Technological University Dublin

ARROW@TU Dublin

Dissertations

School of Electrical and Electronic Engineering

Winter 2011-01-01

# The Application of Fuzzy Logic in Determining Linguistic Rules and Associative Membership Functions for the Control of a Manufacturing Process

Marcus Foley
*Technological University Dublin*

John McGrory
*Technological University Dublin*, john.mcgrory@tudublin.ie

Follow this and additional works at: https://arrow.tudublin.ie/engscheledis

Part of the Electrical and Computer Engineering Commons

## Recommended Citation

Foley, M.:The Application of Fuzzy Logic in Determining Linguistic Rules and Associative Membership Functions for the Control of a Manufacturing Process. Masters Dissertation. Technological University Dublin, 2011.

This Dissertation is brought to you for free and open access by the School of Electrical and Electronic Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Dissertations by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, vera.kilshaw@tudublin.ie.

# The Application of Fuzzy Logic in Determining Linguistic Rules and Associative Membership Functions for the Control of a Manufacturing Process

by

Marcus Foley

This Report is submitted in partial fulfilment of the requirements of the Master of Engineering in Pharmaceutical Process Control and Automation of the Dublin Institute of Technology

January 10th 2011

Supervisor: John McGrory

School of Electrical Engineering Systems

## ABSTRACT

Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory. Its methodology aims to provide a definitive solution from information that may be construed as ambiguous, imprecise or noisy. Classical set theory studies the properties of sets, while fuzzy set theory investigates the degree to which an element can be related to a set. The aim of this project is to develop a control strategy for a specific technical challenge relating to the food processing sector based on the deployment of fuzzy logic control concepts. Specifically, in this paper the author is concerned with the ability to control the density input of a variable feed product stream by automatically adjusting the 'thermo pressure' & 'feed flow' within desired limits. For the purpose of this study, the expert knowledge of both senior automation engineers and process operators was procured in order to develop an understanding of the dynamics and the limitations of the manufacturing process. The focus of this study is the development of a fuzzy logic control system for the production of "*Whey Permeate Concentrate*" in the production facilities of Glanbia plc. in Ballyragget, County Kilkenny.

# DECLARATION

I certify that this thesis, which I submit in partial fulfilment of the requirements of the ME Pharmaceutical Process Control and Automation (Programme Ref: DT702/3) of the Dublin Institute of Technology, is entirely my own work and that any content that relates to the work of other individuals, published or otherwise, are acknowledged through appropriate referencing.

I also confirm that this work has not been submitted for assessment in whole or part for an award in any other Institute or University.

Signed: ___________________________

Date: ______________________________

# ACKNOWLEDGEMENTS

I would like to thank all individuals who provided me with the valuable assistance, insight and time required to complete this study.

I am extremely grateful for the support of Glanbia plc, namely Mathew O'Donnell (senior control engineer), who has facilitated me with access to the production plant and all relevant process data, which aided my study in the development a fuzzy control system for the production of whey permeate concentrate (WPC). I gratefully appreciate the opportunity to have worked on such a fascinating subject matter within the field of Engineering, specifically in the branch of process control and automation.

I would also like to thank my supervisor John McGrory and all the lecturers and staff in Dublin Institute of Technology who have provided me with both an excellent education and a quality and friendly environment in which to learn and challenge.

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF APPENDICES

**TABLE OF CONTENTS**

# CHAPTER 1 INTRODUCTION

## 1.1 Introduction

Manufacturing may be defined as the use of machines and labour to produce a product from raw material(s). Traditionally, manufacturing processes were highly dependent on human management and intervention in order to ensure that the product was produced in a safe, efficient and timely manner. As technology evolved, so did the methods for controlling manufaturing processes. Advancements in technology and micorprocessor based control equipment, and associated software systems, have resulted in a massive reduction in human control dependency, leadings to major increases in manufacturing efficiencies relating to volume, yield, cost, energy consumption, waste reduction and time. Globalisation and competition in the marketplace continue to drive an ever increasing demand for low cost, highly efficient production systems and therefore more sophisticated and more robust production planning and process control systems are becoming more and more important for manufacturers.

Process control theory is the branch of engineering and mathematics that is concerned with understanding the dynamic behaviour of complex systems so that alogorithms can be developed to control the response of the system to changes in the system inputs. By observing how the output from a system reacts to a change in input(s), process engineers can design algorithms and control responses to ensure that the output(s) of the system are managed and controlled in a manner which maximises the performance of the process. . The dynamics of any system will be unique to the system and will vary depending on the process in question. Thus, a study of how the process behaves must be performed in order to determine the best control strategy for the process. For the purpose of this study, the expert knowledge of both senior automation engineers and process operators was procured in order to develop an understanding of the dynamics and the limitations of the manufacturing process. The focus of this study is the development of a fuzzy logic control system for the production of "*Whey Permeate Concentrate*" (WPC) in the production facilities of Glanbia plc. in Ballyragget, County Kilkenny. This plant is the largest integrated milk processing facility in Western Europe.

### 1.2 Significance of Study

Wang suggests that "*the desire to make controllers more autonomous and intelligent*" (1) has led to the introduction of control systems that incorporate many of the approaches which underpin artificial intelligence based systems, namely fuzzy control which is supported by fuzzy theory. Fuzzy theory is designed to deal with ambiguous concepts that lack crisp definition, for example;'*water is too cold make it hotter*', and it applies logical reasoning as a means to correct the problem e.g. "*turn on the hot tap until an appropriate temperature is reached*". Fuzzy control theory is designed to replicate human reasoning, thinking and response mechanisms. It is intended to mirror the behaviour of operators or experts to perform effective and timely control over a process.

The study of this topic was chosen as it is based on an existing manufacturing process and the particular challenge presented by the need to control whey permeate concentrate in the dairy manufacturing facility. This particular process has an intrinsically long process delay time of approximately 9 or 10 minutes. For this process, a change in process input will take up to 10 minutes to manifest a change in the process output, due to the fact that the process involves 4 stages and a long residence time. In a less complex, fast response type system, a change in the measured variable would result in a feedback controller making an immediate adjustment to the system's control parameter(s) in order to mitigate the change, and restore the measured variable to the desired level i.e. the setpoint.

However, in the case which forms the basis for this study, a change in the desired system output is in fact a function of change(s) which will have occurred in the process up to a maximum of 10 minutes prior. For example, if material(s) is fed into a multi-step process and a critical control property (e.g. density) changes through the process steps and the sensor is located at the end of the process stages, then the change will not be detected until the material has completed its passage through all of the stages. Therefore in such a case, a simple feedback control strategy will not be viable or robust.

The literature relating to fuzzy logic control in manufacturing industry is sparse on applications and solutions for complex processes, with intrinsic delays, such as the case under consideration in this thesis. More research has been conducted in the area of continuous and discrete dynamic systems using less complex feedback control systems. Karimi and Jahanmiri (2), investigated the control of a process output for multi-effect falling film evaporators using cascade control. However, the process that Karimi and Jahanmiri were dealing with had a relatively small process delay time (100 seconds) in comparison to the case described in this study.

In short, research in this area is still ongoing; Perrot; Ioannou; Allais; Curt; Hossenlopp and Trystram (3) investigate the vast field of study related to fuzzy logic and the different tools that have been developed over a decade up to the year 2003. In short, Perrot et al, discuss the need for advanced "*quality control*" in the food industry and that applications involving fuzzy control "*are still limited and few reviews on this topic are readily available*" (3). **Figure 1.1** below, illustrates the few articles already published in the field of fuzzy control for the food processing industry.

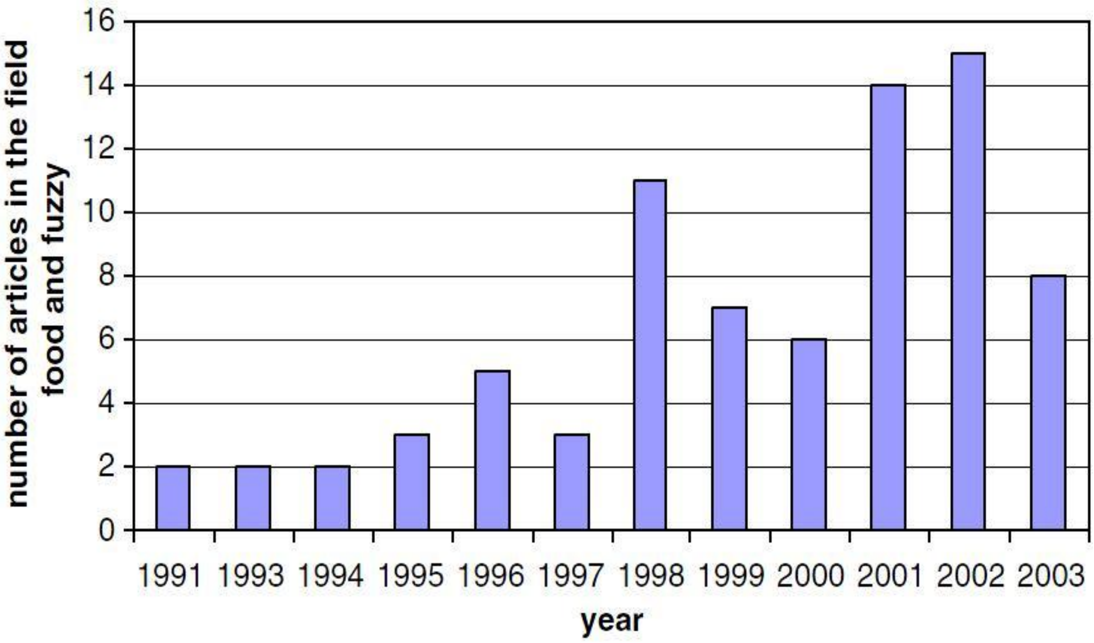


**Figure 1.1:** Number of articles published in the field crossing the fuzzy logic and the control of the food processes **(3)**.

### 1.3 Aim of study

It is the aim of this study, through the research of existing literature and experimentation related to the field of fuzzy logic and fuzzy control systems, to develop a *'closed loop'* control system that is based on the control methodology of fuzzy logic to achieve a target density output for a highly variable density input by controlling feed flow and thermo-pressure in the system. At present, the manufacturing process for whey permeate concentrate is heavily reliant on operator (manual) control. The control in this particular production line is an '*open loop*' system by which an operator will react to a change in output density, and make an adjustment to the system's feed flow and thermo-pressure accordingly.

Corrective response, in this case, is dependent entirely on an operator's reaction to a deviation in the measured output. Any corrective measure is predicated on the experience of the specific operator and observations have confirmed that there is significant variation in response between different operators. Therefore, the control system proposed in this study will implement a logic based control system that mimics the observed response of a range of operators to changes in the measured output of the system. The aim is to control the density output of the product to with +/- 2 $kg/m^3$ of the setpoint, 1230 $kg/m^3$.

### 1.4 Summary

The focus of this study is the development of a fuzzy logic control system for the production of WPC in the production facilities of Glanbia plc. in Ballyragget, County Kilkenny. This study was chosen as it is based on an existing manufacturing process issue. In the next chapter, literature detailing classical sets; fuzzy sets ; fuzzy systems and models used in fuzzy systems is reviewed.

# CHAPTER 2 LITERATURE REVIEW

## 2.1 Introduction

Huub H.C. Bakker, Clive Marsh, Shabeshe Paramalingam and Hong Chen (4) discuss the issues regarding "*tight control*" for product concentration in multi-effect falling-film evaporators. They state that achieving quality control "*in evaporators is difficult due to disturbances, large time delays and other plant constraints*". Furthermore, Huub H.C. Bakker et al (4) suggest that "*the use of a single feedback PI (Proportional Integral) control is not sufficient for this application*". Experience has shown that PI control has a limited disturbance rejection bandwidth (4). Therefore, an alternative control operation method must be investigated in order to obtain consistent performance in these styles of evaporators.

Fuzzy control operates on a continuous value basis (between 0 and 1) by means of converting a linguistic control strategy, based on human expert knowledge, into an automatic control strategy (5). Fuzzy logic is used by the controller to apply reasoning to an error and attempts to rectify it through a rule based algorithm. Rules are often formatted using 'if-then' statements to perform corrective action based on the *'measured input error'* or *'change in error*' of a system (6).

Fuzzy systems, more often than not, are found to be nonlinear and so the concept of stabilising a system like this is more difficult. Jantzen states that it is possible "to approximate a fuzzy controller with a linear controller and then apply the conventional linear analysis and design procedures on the approximation" (6). The theory based on this control system centres around the following control rules:

a) Expert experience and control engineering knowledge – this approach ascertains a collection of rules based on carefully organised answers from experts and operators in the field (6).

b) Based on the operator's control actions – fuzzy 'if-then' rules are established from an operator's control action or learned from previous issues logged (6).

c) Based on a fuzzy model of the process – linguistic rules are considered to be the inverse model of a controlled process. Instead of using numerical values to represent control actions for a disturbance, logical statements are preferred. Unfortunately, this method can only be applied to low order systems assuming that fuzzy models of the open and closed systems are available (6).

d) Based on learning – the controller will determine the rules itself based on a neural network of information (6).

## 2.2 Classical sets

Fuzzy logic is a concept that is founded on classical, or crisp set theory. Crisp set theory defines a universe, say Z, in which a collection of objects, also known as elements, exist within this universe. Often these elements share a similarity that allows them to be grouped together for simplicity or convenience. For example, in a universe whose characteristic elements are whole numbers ranging from one to ten inclusive; there exists a set titled '*prime numbers*'. Therefore the elements that are unique to this set are two, three, five and seven. Every other number would fall outside the boundary of this set. This is classified as a classical or crisp set, as we know '*without-a-doubt*' that these four numbers belong to this set. However, a fuzzy set will contain elements whose membership will evoke a certain level of ambiguity.

In other words, they are sets with indistinct boundaries or '*fuzzy*' boundaries. By all rights, fuzzy logic is an extension of classical, or crisp, set theory and so it would be pertinent to briefly review the rules and notation that exemplify crisp set theory. Classical set theory studies the properties of sets. Its methods span over vast fields in mathematics and are applied to a variety of applications (e.g. fuzzy logic). Set theory uses a language based on a single fundamental relation called membership, denoted '∈'. If an element 'x' is a member of set 'A' then the relationship can be expressed as follows:

$$x \in A$$

In classical set theory the membership of elements in a set is assessed in binary terms (1 or 0) according to a *bivalent condition*[1] (7).

In addition, both sets and elements are located in a region known as a universe. The total number of elements in a universe Z is denoted by the symbol '$n_z$', and is referred to as the cardinal number (7). Discrete universes that contain a countable finite number of elements will have a finite cardinal number. It follows, that a continuous universe that comprises an infinite collection of elements will have an infinite cardinality (7).

A set can be defined as a collection of elements within a universe that are related by a common characterisation that can be attributed to that set specifically. A subset is classified as a collection of elements within a set. Sets and subsets are terms that can be used synonymously, since any subset of a set is also a member of the universal set Z. A whole set describes the collection of all possible sets within the universe (7). For a crisp set A that contains a range of elements in a universe X, the following notation is used to define the relative membership:

**Table 2.1: Crisp set notations and definitions (7).**

| Notation | Definition |
|---|---|
| $x \in Z$ | x is a member of Universe Z |
| $x \in A$ | x is a member of set A |
| $x \notin A$ | x does not belong to set A |

A characteristic function also known as a membership function, $\mu_A$ (x), is defined as an element in the universe Z having a crisp value of either 1 or 0. This is the premise of the bivalent condition, mentioned previously (8). By way of explanation, an element x that belongs to set A will have a crisp value of 1. However, an element x that does not belong to set A will have a crisp value of 0. For every element in the universe Z, each can be defined in accordance with their degree of membership relative to a set A as follows (8):

[1] Is a term used in classical logic to express if a proposition is either *true* (1) or *false* (0).

$$\mu_A(x) = \begin{cases} 1 & for\ x \in A \\ 0 & for\ x \notin A \end{cases}$$

The null set, denoted Ø, is a unique set that contains no elements within its boundary. In contrast, the whole set is considered comparable to a certain event whereas the null set is comparable to an impossible event (7).

