

POLITECNICO DI TORINO

Collegio di Ingegneria Chimica e dei Materiali

**Corso di Laurea Magistrale
in Ingegneria Chimica e dei Processi Sostenibili
/dei Materiali**

Master Thesis

ANFIS modelling of PIV tests



Supervisors

Prof. Bernardo Ruggeri
Ing. Carlos Enrique Gomez Camacho

Candidate

Leonardo Clemente

October 2019

Table of contents

1. Introduction	1
1.1 Objectives	2
1.2 Thesis Layout.....	3
2. Theoretical framework	5
2.1 Particle Image Velocimetry	5
2.2 Non – Deterministic approaches for experimental data modelling	6
2.2.1 Fuzzy Logic Models.....	6
2.2.2 ANN: Artificial Neural Networks	12
2.2.3 ANFIS: Adaptive Neuro – Fuzzy Inference System.....	17
2.2.3.1 Grid Partitioning	20
2.2.3.2 Subtractive clustering	21
2.2.3.3 ANFIS Hybrid learning method	22
2.3 Correlation Indexes.....	23
2.3.1 RRMSE: Relative Root Mean Square Error	23
2.3.2 R ² : Coefficient of Determination	24
2.3.3 IA: Index of Agreement	24
2.3.4 GI: Goodness Index.....	25
3. Experimental methods.....	29
3.1 Particle Image Velocimetry Experimental Data	29
3.2 ANFIS MatLab® Toolbox.....	32
4. Velocity Field Modelling	37
4.1 Comparison between Fuzzy and Neuro – Fuzzy Modelling with big data.....	37
4.2 Comparison on performance between different types of MF	38
4.3 Variation of quantity of training data	42
4.4 Alternative approach for modelling.....	46
4.5 Analysis on correlation and computational performances with variation on antecedent membership functions number.....	52

4.6 Investigation on applicability of ANFIS modelling on BiLoop impeller velocity field	57
4.7 Contour plots processing in verification of obtained model correlation power	59
5. Conclusions and Future Outlook	65
Bibliography.....	70
List of symbols.....	72

List of figures

Figure 2.1 – Main components of a Particle Image Velocimetry measurement and analysis apparatus. _____	5
Figure 2.2 - Comparison between Fuzzy Set and Classical Crisp Set _____	7
Figure 2.3 – OR connective between two antecedents in a single rule, performing a Maximum operation (adapted from “Fuzzy logic with engineering applications”, Ross, 2010) _____	8
Figure 2.4 – AND operation as Minimum, for a single rule in a Mamdani FIS (adapted from “Fuzzy logic with engineering applications”, Ross, 2010) _____	8
Figure 2.5 – Example showing a two rules Mamdani FIS functioning, comprehending rules connective operation, rules aggregation and Centroid Defuzzification (adapted from “Fuzzy logic with engineering applications”, Ross, 2010) _____	10
Figure 2.6 – Example showing a two rules Sugeno FIS functioning, comprehending weights obtainment and Weighted Average (adapted from “Fuzzy logic with engineering applications”, Ross, 2010) _____	11
Figure 2.7 – Scheme of Fuzzy Inference Systems functioning from input insert to output obtainment _____	12
Figure 2.8 – Simplified layout of an Artificial Neural Network, consituted by an input layer, an output layer and a single hidden layer (adapted from “Artificial neural networks are changing the world. What are they?”, Graham, 2015). _____	13
Figure 2.9 – Written numbers (from 1 to 9) recognition model with trained ANN ____	14
Figure 2.10 – Graphical illustration of Gradient Descent Method in minimum search for a Cost function (adapted from “Gradient Descent with Momentum”, 2019) _____	16
Figure 2.11 – Explicative example of an Adaptive Neuro – Fuzzy Inference System structure for a Sugeno type (adapted from “Optimization of EPB Shield Performance with Adaptive Neuro-Fuzzy Inference System and Genetic Algorithm”, K. Elbaz et al., 2019”.) _____	18

Figure 2.12 – Grid Partitioning for an input space composed by two inputs x and y, each one defined by three membership functions (adapted from “Estimating Development Time and Effort of Software Projects by using a Neuro Fuzzy Approach”, V. Marza et al., 2009)	20
Figure 2.13 – Subtractive clustering for an input space composed by two inputs x and y (adapted from “Performance Comparison of ANFIS Models by Input Space Partitioning Methods”, Yeom et al., 2018)	21
Figure 2.14 – Membership functions of Coefficient of determination antecedent	25
Figure 2.15 – Membership functions of Index of Agreement antecedent	25
Figure 2.16 – Membership functions of RRMSE antecedent	25
Figure 2.17 – Membership functions of Goodness Index antecedent	25
Figure 3.1 – Image and scheme of flow pattern for: a) Ring Propeller; b) BiLoop impeller	30
Figure 3.2 - Example of a graph obtained through PIV experimental data	32
Figure 3.3 – Neuro Fuzzy designer MatLab toolbox interface	32
Figure 3.4 – Selection of number and type of membership functions	33
Figure 3.5 – FIS structure generated from Grid Partitioning	34
Figure 3.6 – Training error dynamically changing through data processing	35
Figure 3.7 – Quick comparison between observed and predicted values from produced model	35
Figure 4.1 – Coefficient of determination of models presenting differently shaped membership functions	40
Figure 4.2 – Index of agreement of models presenting differently shaped membership functions	40

