels instead of three in the previous experiment. Similar to the previous set of experiments, the results found are encouraging because a peak in performance is again found at an

FOU size of 20. However, they still do not reflect the correlations anticipated that could be applied to a more general setting. The hypothesis for this investigation was that the desired outcome would be a decrease in performance as the amount of variation present in the environment increases and the controller configuration and course difficulty are kept constant. However, this was not always the case, for example: the RMSE change between the type-1 and type-2 systems do not exhibit any regular observable pattern.

In the results of both experiment sets three and four, show some similar patterns observed. However, overall there is not enough statistical evidence to support a direct causation of either parameter value or environmental variation upon RMSE value obtained. The nature of the application used here is fairy complex coupled with a fairly sophisticated controller may mean that there is no direct and obvious way to reliably predict performance levels.

In order to counteract the possible effects of having two sources of variation, such as in the Section 5.5, the analysis must take into account each of the sources individually, before attempting to observe their joint effects. From experiment two, configurations 'A', 'B' and 'C' should be grouped for analysis, as they only regard changes in the wind direction. Subsequently, its necessary to observe how the differences in the first group compare with configurations 'A', 'D' and 'G', where only the wind speed is changed. With these analysis, finally, the performance of configurations 'E', 'F', 'H' and 'I' can be effectively assessed where the combination of changes in speed and direction come into effect.

One approach to solving the problem of operating within noisy and uncertain environments is discussed by Brooks [18] where 'Relational Maps' are introduced as a means to modelling environments and incorporating the related uncertainties thus allowing robots to reduce the effect upon performance. These 'Relational maps' are described as *rubbery and stretchy* rather than using a traditional fixed coordinate system. This implies that Brooks believes that flexibility in such systems is an important feature for good performance within uncertain environments. In general features of the more sophisticated types of fuzzy logic systems, such as the FOU size of interval type-2 systems can be shown to provide this sort of flexibility.

The interaction of multiple sources of variation and randomness is also a point that merits discussion. It is hypothesized that, as the number of sources of variation increases, the complexity of modelling the combined effects upon performance will expand exponentially, making understanding and adapting to such situations more and more difficult. An interesting development would be to use it in a real world situation where noise, and variation cannot be controlled or accurately measured in every dimension. Alternatively, a more sophisticated physics engine could be integrated into either Tracksail or some other simulator, however this would be a difficult task with many considerations including processing and memory requirements and the complexity of the maths involved.

All of the three approaches to comparison (Rule fire experiments, studying the Control surface and RMSE comparisons) used in this chapter have shown to have some ability to differentiate between differing fuzzy logic controllers configurations. Overall the rule fire experiments show the least useful information as the number of rules which fire cannot be directly linked to performance in a real situation, in addition it seems that this technique is the least sensitive to change, showing the smallest differences between the tested configurations. The control surface study results in interesting graph shapes in which the change in the smoothness of the transitions could be easily seen, however this does not provide any quantitative evidence as the RMSE study does, which shows that the RMSE method is the most useful overall.

The effect of FOU size in these experiments is considerably more noticeable than the results obtained in the previous chapter. This supports the theory previously proposed in Section 3.6.3 (Page 68) where the difference between the largest and smallest FOU sizes was considered too small to cause trigger significant differences. The differences here were 25 — a approximately 5 times greater difference, with differences generally becoming apparent when the FOU size exceeds size 10. While it is difficult to generalise this sort of result, it does give a starting point for discussion.

The methods used for comparison in Sections 5.2 and 5.3 operate across the entire possible input space with no concept of variation and therefore the idea of task difficulty cannot easily be applied. The sections which follow however in which increasing numbers of turns, with increasing turn angle offer a simple mechanism to model such a concept.

Using wheeled robots for a similar sorts of investigation has been discussed by Birkin [13] and Hagras [41] (which both find Type-2 based control to be superior to type-1 in a subset of cases). Wheeled robotics however, do not rely on external processes such as the wind for their movement. In order to use a wheeled robot, a mechanism by which the amount of forward motion obtained from a single value of input into the motors vary by some degree. This may be achieved, for example, by using different surfaces upon which the robot would move — a hard firm surface, which would provide sufficient friction to allow good movement would vary significantly with, for example, deep pile carpet, sandy or icy surfaces. For a given amount of movement, such as a complete turn of a motor, each surface would cause a different amount of forward motion to be achieved, introducing a source of variation into the environments in which wheeled robots operate.

### 5.6.1 Conclusions

From the first two approaches certain conclusion can be drawn, specifically: The differences in the number of rules which in type-1 and type-2 configurations is significantly higher than in the previous chapter, showing that increasing the complexity of the set-up has achieved one of its goals of making differences more obvious and observable in each comparative approach; secondly larger FOU values seem to give a smoother transition across the control surface implying that a controller with a larger FOU will give a smoother response to inputs across specific areas of the input space, potentially giving better performance when the inputs are within this one of these regions.

One of the main objectives for this work was to observe if a more complex experimental environment would enable observation of more obvious differences between different configurations of controller. Including both different configurations of the same controller (such as varying the FOU size of a type-2 fuzzy logic controller) and between different types entirely, such as between type-1 and type-2. The results in this chapter shows a significantly larger range than in the previous chapter, so this objective has been achieved to some degree. However there are still many combinations in which performance is not significantly different, which suggests that there is still development that can be done in order further improve findings. In the next chapter this is attempted by moving the controller into a real world environment in which the variation and randomness should be significantly higher, making the environments in which the experiments are performed even more complex. It is hoped that at this level, different configurations will present significant differences in performance allowing more solid conclusions to be drawn.

Secondly, it must be noted obtaining optimal performance from a given controller was not a goal of this work, instead the goal being to try and observe if and when type-2 fuzzy logic would outperform type-1 based systems and under what conditions this would occur. It has been shown in this work that this certainly does occur using the experimental set-up, it is not a given outcome. Further work is required to determine the exact reasoning and how and why this occurs at the levels it did.

## 5.7 Summary

In this chapter the methodologies used to compare fuzzy logic controllers within the context of two sets of simulation experiments are presented. A new fuzzy logic controller is introduced in order to address some of the shortcomings found and discussed in the previous chapter. The first experiments investigates which rules fire and how this changes as the FOU size increase. This is followed by a short study into the control surface of each of the investigated fuzzy logic controllers. Both of these begin to show the sort of differences that can be observed when the controllers are used directly, without a surrounding control task. As the differences between the various controllers are usually still relatively small, the decision is made to move into a real world environment. Real world environments are generally considered to be more complex in terms of the amount of variation present in the environment and therefore present more of a challenge to fuzzy logic control systems, giving better controllers more scope to present differences in performance.

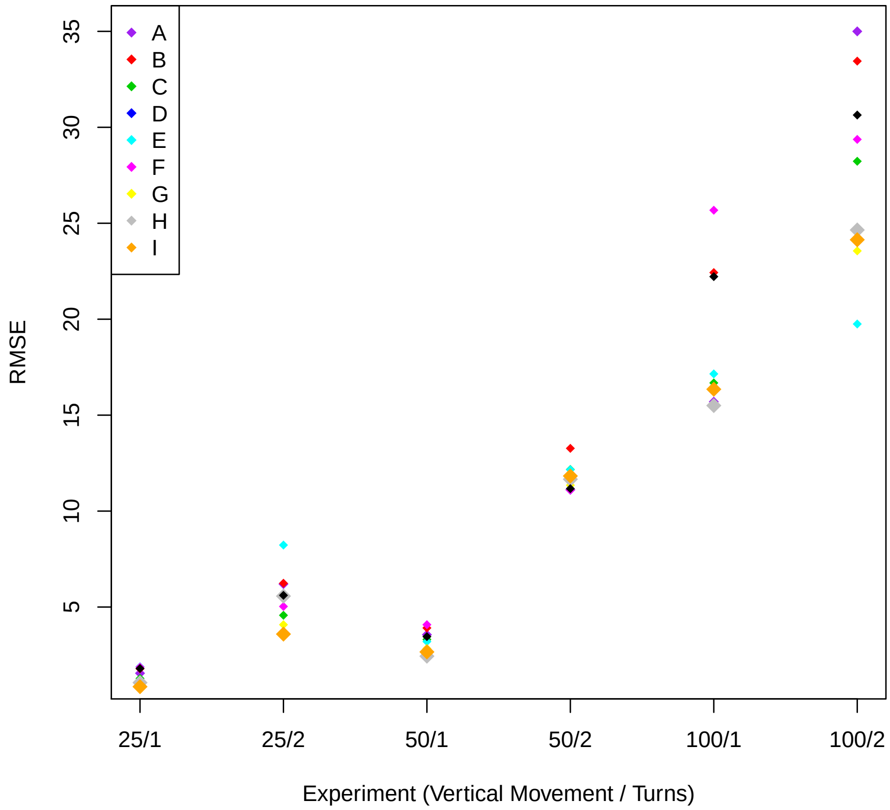



**Figure 5.14:** RMSE values for each wind configuration for each experiment. Points $x$-axis represents a given course layout from easier to complete to most difficult. $y$-axis indicates the RMSE for a given experimental set-up. Each coloured point represents the RMSE of given level of environmental variation for the specified course layout