### 2.2.1 Classical Set Operations

Having discussed the important aspects of classical set notation that will later be applied to fuzzy logic, it would also be significant to consider how sets can also operate in relation to one another. The following example takes two sets, A and B, which exist in a Universe Z, as seen in **Figure 2.1** below.

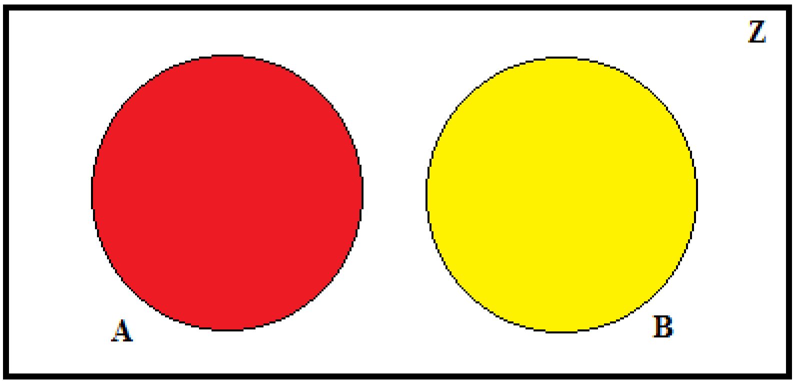


**Figure 2.1: Set A and set B in universe Z.**

**Example: Figure 2.2** below, illustrates the union between the two sets, often denoted in classical set theory as A ∪ B. The union of two sets, such as the Venn diagram depicted below, represents all elements in the universe Z that reside in either set A, set B or both sets A and B.

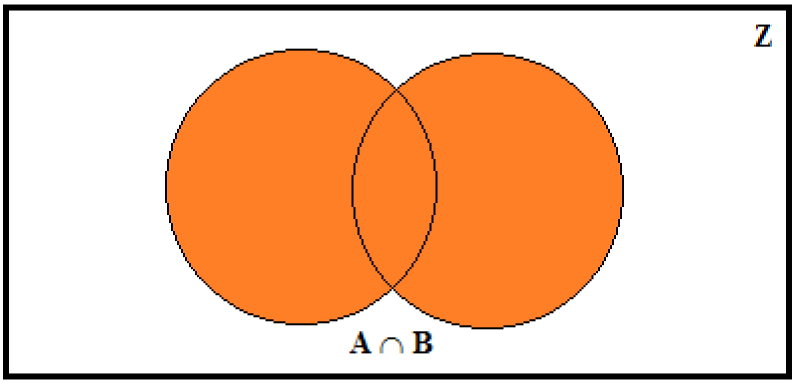


**Figure 2.2: Union of set A and set B in universe Z.**

**Figure 2.3** below, portrays the intersection of the two sets A and B, often denoted A ∩ B in classical set theory. The intersection principle represents all elements in the universe Z that belong to both sets A and B. In other words, the elements of the intersection must simultaneously belong to both set A and set B, like an overlap.

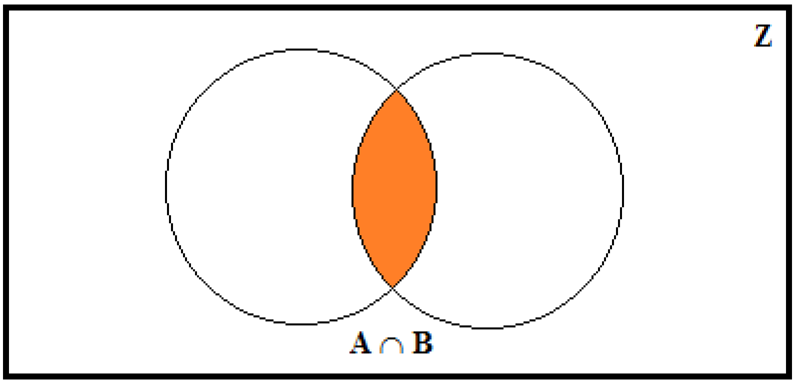


**Figure 2.3: Intersection of set A and set B in universe Z.**

The complement of a set is defined as the collection of all elements in the universe that do not reside in that set. In other words, everything outside that set as portrayed by the grey shading in **Figure 2.4** below. The complement of set A below is often denoted in classical set theory as $\bar{A}$.

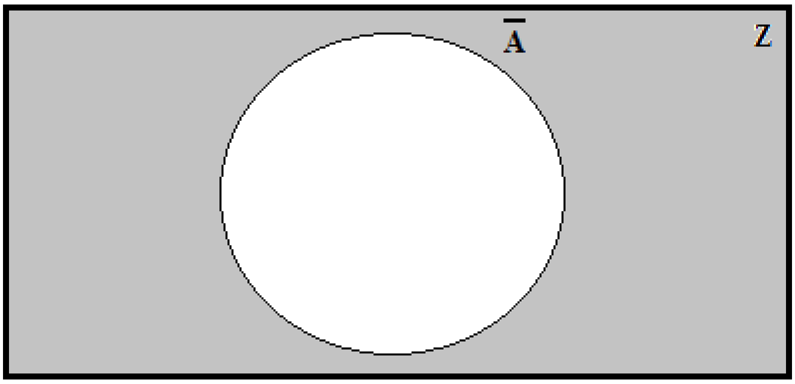


**Figure 2.4: Complement of set A in universe Z.**

The three operations described above can be written as follows (7):

- *Union* $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
- *Intersection* $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$
- *Complement* $\overline{A} = \{x \mid x \notin A, \ x \in Z\}$

### 2.2.2 Properties of classical sets

There exist certain properties of sets that occupy a role of great importance due to their influence on the mathematical manipulation of sets. In effect, these properties define the classical set and provide the fundamental stepping stones for the development of fuzzy logic rules. Some of these properties operate as follows (7):

- *Commutativity* $A \cup B = B \cup A$
- *Associativity* $A \cup (B \cup C) = (A \cup B) \cup C$

  $A \cap (B \cap C) = (A \cap B) \cap C$
- *Distributivity* $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

- *Idempotency* $A \cup A = A$

  $A \cap A = A$

- *Identity* $A \cup \phi = A$

  $A \cap Z = A$

  $A \cap \phi = \phi$

  $A \cup Z = Z$

- *Transitivity* If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$
- *Involution* $\bar{\bar{A}} = A$

The relevance of these mathematical operators with regards to fuzzy logic are further discussed at a later stage in the next section.

### 2.3 Fuzzy Sets

The notion of '*fuzzy sets*' was first introduced by Lofti A. Zadeh. They were derived from the concept of classical set theory. Fuzzy sets can be considered as an extension of crisp sets. As discussed in the previous section, objects within a set are referred to as members or elements of a set. For a fuzzy set A, the function $\mu_A$ represents the membership function for which $\mu_A$ (x) measures the degree to which an absolute value x, of the universal set Z, belongs to set A (8). For a classical set, the membership function follows conventional Boolean logic in that an element either does or does not belong to a set. Therefore, its membership value will either be 1 (true) or 0 (false).

However for a fuzzy set, the membership function can take a value in the interval ranging between 0 and 1. This interval is referred to as the membership grade or the degree of membership. A fuzzy set A can be represented as follows (8):

$$A = \left\{ \left( x, \mu_A(\mathrm{x}) \right) \middle| \, x \in A,\ \mu_A(\mathrm{x}) \in [0,1] \right\}$$

The degree of membership can also be described as a degree of measurability of which '*x*' is described by set A. The process of deriving the measurability values for a given value of '*x*' is known as *fuzzification,* which is discussed in **section 2.4**. Fuzzy sets can be categorised as either continuous or discrete. If a discrete set A, has a member '*x*', to which there is a relative membership '$\mu$' then '*x*' is a member of the set to degree '$\mu$' and can be represented as μ/x (7). Discrete sets can be written as follows:

$$A = \frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \ldots\ldots\ldots + \frac{\mu_n}{x_n}$$

Or

$$A = \sum_{i=1,n} \frac{\mu_i}{x_i}$$

Where $x_1$, $x_2$ ... $x_n$, are members of discrete set A and $\mu_1$, $\mu_2$ ... $\mu_n$, are the degrees of membership. However, a continuous fuzzy set is infinite and can be expressed as follows (7):

$$A = \int_X \frac{\mu(x)}{x}$$

The following terms are used to describe various features that relate to membership functions:

- **Core:** the core of a membership function for a fuzzy set A can be defined as the region of the universe that is characterised by complete and full membership in the fuzzy set A (7). In other words, the core comprises of those elements '*x*' within the universe; such that $\mu_A(x) = 1$.

- **Support:** the support of a membership function for a fuzzy set A can be defined as the region of the universe that is characterised by all non-zero memberships in the fuzzy set A (7). In other words, the support comprises of those elements '*x*' within the universe; such that $\mu_A(x) > 0$.

- **Boundaries:** the boundaries of a membership function for a fuzzy set A are defined as a region of the universe containing elements that have a non-zero membership but not complete membership (7). In other words, the boundaries comprise of those elements '*x*' within the universe; such that $0 < \mu_A(x) < 1$. The elements that fall within this classification are elements with some degree of fuzziness, or partial membership within the fuzzy set A.

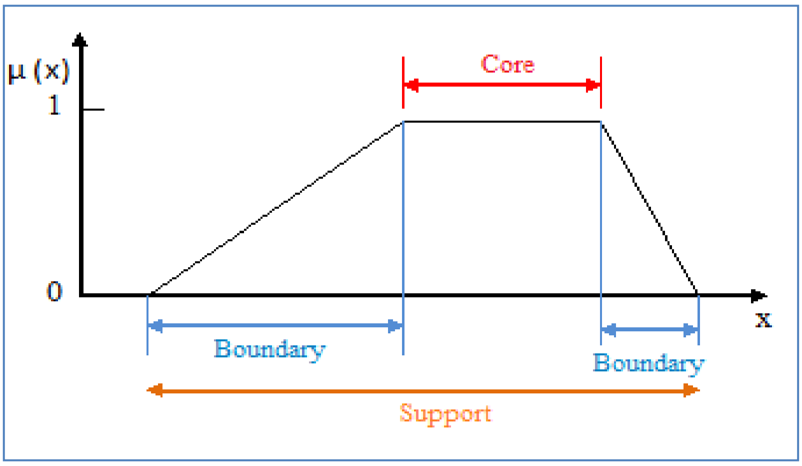


**Figure 2.5: Membership features of a fuzzy set (7).**

- **Normal:** a normal fuzzy set is one whose membership function contains at least one element '*x*' whose membership value is equal to one. In a fuzzy set where there is only one element that has a membership of one, it is typically referred to as the prototype of the set, or the prototypical element (7).

- **Sub-normal:** a subnormal fuzzy set is one whose membership function contains no element 'x' and whose membership value is equal to one.

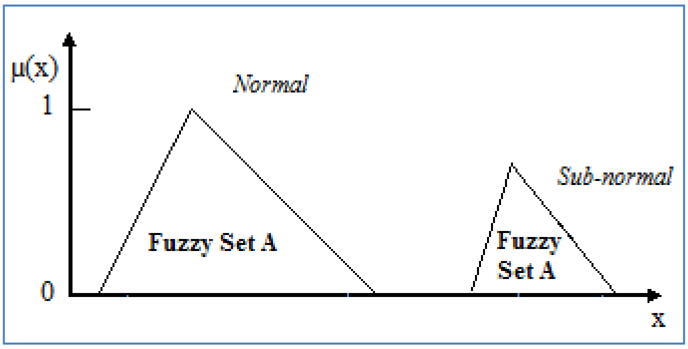


**Figure 2.6: Fuzzy set Normal & Sub-normal (7).**

- **Convex:** A convex normal fuzzy set is defined by its membership function whose values increase or decrease monotonically, or increase then decrease while the elements in the universal set increase only. In other words, a convex set contains elements x, y, and z; where the relation $x < y < z$ implies that $\mu_A(y) \geq \min [\mu_A(x), \mu_A(z)]$ (9).

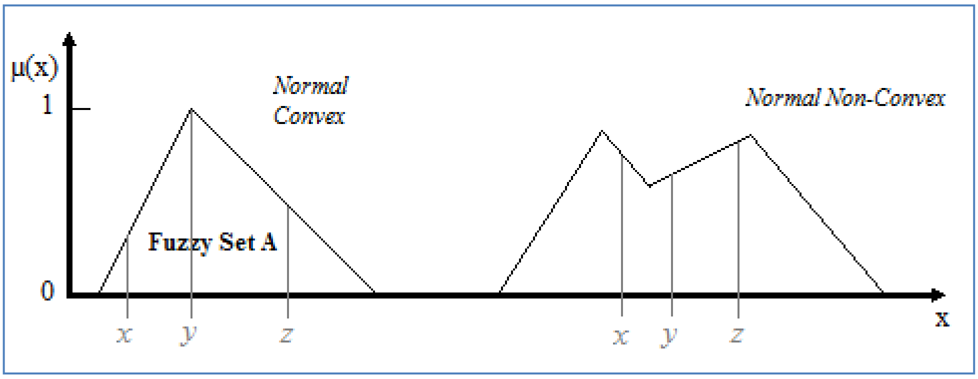


**Figure 2.7: Convex & non-convex normal fuzzy sets (9).**

One of the primary issues with regards to developing a fuzzy set is determining the associative fuzzy membership function. The membership function provides a measure of the degree of similarity of an element to a fuzzy set. Membership functions can be chosen in one of two ways:

1. **Userdefined:** the membership functions are chosen arbitrarily based on the users experience, but this is often quite subjective and can be very time consuming.

2. **Learned:** an adaptive learning system can be designed to automatically choose the most accurate parameters by observing the relationship between a series of input/output data that has been collected using artificial neural networks (this will be discussed further in **section 2.6**).

### 2.3.1 Types of membership functions

There are different shapes of membership functions, such as triangular, Gaussian, bell-shaped etc. Membership functions can have a variety of different forms to describe the same function; however the membership functions used in this study will be standardised throughout. The simplest membership functions are those who are formed using straight lines.

**Triangular Membership Functions**

One of the most basic piecewise linear function is the triangular membership function. **Figure 2.8** below, illustrates the membership function where a, b and c represent the x coordinates of the three vertices of $\mu_A(x)$ in the fuzzy set A. The coordinate 'a' is defined as the lower boundary in set A whose degree of membership is zero. The coordinate 'c' is defined as the upper boundary whose degree of membership is also zero. Finally, coordinate 'b' is the third apex of the triangle whose degree of membership is one (7).