Figure 4.3 – RRMSE of models presenting differently shaped membership functions	41
Figure 4.4 – Goodness index of models presenting differently shaped membership functions	41
Figure 4.5 – Coefficient of determination of models with gaussian membership functions, trained with different percentages of data	43
Figure 4.6 – Index of agreement of models with gaussian membership functions, trained with different percentages of data	43
Figure 4.7 – RRMSE of models with gaussian membership functions, trained with different percentages of data	44
Figure 4.8 – Goodness Index of models with gaussian membership functions, trained with different percentages of data	44
Figure 4.9 – Raw model antecedent membership functions: a) Power input; b) Radial coordinate; c) Axial coordinate.	45
Figure 4.10 – Comparison between 5% (left) and 30% (right) trained models in terms of membership function adaptivity: a) Power input; b) Radial Coordinate; c) Axial Coordinate.	46
Figure 4.11 – Passage from 7/7 model (left) to 6/6 model (right) erasing membership functions highlighted in red on the left	47
Figure 4.12 – Coefficient of determination of 6/6 model (left) and 7/7 model (right)	47
Figure 4.13 – Index of Agreement of 6/6 model (left) and 7/7 model (right)	48
Figure 4.14 – RRMSE of 6/6 model (left) and 7/7 model (right)	48
Figure 4.15 – Goodness index of 6/6 model (left) and 7/7 model (right)	49
Figure 4.16 – Comparison between experimental values and two antecedents model predicted values	50
Figure 4.17 – Coefficient of determination of 2 -3 model (left) and 3 model (right)	51

Figure 4.18 – Index of agreement of 2 -3 model (left) and 3 model (right) _____	51
Figure 4.19 – RRMSE of 2 -3 model (left) and 3 model (right) _____	52
Figure 4.20 – Goodness Index of 2 -3 model (left) and 3 model (right) _____	52
Figure 4.21 – Coefficient of determination of models with varying number of antecedents' membership functions _____	53
Figure 4.22 – Index of agreement of models with varying number of antecedents' membership functions _____	53
Figure 4.23 – RRMSE of models with varying number of antecedents' membership functions _____	54
Figure 4.24 – Goodness index of models with varying number of antecedents' membership functions _____	54
Figure 4.25 – Number of rules and output functions of models with varying number of antecedents' membership functions _____	55
Figure 4.26 – Time for each epoch during training of models with varying number of antecedents' membership functions _____	56
Figure 4.27 – Epoch for convergence during training of models with varying number of antecedents' membership functions _____	56
Figure 4.28 – PIV Velocity field for Ring Propeller at 100 W/m ³ _____	57
Figure 4.29 – PIV Velocity field for BiLoop at 100 W/m ³ _____	58
Figure 4.30 – Two antecedents' ANFIS model output as velocity field for BiLoop at 100 W/m ³ _____	58
Figure 4.31 - – Experimental data of PIV measurements with power input equal to 20 W/m ³ for a Ring Propeller _____	59
Figure 4.32 – Output of ANFIS model with power input equal to 20 W/m ³ for a Ring Propeller _____	60

Figure 4.33 – Experimental data of PIV measurements with power input equal to 200 W/m ³ for a Ring Propeller	60
Figure 4.34 – Output of ANFIS model with power input equal to 200 W/m ³ for a Ring Propeller	61
Figure 4.35 – Low mixing zones percentages for experimental data and model output, evaluated at five power inputs	61
Figure 4.36 – High mixing zones percentages for experimental data and model output, evaluated at five power inputs	62
Figure 4.37 – Output of ANFIS model with non -experimented power input equal to 35 W/m ³ for a Ring Propeller	63
Figure 4.38 – Output of ANFIS model with non -experimented power input equal to 350 W/m ³ for a Ring Propeller	64
Figure 5.1 – Membership functions for power input antecedent of best obtained model	65
Figure 5.2 – Membership functions for dimensionless radial coordinate antecedent of best obtained model	66
Figure 5.3 – Membership functions for dimensionless axial coordinate antecedent of best obtained model	66

List of Tables

Table 2.1 – Synthesis of the procedure of ANFIS hybrid training method _____	22
Table 2.2 – Rules composing inference system in Fuzzy Goodness Index Model ____	27
Table 3.1 – Available power input values regarding vessel filled with pure water, agitator equipped with a ring propeller, with a 2D camera disposition. _____	30
Table 3.2 – Example of data set given to the ANFIS training algorithm _____	31
Table 4.1 – Adopted conditions in a comparison between Fuzzy and Neuro – Fuzzy modelling performances _____	37
Table 4.2 – Examined membership functions’ shapes _____	39
Table 4.3 – Correlation indexes regarding Low and High mixing zones projection through ANFIS model _____	62
Table 4.4 – Low and high mixing zones percentages for two non – experimented power input values _____	63
Table 5.1 – Sample of randomly picked 10 rules from a total of 180 _____	67

1. Introduction

Nowadays, many improvements regarding measurement methods and analysis tools are being made every day to provide more accurate ways to evaluate all sorts of interesting parameters. This has to be done in order to achieve a better comprehension of certain phenomena, observed during researches regarding technological studies. In this sense, big data are being used every time more frequently to have an extended and thorough evaluation in quantitative terms. This is due to the fact that by scanning such huge amount of information, hidden patterns can be revealed (Sagiroglu & Sinanc, 2013). However, it is not sufficient, in most cases, to acquire massive quantities of experimental values. As a matter of fact, once these are obtained from scientific observations, they could be characterised in certain ways or, in other words, they could be modelled. This is done with the purpose of giving qualitative and quantitative conclusions regarding observed phenomena. Said so, several genres of machine learning turn out to be very useful when it comes to give a projection and a logical explanation to certain non – linear problems, such as the one which will be encountered in this study. This, in simpler terms, means having the ability to fully understand the behaviour of observed phenomena (Patterson & Gibson, 2008). In addition to this, the availability of a mathematical model allows, in most cases, to obtain unseen values for parameters of interest through extrapolation.