# 6

# Real-World Experiments

## 6.1 Introduction

Experimentation within a simulation environment is a technique often used to complement real-world trials. Simulation is a collection of procedures mimicking reality with the purpose of providing further insights into the real system, in a controlled environment, through the use of ‘what-if’ scenarios. These techniques, therefore, present some advantages when compared with real-world experimentation, as discussed by Miglino *et al.* [71]. In relation to the problem of autonomous sailing boats, for example, simulation has the benefit of being time and cost-effective. Within the simulation environment, sensors and robots are never faulty unless these errors have been purposely added. In addition, it is possible to control and tune sources of variation for a given experiment, which is impossible to do in reality. Additionally, there is no need to rely on weather conditions to perform a run; replicating the identical experiments under the same parameters is a straightforward task. However, by their nature, simulators are restricted in their scope and even the best systems do not encompass every possible scenario that could occur in the real world. As a consequence, the experiments in the simulation environment, such as those performed in our previous chapters, are limited to pre-defined scenarios, in which uncertainty sources are derived from a single random number generator.

The previous chapters have shown that the different fuzzy logic controllers considered in this thesis, specifically type-1, type-2, non-stationary and dual surface controllers, can present different levels of performance as the environment in which they operate change. However, in the first experimental chapter 3, the differences found were minimal, which have been attributed to the simplicity of the application. This is addressed in Chapter 5 where a more sophisticated controller and application were introduced, which gave improved results. In this chapter the same controllers are used in the context of real-world experiments which, as discussed above, introduce additional variability, making the task in question more difficult to complete than in simulation. This is done in an attempt to show greater differences and therefore assist in answering our research questions.

This chapter is organised as follows. In Section 6.2 the nature of real-world environments and the differences present in comparison to simulation are discussed. Section 6.3 describes a pilot real-world study together with the preliminary results obtained. This section concludes with a discussion of the potential shortcomings and improvements to the pilot methodology. A larger study that includes the improvements proposed in the previous section are presented in Section 6.4. The experiments of this section focus on the interval type-2 fuzzy controller. Finally, the outcomes of these studies are discussed and summarised in Section 6.5 and the chapter is summarised in Section 6.6.

## 6.2 Real-World Sailing

### 6.2.1 Hardware

The robotic hardware used is described in detail in Section 4.4. The sailing robot, as shown in Figure 6.2, has a wind direction sensor on top of the front mast, to which the front sail is attached. The boat is also equipped with a GPS receiver and digital compass modules. These devices calculate the position and the bearing required as inputs to the controller.


**Figure 6.1:** The robotic boat platform used in the experiments.

### 6.2.2 Location Selection

Selecting an appropriate location for performing real-world experiments is of great importance and several requirements must be considered:

**Good view of the sky** The location data is provided by (non assisted) GPS and the receiver is located on board of the boat. Therefore it must have an unobstructed view of the sky. Buildings, large trees and other large structures can cause reflection of the GPS signal, and this in turn can cause a drop in accuracy. The GPS system is further validated by using a phone-based GPS receiver, which employs an A-GPS (assisted) system with GPRS and 3G technologies used in conjunction with a standard GPS receiver in order to increase the accuracy, consistency and robustness of the location reading.

**Easy access to waterside** Natural lakes often contain reeds, weeds and mud banks, which impede the launch and collection of the boat at the start and end of experiments. The difficulty in reaching the boat interferes with the consistency of each experiment and reduces the number of experiments that can be achieved in a given time frame.

**Minimal currents and water-based disturbances** Rivers, streams and other flowing bodies of water introduce variation that currently cannot be handled using the current hardware. Objects such as waterfalls, fountains and inflow or outflow pipes cause similar disturbances in a smaller scale. Similarly, additional variation may be introduced from sites with large amounts of local wildlife, including ducks, swans and fish.

**Unobstructed wind access** Not only do tall buildings, lines of trees and other high structures create reflection of GPS signals, but they can also cause the wind to act in a very unpredictable manner. While these experiments are a study in environmental variation, attempts are made to minimise it to some degree.

When considering this list, the use of an indoor swimming pool, in which wind could be controlled by a number of large electric fans, was found to not to be feasible, as in there would be no view of the sky. After considering several alternatives, a location matching all of the requirements was found in the city of Norwich (Norfolk county) [1]. The location is a purpose-built boating pond. An aerial view of the location is shown in Figure 6.2 (page 143) with a view from a nearby building shown in Figure 6.3 (page 144). The pond is 20m wide by 30m long, which gives a 70cm boat (the length of the robot used in our experiments) plenty of space to manoeuvre and turn. Twice a week the pond is used by model boat enthusiasts for a large variety of different types of craft, which gives confidence that it is suitable for our use.

[1] http://www.norwichmodelboatclub.com/

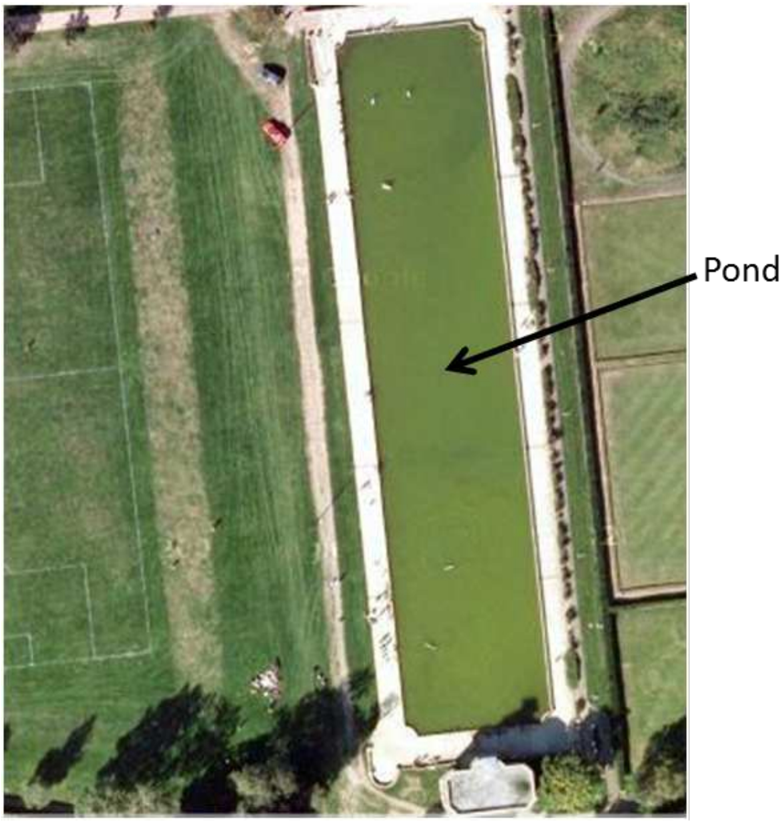



**Figure 6.2:** Source: Google Maps. Aerial view of the model boat pond.

### 6.2.3 Experimental Design: from Simulation to Real-World Experiments

Once experiments move from a simulated environment to the real world using a physical robot, a considerable number of additional variables are introduced. These variables introduce far more sources of variation making the environment considerably more complex. As previously discussed in Section 2.7, there are many different sources in such environments, including:

**Sensor variation** The sensors on the boat have limited accuracy. It is therefore expected that considerable variation and randomness will be introduced by this device.