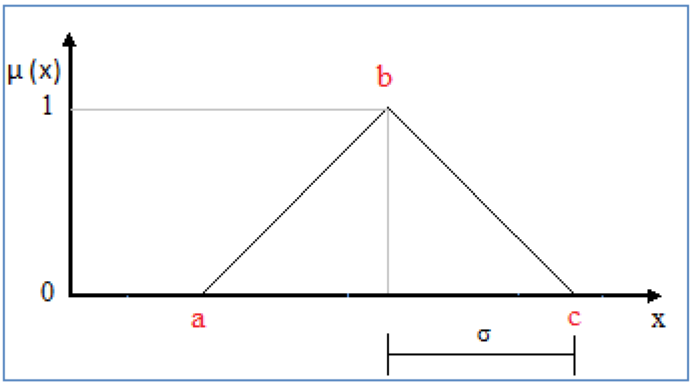


**Figure 2.8: Triangular Membership Function (10)**

The following equation represents the mathematical formula used to calculate the degree of membership for any element 'x' in a fuzzy set A (11):

$$\mu\ (x) = \begin{cases} 0, & if\ x\ \leq a \\ 1 + \dfrac{x - b}{0.5\sigma}, & if\ x\ \leq b \\ 1 + \dfrac{b - x}{0.5\sigma}, & if\ b\ \leq x \leq c \\ 0, & if\ x\ \geq c \end{cases}$$

### Gaussian Membership Function

Another fuzzy membership function that is used in fuzzy logic is the Gaussian membership function, which is represented according to the following equation (7):

$$\mu\ (x) = \left[\frac{1}{2}\left(\frac{x-b}{\sigma}\right)^{2}\right]$$

Where x is the input variable, b is the centre of the membership function and σ is the constant that represents the width of the membership function. Gaussian fuzzy membership functions are quite common with regards to fuzzy logic systems. **Figure 2.9** below, illustrates a typical Gaussian membership function.

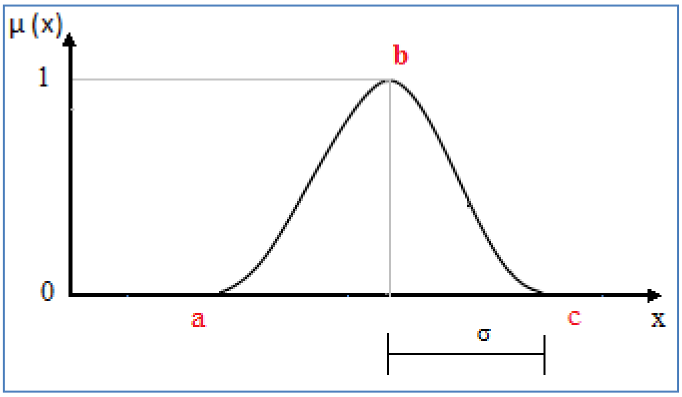


**Figure 2.9: Gaussian Membership Function (10).**

### 2.3.2 Fuzzy Set Operations

Fuzzy set operations are derived from classical set theory, as discussed in the previous section. Over the past forty five years, since Dr. Lofti Zadeh first introduced the notion of fuzzy logic in his seminal paper, fuzzy logic has developed a well-established theoretical base. However, for practical implementations there are a reasonably small amount of operations required to develop a fuzzy system. This section investigates the necessary operations required to successfully implement a computer system that utilises fuzzy logic. Furthermore, this provides the foundation for the implementation of the design model developed in this study and discussed in **Chapter 3**.

Three particularly important operations that are frequently utilised in a fuzzy logic system are the union, intersection and complement. Specifically, the union and intersection operators are often described as the fundamental building blocks that compute the fuzzy if-then rules. Both these operators operate in a similar fashion as Boolean logic to perform a calculation. The union operator uses the Boolean term '**OR**', where as the intersection operator uses the term '**AND**' when executing the fuzzy rules (7). The following example describes the fuzzy set operations for three fuzzy sets A, B and C:

**Example:** For a given element, *x*, of the universe Z the operations union, intersection and complement are defined for fuzzy sets A, B and C as follows (9):

- **Union (OR)** $\mu_{\mathrm{A} \cup \mathrm{B}}(x) = \mu_{\mathrm{A}}(x) \cup \mu_{\mathrm{B}}(x)$
- **Intersection (AND)** $\mu_{\mathrm{A} \cap \mathrm{B}}(x) = \mu_{\mathrm{A}}(x) \cap \mu_{\mathrm{B}}(x)$
- **Complement (NOT)** $\mu_{\bar{\mathrm{A}}}(x) = 1 - \mu_{\mathrm{A}}(x)$

Venn diagrams for these fuzzy operations are shown below in **Figures 2.10 – 2.12**. The operations shown in these diagrams are based on a triangular membership function, as discussed in the previous section.

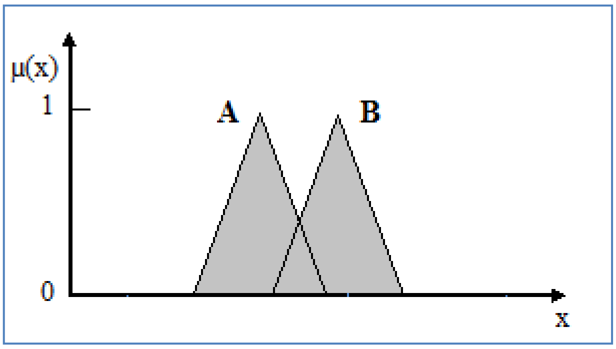


**Figure 2.10: Union (OR) of fuzzy set A and B (7).**

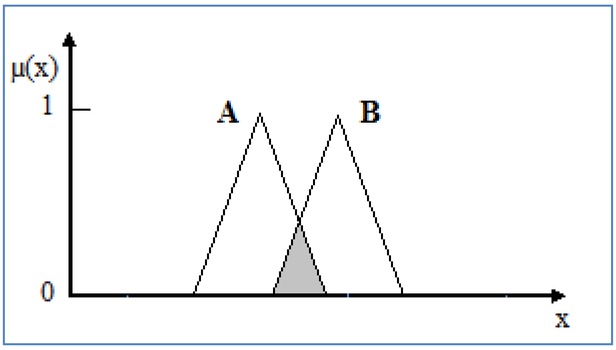


**Figure 2.11: Intersection (AND) of fuzzy set A and B (7).**

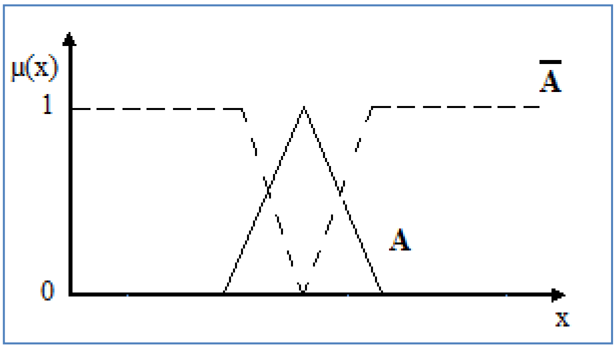


**Figure 2.12: Complement (NOT) of fuzzy set A (7).**

Any fuzzy set (A, B or C) defined in the universe Z is a subset of that universe. According to classical set theory, the membership value of any element ($x$) that exists in the null set (ϕ) is 0. Also, any element (x) that exists in the whole set Z will have a membership value of 1. The acceptable notation for these ideas can be defined as follows (7):

$$A \subseteq Z \Rightarrow \mu_A(x) \leq \mu_Z(x), \text{ For all, } x \in Z, \ \mu_\phi(x) = 0$$

$$\text{For all, } x \in Z, \ \mu_Z(x) = 1$$

### 2.3.2.1 Fuzzy Intersection

In fuzzy logic, the intersection (AND) is calculated using t-norms. A t-norm operator is a form of binary operation used in the multi-valued logic. The term t-norm is an abbreviation for triangular norm, which is used to generalise triangle inequality of ordinary metric spaces. In other words, a t-norm is a function of the type (7):

$$T: [0, 1] \times [0, 1] \rightarrow [0, 1]$$

Where the following conditions are satisfied (7):

- **Commutativity:** $t(a, b) = t(b, a)$
- **Associativity:** $t(a, t(b, c)) = t(t(a, b), c)$
- **Monotonicity:** $t(a, b) \leq t(c, d); \quad if\ a \leq c\ and\ b \leq d$
- **Identity:** the number 1 acts as an identity element so that $t(a, 1) = a$ .

The most commonly adopted t-norm in fuzzy logic is the minimum. In other words, the intersection of two fuzzy sets A and B both with respective membership functions $\mu_A(x)$ and $\mu_B(x)$ can be represented as follows (7):

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

### 2.3.2.2 Fuzzy Union

The union (OR) is calculated using t-conorms, also referred to as s-norms. A t-conorm, or s-norms, are dual to t-norms under the order-reversing operation which assigns $1 - x$ on [0, 1]. S-norms are used to represent logical disjunction in fuzzy logic and union in fuzzy set theory. Given a t-norm, the complementary conorm is defined as (7):

$$s(a,b) = 1 - s(1-a, 1-b)$$

Where the following conditions are satisfied (7):

- **Commutativity:** $(a,b) = s(b,a)$
- **Associativity:** $s\big(a, s(b,c)\big) = s(s(a,b),c)$
- **Monotonicity:** $s(a,b) \leq s(c,d); \quad if\ a \leq c\ and\ b\ \leq d$
- **Identity:** the number 0 acts as an identity element so that $s(a, 0) = a$ .

The most commonly adopted s-norm in fuzzy logic is the maximum. In other words, the union of two fuzzy sets A and B both with respective membership functions $\mu_A(x)$ and $\mu_B(x)$ can be represented as follows (7):

$$\mu_{A \cap B}(x) = \max\big(\mu_A(x), \mu_B(x)\big)$$

## 2.4 Fuzzy Systems

A fuzzy system is a control system that utilises the fundamental principles of fuzzy logic to deliver a definitive conclusion to a problem that is characterised by vague, ambiguous, imprecise, noisy, or even missing information. Systems of this nature are often referred to as fuzzy systems (FS), fuzzy knowledge based systems (FKBS) and fuzzy inference system (FIS); all of which are relatively interchangeable and amount to the same thing. Fuzzy systems use fuzzy sets and fuzzy if-then rules as a part of a computer systems' decision making process in order to draw conclusions.

According to Jantzen (6), in a fuzzy system there exist specific steps fundamental to the design procedure. The diagram below, **Figure 2.13**, illustrates the steps taken during this procedure. The steps are listed and discussed as follows:

1. Pre-processing
2. Fuzzification
3. Rule Base
4. Inference Engine
5. Defuzzification
6. Post-processing

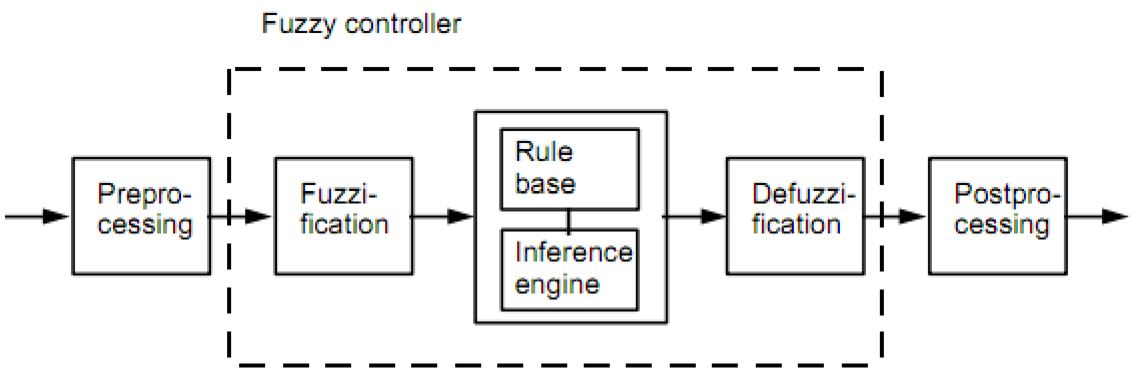


**Figure 2.13: Structure of a fuzzy controller (6).**

### 2.4.1 Pre-processing

In this step, the measured or control variable from the process (often a crisp value) becomes the controllers input. This value is conditioned in the pre-processor block before it enters the controller. In a linear system, the control variable is converted to a crisp discrete figure (6). In other words, an error value of 3.8 is rounded to 4 to fit it to the nearest discrete level. In a non-linear system, it is necessary to process the value using non-linear scaling. Taking three measurements where each value represents a small, medium and large (i.e. fuzzy sets) process condition that occurred at some point; a curve is then constructed. These measurements become the break-points on curve that scales future measurements. The pre-processor then passes the conditioned data on to the controller (6).

### 2.4.2 Fuzzification

The first block inside the fuzzy controller is fuzzification. Fuzzification uses the concepts of fuzzy set theory and specificallly fuzzy set operations, mentioned earlier. The fuzzification block is used to transform the crisp values obtained from the input signal into grades of membership for linguistic terms of fuzzy sets (6). For example, the fuzzification of a man who is six feet in height may belong to two fuzzy sets '*average*' and '*tall*'. The membership functions $\mu_A$ and $\mu_B$ are the terms used to characterise the two fuzzy sets '*average*' and '*tall*', respectively. The man's height, 6 feet, belongs with a grade of 0.75 to the fuzzy set '*average*' and with a grade of 0.25 to the fuzzy set '*tall*'. The fuzzification step involves transforming the input value (6 feet) into the grades of membership (0.75 for '*average*' and 0.25 for '*tall*').

### 2.4.3 Rule Base

This step involves regulating a process output around a desired setpoint or reference value. There are a variety of different methods available for presenting the if-then rule format. The following examples are presented by Jantzen (6), in a technical report titled "*Design of Fuzzy Controllers*". The first example is the most common, and basic, of any of the linguistic rule structures and is used for the design of the fuzzy controller in **Chapter 3** of this study (6):

1. If error is Neg and change in error is Neg then output is Negative Big (NB).
2. If error is Neg and change in error is Zero then output is Negative Medium (NM).
3. If error is Neg and change in error is Pos then output is Zero.
4. If error is Zero and change in error is Neg then output is NM.
5. If error is Zero and change in error is Zero then output is Zero.
6. If error is Zero and change in error is Pos then output is Positive Medium (PM).
7. If error is Pos and change in error is Neg then output is Zero.
8. If error is Pos and change in error is Zero then output is PM.
9. If error is Pos and change in error is Pos then output is Positive Big (PB).

In the example above, the names Zero, Pos, Neg, NB, NM and PM are labels given to fuzzy sets. This information can be presented in the following relational format (6):

**Table 2.2: Fuzzy rules presented in a relational format (6).**

| **Error** | **Change in Error** | **Output** |
|---|---|---|
| Neg | Pos | Zero |
| Neg | Zero | NM |
| Neg | Neg | NB |
| Zero | Pos | PM |
| Zero | Zero | Zero |
| Zero | Neg | NM |
| Pos | Pos | PM |
| Pos | Zero | PM |
| Pos | Neg | Zero |

Each column in the table above represents the variables associated with this process. The first two columns represent inputs, while the third column is the output. Each row corresponds to a rule. This layout is useful in order to gain a concise overview of the rule base. A third format, even more compact than the last, is the tabular linguistic format.

**Table 2.3: Fuzzy rules in a tabular linguistic format** (6)**.**

| | | **Change in Error** | | |
|---|---|---|---|---|
| | | **Neg** | **Zero** | **Pos** |
| **Error** | **Neg** | NB | NM | Zero |
| | **Zero** | NM | Zero | PM |
| | **Pos** | Zero | PM | PB |

The input variables are laid out along the outside of the table, while the output variable is located inside the table. This method is quite useful for identifying any missing information. This would appear in the form of an empty cell indicating that a rule is missing (6).