In this work, it is desired to verify the effectiveness of certain modern non – deterministic approaches towards data modelling, in their manner of verbally describing given parameters behaviour depending on given variables. In this case, certain modelling procedures will be described and adopted in order to characterise velocity fields observed for agitated vessels. Such velocity fields are characterised by several parameters. Therefore, their description through conventional modelling would result very difficult and not accurate. The importance of giving as precise as possible details about such variables stands on the need for knowledge of a very unpredictable thing: Life. As a matter of fact, microorganisms are very difficult to fully understand in their interactions with surrounding environment. This gives birth to the necessity of making experiments which could possibly give, through mathematical modelling, a representation of their behaviour in the most concise way. Non – conventional methods adoption in this case results necessary, given that great quantities of datasets, exhibiting multiple non – dependent variables, have to be manipulated. In the present work, a case of dealing with quantitative information, given by a modern experimental method called PIV: Particle Image Velocimetry. In the next chapter this topic will be briefly illustrated in order to have enough notion to understand what kind of data sets are being modelled in this work. The specific machine learning tool which was mentioned before is known by its acronym

ANFIS: Adaptive Neuro Fuzzy Inference System, which is getting very popular nowadays in control systems and support to decision making in multiple applications and fields.

1.1 Objectives

A mathematical model correlating certain sets of data has to be as precise as possible, guaranteeing a high degree of accuracy in representing employed data. On the other hand, it is also advised to keep it simple, in order to let any user being able to have as easy as possible interpretation of modelled data and model itself. These aims can be reached by means of ANFIS. In fact, this type of modelling will be used in this study, in order to show its effectiveness in manipulating a great mole of different data containing multiple variables, maintaining at the same time a certain degree of simplicity and conciseness in its representation. Further details and explanations about such hybrid logical systems will be given in the next chapter.

First of all, a detailed investigation will be done on ANFIS tool itself in order to construct a solid model on previously mention datasets. This means following different procedures to have a better comprehension of the influence on the modelling performances of such inference systems from the various parameters contained in them. It will be done performing a sensitivity analysis, fully described by model fitting indexes. In this case, such parameters consist of R^2 (Correlation index), IA (Index of Agreement), RRMSE (Relative Root Mean Square Error) and finally, a “Fuzzy” combination of these three indicators, created on the spot, which will be named after GI, “Goodness Index”.

Secondly, after having achieved a satisfyingly robust model, many considerations will be made towards the algorithms involved in such machine learning systems, characterising them not only on the produced model starting from a raw one, but also in terms of computational efficiency. This has the purpose of characterising this machine learning method in processing speed terms, relative to used computer processor characteristics.

Moreover, ANFIS modelling will be tested with two different types of flows, produced by different impellers, in order to see if such models are capable of describing different flow patterns.

Final aim of this work consists in characterising a certain model through its correlation characteristics, but with image processing of certain “Low mixing zones” and “High mixing zones”. This has the purpose of testing the created model on non – experimentally observed power inputs, to see if it is able to produce consistent output even on new values of inputs dominion.

1.2 Thesis Layout

After this introduction, Chapter 2 will be employed in order to give theoretical explanation about Particle Image Velocimetry, Fuzzy Logic, Neural Networks and their hybrid union Neuro – Fuzzy. An overview of involved fitting indexes will be inserted, so to understand their usefulness and general characteristics. Moreover, in such chapter a newly generated Fuzzy index will be presented.

In Chapter 3, experimental methods regarding this study will be thoroughly exposed. First, a basic explanation of particle image velocimetry experimental technique will be given, just with the purpose of understanding the apparatus which concerns the foreseen data. Then, the programming frame regarding Neuro – Fuzzy training implementation will be shown, defining its main functionalities and setting certain constraints on operative conditions used through this thesis paper.

In Chapter 4, a sensitivity study regarding neuro fuzzy modelling of PIV raw data will be shown, followed by its necessary considerations. Of course, such data sets will be contextualised starting from their origin where they come from, explaining the examinations made in terms of observed fluids and other varied parameters. Different membership functions contained in produced inference systems will be investigated in order to see how they perform in terms of precision, convergence speed and simplicity. Moreover, a simple comparison between Fuzzy and Neuro Fuzzy modelling of presented data will be shown, in order to motivate the need for using the second one over the first one, proving machine learning ability of the hybrid method as superior. Moreover, different observations will be done in order to increase the usefulness of the fulfilled model, in terms of predictability towards agitated vessels study. In particular, the aim will be to obtain a model capable of giving velocity vector norm as outcome, given a certain position in terms of radial and axial coordinate of a cylindrical vessel and even a power input value measurable as W/m^3 . Finally, a paragraph will be dedicated to the investigation of ANFIS capability of modelling other types of flow patterns, different from the ones modelled in the entire thesis.

Then, Chapter 5 will be used as Conclusion and Future Outlook section, general deductions coming from the analysis made through this study will be resumed. Successively, the best obtained model will be presented, enunciating its advantages and drawbacks. Considerations will be made around possible developments of such ANFIS model, in order to make it useful for future technological applications.

2. Theoretical framework

2.1 Particle Image Velocimetry

This section has the purpose of showing, even though quite briefly, the main characteristics, as well as the background, concerning the technology which allowed to obtain the velocity field values which will be modelled. Let's start saying that this experimental approach gives the possibility to have velocity values in high quantity of positions, in a certain system where a fluid flows by means of a driving force, for instance an agitator moving a liquid in a vessel (Prasad, 2000). This gained popularity to visualise local liquid motion in a non – invasive way. In Figure 2.1 the main tools constituting a generic PIV apparatus is shown.