**Figure 6.3:** A view from the boat house.

**Actuator variation** The boat has motors which can only be set within a certain degree of accuracy. It is also possible for overshoot to occur; this commonly happens when there is a large change in the sail position, and the momentum of the sail causes it to move past the endpoint. While it is not anticipated that this form of randomness would have a significant effect, it should, however, still be considered.

It is expected that with the transition from the simulation environment to the real world, the experimental scenarios would become considerably less predictable. The wind readings should therefore reflect a far more dynamic and unpredictable environment than under simulation. This is due to the real-world nature of variable wind and the fact that the wind sources are not restricted to a single random variable, such as the random number generator used in simulation.

As discussed previously, in controlled simulation scenarios the generations of thousands of runs at a time is easily achievable. In the real world, however, it is significantly more laborious to perform a single run where each run takes an estimated at thirty minutes. It is infeasible therefore, to perform the thirty runs for each combination (requiring an estimated 15 hours per configuration) as it has been carried out in the simulation environment. On the other hand, a minimum number of repeats are required for the purpose of making more general conclusions with a certain degree of confidence. Ten repeats are therefore established as the optimal number with a good balance between having sufficient repeats and ensuring enough variety in tested configurations.

Due to the amount of time required to conduct real-world experiments, fewer experiments can be run, and this has led to increasing the interval between the different sizes of the FOU, compared to what was used in the simulations. FOU sizes $0$, $10$, $20$ and $40$ are therefore tested, as opposed to $0$, $5$, $10$, $15$, $20$ and $25$ used in the previous chapter. These values are selected because it has been found that a size of 20 often gives good performance, and time constraints only allow four different sizes to be tested; this leads to selection of values half and double the size of the best performing size, as well as size 0 to allow comparisons with type-1 performance. The finer increase in FOU sizes used in simulation is selected because the level of variation in the environment is controlled and therefore a correlation between the FOU size could be sought. However, this control is less achievable in the real world without a large investment in time for additional experiments.

### 6.2.4 Hypothesis

It is hypothesised that this application will present the most difficult task for the fuzzy logic controllers attempted in this thesis so far. Previous work in this thesis has shown that differences can certainty be found between the performance levels type-1 and (certain configurations of) type-2 fuzzy control but that it can be difficult to observe depending on the exact environmental scenario. This leads to the hypothesis that this additional difficulty and variation found in real-world experiments will allow the differences between type-1 and type-2 to become more obvious. This in turn will potentially allow the identification of the values for task difficulty, environmental variability and controller configuration that are most likely to cause type-2 control to outperform type-1 fuzzy control.

## 6.3 Pilot Study

In order to test the methodology and to determine values such as the time taken to complete an individual experiment, a pilot study was performed. The purpose of this study was to make sure that the correct data is being collected. Furthermore, the pilot provided verification that when a larger study was performed, it can be streamlined and therefore much data as possible collected. Additionally, potential problems in the experimental method, such as a too simple a course, can be identified and addressed.

### 6.3.1 Methodology

The course that the boat must complete is simple and is shown in Figure 6.4, with the defined course superimposed on the aerial view of the pond. The course is equivalent in distance to the width of the pond. This has been chosen for several reasons. First, observations of the wind show that the most common prevailing direction is that marked as a red arrow in the figure; Secondly, the relatively short distance means that a well-performing controller will complete the course quickly, allowing more runs to be performed within the time frame. However, it is anticipated that a badly performing controller will have difficulty managing even a short course, such as the one defined.

The procedure for performing the experiment is defined as follows:

At the start of the day, the exact end point is determined using the A-GPS receiver. This is performed each day to ensure that effects such as atmospheric conditions, which can cause slight variations in GPS readings to occur, are minimised. This way point is programmed into a way point parameter file on-board the boat. For each separate experiment the following procedure is executed:

1. The boat hardware system is reset to realign all motors. The controller is initialised and begins to read sensor values and change actuator positions in response.

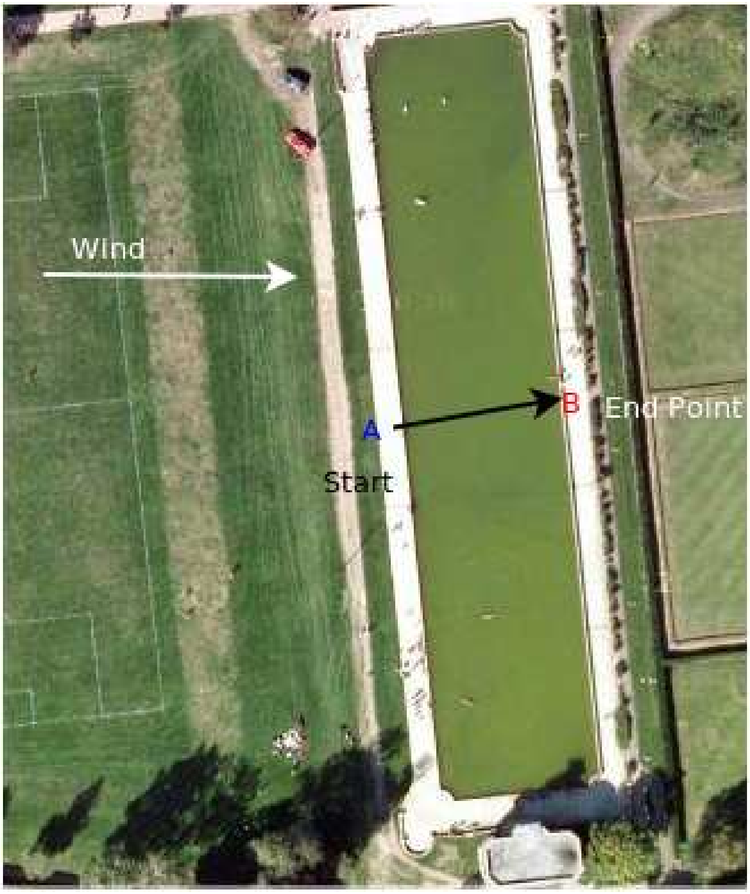



**Figure 6.4:** Pilot course. Start point is shown by ‘A’ and end point shown by ‘B’. Prevailing wind shown by white arrow.

2. The robot is lowered into the water and aligned by the operator to face directly towards the opposite side (e.g. towards point B). The boat is released.
3. The operator uses a video camera to record the boat’s progress.
4. When the boat reaches the opposite side of the pond the run time is noted and the run is considered complete.
5. The boat is removed from the water and walked back to the starting point, where the controller is stopped and the data transferred.

### 6.3.2 Experiments

These experiments were performed during the week of 11th June of 2012. The weather was warm but there was sufficient wind to sail the boat successfully. A total of eight experiments were successfully run. Three fuzzy controllers were used: type-2, DS and PI (Proportional Integral).

The PI controller was used both as a hardware and software test case, as it had previously been used by the system builders to test the platform and shown to have reasonable performance. Type-1 and type-2 based controllers are the major focus of study in this thesis, the aim being to determine how they perform under real-world conditions; they will therefore be the most studied varieties in the preliminary investigation. DS and NS controller types were included in order to test if they demonstrated significant differences from the more standard type-1 and type-2 varieties.

As the type-2 controllers were the main focus, it was decided to spend more time running experiments with this type of controller, allowing several different parameter values to be tested. However, this was at the expense of the NS and DS controller types where only one parameter value for each controller could be attempted. During the study, no runs using the NS controller were successfully completed due to hardware and software initialisation problems.

Data was collected by the systems on-board the boat, including all of the sensor readings which allowed the RMSE and the average speed to be calculated. The runs were also recorded using a video camera so that any obvious problems, such as sail overrun (where the sails would attempt to cross paths and lead to hardware damage) could be observed. It also gave the ability for the numerical data to be more easily associated with observational data. For example, if the boat was to hit debris, thus cause a performance change, this could be observed in the video.