### 2.4.4 Inference Engine

The inference engine is the core of the controller. As discussed in the previous section, the rules of the fuzzy controller map the strategy course to be undertaken by the inference engine. Should an error exist, the inference engine looks up the corresponding membership values as defined by the condition of the rule and maps it to the appropriate output membership function to be defuzzified (i.e. converted to a crisp output) (6).

### 2.4.5 Defuzzification

Without defuzzification, the final output from the inference stage would remain a fuzzy set. In most process applications, there is a requirement for a crisp control signal. In this step a fuzzy set is reduced to a single numbered output. There are a number of defuzzification techniques available for this operation, some of which are described below (6):

**a) Centre of Gravity (CoG) Method**

The CoG, also known as the centre of area, method is a technique for finding a crisp value ($u$) from the mid-point of the output fuzzy set using a weighted average of the membership grades. Suppose, there exists a fuzzy set within a discrete universe, and $\mu(x_i)$ is its membership value in the membership function. The following expression can be used to represent the weighted average of the elements in the support set (6):

$$u = \frac{\sum_i \mu(x_i)\, x_i}{\sum_i \mu(x_i)}$$

For a continuous universe, the summation symbols are replaced by integrals.

**b) Mean of Maximum (MoM) Method**

The MoM method is an approach used to find the average *z* where the membership of the fuzzy set is at a maximum. It may occur that several maximum points exist and so, a common practice is to take the mean of all maximum values. This particular method disregards the shape of the fuzzy set entirely, but the computational complexity is simpler than other methods and yields relatively good results.

As mentioned previously, fuzzy systems use fuzzy rules and fuzzy reasoning to draw upon a conclusion for a given scenario. Fuzzy reasoning is based on a principle that allows a systems' developer to map a function between two fuzzy sets, this is known as the extension principle. For a given scenario there exists a fuzzy set A in a universe Z. The extension principle states that if there is a function, *f*, then the fuzzy set B is given by (6):

$$B = f(A) = \sum_i \frac{\mu_A(x_i)}{f(x_i)}$$

The extension principle operates at the most fundamental level of all fuzzy inference systems. However, due its complexity and vast mathematical detail, this paper will only deal with its practical effect in computer systems.

### 2.4.6 Post-processing

Post-processing is used to scale the output of the controller into its operational engineering units. Not every control signal sent from the controller to the post-processing block will require scaling, therefore this block would be defined by a process engineer according to the process dynamics of the system (6).

## 2.5 Models used in Fuzzy Systems

### 2.5.1 The Mamdani Model

The mamdani model uses rules where by the antecedent and the consequent are both fuzzy. Consider the following two rule system:

- Rule 1: **IF** *x* is $A_1$ and *y* is $B_1$,
- Rule 2: **IF** *x* is $A_2$ and *y* is $B_2$,

where $A_1$, $A_2$, $B_1$, $B_2$, $C_1$ and $C_1$ from the expression above are fuzzy sets. Figure #.# below, illustrates how a Mamdani model takes two inputs *x* and *y*, applies the two rules in order to come to a logical solution based on these inputs.

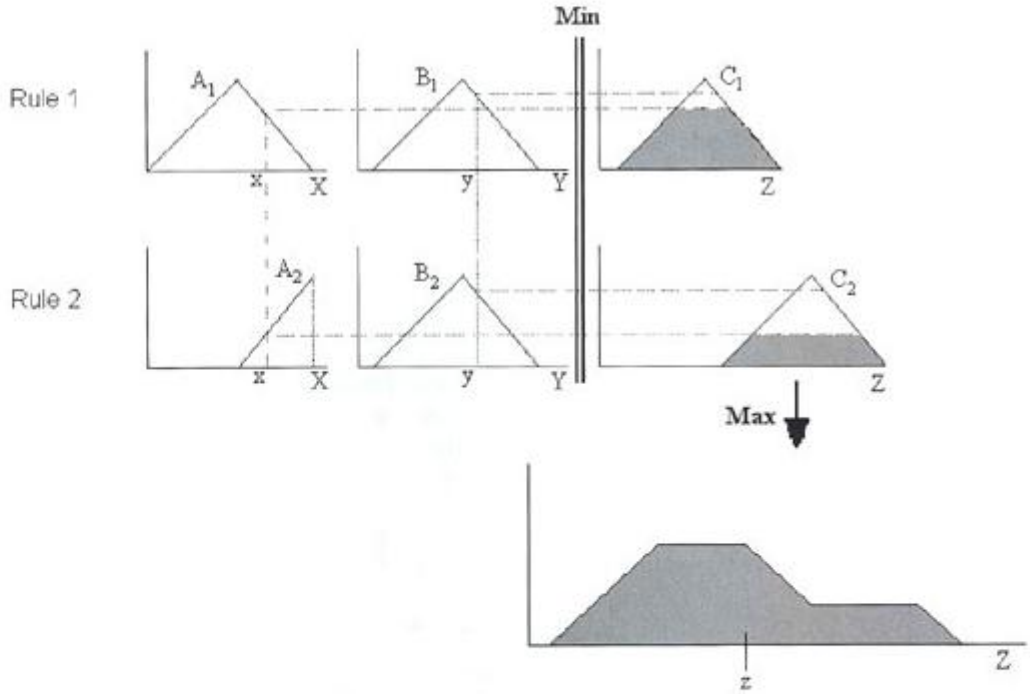


**Figure 2.14: Mamdani model for two inputs x and y**

The following steps were taken from John R's paper "*Fuzzy Logic and Knowledge Based Systems*" (12). In this paper, he lists the steps of the procedure performed by the Mamdani model for a two rule system. For the given values *x* and *y* (as depicted in **Figure 2.14** above), the following procedure was carried out (12):

a) For the value *x* find the membership values associated with fuzzy sets $A_1$ and $A_2$.

b) For the value *y* find the membership values associated with fuzzy sets $B_1$ and $B_2$.

c) For each rule, stated above, take the minimum of the membership values in $A_i$ and $B_i$.

d) Use this value to '*truncate*' the fuzzy set $C_i(i = 1, 2)$ to produce a new set $C'_i(i = 1, 2)$.

e) For each value of *z* in the truncated sets, take the maximum to produce the final output fuzzy set.

f) Optionally, '*defuzzify*' the output set to produce a single, or crisp, number.

### 2.5.2 The Takagi-Sugeno Model

This particular model type is discussed in greater detail in **section 2.6.1**. However, to illustrate the use of the Takagi-Sugeno approach in fuzzy systems, a brief explanation is described below. The Takagi-Sugeno model also uses if-then rules similar to the Mamdani model. They are presented in the following form (12):

**IF** *x* is *A* and *y* is *B*, **THEN** $z = f(x, y)$

where A and B are fuzzy sets and *z* is a crisp function in *x*, *y*. The antecedent is quite often more complicated than it appears above, normally containing more AND statements. The function in the consequent can be any function. However, in a first order Tageaki-Sugeno model the function normally takes the form (12):

$$f(x, y) = px + qy + r$$

where *p*, *q* and *r* are constants. In this particular model, the fuzzy rules contain a fuzzy antecedent and a crisp consequent. The two rules appear in the form (12):

- Rule 1: **IF** $x$ is $A_1$ and $y$ is $B_1$, **THEN** $f_1 = p_1x + q_1y + r_1$
- Rule 2: **IF** $x$ is $A_2$ and $y$ is $B_2$, **THEN** $f_2 = p_2x + q_2y + r_2$

For each input, $x$ and $y$, the membership values are found in $A_1$, $A_2$, $B_1$ and $B_2$. For each rule, the antecedent 'AND' is then found by taking the minimum of the membership grades in each rule (any t-norm would suffice). This operation yields two weighting values, $w_1$ and $w_2$, both of which are associated to each function $f_1$ and $f_2$. A weighted average of the two functions, $f_1$ and $f_2$, produces the following final output (12):

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2}$$

## 2.6 Adaptive Neuro-Fuzzy Inference System

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) are a class of adaptive networks that use a given input/output data set to construct a fuzzy inference system (FIS) whose membership functions are tuned using either an algorithm that operates on a 'backpropagation' principal alone, or in combination with a least squares estimation method (7). Adaptive neuro-learning systems are a highly efficient means for developing a learned fuzzy model structure, as this technique can develop membership function parameters that best allow the associated FIS to track the input/output data over a large operating range. Ultimately, ANFIS operates on the principle that a fuzzy systems can be formulated, or learned, via a data set obtained from an existing process model.

Using a network structure that operates similar to that of a neural network (a computational model consisting of an interconnected group of artificial neurons to process information), the input data values are mapped across a layered network; through input membership functions and their associated parameters, and then through output membership functions and their associated parameters in order to obtain a desired conclusive output (7). The parameters associated with both input and output membership functions will change through the adaptive learning process.

The ANFIS approach is designed to replicate human-like experience within a specific domain. ANFIS is capable of adapting itself in order to better its control capabilities in changing environments. Quite often, the majority of manufacturing processes have complex reaction mechanisms and non-linear, time-variant process dynamics that make their modelling, monitoring and control challenging. According to Jang et al. (8), ANFIS has shown significant results in modelling non-linear functions. Adaptive networks can cover a vast range of different approaches but for the purpose of this study, the '*Two Rule Sugeno ANFIS*' method proposed by Jang will be discussed.

### 2.6.1 Two Rule Sugeno Model

**Figure 2.15** below, represents a '*Two Rule Sugeno*' ANFIS architecture model. Assume that the fuzzy inference system has two inputs '*x*' and '*y*', and only one output 'z'. Takagi, Sugeno and Kang proposed that "in an effort to develop a systematic approach to generating fuzzy rules from a given input – output data set for such a model" (9), then the following rules can be applied to a first-order Sugeno fuzzy model:

- Rule 1: **IF** $x$ is $A_1$ and $y$ is $B_1$, **THEN** $f_1 = p_1x + q_1y + r_1$
- Rule 2: **IF** $x$ is $A_2$ and $y$ is $B_2$, **THEN** $f_2 = p_2x + q_2y + r_2$

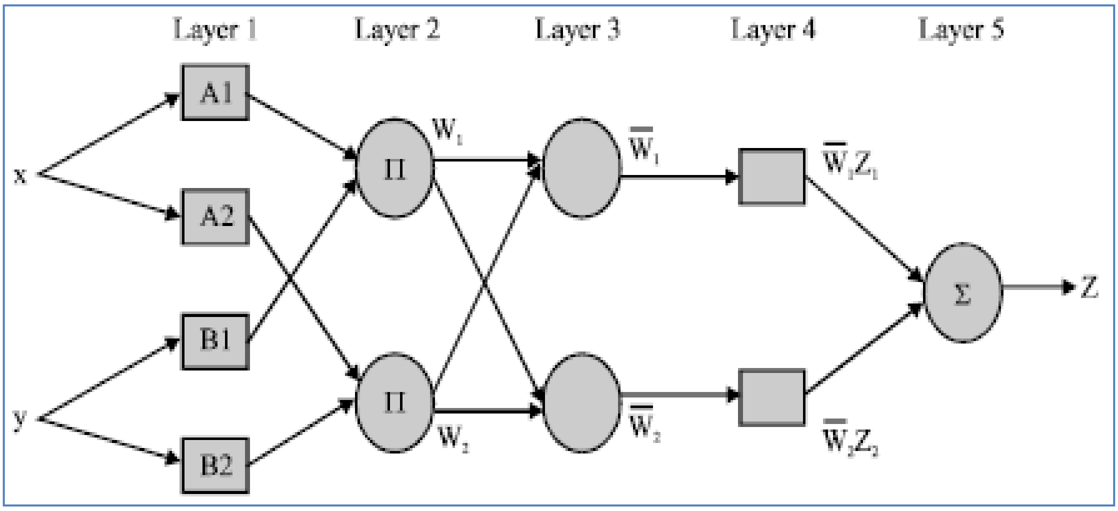


**Figure 2.15: An ANFIS structure for a two rule Sugeno system (13).**

In the diagram above, the circular nodes in Layer 2, 3 and 5 represent nodes that are fixed and the square nodes in Layer 1 and 4 represent nodes that require parameter that are learnt. In order to train this particular network both a forward pass and a backward pass is performed over the system. The forward pass propagates the input vector through the network layer by layer in ascending order. The backwards pass, takes the error and sends it back through the network using backpropogation. The following describes how each layer operates:

- **Layer 1:** this layer comprises of square nodes with a node function of either (10):

  $$O_{1,i} = \mu_{A_i}(x) \qquad for\ i = 1, 2$$

  Or,

  $$O_{1,i} = \mu_{B_{i-2}}(x) \qquad for\ i = 3, 4$$

  Where $x$ (or $y$) is the input to node *'i'* and $A_i$ (or $B_{i-2}$) is a linguistic label (e.g. small, medium, large) associated with this node function. $O_{1,i}$ represents the membership grade, or the membership function, of a fuzzy set that contains the elements $A_1$, $A_2$, $B_1$, $B_2$. The membership function can vary from triangular, trapezoidal, S-shaped or L-shaped, as long as it a characteristically continuous and piecewise differentiable function. However, for illustration purposes the Gaussian (bell-shaped) function will be used. This is demonstrated by the following equation (10):

  $$\mu_{A_i}(x) = \frac{1}{1 + \left|\frac{x - c_i}{a_i}\right|^{2b_i}}$$

  In the equation above, characters $a_i$, $b_i$ and $c_i$ are a parameter set known as the premise parameters. These particular values will vary over time and so; the bell-shaped function will vary accordingly. As a result, a variety of membership functions on the linguistic label $A_i$ will be formed to accurately develop an output for each node *'i'* (10).

- **Layer 2:** Each node in this layer is a circular (or fixed) node, labelled *prod*. An incoming signal from the previous layer is multiplied with another signal from the previous layer to produce the output of the second layer. This can be seen in the equation below:

$$O_{2,i} = w_i = \mu_{A_i}(x) \, . \, \mu_{B_i}(y), \text{ where: } i = 1, 2$$

Each node in this layer represents the so called '*firing strength*' of the rules (10).

- **Layer 3:** Again, each node in this layer is a circular or fixed node, labelled norm. Here, the $i^{th}$ node calculates the ratio of the $i^{th}$ rule's firing strength to the sum of all rules' firing strengths. The outputs of this layer are represented according to the following equation:

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \qquad i = 1, 2$$

The outputs, $O_{3,i}$, are called the '*normalized firing strengths*' as they play a normalisation role to the firing strength from the previous layer (10).

- **Layer 4:** All nodes in this layer are square (variable) nodes with a node function of the form:

$$O_{5,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i)$$

In this layer the nodes are adaptive nodes. From the equation above, the output of each node in this layer is the product of the normalised firing strength (from layer 3) and a first order polynomial (for a first order Sugeno model). The characters $p_i$, $q_i$ and $r_i$ are a parameter set referred to as the consequent parameters (10).

- **Layer 5:** There is only one circular (fixed) node, labelled $\sum$ that computes the overall output as the summation of all incoming signals. The overall output, $O_{5,i}$ in this layer is calculated according to the following equation (10):

$$O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

Having discussed each layer in detail, it can be observed that there are two adaptive layers to this ANFIS method, specifically the first and fourth layers. In the first layer the *premise* parameters, identifiable by the set: {$a_i$, $b_i$, $c_i$}, and in the fourth layer the *consequent* parameters, identifiable by the set { $a_i$, $b_i$, $c_i$ }, are tuned continuously to make the ANFIS output match the training data (10).