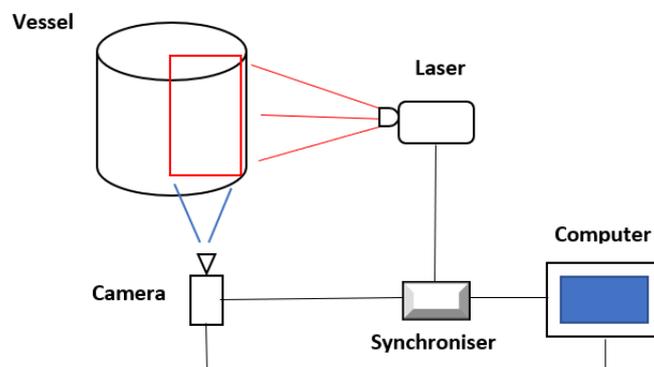


Figure 2.1 – Main components of a Particle Image Velocimetry measurement and analysis apparatus.

Basically, a particular specimen of solid particles called “Seeding Particles” are inserted in a fluid, contained in a stirred vessel. They necessarily have to present a density very similar to the fluid which has to be examined. Then, a laser illuminates a specific section of the vessel, with two consecutive impulses. Particles reflect such intermittent laser light, which is captured by a camera, through a synchronizer. This procedure allows to obtain two images of these particles in two instants, very close to each other. At the end of the procedure, instantaneous velocity values will be acquired in certain positions, given that they will be more accurate if the time difference between the two moments of acquisition is smaller. Of course, this ensemble has to be attached to a computer, capable of processing this information with a suitable software. However, there are some necessary conditions for this method to work appropriately, and these obviously lead to some limitations. The most important of these certainly consists in the fact that, since there must be optical accessibility, it is impossible to load the explored medium up to a high concentration of solids. As a matter of fact, solids and even bubbles would obscure

image capture (Li et al., 2018). This does not give the chance to have a PIV experimental representation of an aerated stirred vessel. Nonetheless, it is still useful to make considerations about fluid dynamics in some systems, using simple calculations between observed points. In some cases, PIV was used to verify theoretical results coming from computational fluid dynamics simulations, even of agitated tanks (Sheng, Meng, & Fox, 1998). In Chapter 3, further details about the specific case of stirred vessel experimental observations will be given.

2.2 Non – deterministic approaches for experimental data modelling

In this section, there will be a resume of the basic rules and functioning of Fuzzy Logic, Neural Networks and finally, most important, Adaptive Neuro Fuzzy Inference Systems.

2.2.1 Fuzzy Logic Models

In the early sixties, Lofti Zadeh, professor at Berkeley University, known for its contributions to Systems Theory, began to feel that traditional techniques for system analysis were excessively and uselessly accurate for many typical problems of the real world. The idea of “Membership Degree”, fundamental concept of fuzzy sets theory, was introduced by him in 1964. Successively, in 1965 Fuzzy logic was born officially, as a topic exposed in an article publication. From then, initially criticised, this logic became more and more popular and became object of many studies, getting the form which is known nowadays. Fuzzy logic is a valid tool when it comes to represent not completely well - defined relationships in non – linear systems, being therefore very common in control systems implementation. It is even found as mathematical transcription of expertise knowledge (Ii & Ground, 1998). Building correlations from experimental raw datasets means obtaining various types of mathematical functions. Usually, developing such kind of correlations immediately gives the perception of dealing with something very easy to use. However, they often turn out to be not so labile and not very representative of the pictured system (Ii & Ground, 1998).

Fuzzy Logic is a logical system related to the theory of fuzzy sets. These differ from their counterpart, crisp sets, as they have unsharp boundaries (J.-S. R. Jang & Gulley, 2015). This is one of the main characteristics of these sets, as they can “Partially” contain an object. In other words, a certain predicate can be owned by a fuzzy set with a certain *Membership Degree*, defined by an appropriate Membership function. Membership functions can be derived from many different shapes, such as triangular, trapezoidal, and many more. They will be exposed more in detail in next chapters. In Figure 2.2 a fuzzy set (membership function) is pictured along with a classical crisp set. The x – axis is where a certain parameter is entered, then membership function μ returns a membership degree to it. Such degree varies from 0 to 1, depending on respective crisp value. Instead,

classical sets only allow full membership (equal to 1) or no membership at all (equal to 0).

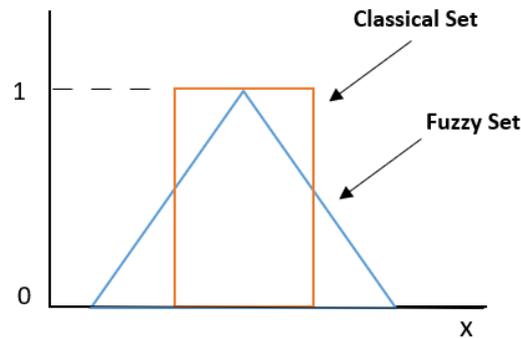


Figure 2.2 - Comparison between Fuzzy Set and Classical Crisp Set

This is the reason why they are eventually less “Adaptive” compared to their fuzzy relatives. Fuzzy inference stands on a rule – based system in order to express relationships between antecedents (inputs) and consequents (outputs). These rules are structured as IF – THEN propositions, where single or multiple inputs are combined in certain manners from the IF part of the rule, being transformed in a certain output by a THEN logical statement. An example of fuzzy rules mapping antecedents to consequents is shown. In order to keep it simple, a single input – single output rule is presented.