### 6.3.3 Results

Table 6.1 shows the summary of the data collected during the pilot study, which lasted for one week in total. The data in the table has been normalised so that it is all on a common scale. It can be observed that a small amount of data has been collected, totalling eight complete successful runs. A large number of unanticipated software problems and errors occurred during the first runs and this slowed down the experimentation process. These issues were fixed and the stability of the system was therefore found suitable for the main study described in the next section. This was reinforced by the fact that during the last two days of the study the amount of data collected, after the fixes were applied, was considerably higher.

| Experiment | FOU | Controller | RMSE |
|---:|---:|---|---:|
| 1 | 5 | Type-2 | 0.55 |
| 2 | 5 | Type-2 | 0.45 |
| 3 | 10 | DS | 0.47 |
| 4 | 10 | DS | 0.45 |
| 5 | 20 | Type-2 | 0.44 |
| 6 | 40 | Type-2 | 0.54 |
| 7 | N/A | PI | 0.52 |
| 8 | N/A | PI | 0.56 |

**Table 6.1:** Pilot study collected data. Each row represents a single experimental run.

### 6.3.4 Pilot Analysis and Discussion

The amount of data gathered during this pilot study is insufficient to be in anyway conclusive, though it is enough to show that the base methodology is acceptable, although small modifications, as outlined in Section 6.4.1, are required to ensure that a large data set is collected and the data collected is as high a quality as possible.

In the pilot experiments there was no fail condition defined, since the boat was given as much time as it needed to reach the opposite shore. This varied between four minutes in fast cases, and over thirty minutes for the slower instances. The limited amount of daylight reduced the number of runs in total that could be achieved each day. In a future investigation a time limit should be introduced after which the run should be considered a fail. This should maximise the number of runs that can be performed.

It was observed during the pilot study that it was possible for the robot to become wedged between an overhang on the pond edge and the waters surface. This can cause the boat to stop moving though the controller and actuators are still running. This observation needs to be taken into account when introducing any changes to the experimental methodology. This is considered in Section 6.4.1.

## 6.4 Type-2 Main Study

In this study, the focus is on a comparison of type-1 and type-2 fuzzy logic controller types, with the aim of establishing if in real-world experiments, the differences between performance levels are more easily observable and can further strengthen the original hypothesis made in Section 6.2.4.

### 6.4.1 Changes in Methodology

As in the previous chapter, an investigation is conducted into type-2 fuzzy logic and specifically how the change in the FOU size affects performance. However, as parameters of the wind cannot be controlled as in simulation they will simply be recorded. This means that every run has a unique level of variation and wind levels that cannot be reproduced.

This work has three objectives, namely:

1. to observe the effect of FOU size upon performance;
2. to demonstrate how real-world environments present a difficult task for controllers under test in comparison to the simulation environment previously used.
3. to show which configurations perform best at differing degrees of variability within the environment.

In order to minimise the differences in the level of variation in the environment between the each run, all runs of each controller type are conducted sequentially, instead of grouping the experiments per FOU sizes. It is anticipated that the weather conditions will not change significantly over the time it takes to run one experiment for each considered controller configuration, estimated at 3 hours.

Based on shortcomings identified when evaluating the results of the pilot study, three changes have been made:

1. A required turn in order to complete the course is introduced. The new course is shown in Figure 6.5. This turn has been added in order to increase the amount of work that is required for each controller to complete the experiment and therefore allow 'better' controllers to differentiate themselves from those which perform poorly.

2. The filming component is eliminated. In these experiments, the operator does not record the course. Recording adds complexity to the experiment, as the operator must control the boat from a laptop, while simultaneously recording it and being in the correct position to stop the boat as it reaches the end point. The videos made as part of the pilot study did not really assist the task and therefore they will be eliminated.

3. A time limit of fifteen minutes to complete the run is added. After this time, the run is considered a failure and the boat is removed from the water as soon as it reaches any side of the pond and the data is discarded.

### 6.4.2 Experiments

The way point data is collected every day and uploaded to the boat as discussed in Section 6.3.1. Each experiment is then performed following the modified run procedure:

1. The boat hardware system is reset, realigning all motors. The controller is initialised and begins to sense values and change actuator positions in response.

2. The robot is lowered into the water and aligned by the operator to face directly towards the opposite side (e.g. towards point B). The boat is released and a timer is started.


**Figure 6.5:** Modified boat course including newly added turn indicated by $\theta$. Start point is shown by 'A' and end point shown by 'B'

3. When the boat reaches the end point, the end time is noted and the run is considered finished.
4. If the time exceeds the limit, the boat is removed from the water as soon as it is possible to do so (i.e. it reaches an edge).
5. If the boat reaches the end point, it is removed from the water and walked back to the starting point, where the controller is halted and the data stored.

These experiments were run during the week of 3rd December of 2012, when the overall weather was cold but relatively calm. There were no adverse weather conditions such as storms. However, sufficient wind to sail the boat was present with overall levels slightly higher than those observed during the pilot study.

### 6.4.3 Results

Figure 6.6 depicts how the calculated RMSE changes as the FOU size is increased. It can be observed that there is a small increase as the FOU size moves from size 0 to 10, and after this point it starts to decrease. Additionally from this graph it can be observed that the RMSE for FOU size 40 decreases by a noticeable amount. This further supports the hypothesis that a real-world environment will allow better or worse performing controllers to become apparent, more so than in the experiments performed in previous works.

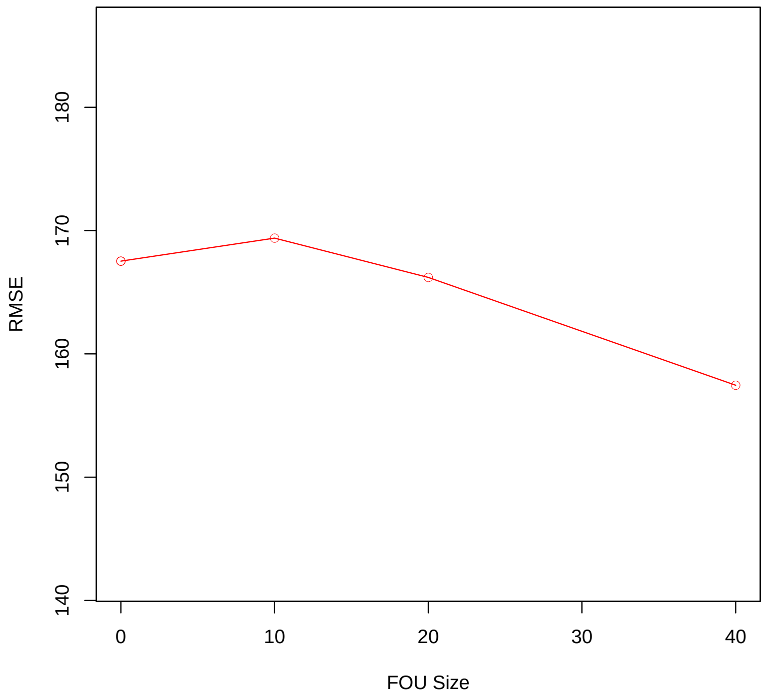



**Figure 6.6:** RMSE of the different sizes of FOU under test

Table 6.2 provides a more quantitative outlook on the data obtained from the experiments. The runs column indicates how many runs were performed for each FOU size, with the goal being 10 runs. The p-value shown here is the value obtained by performing T-test between the rows FOU size and FOU size 0, with most p-values resulting a

| FOU | Runs | Mean RMSE | Std Dev | P-Value |
|---|---|---|---|---|
| 0 | 12 | 162.95 | 13.81 | N/A |
| 10 | 11 | 162.93 | 10.18 | 0.27 |
| 20 | 14 | 157.57 | 15.86 | 0.15 |
| 40 | 10 | 157.04 | 12.36 | 0.07 |

**Table 6.2:** Mean RMSE with its variance and the p-value result of a t-test between type-1 and the indicated FOU size. A smaller p-value indicates a less significant difference.

statistically significant difference. In order to keep the input data sets the same size for the t-tests, the first 10 runs of each configuration were used.

The RMSE values were calculated using the same calculation as those used throughout this thesis. Overall, the results show the RMSE decreasing, meaning improved performance as the FOU grows. It can also be noted that the RMSE values are much greater than those obtained in previous works — showing that the environment does have more variation present, which makes it a more difficult task to complete. The standard deviation is also much larger in these experiments than those in the previous experiments. It is believed that this occurs for the same reasons as for the RMSE value — the environment introduces more variation, within a smaller data set.