**Scenario 1:** When the premise parameters of the membership function are fixed, the output of the ANFIS model can be written as follows (14):

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2$$

Substituting the equation from layer 3 into the equation above gives (14):

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2$$

Substituting the fuzzy '*if-then*' rules defined at the beginning of this section, namely Rule 1 and 2, the equation now becomes (14):

$$f = \overline{w}_1(p_1 x + q_1 y + r_1) + \overline{w}_2(p_2 x + q_2 y + r_2)$$

Rearranging, the output becomes (14):

$$f = (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 + (\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2$$

This yields a linear combination of the modifiable consequent parameters $p_1$, $q_1$, $r_1$, $p_2$, $q_2$ and $r_2$. The least squares method is then used to identify the optimal values of these parameters.

**Scenario 2:** According to S.J. Iqbal and S.A. Miratashi (14), "*when the premise parameters are not fixed, the search space becomes larger and the convergence of the training becomes slower*". In this case, a hybrid algorithm combining the least squares method and the gradient descent method is utilised to solve this issue. The forward pass operates using the least squares method to optimise the consequent parameters with the premise parameters. Once the optimal consequent parameters are found, the backwards pass begins immediately (14).

The gradients decent method is used to adjust optimally the premise parameters that correspond to the fuzzy sets in the input domain. The output of the ANFIS is then calculated by employing the consequent parameters found in the forward pass. The output error is used to adapt the premise parameters by means of a standard back-propagation algorithm. According to S.J. Iqbal and S.A. Miratashi (14), "*It has been proven that this algorithm is highly efficient in training the ANFIS*". The precise computation of both the forward pass and the backward pass can be very complicated and not entirely relevant to the overall theme of this study. Therefore, it will not be discussed.

# CHAPTER 3 METHODOLOGY

## 3.1 Introduction

This chapter aims to provide an overview of the methodological approach and design implementation selected for the development of a fuzzy control system that is supported by fuzzy theory. In this chapter, the methodology applied to the development of the control consists of the following stages:

1. Method of delivery
2. Basis for design model
3. Analysing data
4. Generating membership functions
5. Fuzzy control system

The design criteria for this fuzzy system are reliant on the control of both the flowrate of the feed stream and the thermo-pressure in the evaporator in order to achieve a desired density output. For this particular design the following process conditions are assumed:

a) The manufacturing process has completed its initial start-up phase and is operating within normal operating parameters. This implies that the storage tank is at sufficient operating level, that the plant has been sufficiently cleaned prior to production and that the evaporators are at sufficient temperature and wetness.

b) All process instrumentation devices (such as valves, flow meters and temperature controls) are operating at normal capacity and without complications.

## 3.2 Method of Delivery

The design of this control system is simulated entirely using a sophisticated language and interactive environment known as '*Matlab*', which enables the user to perform computationally intensive tasks and to implement numerical algorithms for a variety of process applications. Matlab is a technical computing environment that incorporates its own programming language similar to *C* or *C++*. Matlab is a programming language often used by control engineers "*to model physical plants, to design control systems, and to evaluate their performance by simulations*" (15).

The software offers the capability to simulate real-time control systems whose process dynamics may be intrinsically discrete, continuous, linear or non-linear. Matlab offers numerous applications such as signal and image processing, communications, control design, test and measurement etc. Add-on toolboxes (used specifically in the design of the control system in this paper) are a collection of specific functions used to extend the Matlab environment to solve a particular problem class relative to the application area.

The fuzzy logic system designed in this thesis was developed using a collection of data obtained from an information technology system called '*Manufacturing Execution System*' (MES). MES is designed to monitor the production processes in a manufacturing facility. This includes:

- Presentation of schedules to work-centres
- Collection of production information, including time, quantity, quality and operator behaviour
- Analysis of production information, such as volume of product produced over a specified time frame
- Shipping and dispatch records and
- Product traceability

Using the MES information system, a collection of data was obtained from the Glanbia plc dairy manufacturing facility located in Ballyragget, Co. Kilkenny (see **Appendix B**). Density input, feed flow (i.e. flowrate), thermo-pressure and density output are continuously measured during production. The MES system records this data for analysis. Also, any alterations made by the operators to the controlled variables '*feed flow*' and '*thermo-pressure*' were recorded by this system. Therefore, the data was collated and analysed to develop a fuzzy control strategy based on the behaviour of process operators.

### 3.3 Basis for Design Model

The aim of this study is to design a control strategy that reads a measured variable in the process and makes adjustments to the comparative process parameters accordingly. As mentioned in **Chapter 1**, this study is founded on an existing process issue found in Glanbia's dairy production facility, located in Ballyragget, Co. Kilkenny. The process parameters that this design is concerned with are as follows:

1. Density input, measured in kilograms per meter cubed (kg/$m^3$), is an uncontrolled process parameter. Throughout the process, this value can typically range from 1,080 kg/$m^3$ → 1,180 kg/$m^3$, according to the data. The value of this parameter will determine the assessment of appropriate settings for both the thermo-pressure and feed flow parameters in order to achieve the desired density output.

2. Density output, measured in kilograms per meter cubed (kg/$m^3$), is the process output that the control system is required to control within the range 1,180 kg/$m^3$ → 1,280 kg/$m^3$, according to the data.

3. Thermo-pressure, measured in bar, is a controlled parameter.

4. Feed flow or flowrate, measured in meter cubed per hour ($m^3$/hr), is a controlled parameter and must be maintained within specified operating limits to prevent dry spots (caused by low flow) or flooding (caused by high flow). Therefore, the minimum and maximum limits are 27 $m^3$/hr and 32 $m^3$/hr, respectively[2].

[2] These values represent actual operating conditions and were set by senior process engineers in Glanbia.

Figure 3.1 below, is a 'screen shot' of the operators supervisory control and data acquisition (SCADA) screen used to control the process in the Glanbia plant. The feed material is stored in the tank, situated on the left-hand side of the image and titled permeate tank. Upon completion of the 'start-up phase', the operator inputs the desired feed flow (located on a pop up window designed for parameter control) from the storage tank. The feed enters the system and passes through a density meter to calculate the density of the feed.

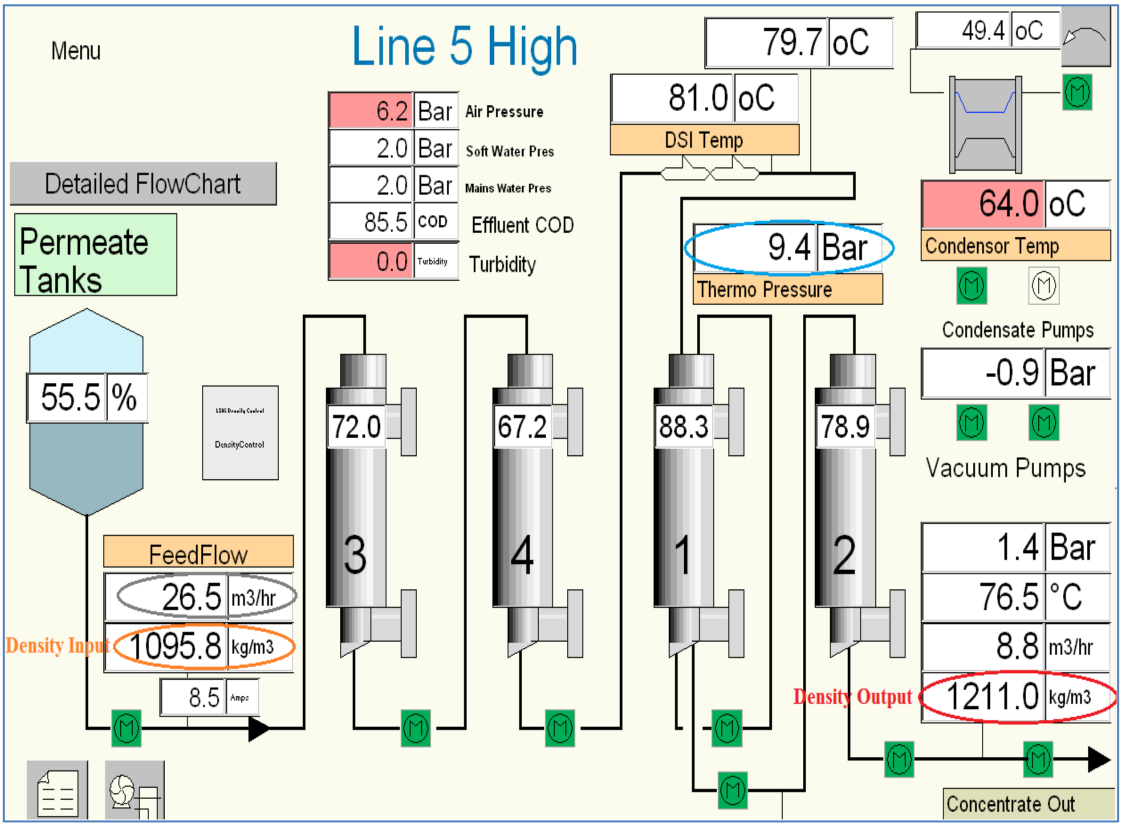


**Figure 3.1: Operators SCADA screen with relevant (circled) process parameters.**

During "*start-up*", the evaporators (labelled 1 – 4) are brought up to operating temperature and sufficiently '*wetted*' to cook the feed material upon entering the evaporators. The term '*thermo-pressure*' is used to describe the pressure of the steam that resides in the evaporator. The thermo-pressure is directly related to the temperature in the evaporator so, an increase in pressure results in an increase in temperature. Ultimately, this affects the density of the final product. Once the feed has passed through the system, its density is calculated using another density meter.

### 3.4 Analysis of Manufacturing Data

The data collected from Glanbia plc were grouped according to the following process measurements:

- Density input
- Feed flow
- Thermo-pressure and
- Density output

Using MES, the process operating values for each of the parameters mentioned above were collected and listed in an excel file. All density output values that occurred over a specified operating period were collated for analysis. For each density output value, its corresponding thermo-pressure, feed flow and density input[3] were also collected. This data was then analysed to determine how the operator reacted should the measured variable rise or fall above the setpoint.

Once an understanding of the operators' behaviour was developed, the data was filtered[4] further to only include values that fell within a range of +/- 2 kg/m$^3$ of the setpoint, 1,230 kg/m$^3$. This significantly reduced the level of data and provided a clearer representation of the values required to control the process at its optimal operating range. **Appendix B** illustrates a data set that lists a varying density output and its corresponding operating parameters over a large difference range with respect to the setpoint. The data featured in the table below, is a representation of an increasing density input and the corresponding thermo-pressure and feed flow that gave the desired density output. Subsequently, this reference data

[3] Process delay times had to be accounted for when selecting the data, this operation was carried out by senior process engineers who are familiar with the process.

[4] The range of +/- 2 kg/m$^3$ was included to the filter criteria as operators have stated that little to no change is made if the density output deviates from the setpoint by this much.

was used to formulate the membership functions for the fuzzy control system developed in **Section 4.6** below.

**Table 3.1: MES data illustrating required operating parameters to achieve desired density output.**

| **Parameters:** | **Density Input** | **Feed Flow** | **Thermo Pressure** | **Density Output** | **Set Point** | **Difference** |
|---|---|---|---|---|---|---|
| **Units:** | ($kg/m^3$) | ($m^3/hr$) | (bar) | ($kg/m^3$) | ($kg/m^3$) | ($kg/m^3$) |
| **Variables:** | 1110 | 28 | 9.3 | 1230 | 1230 | 0 |
| | 1112 | 28 | 9.3 | 1230 | 1230 | 0 |
| | 1114 | 28 | 9.3 | 1230 | 1230 | 0 |
| | 1116 | 28.5 | 9.3 | 1230 | 1230 | 0 |
| | 1118 | 28.5 | 9.3 | 1230 | 1230 | 0 |
| | 1120 | 28.5 | 9.3 | 1230 | 1230 | 0 |
| | 1122 | 29 | 9.2 | 1230 | 1230 | 0 |
| | 1124 | 29 | 9.1 | 1230 | 1230 | 0 |
| | 1126 | 28 | 8.7 | 1230 | 1230 | 0 |
| | 1126 | 29.5 | 9 | 1230 | 1230 | 0 |
| | 1128 | 29.5 | 8.9 | 1230 | 1230 | 0 |
| | 1129 | 30 | 8.6 | 1230 | 1230 | 0 |
| | 1130 | 30 | 8.7 | 1230 | 1230 | 0 |
| | 1131 | 29 | 8.3 | 1230 | 1230 | 0 |
| | 1131 | 30 | 8.5 | 1230 | 1230 | 0 |
| | 1132 | 30 | 8.6 | 1230 | 1230 | 0 |
| | 1134 | 30 | 8.5 | 1230 | 1230 | 0 |
| | 1136 | 30 | 8.4 | 1230 | 1230 | 0 |
| | 1138 | 30.5 | 8.3 | 1230 | 1230 | 0 |
| | 1140 | 30.5 | 8.2 | 1230 | 1230 | 0 |
| | 1142 | 31 | 8.1 | 1230 | 1230 | 0 |
| | 1144 | 31 | 8 | 1230 | 1230 | 0 |
| | 1146 | 31 | 7.9 | 1230 | 1230 | 0 |

## 3.5 Generating Fuzzy Membership Function

As previously discussed, the membership functions used in this fuzzy system are constructed from the data set above, obtained from Glanbia plc. Two types of membership functions were used to generate the fuzzy system: the triangular membership function and the trapezoidal membership function. In the proposed fuzzy system, there are two fuzzy controllers required to control the process. The first controller was designed to control the feed-flow. This required a total of sixteen membership functions for the input block and thirteen membership functions for the output block. The second controller was designed to control the themo-pressure. This required a total of fourteen functions for the input block and fourteen membership functions for the output block.

According to Jantzen (6), fuzzy set theory suggests that there is no one practiced method for determining the shape and width of a fuzzy membership function. It is a subjective process that will vary with the designer of the control system. However, a few rules of thumb should be considered when trying to formulate the membership functions for a design model:

- Each set should be wide enough to allow for measurement noise.
- A certain amount of overlap should exist between membership functions; this prevents the controller from returning a poorly defined manipulated variable (output).
- Start with triangular sets, as these are the most basic form of membership function and the easiest to design. Should it not offer the desired control, then more complicated membership functions be considered.

The membership functions developed for the control strategy in this paper are illustrated in the tables below. The tables below list the following information: the name of the membership function used in the Matlab programme; the membership function type (i.e. triangular, trapezoidal, Gaussian etc.) and the parameters that define the membership function. This is done for both controllers titled ‘feed flow’ and ‘thermo-pressure’.