IF x is A THEN y is B

In this case, x is a crisp input, being located in the fuzzy universe of discourse by means of a generic fuzzy set A, with a certain membership degree. This is then converted in a fuzzy output through another fuzzy set B, characterising the output y with a membership degree. After this passage, the fuzzy output is defuzzified in order to acquire a crisp result at the exit of the system in use. In other terms, *Defuzzification* allows to convert the output from fuzzy to crisp, so to have a solution in its original measurement units. Even though there is more than one procedure of Defuzzification, in this study only one of them will be defined: Centroid method (Mamdani FIS). Multiple inputs are usually implemented to define a single output. The description of how a single rule produces an interaction among multiple inputs is more complex and needs a deeper explanation. First, it depends on how inputs are related with each other. In fact, so called *Connectives* play an important role during the inference process. These can be either in form of “and”, “or” or “else”. Different operations can be performed by these three, depending on the user preferences. For OR connective, the most common operation made is Union between antecedents, which means taking the maximum membership degree among

antecedents contained in a rule to transfer it to the output. For example, with two antecedents x and y and one consequent z , it would be:

IF x is A OR y is B THEN z is C

In Figure 2.3 there is a scheme of what happens in the enunciated rule above, given that A, B and C are triangular membership function.

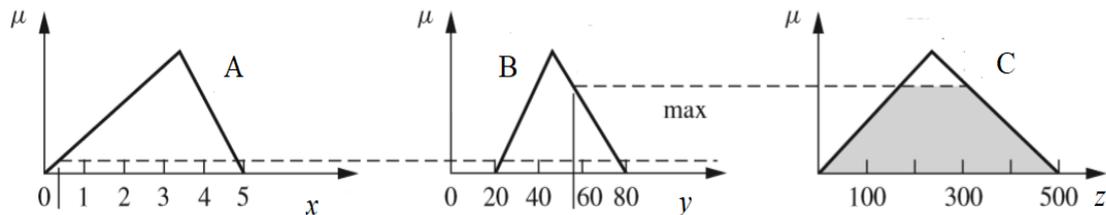


Figure 2.3 – OR connective between two antecedents in a single rule, performing a Maximum operation (adapted from “Fuzzy logic with engineering applications”, Ross, 2010)

The antecedents x and y , along with the consequent z , have each one its own crisp range, mapped into a fuzzy set. The AND operator works in a very similar way, but it is usually found in two shapes. The first one, commonly adopted in Mamdani FIS is the Minimum operation, while in Sugeno FIS a Product operation occurs. This means that having a rule of this kind:

IF x is A AND y is B THEN z is C

Typical of a Mamdani FIS, it would be graphically translated as in Figure 2.4. Minimum operator could be used in Sugeno – type inference system as well. However, in such FIS the product operator is more popular, especially when it comes to use a fuzzy system produced by a neuro fuzzy training algorithm. Substantially, it consists in operating a product between membership degrees of the connected antecedents of a rule.

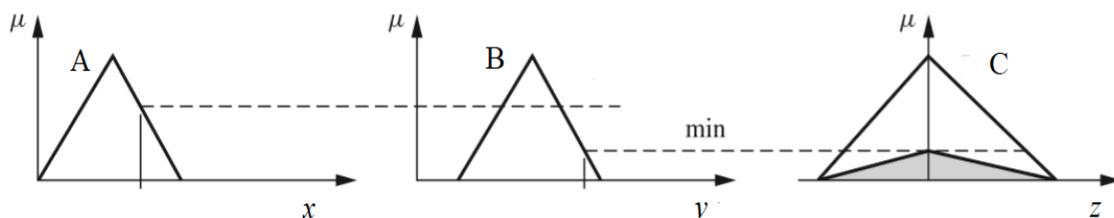


Figure 2.4 – AND operation as Minimum, for a single rule in a Mamdani FIS (adapted from “Fuzzy logic with engineering applications”, Ross, 2010)

Up to now, Mamdani and Sugeno Fuzzy Inference Systems (FIS) were mentioned. They show some similarities, but also some differences. Input fuzzification and the application of the fuzzy operators are applied for both (J.-S. R. Jang & Gulley, 2015). When it comes

to dealing with the output, or consequent, there are critical differences which confer certain peculiarities to one respect to the other. Mamdani still has a fuzzified output, which is necessarily followed by a Defuzzification method. In Sugeno FIS, instead, there is a crisp output coming out from each rule, represented as a pre – specified degree polynomial function of the inputs. An example of Sugeno FIS rule is shown below just for clarity.

IF x is A AND y is B THEN z is f(x,y)

Most used functions in such inference systems are zero – degree (constant) and first – degree (linear) polynomials. Using crisp outputs partially eliminates consumption of computational processing time involved in defuzzification methods, applied in Mamdani FIS (Ross, 2010). In fact, Sugeno FIS final solution is obtained via *Weighted Average Method*. This procedure allows to combine more rules to create an aggregated value for the output. In fact, a typical FIS of any kind is composed by several rules and each one of them produces an output, fuzzified or not (depending on the kind of FIS), which is compared to the others. Rules can get combined with each other in different ways. Basically, it depends on the type of inference system one is dealing with. Let’s see this first for Mamdani method. Let’s imagine a simple FIS, constituted by only two rules:

Rule 1 → IF x is A₁ AND y is B₁ THEN z is C₁

Rule 2 → IF x is A₂ OR y is B₂ THEN z is C₂

Then, let’s say that AND connective operates as a minimum in the first rule, while OR connective operates as a maximum in the second rule (as usual for a Mamdani FIS). In mathematical terms, it would be (Equations 2.1, 2.2):

$$\mu_{C_1} = \min (\mu_{A_1}, \mu_{B_1}) \quad (2.1)$$

$$\mu_{C_2} = \max (\mu_{A_2}, \mu_{B_2}) \quad (2.2)$$

Now, since usually it is Disjunctive system of Rules the kind which is most dealt with, a Union operator is applied between all individual rule outcomes. Finally, obtained rules aggregation is Defuzzified with previously mentioned Centroid Method. This is given by Equation 2.3 and graphically consists in finding the centre of gravity of the created figure.