## 6.5 Discussion

A significant number of issues arose while working in the real world that were absent during the simulation experiments. Some of these issues were anticipated, some not. Overall, the hardware used in the boat was reliable and worked as expected. However, some relatively minor problems occurred, such as the battery life of the platform, which did not provide enough power for a complete day of experiments on a single charge. It has been suggested by Doel and Pai [95] that battery usage is another metric of performance that should to be considered. The goal would be minimising the battery usage by reducing the amount of motor usage used during an experiment. This obviously involve a trade-off, especially when significant variation is present, as each course correction would require motor movement which in turn would use battery power.

The behaviour of the boat in the water was much more variable than anticipated, with many small movements occurring in all directions. For example, the tilt of the boat seems to have a significant impact on how much control the rudder has upon the boat in a given direction, as well as the amount of wind that is converted into forward motion. As the boat has a tilt sensor as part of the digital compass, this could be used in future work to further increase performance, using the tilt as an additional input to the fuzzy system.

Performance and localisation of the GPS sensor on-board the robot was satisfactory, and in general it matched the measurements of the external A-GPS device. However, the time taken to obtain a good satellite fix was considerably longer than anticipated. A-GPS chips are becoming more widespread and hence the replacement of the current GPS receiver with a more accurate device would benefit the accuracy of the position data generated in future work.

One potential issue, to be subject of future investigation and study, is the effect of the update rate of the sensors/actuators. It has already been explained in Section 4.4.1 why the selected rate was used. However, observations show that at some points in real-world experiments, the controller was not able to handle certain conditions, such as when a large gust would blow, with its direction significantly different from the current direction. This would cause the boat to over or under-shoot a turn and therefore dramatically alter the speed.

The FOU sizes used in the main study of this chapter show some changes in results as they increase. However, these changes are smaller than would be expected in a real world with large levels of variation. This supports the idea that the levels of variation present in this real-world environment are not as large as anticipated, either due to the prevailing conditions or the location selected. The relatively small differences obtained between the different FOU sizes seem to imply that FOU choice is a small factor in performance of a fuzzy-based system.

The sample size used in this experiment was established by the methodology and limitations upon potential duration of the study. As the statistical test results (Table 6.2) generally indicate no significant changes, it may be possible that the sample size needs to be significantly larger, in order for the changes to become apparent. This was attempted to be partially corrected by increasing the difficulty of the task between the pilot and main studies — that of adding in a turn into the course. However, this did not alter the results.

The complexity and difficulty of the course (i.e the number and size of each turn), is one potential reason for the similar performance levels of each different configuration. Additionally, the tendency of the wind to blow as shown by the white arrow in Figure 6.5 (page 152) should also be considered. The fact that the wind most often blows perpendicular to the route required means that overall the course is fairly easy — a human sailor, for example, would have little difficulty in completing a similar course. It may be that orienting the course so that the boat must move into the wind or at a more difficult angle may further differentiate between different controller configurations and ease comparisons.

In the pilot study, the results presented very little variation regarding the RMSE values, and this caused changes to be incorporated into the main study, where an extra turn was added to the course. This should have made the route more difficult to complete, allowing better controllers to show correspondingly better results. This was indicated by a larger spread of RMSE values across different configurations. While the small amount of data collected in the pilot study makes direct comparisons difficult, it was hypothesized that the larger study, with the added turn would show a larger gap in performance between better and worse controllers. This was somewhat supported by the results in the main study.

As has previously been stated, the goal of the work in this thesis is not to develop the best controller possible but to understand the relative performance of each. Due to this, the tuning of the fuzzy controllers under test was not considered important, as each was derived in the same manner. Due to the similarity of the RMSE values obtained, the effect of additional tuning of each of the fuzzy systems may have been a useful area of study. However, this would have been a considerable investment in time and will be left as an avenue for future work.

Overall, this chapter shows that more difficult and variable real-world experiments cause more variable results to be observed. The hypothesis made in Section 6.2.4 is supported by the results in that larger FOU sizes seem to give better performance than those with smaller FOU sizes. However, the results obtained still not fully support this hypothesis, as the differences observed, while larger than those previously found, still do not achieve the magnitude of changes that were anticipated. This has lead to considering other factors for the reasons for this lack of differences. The main considerations are (i) the original type-1 fuzzy sets may have been far from optimal, making it hard to improve upon performance; and (ii) the method by which the type-2 fuzzy logic controllers were derived from the type-1 may have been too simple to ever give good results. Both of these considerations are out of the scope of this thesis but present good opportunities for future work.

## 6.6 Summary

In this chapter, the same fuzzy logic controllers in the previous chapter were applied to a real-world autonomous sailing boat context, as opposed to simulation. The results obtained show some additional differences than those observed in the previous chapter. The reasons for this are discussed along with reasoning as to why these differences have been found, possible solutions and avenues of future work are then identified.

In the next chapter an in-depth discussion about the work and findings of this thesis is presented. In addition, the ideas for future work and improvements to address some shortcomings are presented.

# 7

# Discussion

## 7.1 Introduction

In this thesis, the topic of fuzzy logic and specifically, how its behaviour changes across different scenarios was investigated using three case studies: the tipping problem and autonomous sailing robots in simulation and real-world environments. The effects of variation in the environment, how it can affect performance and how it can be introduced into the environment was studied. Multiple varieties of fuzzy logic control were investigated, with comparisons between them being the major focus of study. Each variety was evaluated using several different internal configurations, generally determined by the FOU size.

The main motivation behind this work was to be able to identify which factors are likely to cause type-2, dual surface and non-stationary fuzzy logic types to outperform type-1 and the relative import of such factors, with most focus applied to interval type-2 control. This was intended to act as a starting point for being able to develop techniques for the selection and justification of the type of fuzzy logic control for a given application. Task difficulty in the context of the sailing robot application was defined by (1) the sailing boats defined course, including how many turns and the total cumulative angle; and (2) the conditions under which the sailing occurs, including the wind, water and other sources of environmental variation.

## 7.2 Evaluation of Aims

The aims stated in Chapter 1 (shown in boldface) have been addressed as follows:

- **To show that variations on standard type-1 fuzzy logic control such as type-2, Dual Surface, and Non-stationary fuzzy logic control can provide significantly improved performance over standard type-1 fuzzy logic based control systems.** This was addressed by using the different varieties of fuzzy logic across experiments of increasing difficulty and complexity utilising the simulation environment in Chapter 5. Within each experimental set-up, many variables were kept constant, which enabled us to perform meaningful comparisons. The differences were most obviously shown in the experiments within sections 5.4 and 5.5 (pages 107, 115) where differences in performance are found throughout several different set-ups.

- **To study how performance changes as the environment is made more or less complex, by changing the degrees of environmental variation and the task difficulty defined.** While moving through the case studies, the experimental environment generally increased in complexity from the very simple Tipper experiments in Section 3.4 (Page 54) to the real-world experiments in Section 6.4 (Page 150). In addition, within the simulated sailing study the environment was studied with several combinations of task difficulty and environmental set-up.

- **To investigate how the internal configuration of a given controller (referred to as the FOU size) changes the level of performance of type-2, DS and NS based fuzzy systems in comparison with the more standard type-1 based configuration.** This was achieved by gradually increasing the range of the FOU sizes used in each case study. The Tipper experiments in Section 3.3 (Page 50) used a narrow range of values (sizes 1 to 4) of FOU, which is increased to a range of 10 to 40 in the real-world experiments. It was anticipated due to the greater variation present in the real world.

- **To determine the combination of factors (FOU Size, environmental variation and task difficulty) with which type-2 fuzzy logic would consistently outperform type-1 based control.** Each of the case studies in the thesis utilised different combinations of the above factors in an attempt to address the stated aim. For example, in the experiments in Section 5.5 (Page 115) nine different configurations of wind variables (defining the environmental variation), several courses (defining task difficulty) and multiple FOU values for each fuzzy variety are used with the results, giving an indication of where peaks and troughs in performance lie.

## 7.3 Contributions and Findings

In addressing the aims discussed above, the following contributions have been made:

- A methodology and supporting framework that enabled effective comparison of fuzzy logic controllers of multiple different varieties has been developed. It takes into account the following factors that may alter behaviour:

  1. Task difficulty, made up of both the direct task difficulty (such as the course layout in the sailing boat simulation) and the environment in which the task is performed. Specifically the variation present in the environment, such as the changes in wind speed and direction for the sailing boat experiments were employed.
  2. Fuzzy controller configuration. The FOU size of the type-2 and dual surface controllers was varied in an attempt to observe how this altered the behaviour of the system — specifically, how the RMSE value was obtained. A similar value, the standard deviation of the random number generator, was used in the case of the non-stationary controllers.
  3. Means of comparison. The determination of which rules fire, shapes of control surface and calculated RMSE values were investigated to observe which factor or combination of factors give the most effective means of comparison between controllers.