**Table 3.2: Membership functions for the feed flow controller.**

| **Feed Flow Controller** | | | | | |
|---|---|---|---|---|---|
| **Input (Density Input)** | | | **Output (Feed Flow)** | | |
| **Name:** | **Function Type:** | **Parameters:** | **Name:** | **Function Type:** | **Parameters:** |
| MF1 | Trapezoidal | [1080 1080 1112 1112] | MF1 | Triangular | [28 28 28] |
| MF2 | Triangular | [1112 1114 1116] | MF2 | Triangular | [28  28.25 28.5] |
| MF3 | Triangular | [1114 1116 1118] | MF3 | Triangular | [28.25 28.5 28.75] |
| MF4 | Triangular | [1116 1118 1120] | MF4 | Triangular | [28.5 28.75 29] |
| MF5 | Triangular | [1118 1120 1122] | MF5 | Triangular | [28.75 29 29.25] |
| MF6 | Triangular | [1120 1122 1124] | MF6 | Triangular | [29 29.25 29.5] |
| MF7 | Triangular | [1122 1124 1126] | MF7 | Triangular | [29.3 29.6 29.8] |
| MF8 | Triangular | [1124 1126 1128] | MF8 | Triangular | [29.5 29.75 30] |
| MF9 | Triangular | [1126 1128 1130] | MF9 | Triangular | [29.75 30 30.25] |
| MF10 | Triangular | [1128 1130 1132] | MF10 | Triangular | [30 30.25 30.5] |
| MF11 | Triangular | [1130 1132 1134] | MF11 | Triangular | [30.25 30.5 30.75] |
| MF12 | Triangular | [1132 1134 1136] | MF12 | Triangular | [30.5 30.75 31] |
| MF13 | Triangular | [1134 1136 1138] | MF13 | Triangular | [31 31 31] |
| MF14 | Triangular | [1136 1138 1140] | - | - | - |
| MF15 | Triangular | [1138 1140 1142] | - | - | - |
| MF16 | Trapezoidal | [1142 1142 1180 1180] | - | - | - |

Table 3.3: Membership functions for the thermo-pressure controller.

| Thermo-Pressure | | | | | |
|---|---|---|---|---|---|
| Input (Density Input) | | | Output (Thermo-Pressure) | | |
| Name: | Function Type: | Parameters: | Name: | Function Type: | Parameters: |
| MF1 | Trapezoidal | [1080 1080 1120 1120] | MF1 | Triangular | [7.8 7.8 7.8] |
| MF2 | Triangular | [1120 1122 1124] | MF2 | Triangular | [7.8 7.9 8] |
| MF3 | Triangular | [1122 1124 1126] | MF3 | Triangular | [7.9 8 8.1] |
| MF4 | Triangular | [1124 1126 1128] | MF4 | Triangular | [8 8.1 8.2] |
| MF5 | Triangular | [1126 1128 1130] | MF5 | Triangular | [8.1 8.2 8.3] |
| MF6 | Triangular | [1128 1130 1132] | MF6 | Triangular | [8.2 8.3 8.4] |
| MF7 | Triangular | [1132 1134 1136] | MF7 | Triangular | [8.3 8.4 8.5] |
| MF8 | Triangular | [1134 1136 1138] | MF8 | Triangular | [8.4 8.5 8.6] |
| MF9 | Triangular | [1136 1138 1140] | MF9 | Triangular | [8.6 8.7 8.8] |
| MF10 | Triangular | [1138 1140 1142] | MF10 | Triangular | [8.8 8.9 9] |
| MF11 | Triangular | [1140 1142 1144] | MF11 | Triangular | [8.9 9 9.1] |
| MF12 | Triangular | [1142 1144 1146] | MF12 | Triangular | [9 9.1 9.2] |
| MF13 | Triangular | [1144 1146 1148] | MF13 | Triangular | [9.1 9.2 9.3] |
| MF14 | Trapezoidal | [1148 1148 1180 1180] | MF14 | Triangular | [9.3 9.3 9.3] |

The two aforementioned tables were developed using the information obtained from the tabled data in **Table 3.1**. The membership functions were constructed by comparing the density of the feed material coming into the plant with the output density. When there was a difference of zero between the actual density output and the setpoint density output, then all values were recorded. These values represented the required settings for both feed flow and thermo-pressure that achieved a density output of 1230 kg/m$^3$.

### 3.6 Fuzzy Control System

Based on the analyses of the original data set in **Appendix B**, three vague linguistic statements were developed to illustrate the appropriate corrective measures that the control system should perform if the measured density output varies from the target setpoint:

a) IF density is lower than the setpoint THEN decrease the feed flow and increase the thermo-pressure.

b) IF density is within operating range THEN no alterations are performed to the controlled parameters.

c) IF density is higher than the setpoint THEN increase the feed flow and decrease the thermo-pressure.

The above statements are characteristic of a typical Mamdani fuzzy system, discussed previously in **Section 2.5.1**. **Figure 3.2** and **Figure 3.3** below, illustrate a basic schematic diagram of the two fuzzy controllers, for both feed flow and thermo-pressure. Using Matlab's toolbox function for fuzzy logic, two controllers were designed on the principle of the linguistic statements defined above. The illustrations below depict the constructed fuzzy system (as represented by the Matlab programme) developed using the membership functions. It contains two fuzzy controllers for controlling feed flow and thermo-pressure separately. The input block, '*Density Input*', is slightly different in each case. The input block utilised by the feed flow controller contains a total of thirteen[5] membership functions. The input block in utilised by the thermo-pressure controller contains fourteen membership functions.

[5] In this case, some of the membership functions overlapped when being mapped to the resultant output.

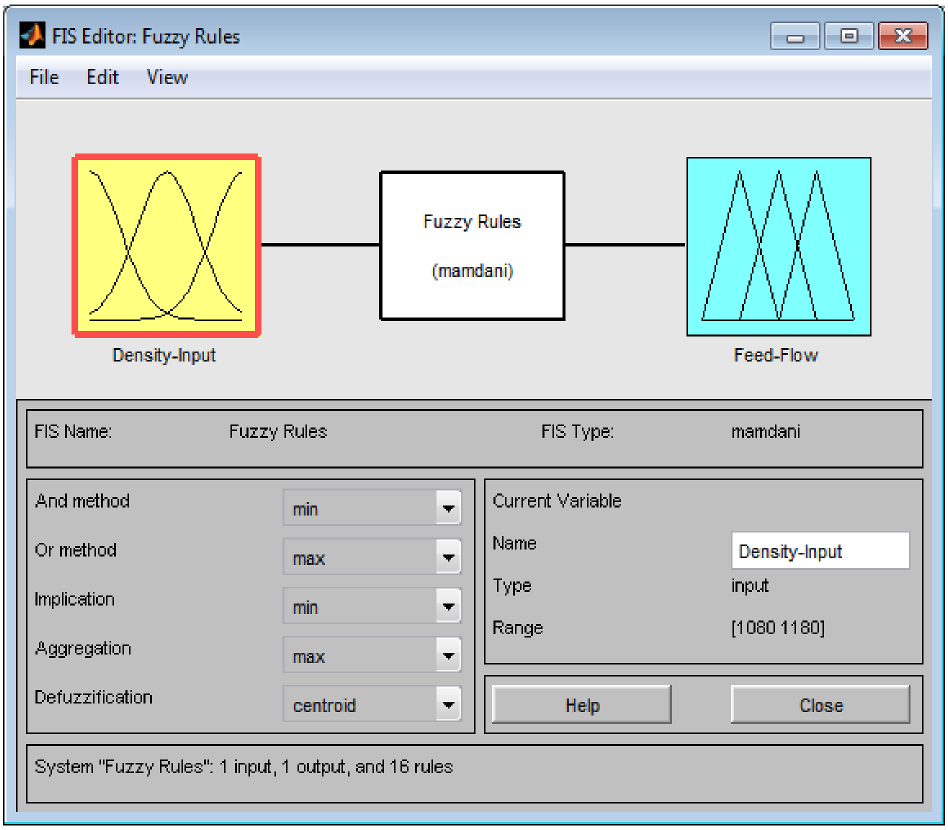


**Figure 3.2: Basic fuzzy control system for the control of feed flow.**

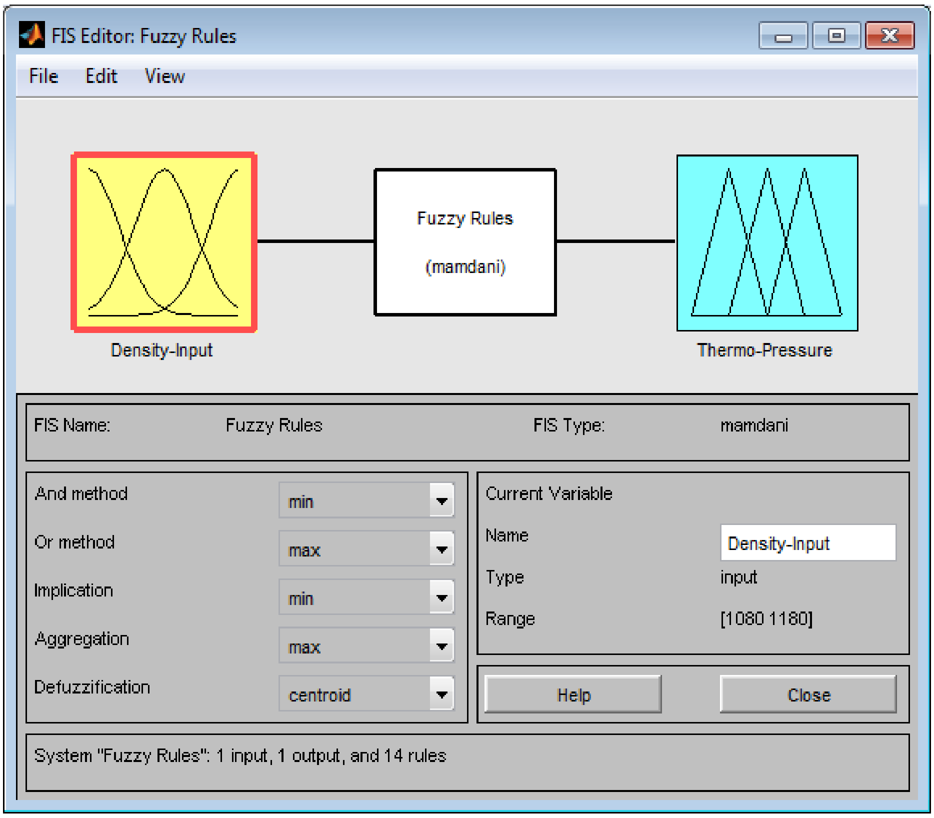


**Figure 3.3: Basic fuzzy control system for the control of thermo-pressure.**

Using a similar logic based rule structure outlined by the three ambiguous statements mentioned at the beginning of this section, the controller rules for this control strategy can be defined using the membership functions from **Table 3.2** and **Table 3.3**. These rules are formulated in the middle block of the control system depicted in the images above, and read as follows:

a) For the feed flow controller:

IF density input reads MF1 THEN desired feed flow is MF1.

IF density input reads MF2 THEN desired feed flow is MF2.

IF density input reads MF3 THEN desired feed flow is MF3.

IF density input reads MF4 THEN desired feed flow is MF3.

IF density input reads MF5 THEN desired feed flow is MF4.

IF density input reads MF6 THEN desired feed flow is MF5.

IF density input reads MF7 THEN desired feed flow is MF6.

IF density input reads MF8 THEN desired feed flow is MF7.

IF density input reads MF9 THEN desired feed flow is MF8.

IF density input reads MF10 THEN desired feed flow is MF9.

IF density input reads MF11 THEN desired feed flow is MF9.

IF density input reads MF12 THEN desired feed flow is MF9.

IF density input reads MF13 THEN desired feed flow is MF10.

IF density input reads MF14 THEN desired feed flow is MF11.

IF density input reads MF15 THEN desired feed flow is MF12.

IF density input reads MF16 THEN desired feed flow is MF13.

b) For the thermo-pressure controller:

IF density input reads MF1 THEN desired feed flow is MF14.

IF density input reads MF2 THEN desired feed flow is MF13.

IF density input reads MF3 THEN desired feed flow is MF12.

IF density input reads MF4 THEN desired feed flow is MF11.

IF density input reads MF5 THEN desired feed flow is MF10.

IF density input reads MF6 THEN desired feed flow is MF9.

IF density input reads MF7 THEN desired feed flow is MF8.

IF density input reads MF8 THEN desired feed flow is MF7.

IF density input reads MF9 THEN desired feed flow is MF6.

IF density input reads MF10 THEN desired feed flow is MF5.

IF density input reads MF11 THEN desired feed flow is MF4.

IF density input reads MF12 THEN desired feed flow is MF3.

IF density input reads MF13 THEN desired feed flow is MF2.

IF density input reads MF14 THEN desired feed flow is MF1.

### 3.7 Summary

In this chapter the methodology was described with emphasis on the method of delivery, the basis for design model, analysis of manufacturing data, generating fuzzy membership function and fuzzy control system. The next chapter review the results attained using the above methodology.

# CHAPTER 4 RESULTS

## 4.1 Introduction

The hypothesis of this study was tested using a computational, simulation software called Matlab. Using a fuzzy logic toolbox, one of the many applications available with this software, a simple input/output control strategy was developed. The control strategy consists of two fuzzy logic controllers designed to control the feed flow and thermo-pressure for a variable density input. Each controller uses a single-input, single-output design mechanism. In this section the results are discussed as follows:

1. Modelling of the feed flow controller
2. Modelling of the thermo-pressure controller
3. Analysis of results

## 4.2 Modelling of the Feed Flow Controller

In this test an array of density input operating values, ranging from 1,080 to 1,180 kg/m$^3$, were evaluated by the controller to test what feed flow output signal would be sent to the plant. In this test a density input series was considered (containing 101 points) where all values for the input variables are increasing at a constant rate of 1 kg/m$^3$. Using the 'surface viewer' function of the fuzzy logic toolbox, a two-dimensional curve was constructed as illustrated in **Figure 4.1** below. This diagram represents the mapping from density input to feed flow within the controller over the entire series of variables.

From the diagram, it can be observed that there is a strong linear relationship between the input and output parameters of the controller. For an increasing density input, the process requires an increase in flowrate of the material entering the process. However, this increase can only operate within the minimum and maximum process limits of 27 $m^3$/hr and 32 $m^3$/hr, respectively, to prevent dry spots resulting from low flow rates or flooding resulting from excessivley high flow rates. The derivation of the maximum and minimum limits are the product of years of process knowledge and experience on the part of process engineers and operators. The membership functions designed in the previous section had to adhere to these constraints.

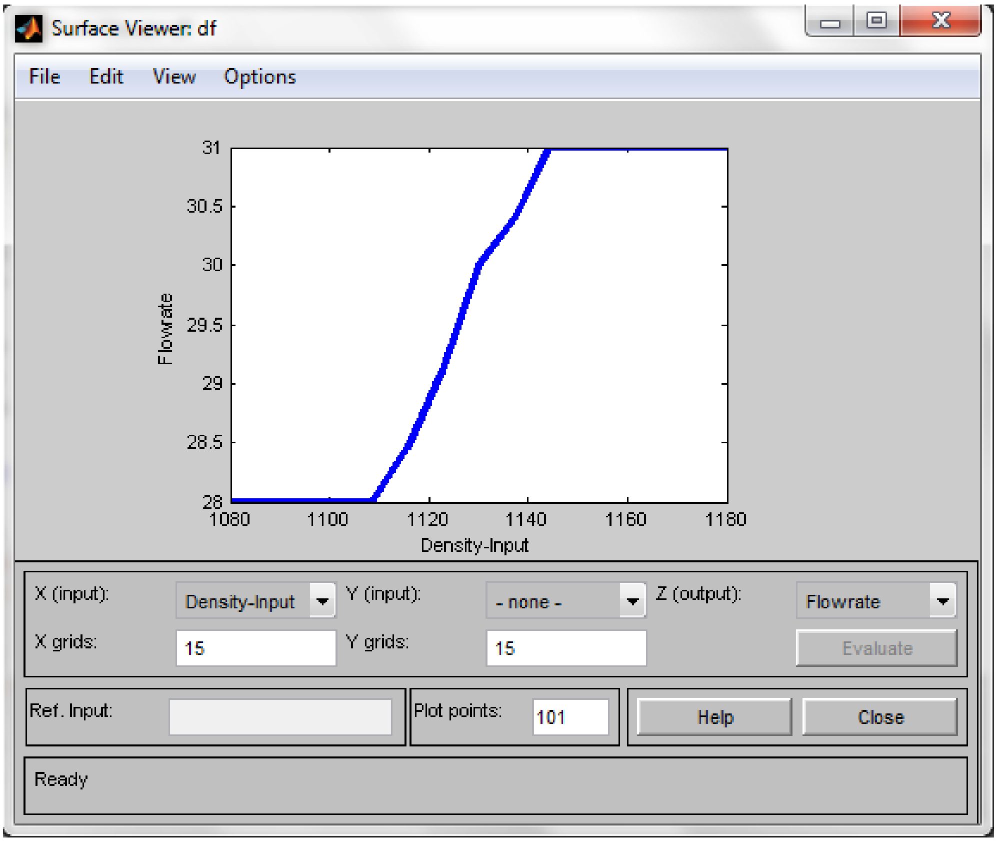


**Figure 4.1: Controller output for the mapping of density input with feed flow.**

## 4.3 Modelling of the Thermo-Pressure Controller

The second controller in the fuzzy control system is modelled in the same fashion as the first. Again, an array of density input operating values, ranging from 1,080 to 1,180 kg/m$^3$, were evaluated by the controller to test what thermo-pressure output signal would be sent to the plant. The test used a density input series (containing 101 points) where all values for the input variables are increasing at a constant rate of 1 kg/m$^3$. As stated before, the 'surface viewer' function of the fuzzy logic toolbox was used to construct a two-dimensional curve for the input/output relationship (illustrated in **Figure 4.2** below). The diagram represents the mapping from density input to thermo-pressure within the controller over the entire series of variables.