$$z^* = \frac{\int \mu(z) \cdot z \, dz}{\int \mu(z) \, dz} \quad (2.3)$$

For clarity, followed procedure is illustrated graphically in Figure 2.5. Such procedure works for Mamdani FIS. There is interested also on Sugeno FIS, since it will be used to

develop several ANFIS models in the next chapters. A similar process of input insertion and output crisp value obtainment is going to be described for a Sugeno FIS. Let's suppose having a first – order Sugeno FIS, this means obtaining from each rule a first – order polynomial function of the inputs.

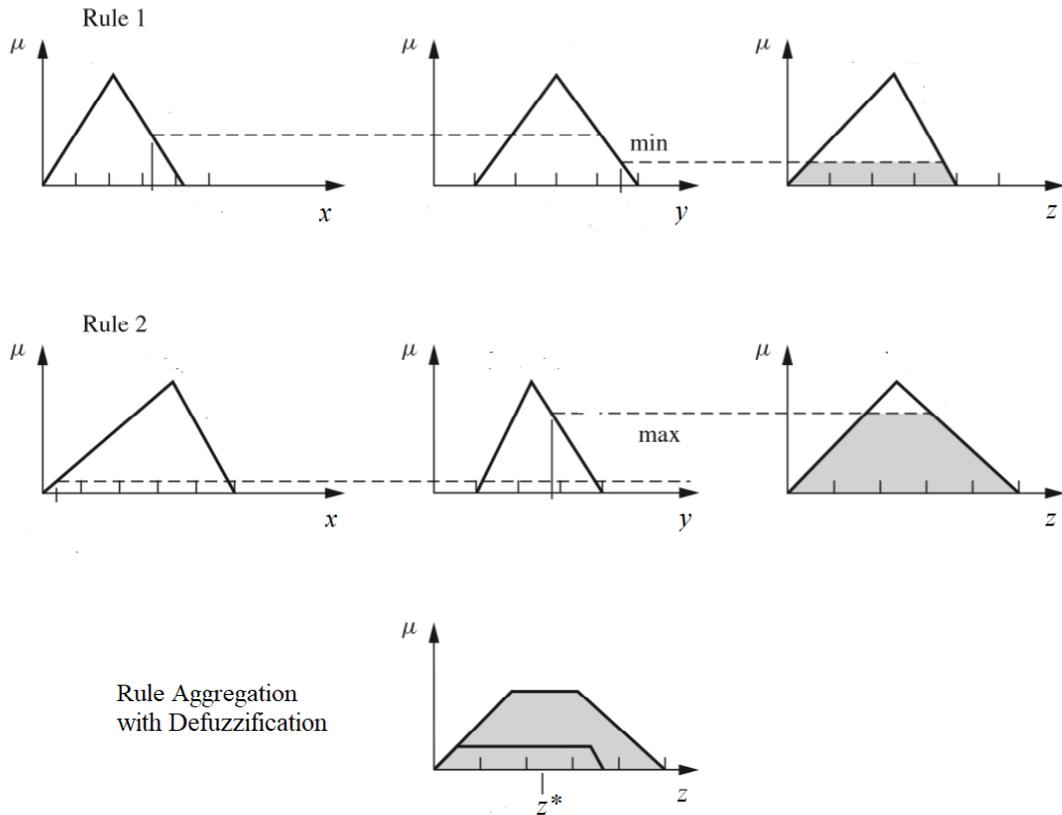


Figure 2.5 – Example showing a two rules Mamdani FIS functioning, comprehending rules connective operation, rules aggregation and Centroid Defuzzification (adapted from “Fuzzy logic with engineering applications”, Ross, 2010)

Then, for simplicity, let's just use two rules for two inputs and a single output. In this case, inputs will have Gaussian – shaped fuzzy sets, just to show another example of Membership function. Two rules are shown:

Rule 1 → IF x is A_1 AND y is B_1 THEN z is $z_1 = p_1 x + q_1 y + r_1$

Rule 2 → IF x is A_2 AND y is B_2 THEN z is $z_2 = p_2 x + q_2 y + r_2$

It was previously mentioned that, with this type of FIS, AND connectives work as product operators between inputs membership degrees. Moreover, these will produce so – called weights w_i (Equations 2.4, 2.5), which are needed to apply the Weighted Average Method (Equation 2.6):

$$w_1 = \text{prod} (\mu_{A_1}, \mu_{B_1}) = \mu_{A_1} \cdot \mu_{B_1} \quad (2.4)$$

$$w_2 = \text{prod} (\mu_{A_2}, \mu_{B_2}) = \mu_{A_2} \cdot \mu_{B_2} \quad (2.5)$$

$$z = \frac{w_1 z_1 + w_2 z_2}{w_1 + w_2} \quad (2.6)$$

From last expression, a final polynomial function is obtained by means of weights, then initially inserted input values are entered in such function to find the final solution. In Figure 2.6 a graphical explanation of previous expressions is exploited.