- The application of non-stationary fuzzy control to robotic control problems. To the best of our knowledge, the current literature does not contain any similar work. The use of non-stationary provides a stepping point in complexity between standard type-2 and the more sophisticated (and therefore expensive) interval (or indeed general) type-2 control.

- The application of the same fuzzy controller to both simulated and real-world environments has not been extensively examined within the literature. Performing this sort of comparison allows the ability to examine how two very different environments can alter performance while keeping as many factors constant as possible.

The methodology described in point 1 above has been applied to multiple case studies and has resulted in being able to describe the following findings:

1. Control surfaces inspection can highlight some potential differences, such as the increased smoothing discussed in detail in Section 5.3.1 (Page 104).
2. The use of the rule fire comparison method for performance evaluation has shown to be the least effective means of comparison of those used in this thesis. However it has still highlighted some issues with the design of rule bases for fuzzy systems, which are discussed in further detail in Section 3.3.5 (Page 53).
3. Each of the more sophisticated types of fuzzy control have been shown to be capable of producing better (lower RMSE) values than type-1 under certain conditions:
    - Interval Type-2 outperforms type-1 less frequently than it was anticipated. In the large simulation experiment in Section 5.5.6 (Page 131, this occurs in approximately 8% of the the experiments.
    - Non-stationary control generally presents improvements at lower levels of complexity (low levels of environmental variability and task difficulty) than Interval type-2. This occurs in approximately 5% of cases in the same study as above.
    - Dual surface fuzzy control improvements occur the least frequently of the three types studied and it is more difficult to predict when this may occur. The reasoning for this is discussed by Birkin in [13].

These findings can be generalised to suggest that overall the more sophisticated fuzzy controllers can only show improvements in performance if the environment of the experiment is of a suitable level of complexity. In the most simple of situations, such as the Tipper in Chapter 3, very few differences were observed as expected — the task was so simple any controller was able to complete it without difficulty. As more complex case studies were introduced, the differences became more obvious. It is hypothesised that if even more complex case studies were used, there would be an upper bound where no controller would be able to complete the task, resulting in a drop in performance.

A possible reason for the the more sophisticated controllers only showing a small improvement over type-1 is the number of parameters available in the more sophisticated varieties. An example of this would be when moving from a type-1 based design to interval type-2 based design. The system designer must, in addition to the type-1 parameters, define a membership function for each fuzzy set. This gives flexibility, but requires further effort to select the appropriate values. Therefore, with increasing numbers of parameters to define, without using a systematic method (such as a genetic algorithm), the chance of selecting a set of parameters which result in improved performance decreases.

## 7.4 Shortcomings and Limitations

There were shortcomings in the work performed, which are discussed below per case study:

- Tipper case study: While a useful introductory work used to successfully validate the methodology, overall the experimental set-up was found to be too simple to show significant differences between the different controllers under test.

- Simulated sailing case study: The Tracksail simulator is very simple in nature with a minimal model for the simulation of wind and water dynamics. In some ways the simplicity helped, allowing considerable control of the experiment. However, it also limited the utility and realism of the simulator.

- Real-world case study: The amount of data collected was small, due to hardware issues and the inherent difficulty in working in real-world situations. This has led to difficulties in drawing firm conclusions from this case study.

Aside from the shortcomings in the case studies, more general limitations of the thesis as a whole have been identified as follows:

- The comparison of RMSE values, while shown to be effective, is limited in its ability to compare between different experimental set-ups, in which aspects such as the variation present changes. For instance, a given RMSE in experiment 'A' may represent a well-performing controller, while in experiment 'B' may be significantly worse.

- The update rate of the controller was very slow for such a control system. The rate selected was due to hardware constraints of the sailing robot and the desire to maintain the same update rate across the different environments (simulation and real world). It is believed that this adversely affected the RMSE value. With a faster update rate the differences between better and worse performing controllers would have become more apparent. This is because in a given time (e.g. a 30 second window) an update rate of 1Hz would allow 30 changes of rudder position while a faster update rate of 10Hz would allow 300 changes — giving better controllers more scope to respond to changes in conditions.

## 7.5 Future Work

In this thesis, four main controller types were studied (type-1, interval type-2, type-1 non-stationary and dual surface). However this is not an exhaustive list of types of fuzzy controllers described in the literature. There are several others which present interesting avenues for future work, including:

- Non-Singleton, which modifies the standard method for fuzzification to use a shape such as triangle instead of a single line to determine the membership. It has been shown by Mouzouris and Mendel [72] that this type of fuzzy logic can minimise the effect of noise and therefore lead to increased performance with minimal changes to existing control systems.

- Interval type-2 based non-stationary. The non-stationary controller used in this thesis was based on type-1 fuzzy logic and showed a small number of cases having better performance than standard type-1. The move to interval type-2 based non-stationary may further increase performance levels.

- General type-2 fuzzy logic (briefly described in Section 2.5 (Page 20)) is a very interesting avenue for future work, as it represents the most sophisticated type of fuzzy logic currently in use. However, it comes with a high processing cost and it was found to be unsuitable for the robot used in this thesis — these issues would need to be solved prior to its usage.

The case studies used in this thesis cover a large range of different levels of complexity, from the Tipping example to the real-world sailing study. Additional case studies, such as wheeled robots or changing existing experiments using the sailing simulator to include more complex and realistic models could give the additional scope that is needed to show performance differences between the simple and more sophisticated control methods.

The comparison of rules which fire and of control surface shape as done in this thesis were not well developed. We believe that additional analysis of the results (in a suitably complex experimental setup) may lead to additional methods of comparison. For example, a mathematical analysis of the control surface including the gradient of transitions and where they occur may be able to give a predictor of performance.

All of type-2 FOUs in our work have been derived from the type-1 membership functions using a standard horizontal movement mechanism. This means of derivation is not the only way of performing such adaptations, and the effect of the method selected is currently not investigated in the literature. This is another subject which may greatly affect the performance of the fuzzy controllers to be investigated.

# References

[1] J. Abril, J. Salom, and O. Calvo. Fuzzy control of a sailboat. *International Journal of Approximate Reasoning*, (96):359–375, 1997. 37, 39

[2] M. Aliasghary, . Eksin, M. Güzelkaya, and T. Kumbasar. Design of an interval type-2 fuzzy logic controller based on conventional PI controller. In *Mediterranean Conference on Control & Automation (MED)*, 2012. 30

[3] S. M. Antoun and P. J. Mckerrow. Landmark navigation with fuzzy logic. In *Australasian Conference on Robotics and Automation, Australian Robotics & Automation Association*, 2006. 41

[4] B. H. Arnold. *Logic and Boolean Algebra*. 2011. 16

[5] K. Arnold and J. Gosling. The java programming language (2nd ed.), 1998. 87

[6] A. Bastian. Handling the nonlinearity of a fuzzy logic controller at the transition between rules. *Fuzzy Sets and Systems*, 71(3):369–387, May 1995. 105

[7] N. Benatar, U. Aickelin, and J. M. Garibaldi. A comparison of non-stationary , type-2 and dual surface fuzzy control. In *FUZZ-IEEE*, 2011. 73

[8] N. Benatar, U. Aickelin, and J. M. Garibaldi. An investigation into the relationship between type-2 FOU size and environmental uncertainty in robotic control. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, pages 1–8, June 2012. 73

[9] R. C. Berkan and S. Trubatch. *Fuzzy System Design Principles*. 1997. 45

[10] B. M. Bernadette, M. Dotoli, and B. Maione. On the choice of membership functions ina a Mamdani-type fuzzy controller. 22

[11] C. Bing-Yuan, G. Wang, S. Chen, and S. Guo. *Fuzzy Information and Engineering*. 2010. 1

[12] P. A. Birkin and J. M. Garibaldi. A novel dual-surface type-2 controller for micro robots. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8, July 2010. 91