From the diagram, it can be observed that there is a strong linear relationship between the input and output parameters of the controller. For an increasing density input, the process requires a decrease in thermo-pressure in the evaporators. The optimum thermo-pressure output ranges from 7.9 to 9.3 bar, according to the data collected by Glanbia. The diagram below illustrates a maximum value of 9.6 bar and a minimum value of 7.9 bar. In the case of low density input (sub 1120 kg/m$^3$) and high density input (greater than 1148 kg/m$^3$), no variation in thermo pressure should occur and the maximum and minimum levels should be used. This is conclusive in the illustration below which is represented by the horizontal blue lines that run parallel with the density input axis.

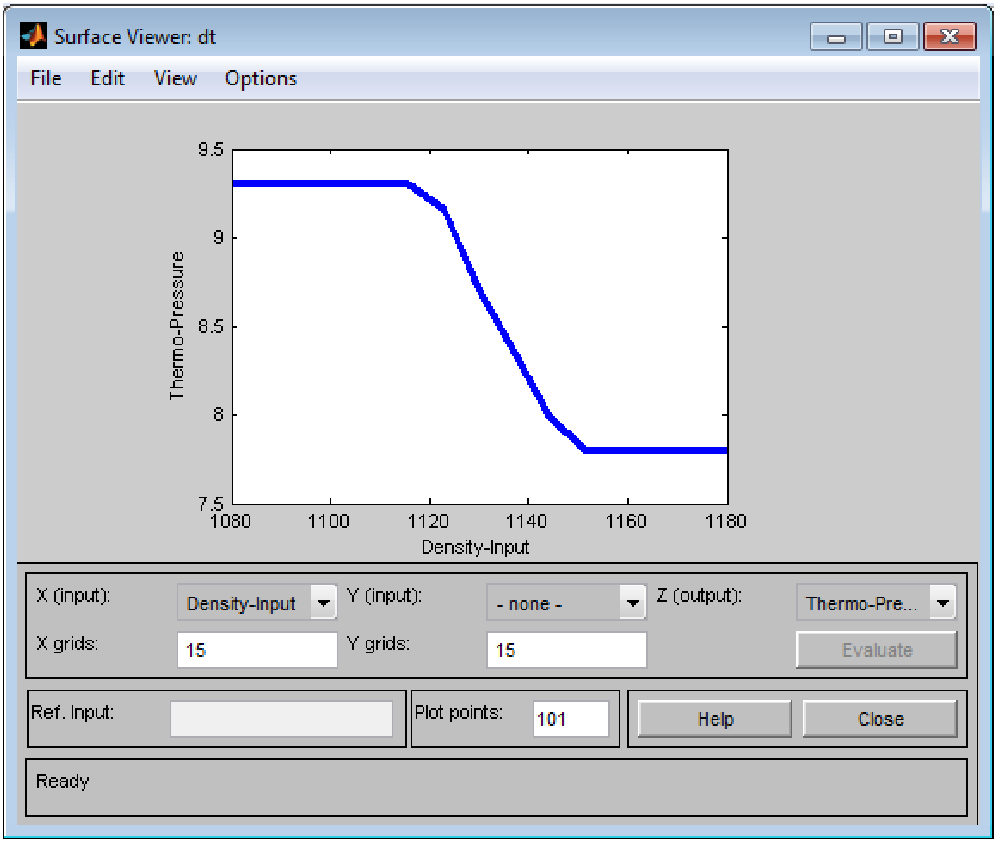


**Figure 4.2: Controller output for the mapping of density input with thermo-pressure.**

## 4.4 Analysis of Results

The table below, represents a selected portion of the results for the output of the controller compared to its setpoint for a varying flow (for full results, see **Appendix C**). In most cases, the output from the controller matches the set point exactly. And in the cases where the controller output does not match the output setpoint, there is a small difference of less than one percent. The data below exemplifies an accurate control strategy for this particular manufacturing issue.

**Table 4.1: Results obtained from testing the fuzzy logic control system**

| Density Input | Thermo-Pressure | Setpoint | Feed Flow | Setpoint |
|---|---|---|---|---|
| $kg/m^3$ | bar | bar | $m^3/hr$ | $m^3/hr$ |
| 1080 | 9.3 | 9.3 | 28 | 28 |
| 1092 | 9.3 | 9.3 | 28 | 28 |
| 1104 | 9.3 | 9.3 | 28 | 28 |
| 1110 | 9.3 | 9.3 | 28 | 28 |
| 1114 | 9.3 | 9.3 | 28.2504 | 28.25 |
| 1116 | 9.3 | 9.3 | 28.4996 | 28.5 |
| 1118 | 9.3 | 9.3 | 28.4996 | 28.5 |
| 1120 | 9.3 | 9.3 | 28.75 | 28.75 |
| 1122 | 9.1997 | 9.2 | 29.0004 | 29 |
| 1124 | 9.1003 | 9.1 | 29.2496 | 29.25 |
| 1126 | 9 | 9 | 29.5662 | 29.5 |
| 1128 | 8.8997 | 8.85 | 29.7504 | 29.75 |
| 1130 | 8.7 | 8.7 | 29.9996 | 30 |
| 1132 | 8.55 | 8.6 | 29.9996 | 30 |
| 1134 | 8.5003 | 8.5 | 29.9996 | 30 |
| 1136 | 8.4 | 8.4 | 30.25 | 30.25 |
| 1138 | 8.2997 | 8.3 | 30.5004 | 30.5 |
| 1140 | 8.2003 | 8.2 | 30.7496 | 30.75 |
| 1142 | 8.1 | 8.1 | 31 | 31 |
| 1144 | 7.9997 | 8 | 31 | 31 |
| 1146 | 7.9003 | 7.9 | 31 | 31 |
| 1148 | 7.8 | 7.8 | 31 | 31 |
| 1160 | 7.8 | 7.8 | 31 | 31 |
| 1166 | 7.8 | 7.8 | 31 | 31 |
| 1178 | 7.8 | 7.8 | 31 | 31 |
| 1180 | 7.8 | 7.8 | 31 | 31 |

## 4.5 Summary

In this section, testing was conducted on the designed fuzzy logic control system to determine if the controller produced the required feed flow and thermo-pressure output for a varying density input. It was clear from the results obtained that relatively tight control around the setpoints was achieved. It was also evident from the two diagrams produced in this section that a linear relationship was found to exist between the input and two output parameters.

# CHAPTER 5 Conclusion

## 5.1 Introduction

In this study, a fuzzy system was implemented using fuzzy logic theory to obtain a target '*density output*' from a variable '*density input*' by varying the feed flow into the process and thermo-pressure of the evaporators. Using Matlab and its associated fuzzy logic toolbox to simulate the process control system, it was concluded that the triangular and trapezoidal membership functions offered precise control for a varying density input.

## 5.2 Implications of study

The findings of this study allow for the research to conclude, with certainty, the assertion that fuzzy logic control offers favourable results for processes with large dead times. The hypothesis of this study was tested with the aim of developing a control strategy that replicated the rationale of human behaviour. Results showed that for a varying density input, both feed flow and thermo-pressure changed to the desired operating condition accordingly. A density input of 1080 kg/m$^3$ yielded an output of 9.3 bar in the thermo-pressure controller and an output of 28 m$^3$/hr in the feed flow controller. This met the required operating conditions expected by both operators and process engineers for this process.

Furthermore, the results illustrated excellent control for a varying density input over a range of 1,080 kg/m$^3$ to 1,180 kg/m$^3$ yielded. One of the objectives of this study was to develop a controller that manipulated the thermo-pressure AND feed flow for a change in feed density. Instead, two controllers were designed to control both parameters separately. A single controller alone could not determine an accurate response for both thermo-pressure and feed flow due to the number of membership functions associated with the density input in each controller. The density input block for the feed flow controller required 16 membership functions. Whereas, the density input block for the thermo-pressure block only necessitated 14 membership functions.

This level of complexity was a direct consequence of the analysis of the data collected from the Glanbia system. The design of the control strategy for the feed flow controller was more complex due to the high number of unique membership functions required to achieve the desired output. Based on the data from Glanbia the thermo-pressure required significantly less adjustment during the manufacturing process, thus the control strategy required less membership functions. The recommendation of the operators stated that for a density input ranging from 1,080 – 1,118 kg/m$^3$ a thermo–pressure setting of 9.3 bar should be used. However, for the same density range, a feed flow range of 28 - 28.75 m$^3$/hr was required to achieve a density output of 1,230 kg/m$^3$.

## 5.3 Limitations of study

At this point, it may seem that fuzzy logic it is the answer to all control issues with the attributes of this type of manufacturing process. This is not the case. Fuzzy logic can best be described as "*a convenient way to map an input space to an output space*" (16). Other than trying to understand the concepts of fuzzy theory, there is no one right way to develop a fuzzy system. However, for this particular study the drawback and limitations of the design can be described as follows:

- The control strategy required a vast amount of data to produce an accurate data set that accurately represents the ideal operating conditions being modelled.
- Defining fuzzy sets and membership functions can be an extremely tedious task. It is normally performed by a collaborative effort of both operators, process engineers and those who posses expert knowledge in the relevant process field. Even then, there is a reasonable chance of debate among each party when attempting to formulate the fuzzy sets and membership functions.

- There is no one correct design procedure for developing a fuzzy system. Selecting a system type, Sugeno or Mamdani, comes down to the designer's interpretation of and preference for a particular form of fuzzy theory.

No process is ever perfect and there will always be unpredictable disturbances that may cause the dynamics of the process to change suddenly. The major issue regarding a process such as this one is its inherent process delay time, approximately 9 – 10 minutes in the case of Glanbia's process. As discussed in chapter 1, simple feedback control systems alone are not suitable for this manufacturing system. The proposed control approach is developed on the interpretation of '*expert knowledge*' and operator experience for the existing manufacturing process. Should the process change in any major manner, the control strategy in place may be rendered obsolete and may require re-engeneering.

## 5.4 Future Developments

Therefore, it is important to identify the limitations of design in order to establish future issues that still require a resolution. Based on the limitations discussed in the previous section above, further work is required in the development of fuzzy control systems that have the capability of controlling more than one controller e.g. a single controller with the capability to vary both the feed flow and/or the thermo-pressure within the same device. Therefore, one controller would be required instead of two.

For this particular manufacturing process, further investigations could be performed in the area of ANFIS, mentioned in **Chapter 2**. Combining the MES system that is already in place, along with an adaptive neuro-fuzzy interference system would assist in defining the fuzzy sets and associative membership functions thus, reducing the dependency on expert knowledge from the operators and process engineers. This would also offer a more accurate means of developing the membership functions associated with the process, as the data obtained would be analysed by this online system.

BIBLIOGRAPHY


**1.** Wang, Hong Guang. ***Fuzzy Control in Manufacturing Systems.*** **Eindhoven : Eindhoven University of Technology, 1997. 90-386-0540-4.**

2. *Nonlinear Modeling and Cascade Control Design for Multi Effect Falling Film Evaporators.* **Karimi, M. and Jahanmiri, A.** 2, Shiraz : Siraz University, 2006, Vol. 3.

3. *Fuzzy concepts applied to food product quality control: A review.* **Perrot, N., et al.** 9, s.l. : Science Direct, 2006, Fuzzy Sets and Systems, Vol. 157, pp. 1145 - 1154. 0165-0114.

4. *Cascade controller design for concentration control in a falling-film evaporator.* **Bakker, Huub H.C., et al.** 5, 2006, Food Control, Vol. 17, pp. 325 - 330. 0956-7135.

5. *Fuzzy Logic in Control Systems: Fuzzy Logic Controller I.* **Lee, C.C.** 2, 1990, Systems, Man and Cybernetics, Vol. 20, pp. 404 - 418. 0018-9472.

6. **Jantzen, Jan.** *Design of Fuzzy Controllers.* Department of Automation, Technical University of Denmark. s.l. : Technical University of Denmark, 1998. 98-E 864.

7. **Ross, Timothy J.** *Fuzzy Logic with Engineering Applications.* s.l. : John Wiley & Sons Ltd, 2004. 0-470-86075-8.

8. *A fuzzy method for automatic generation of membership function using fuzzy relations from training examples.* **Lucero, Yvonne C. and Nava, Patricia A.** El Paso : University of Texas at El Paso, 2002. 0-7803-7461-4.

9. *Fuzzy Sets.* **Zadeh, L. A.** Berkeley : University of California, 1965, Information and Control, Vol. 8, pp. 338 - 353.

10. **The MathWorks.** Fuzzy Logic Toolbox for Use with MATLAB. *Fuzzy Logic Toolbox User's Guide.* 2002.

11. *Fuzzy membership function in determining statistical process control position.* **Nababan, E.B., et al.** s.l. : IEEE, 2005. Engineering Management Conference. Vol. 3, p. 1066. 0-7803-8519-5 .

12. **John, R.** Fuzzy Logic and Knowledge Based Systems. 2006.

13. *ANFIS: adaptive-network-based fuzzy inference system.* **Jang, J.S.R.** 3, s.l. : IEEE, June 1993, Systems, Man and Cybernetics, Vol. 23, pp. 665 - 685. 0018-9472.

14. *Neuro-Fuzzy and Soft Computing-A Computational Approach to Learning and Machine Intelligence [Book Review].* **Jang, J.S.R., Sun, C.T. and Mizutani, E.** 10, s.l. : IEEE, 1997, IEEE Transactions on Automatic Control, Vol. 42, pp. 1482-1484. 0018-9286.

15. *A Matlab Toolbox for Real-Time and Control Systems Co-Design.* **Eker, Johan and Cervin, Anton.** Hong Kong : IEEE, 1999. Sixth International Conference on Real-Time Computing Systems and Applications. pp. 320 - 327. 0-7695-0306-3 .

16. **The MathWorks.** Help Guide. *MathWorks.* [Online] MathWorks. [Cited: 9th December 2010.] http://www.mathworks.com/help/toolbox/fuzzy/fp72.html.

**APPENDICES**

**Appendix A:** A table related to annotations used to exemplify classical set theory notation in Chapter 2: *Literature Review*.