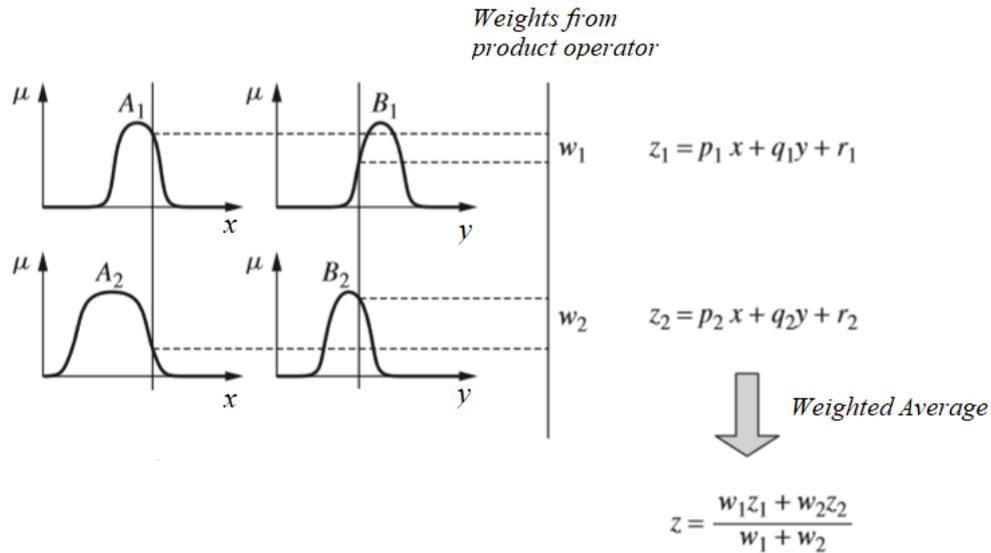


Figure 2.6 – Example showing a two rules Sugeno FIS functioning, comprehending weights obtainment and Weighted Average (adapted from “Fuzzy logic with engineering applications”, Ross, 2010)

There is an utter type of FIS, called “Tsukamoto”. However, this will not be explained, since it is not of interest in this study. After having shown basic features of Mamdani and Sugeno systems structure, it is worth listing some of the main reasons why they are applied in certain circumstances, instead of others. First aspect, which has to be considered, is the presence of membership functions in the output, occurring in Mamdani and not in Sugeno. This certainly confer more interpretability to Mamdani compared to Sugeno, since it is easier to deal with eye – catching fuzzy sets labelled by linguistic expressions rather than oblivious polynomial expressions (Egaji, Griffiths, Hasan, & Yu, 2015). On the other hand, as previously stated, Sugeno skips Defuzzification passage, being computationally lighter in terms of processability. This fact will be demonstrated with a dataset modelling in one of next chapters paragraph. Secondly, Mamdani inference system lacks of flexibility in designing ease compared to Sugeno’s, since this can be implemented with adaptive neuro – fuzzy inference systems in order to detect best values of modifiable parameters, for modelling purposes (Singla, 2015). In conclusion, Mamdani is more suitable when it comes to transferring Human Knowledge to an

inference system, while Sugeno is more suitable for Machine Learning. Figure 2.7 resumes very simply the process of converting a crisp input in a crisp output with Mamdani and Sugeno fuzzy inference systems (used separately).

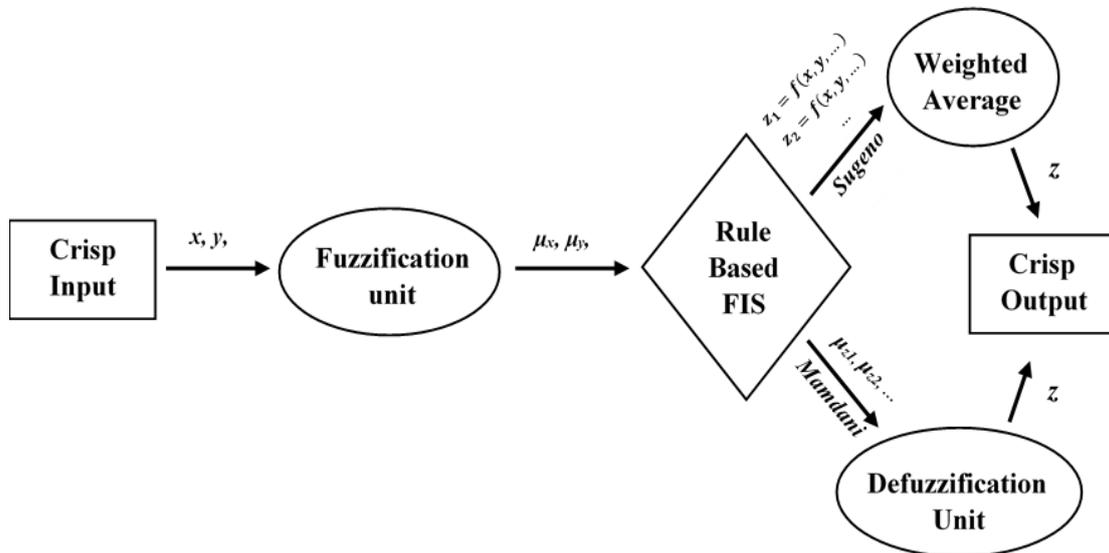


Figure 2.7 – Scheme of Fuzzy Inference Systems functioning from input insert to output obtainment

2.2.2 ANN: Artificial Neural Networks

A brief explanation regarding this very popular machine learning method is needed. In fact, if it is true that Adaptive Neuro Fuzzy Inference Systems adopt fuzzy sets and rules to map inputs into outputs, it still has Neural Network training algorithms. The main reason for this resides in the fact that Artificial Neural Networks have a wide range of model parameters optimisation tools. Fuzzy inference systems do not have such capabilities on their own. Therefore, when a certain precision is required it is very difficult to change FIS modifiable parameters, in order to have an overall better mapping.