[13] P. A. S. Birkin and J. M. Garibaldi. A comparison of type-1 and type-2 fuzzy controllers in a micro-robot context. *2009 IEEE International Conference on Fuzzy Systems*, pages 1857–1862, Aug. 2009. 28, 29, 131, 135, 161

[14] B. Bond. *The Handbook of Sailing*. 1992. 35, 93

[15] P. Bonissone, V. Badami, K. Chiang, P. Khedkar, K. Marcelle, and M. Schutten. Industrial applications of fuzzy logic at general electric. *Proceedings of the IEEE*, 83(3):450–465, Mar. 1995. 33

[16] M. Braae and D. A. Rutherford. Selection of parameters for a fuzzy logic controller. In D. Dubois, H. Prade, and R. R. Yager, editors, *Readings in Fuzzy Sets for Intelligent Systems*, pages 298–305. Kaufmann, San Mateo, CA, 1993. 1

[17] Y. Briere. Iboat: an autonomous robot for long-term offshore operation. In *MELECON*, pages 323–329, 2008. 38

[18] R. Brooks. Visual map making for a mobile robot. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 824–829, Mar 1985. 134

[19] A. B. Cara, I. Rojas, H. Pomares, C. Wagner, and H. Hagras. On comparing non-singleton type-1 and singleton type-2 fuzzy controllers for a nonlinear servo system. In *2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, pages 126–133. IEEE, Apr. 2011. 30

[20] N. Cázarez, O. Castillo, L. Aguilar, and S. Cárdenas. From type-1 to type-2 fuzzy logic control: A stability and robustness study. *Hybrid Intelligent Systems Studies in Fuzziness and Soft Computing*, 149(2007):135–149, 2007. 12, 30

[21] A. Celikyilmaz and I. B. Türksen. *Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications*. 2009. 1

[22] M.-Y. Chen and D. Linkens. Rule-base self-generation and simplification for data-driven fuzzy models. *Fuzzy Sets and Systems*, 142(2):243–265, Mar. 2004. 50

[23] S. L. Chiu. Selecting input variables for fuzzy models. *Journal of Intelligent and Fuzzy Systems*, pages 243–256, 1996. 22

[24] S. L. Chiu. Selecting input variables for fuzzy models. *Journal of Intelligent & Fuzzy Systems*, 4:243–256, 1996. 45

[25] B. J. Choi, S. W. Kwak, and B. K. Kim. Design of a single-input fuzzy logic controller and its properties. *Fuzzy Sets and Systems*, 106(3):299–308, 1999. 26

[26] O. Cordón, F. Herrera, and A. Peregrín. Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy Sets and Systems*, 86(1):15–41, Feb. 1997. 19

[27] S. Coupland and R. I. John. Towards more efficient type-2 fuzzy logic systems. *IEEE International Conference On Fuzzy Systems*, ($\lambda$ 0):236–241, 2005. 32

[28] E. Cox. Fuzzy fundamentals. *IEEE Spectrum*, 29:58–61, 1992. 17

[29] L. Doitsidis, N. C. Tsourveloudis, and S. Piperidis. Evolution of fuzzy controllers for robotic vehicles: The role of fitness function selection. *Journal of Intelligent and Robotic Systems*, 56(4):469–484, Apr. 2009. 12

[30] H. Enderton. *A Mathematical Introduction to Logic*. 2001. 16

[31] U. Farooq, J. Gu, and M. Seto. Design and comparison of Takagi-Sugeno type-1 and interval type-2 fuzzy logic controllers for posture stabilization of wheeled mobile robot. In *International Conference of Robotics and Biomimetics (ROBIO), 2013 IEEE*, pages 2309–2316, 2013. 12

[32] G. Feng, S. Cao, N. Rees, and C. Chak. Design of fuzzy control systems with guaranteed stability. *Fuzzy Sets and Systems*, 85:1–10, 1997. 45

[33] FFLL. The free fuzzy logic library website. *http://ffll.sourceforge.net/*, (last acessed 23/11/2014). 87

[34] J. Figueroa, J. Posada, J. Soriano, M. Melgarejo, and S. Rojas. A type-2 fuzzy logic controller for tracking mobile objects in the context of robotic soccer games. In *Ieee International Conference On Fuzzy Systems*, volume 2, pages 359–364, 2005. 31

[35] A. Fraenkel, Y. Bar-Hillel, and A. Levy. *Foundations of Set Theory*. 1958. 12

[36] J. Garibaldi, S. Musikasuwan, and T. Ozen. The association between non-stationary and interval type-2 fuzzy sets: A case study. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pages 224–229. Ieee, 2005. 28

[37] J. M. Garibaldi and M. Jaroszewski. Generalisations of the concept of a non-stationary fuzzy set – a starting point to a formal discussion. *Ieee International Conference On Fuzzy Systems*, pages 2313–2319, June 2008. 27

[38] J. M. Garibaldi, M. Jaroszewski, and S. Musikasuwan. Nonstationary fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 16(4):1072–1086, Aug. 2008. 27

[39] S. Greenfield, F. Chiclana, S. Coupland, and R. I. John. The collapsing method of defuzzification for discretised interval type-2 fuzzy sets. *Information Sciences*, 179(13):2055–2069, June 2009. 25

[40] S. Guillaume. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Trans. on Fuzzy Systems*, 9(3):426–443, 2001. 26

[41] H. Hagras. Developing a type-2 FLC through embedded type-1 FLCs. In *FUZZ-IEEE 2008*, June 2008. 33, 135

[42] H. Hellendoorn and C. Thomas. Defuzzification in fuzzy controllers. *Journal of Intelligent and Fuzzy Systems*, pages 109–123, 1993. 19

[43] T.-P. Hong and C.-Y. Lee. Induction of fuzzy rules and membership functions from training examples. *Fuzzy Sets and Systems*, 84(1):33–47, 1996. 18

[44] R. Jager. *Fuzzy Logic in Control*. 1995. 33

[45] T. Jech. *Set theory*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2003. The third millennium edition, revised and expanded. 10, 12

[46] N. Karnik and J. M. Mendel. Centroid of a type-2 fuzzy set. *Information Sciences*, 132(1-4):195–220, Feb. 2001. 76

[47] 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. 20

[48] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic*. 1995. 16

[49] T. Kuusela and S. Brockmann. Tracksail, 2002, http://tracksail.sourceforge.net/ (last acessed 30/11/2014). 80

[50] C.-C. Lee. Fuzzy logic in control systems: fuzzy logic controller. ii. *Systems, Man and Cybernetics, IEEE Transactions on*, 20:419–435, 1990. 59

[51] W. V. Leekwijck and E. E. Kerre. Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, 108(2):159–178, 1999. 19

[52] C. Leondes. *Fuzzy Logic and Expert Systems Applications*. 1998. 10

[53] Y. Li, K. H. Ang, and G. C. Y. Chong. Pid control system analysis and design. *Control Systems, IEEE*, 26(1):32–41, 2006. 1, 88

[54] Q. Liang and J. Mendel. An introduction to type-2 TSK fuzzy logic systems. In *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International*, pages 1534–1539, 1999. 20

[55] Q. Liang and J. M. Mendel. Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy Systems*, 8(5):535–550, 2000. 32

[56] R. Liebscher. Pyfuzzy – Python fuzzy package, 2010, http://pyfuzzy.sourceforge.net/ (last acessed 10/03/2013). 87

[57] Y. Lin and I. Cunningham, G.A. A fuzzy approach to input variable identification. In *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*, pages 2031–2036, 1994. 45

[58] F. Liu and J. Mendel. Aggregation using the fuzzy weighted average as computed by the Karnik & Mendel algorithms. *IEEE Transactions on Fuzzy Systems*, 16(1):1–12, 2008. 23

[59] D. Luenberger. *Introduction to dynamic systems: theory, models, and applications*. Wiley, 1979. 34

[60] M. Luhandjula. Fuzzy optimization: Milestones and perspectives. *Fuzzy Sets and Systems*, Jan. 2014. 11

[61] J. Luo and E. Lan. Fuzzy logic controllers for aircraft flight control. In H.-J. Zimmermann, editor, *Fuzzy Logic and Intelligent Systems*, volume 3 of *International Series in Intelligent Technologies*, pages 85–124. Springer Netherlands, 1995. 33

[62] E. H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121(12):1585–1588, 1974. 59