**Table A: Annotations related to classical set theory**

| Set Notation | Pronunciation | Explanation |
|---|---|---|
| x ∈ A. | object 'x' is a member of set 'A' | object 'x' is a member of set 'A' |
| Ø | "null set" | A set with no objects in it |
| A ∪ B | "A union B" | All members of both sets A & B |
| A ∩ B | "A intersect B" | only the things that are in both of the sets |
| A\B, or<br>A - B | "A complement B", or "A minus B" | Everything in A that is not a member of B also |
| ~ (A ∪ B) | "not (A union B)" | Everything outside A & B |
| ~ (A ∩ B) | "not (A intersect B)" | Everything outside the overlap of objects relative to sets A & B |

**Appendix B:** A table containing the data set procured from Glanbia's MES system.

**Table B: Operating data listing density input, feed flow, thermo pressure, density output, density setpoint and setpoint difference.**

| Density Input | Feed Flow | Thermo Pressure | Density Output | Set Point | Setpoint Difference |
|---|---|---|---|---|---|
| 1086 | 28 | 9.5 | 1187 | 1230 | -43 |
| 1110 | 28 | 9.3 | 1230 | 1230 | 0 |
| 1110 | 28.5 | 9.3 | 1221 | 1230 | -9 |
| 1112 | 28 | 9.3 | 1230 | 1230 | 0 |
| 1113 | 29 | 9 | 1207 | 1230 | -23 |
| 1114 | 29 | 8.8 | 1205 | 1230 | -25 |
| 1114 | 29 | 9 | 1207 | 1230 | -23 |
| 1114 | 28 | 9.3 | 1230 | 1230 | 0 |
| 1115 | 28 | 9 | 1215 | 1230 | -15 |
| 1115 | 31 | 8.5 | 1189 | 1230 | -41 |
| 1116 | 28.5 | 9.3 | 1230 | 1230 | 0 |
| 1116 | 28 | 8.7 | 1211 | 1230 | -19 |
| 1117 | 29 | 9 | 1216 | 1230 | -14 |
| 1117 | 28 | 8.8 | 1227 | 1230 | -3 |
| 1118 | 28.5 | 9.3 | 1230 | 1230 | 0 |
| 1118 | 29 | 9 | 1214 | 1230 | -16 |
| 1118 | 29 | 9.1 | 1215 | 1230 | -15 |
| 1118 | 29 | 9.5 | 1221 | 1230 | -9 |
| 1119 | 30 | 8.52 | 1197 | 1230 | -33 |
| 1120 | 28.5 | 9.3 | 1230 | 1230 | 0 |
| 1122 | 29 | 9.5 | 1246 | 1230 | 16 |
| 1122 | 29 | 9.2 | 1230 | 1230 | 0 |
| 1123 | 29 | 8.8 | 1234 | 1230 | 4 |
| 1124 | 29 | 9.1 | 1230 | 1230 | 0 |
| 1125 | 30 | 8.7 | 1225 | 1230 | -5 |
| 1126 | 28 | 8.5 | 1227 | 1230 | -3 |
| 1126 | 28 | 8.7 | 1228 | 1230 | -2 |
| 1126 | 28 | 8.7 | 1230 | 1230 | 0 |
| 1126 | 29.5 | 9 | 1230 | 1230 | 0 |
| 1126 | 30 | 8.5 | 1212 | 1230 | -18 |
| 1127 | 28 | 8.7 | 1234 | 1230 | 4 |
| 1127 | 28 | 8.7 | 1233 | 1230 | 3 |
| 1127 | 28 | 8.7 | 1228 | 1230 | -2 |
| 1127 | 29 | 8.7 | 1220 | 1230 | -10 |
| 1127 | 30 | 8.5 | 1223 | 1230 | -7 |
| 1128 | 28 | 8.5 | 1233 | 1230 | 3 |
| 1128 | 28 | 8.7 | 1232 | 1230 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| 1128 | 28 | 8.6 | 1232 | 1230 | 2 |
| 1128 | 29.5 | 8.9 | 1230 | 1230 | 0 |
| 1128 | 30 | 8.23 | 1218 | 1230 | -12 |
| 1129 | 28 | 8.5 | 1234 | 1230 | 4 |
| 1129 | 28 | 8.7 | 1235 | 1230 | 5 |
| 1129 | 28 | 8.6 | 1232 | 1230 | 2 |
| 1129 | 30 | 8.23 | 1219 | 1230 | -11 |
| 1129 | 31 | 8.5 | 1220 | 1230 | -10 |
| 1129 | 30 | 8.6 | 1230 | 1230 | 0 |
| 1130 | 29 | 8.35 | 1225 | 1230 | -5 |
| 1130 | 30 | 8.4 | 1227 | 1230 | -3 |
| 1130 | 30 | 8.7 | 1231 | 1230 | 1 |
| 1130 | 30 | 9 | 1240 | 1230 | 10 |
| 1130 | 30 | 8.1 | 1221 | 1230 | -9 |
| 1130 | 30 | 8.2 | 1223 | 1230 | -7 |
| 1130 | 30 | 8.5 | 1227 | 1230 | -3 |
| 1130 | 30 | 8.7 | 1230 | 1230 | 0 |
| 1131 | 29 | 8.3 | 1230 | 1230 | 0 |
| 1131 | 29 | 8.35 | 1223 | 1230 | -7 |
| 1131 | 30 | 8.5 | 1230 | 1230 | 0 |
| 1132 | 29 | 8.35 | 1228 | 1230 | -2 |
| 1132 | 30 | 8.2 | 1224 | 1230 | -6 |
| 1132 | 30 | 8.6 | 1230 | 1230 | 0 |
| 1133 | 30 | 8.2 | 1227 | 1230 | -3 |
| 1133 | 28 | 9 | 1252 | 1230 | 22 |
| 1133 | 31 | 8.5 | 1225 | 1230 | -5 |
| 1133 | 31 | 8.5 | 1225 | 1230 | -5 |
| 1133 | 31 | 8.5 | 1227 | 1230 | -3 |
| 1133 | 31 | 8.5 | 1228 | 1230 | -2 |
| 1133 | 31 | 8.5 | 1226 | 1230 | -4 |
| 1133 | 30 | 8.7 | 1238 | 1230 | 8 |
| 1134 | 29 | 8.35 | 1239 | 1230 | 9 |
| 1134 | 30 | 8.5 | 1230 | 1230 | 0 |
| 1134 | 30 | 8.2 | 1227 | 1230 | -3 |
| 1135 | 30 | 8.5 | 1237 | 1230 | 7 |
| 1135 | 28 | 8.8 | 1250 | 1230 | 20 |
| 1136 | 29 | 8.6 | 1242 | 1230 | 12 |
| 1136 | 28 | 8.8 | 1252 | 1230 | 22 |
| 1136 | 30 | 8.4 | 1230 | 1230 | 0 |
| 1137 | 29 | 8.4 | 1241 | 1230 | 11 |
| 1137 | 30 | 8.3 | 1220 | 1230 | -10 |
| 1137 | 30 | 8.1 | 1232 | 1230 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| 1137 | 30 | 8.7 | 1244 | 1230 | 14 |
| 1137 | 30 | 8.5 | 1242 | 1230 | 12 |
| 1137 | 30 | 8.5 | 1238 | 1230 | 8 |
| 1138 | 28 | 8.7 | 1247 | 1230 | 17 |
| 1138 | 30 | 8.5 | 1239 | 1230 | 9 |
| 1138 | 30.5 | 8.3 | 1230 | 1230 | 0 |
| 1139 | 28 | 8.8 | 1256 | 1230 | 26 |
| 1140 | 30.5 | 8.2 | 1230 | 1230 | 0 |
| 1142 | 30 | 8.5 | 1241 | 1230 | 11 |
| 1142 | 31 | 8.1 | 1230 | 1230 | 0 |
| 1142 | 30 | 8.3 | 1238 | 1230 | 8 |
| 1143 | 30 | 8.1 | 1223 | 1230 | -7 |
| 1144 | 31 | 8 | 1230 | 1230 | 0 |
| 1146 | 31 | 7.9 | 1230 | 1230 | 0 |
| 1163 | 30 | 8.6 | 1252 | 1230 | 22 |

**Appendix C:** A table listing the controller output for feed flow and thermo-pressure relative to an increasing density input.

**Table C: Results obtained from the designed fuzzy system.**

| Density Input | Thermo-Pressure | Setpoint | Feed Flow | Setpoint |
|---|---|---|---|---|
| $kg/m^3$ | bar | bar | $m^3/hr$ | $m^3/hr$ |
| 1080 | 9.3 | 9.3 | 28 | 28 |
| 1081 | 9.3 | 9.3 | 28 | 28 |
| 1082 | 9.3 | 9.3 | 28 | 28 |
| 1083 | 9.3 | 9.3 | 28 | 28 |
| 1084 | 9.3 | 9.3 | 28 | 28 |
| 1085 | 9.3 | 9.3 | 28 | 28 |
| 1086 | 9.3 | 9.3 | 28 | 28 |
| 1087 | 9.3 | 9.3 | 28 | 28 |
| 1088 | 9.3 | 9.3 | 28 | 28 |
| 1089 | 9.3 | 9.3 | 28 | 28 |
| 1090 | 9.3 | 9.3 | 28 | 28 |
| 1091 | 9.3 | 9.3 | 28 | 28 |
| 1092 | 9.3 | 9.3 | 28 | 28 |
| 1093 | 9.3 | 9.3 | 28 | 28 |
| 1094 | 9.3 | 9.3 | 28 | 28 |
| 1095 | 9.3 | 9.3 | 28 | 28 |
| 1096 | 9.3 | 9.3 | 28 | 28 |
| 1097 | 9.3 | 9.3 | 28 | 28 |
| 1098 | 9.3 | 9.3 | 28 | 28 |
| 1099 | 9.3 | 9.3 | 28 | 28 |
| 1100 | 9.3 | 9.3 | 28 | 28 |
| 1101 | 9.3 | 9.3 | 28 | 28 |
| 1102 | 9.3 | 9.3 | 28 | 28 |
| 1103 | 9.3 | 9.3 | 28 | 28 |
| 1104 | 9.3 | 9.3 | 28 | 28 |
| 1105 | 9.3 | 9.3 | 28 | 28 |
| 1106 | 9.3 | 9.3 | 28 | 28 |
| 1107 | 9.3 | 9.3 | 28 | 28 |
| 1108 | 9.3 | 9.3 | 28 | 28 |
| 1109 | 9.3 | 9.3 | 28 | 28 |
| 1110 | 9.3 | 9.3 | 28 | 28 |
| 1111 | 9.3 | 9.3 | 28 | 28 |
| 1112 | 9.3 | 9.3 | 28 | 28 |
| 1113 | 9.3 | 9.3 | 28.2504 | 28.25 |
| 1114 | 9.3 | 9.3 | 28.2504 | 28.25 |

| | | | | |
|---|---|---|---|---|
| 1115 | 9.3 | 9.3 | 28.375 | 28.25 |
| 1116 | 9.3 | 9.3 | 28.4996 | 28.5 |
| 1117 | 9.3 | 9.3 | 28.4996 | 28.5 |
| 1118 | 9.3 | 9.3 | 28.4996 | 28.5 |
| 1119 | 9.3 | 9.3 | 28.6252 | 28.75 |
| 1120 | 9.3 | 9.3 | 28.75 | 28.75 |
| 1121 | 9.1995 | 9.2 | 28.8748 | 29 |
| 1122 | 9.1997 | 9.2 | 29.0004 | 29 |
| 1123 | 9.15 | 9.1 | 29.125 | 29.25 |
| 1124 | 9.1003 | 9.1 | 29.2496 | 29.25 |
| 1125 | 9.0502 | 9.1 | 29.4053 | 29.5 |
| 1126 | 9 | 9 | 29.5662 | 29.5 |
| 1127 | 8.9498 | 9 | 29.6557 | 29.5 |
| 1128 | 8.8997 | 8.85 | 29.7504 | 29.75 |
| 1129 | 8.7997 | 8.85 | 29.875 | 29.75 |
| 1130 | 8.7 | 8.7 | 29.9996 | 30 |
| 1131 | 8.7 | 8.7 | 29.9996 | 30 |
| 1132 | 8.55 | 8.6 | 29.9996 | 30 |
| 1133 | 8.5005 | 8.5 | 29.9996 | 30 |
| 1134 | 8.5003 | 8.5 | 29.9996 | 30 |
| 1135 | 8.4502 | 8.4 | 30.1419 | 30.25 |
| 1136 | 8.4 | 8.4 | 30.25 | 30.25 |
| 1137 | 8.3498 | 8.3 | 30.3748 | 30.5 |
| 1138 | 8.2997 | 8.3 | 30.5004 | 30.5 |
| 1139 | 8.25 | 8.2 | 30.625 | 30.75 |
| 1140 | 8.2003 | 8.2 | 30.7496 | 30.75 |
| 1141 | 8.1502 | 8.2 | 30.7496 | 30.75 |
| 1142 | 8.1 | 8.1 | 31 | 31 |
| 1143 | 8.0498 | 8 | 31 | 31 |
| 1144 | 7.9997 | 8 | 31 | 31 |
| 1145 | 7.95 | 8 | 31 | 31 |
| 1146 | 7.9003 | 7.9 | 31 | 31 |
| 1147 | 7.9005 | 7.9 | 31 | 31 |
| 1148 | 7.8 | 7.8 | 31 | 31 |
| 1149 | 7.8 | 7.8 | 31 | 31 |
| 1150 | 7.8 | 7.8 | 31 | 31 |
| 1151 | 7.8 | 7.8 | 31 | 31 |
| 1152 | 7.8 | 7.8 | 31 | 31 |
| 1153 | 7.8 | 7.8 | 31 | 31 |
| 1154 | 7.8 | 7.8 | 31 | 31 |
| 1155 | 7.8 | 7.8 | 31 | 31 |
| 1156 | 7.8 | 7.8 | 31 | 31 |

| | | | | |
|---|---|---|---|---|
| 1157 | 7.8 | 7.8 | 31 | 31 |
| 1158 | 7.8 | 7.8 | 31 | 31 |
| 1159 | 7.8 | 7.8 | 31 | 31 |
| 1160 | 7.8 | 7.8 | 31 | 31 |
| 1161 | 7.8 | 7.8 | 31 | 31 |
| 1162 | 7.8 | 7.8 | 31 | 31 |
| 1163 | 7.8 | 7.8 | 31 | 31 |
| 1164 | 7.8 | 7.8 | 31 | 31 |
| 1165 | 7.8 | 7.8 | 31 | 31 |
| 1166 | 7.8 | 7.8 | 31 | 31 |
| 1167 | 7.8 | 7.8 | 31 | 31 |
| 1168 | 7.8 | 7.8 | 31 | 31 |
| 1169 | 7.8 | 7.8 | 31 | 31 |
| 1170 | 7.8 | 7.8 | 31 | 31 |
| 1171 | 7.8 | 7.8 | 31 | 31 |
| 1172 | 7.8 | 7.8 | 31 | 31 |
| 1173 | 7.8 | 7.8 | 31 | 31 |
| 1174 | 7.8 | 7.8 | 31 | 31 |
| 1175 | 7.8 | 7.8 | 31 | 31 |
| 1176 | 7.8 | 7.8 | 31 | 31 |
| 1177 | 7.8 | 7.8 | 31 | 31 |
| 1178 | 7.8 | 7.8 | 31 | 31 |
| 1179 | 7.8 | 7.8 | 31 | 31 |
| 1180 | 7.8 | 7.8 | 31 | 31 |