[63] R. Martínez-Soto, O. Castillo, and J. R. Castro. Genetic algorithm optimization for type-2 non-singleton fuzzy logic controllers. *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, 547:3–18, 2014. 1

[64] Matlab. Matlab fuzzy tipper example, http://www.mathworks.co.uk/help/fuzzy/building-systems-with-fuzzy-logic-toolbox-software.html (last acessed 30/11/2014). 45

[65] A. B. McBratney and I. O. A. Odeh. Application of fuzzy sets in soil science: fuzzy logic, fuzzy measurements and fuzzy decisions. *Geoderma*, 77:85 – 113, 1997. 40

[66] M. Melgarejo. A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set. In *Annual Meeting of the North American Fuzzy Information Processing Society*, pages 190–194, 2007. 25

[67] P. Melin and O. Castillo. Type-1 fuzzy logic. In *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing*, volume 172 of *Studies in Fuzziness and Soft Computing*, pages 7–32. Springer Berlin / Heidelberg, 2005. 75

[68] J. M. Mendel. *Uncertain rule-based fuzzy logic system: introduction and new directions*. Prentice–Hall PTR, 1st edition, 2001. 12

[69] J. M. Mendel, R. I. John, and F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–821, Dec. 2006. 26

[70] G. M. Méndez, L. Leduc-lezama, R. Colas, G. Murillo-pérez, J. Ramírez-cuellar, and J. J. López. Application of interval type-2 fuzzy logic systems for control of the coiling entry temperature in a hot strip mill. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 352–359, 2009. 31

[71] O. Miglino, H. H. Lund, and S. Nolfi. Evolving mobile robots in simulated and real environments. *Artificial Life*, pages 417–434, 1995. 139

[72] G. Mouzouris and J. Mendel. Nonsingleton fuzzy logic systems: theory and application. *IEEE Transactions on Fuzzy Systems*, 5(1):56–71, May 1997. 163

[73] S. Musikasuwan, T. Ozen, and J. M. Garibaldi. An investigation into the effect of number of model parameters on performance in type-1 and type-2 fuzzy logic systems. In *Proc. 10th Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU 2004)*, pages 1593–1600, 2004. 131

[74] J. Nieminen. On the algebraic structure of fuzzy sets of type 2. *Kybernetika*, 13(4), 1977. 20

[75] K. Nozaki, H. Ishibuchi, and H. Tanaka. A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, 86(3):251–270, 1997. 18

[76] P. Phokharatkul and S. Phaiboon. Mobile robot control using type-2 fuzzy logic system. In *2004 IEEE Conference on Robotics, Automation and Mechatronics*, volume 1, pages 296–299, 2004. 32, 33

[77] A. Sadeghian, J. Mendel, and H. Tahayori. *Advances in Type-2 Fuzzy Sets and Systems: Theory and Applications*. Springer Publishing Company, Incorporated, 2013. 11

[78] A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 1(4):180–197, 1997. 33

[79] S. Sanchez-Solano, A. Cabrera, I. Baturone, F. Moreno-Velo, and M. Brox. FPGA implementation of embedded fuzzy controllers for robotic applications. *Industrial Electronics, IEEE Transactions on*, 54(4):1937–1945, 2007. 33

[80] C. Sauze. Controlling a sailing robot. In *Biologically Inspired Robots BIRO-net*, 2005. 37

[81] C. Sauze and M. Neal. An autonomous sailing robot for ocean observation. In *TAROS*, 2006. 107

[82] C. Sauze and M. Neal. Endocrine inspired modulation of artificial neural networks to aid survivability in mobile robotics. *TAROS*, (1992):3–4, 2007. 37

[83] C. Sauze and M. Neal. A biologically inspired approach to long term autonomy and survival in sailing robots. In *Proceedings of the 2008 International Robotic Sailing Conference*, pages 6–12, 2008. 37

[84] C. Schuh. Fuzzy sets and their application in medicine. In *NAFIPS 2005 - 2005 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 86–91. IEEE. 12

[85] N. H. Seng. *Implementation of various types of fuzzy controls on a mobile robot using sonar sensors*. PhD thesis, 2008. 33

[86] R. Sepúlveda, O. Castillo, P. Melin, and O. Montiel. An efficient computational method to implement type-2 fuzzy logic in control applications. *Advances in Soft Computing*, pages 45–52, 2007. 30, 132

[87] C. Shi, J. Li, and J. Peng. A new approach for ARPA display and collision danger assessment. In *Radar Symposium, 2008 International*, pages 1–4, May 2008. 33

[88] M. Shuttleworth. Ubuntu, 2011, http://www.ubuntu.com/desktop/ (last acessed 30/11/2014). 87

[89] D. Sinha and E. R. Dougherty. Fuzzification of set inclusion: Theory and applications. *Fuzzy Sets and Systems*, 55(1):15–42, Apr. 1993. 18

[90] K. T. Song and J. C. Tai. Fuzzy navigation of a mobile robot. In *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, volume 1, pages 621–627, July 1992. 34

[91] R. Stelzer and T. Proll. Autonomous sailboat navigation for short course racing. *Robotics and Autonomous Systems*, 56:604–614, 2008. 38, 87

[92] R. Stelzer, T. Pröll, and R. I. John. Fuzzy logic control system for autonomous sailboats. In *FUZZ-IEEE 2007*, pages 6–11, 2006. 38, 87, 90

[93] R. R. Stoll. *Set Theory and Logic*. Dover Publications, 1979. 13

[94] A. D. Torshizi, M. H. F. Zarandi, and H. Zakeri. On type-reduction of type-2 fuzzy sets: A review. *Applied Soft Computing*, 2014. 11

[95] K. van den Doel and D. K. Pai. Performance measures for locomotion robots. *Journal of Robotic Systems*, 14(2):135–147, Feb. 1997. 154

[96] T. W. Vaneck. Fuzzy guidence controller for an autonomous boat. *IEEE Control Systems*, (April):43–51, 1997. 37, 39

[97] C. Von Altrock. Fuzzy logic technologies in automotive engineering. In *WESCON/94. 'Idea/Microelectronics'. Conference Record*, pages 110–117, Sept. 1994. 33

[98] C. Wagner. Juzzy – a Java based toolkit for type-2 fuzzy logic. In *IEEE Symposium Series on Computational Intelligence*, pages 45–52, 2013. 87

[99] C. Wagner and H. Hagras. Evolving type-2 fuzzy logic controllers for autonomous mobile robots. In P. Melin, O. Castillo, E. Ramírez, J. Kacprzyk, and W. Pedrycz, editors, *Analysis and Design of Intelligent Systems using Soft Computing Techniques*, volume 41 of *Advances in Soft Computing*, pages 16–25. Springer Berlin / Heidelberg, 2007. 32

[100] L.-X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(6):1414–1427, 1992. 18, 50

[101] D. Wu. An interval type-2 fuzzy logic system cannot be implemented by traditional type-1 fuzzy logic systems. In *World Conference on Soft Computing*, 2011. 106

[102] D. Wu. An overview of alternative type-reduction approaches for reducing the computational cost of interval type-2 fuzzy logic controllers. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, pages 1–8, 2012. 25

[103] D. Wu and J. M. Mendel. Enhanced Karnik–Mendel algorithms. *IEEE Transactions on Fuzzy Systems*, 17(4):923–934, Aug. 2009. 23

[104] D. Wu and J. M. Mendel. Examining the continuity of type-1 and interval type-2 fuzzy logic systems. In *IEEE World Congress on Computational Intelligence*, volume 19, pages 179–192, 2010. 30, 102, 106

[105] H. Wu and J. M. Mendel. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, pages 622–639, 2002. 32

[106] A. I. E. Yakzan, A. A. Khan, E. L. Hines, and D. Iliescu. Adaptive control of an intelligent tank using fuzzy logic. In *Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on*, pages 257–262, Mar. 2012. 33

[107] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965. 1, 10

[108] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8(3):199–249, 1975. 11

[109] L. A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems*, 90(2):111–127, 1997. 1

[110] L. Zhao. Non-stationary interval type-2 fuzzy logic systems. In *Intelligent Control and Information Processing (ICICIP), 2011 2nd International Conference on*, volume 2, pages 1169–1174, July 2011. 28

[111] H.-J. Zimmermann. *Fuzzy Set Theory – and Its Applications*. 1991. 16