“A neural network is a technique that seeks to build an intelligent program (to implement intelligence) using models that simulate the working network of the neurons in the human brain” (Ross, 2010).

This short pass of Ross’ book appropriately introduces this topic. The wide variety of models of this family cannot be regardless of its fundamental component, the *“Artificial Neuron”*, proposed by W.S. McCulloch and Walter Pitts in a famous work in 1943: *“A logical calculus of the ideas immanent in nervous activity”*. This study schematises a linear threshold combinator, with multiple binary data as input and a single data as

output. In that first case, a suitable number of elements of that kind, connected in such way to form a network, could calculate simple Boolean functions.

First, it is important to define Artificial Neural Networks' structure, in order to comprehend involved learning mechanisms. Human neural networks are substantially composed by neurons and dendrites, which are connections between neurons.

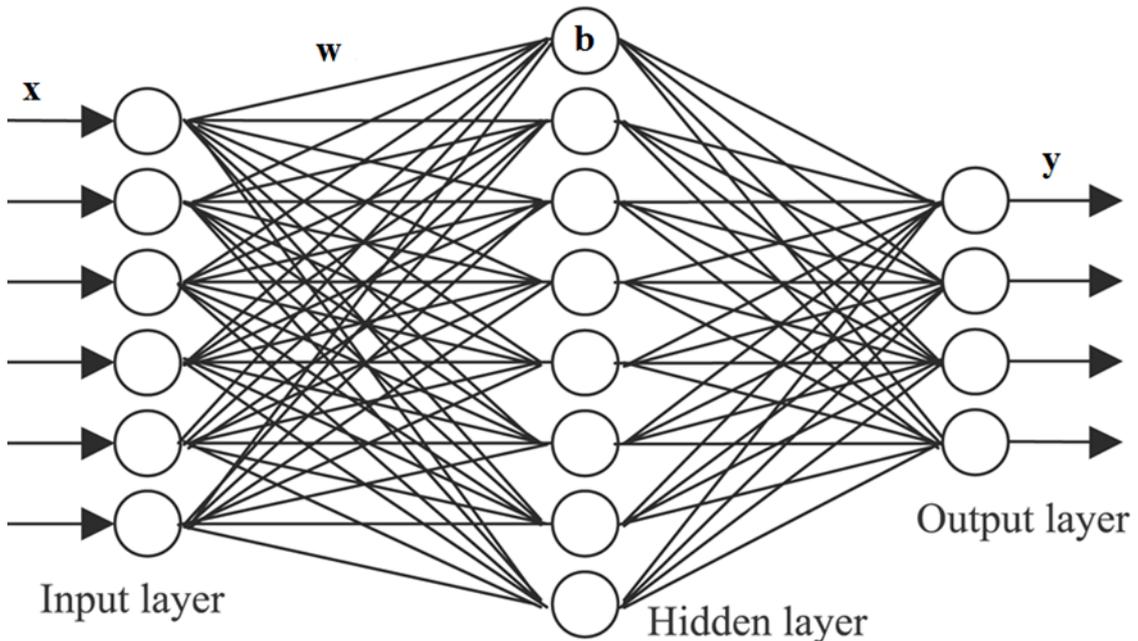


Figure 2.8 – Simplified layout of an Artificial Neural Network, constituted by an input layer, an output layer and a single hidden layer (adapted from “Artificial neural networks are changing the world. What are they?”, Graham, 2015).

ANN work in an analogous way. They have neurons, more often denominated “Nodes”, and connections between them. Such networks are organised in layers of neurons, which transfer information in form of numbers from a layer to the next one. Moreover, each node receives information from each node of the previous layer, then it passes it in another way to each node of the next one. This is going to be seen more in detail. Figure 2.8 allows to make the description easier. There are three layers, comprehending an input layer on the left, a hidden layer in the middle and an output layer on the right. It is important to say that a single hidden layer is used just for simplicity, because often more hidden layers are employed, and they can be looked at as a black box. Input layers have fixed nodes number, equal to the number of input variables. Of course, this is valid also for the output layer. On the other hand, number of hidden layers and nodes per layer are not a constraint. In fact, the choice of these parameters is left to the network builder. Every connection has its own “Weight”, which is a number that could be negative or positive. Then, every node has its own “Bias”, also known as threshold, and its utility will be shown promptly. Finally, every node has the task of elaborating and computing

information from the previous layer, translating it in a space between 0 and 1 through a particular function. They are called “*Activation Functions*” and they are usually characterised by Sigmoidal shapes (Equation 2.7), even if for some researchers they are old fashioned nowadays.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.7)$$

Given these premises, let’s see how a neural network operates from input insert to output obtainment. Let’s consider only one node of the hidden layer. Once inputs values are dictated, they are multiplied for the weights and they are fed to the considered node through its connections with inputs nodes. This node computes received information by making a sum of weights multiplied by previous nodes exit values, minus previously mentioned Bias. This parameter is defined differently for each node and it serves the purpose of acting as a “Threshold”. In fact, this value let the node be activated only if the entered sum is higher than a certain value, and by activated it means that it goes to affect next layers nodes inputs. Equation 2.8 reports a node input in mathematical terms.

$$\text{node input} = \left(\sum_{i=0}^n w_i a_i \right) - b \quad (2.8)$$

where $i = 1, \dots, n$ with n equal to previous layer nodes number. As previously stated, when such node input is given to a sigmoidal function it is compressed into a space between 0 and 1 and sent to the next layer. So, this occurs for all nodes of the hidden layer receiving inputs from input layer. They compute the outcome, which is then sent to the output layer, which itself computes again a summation and insert it along with its biases. Finally, they give a certain network computed output.

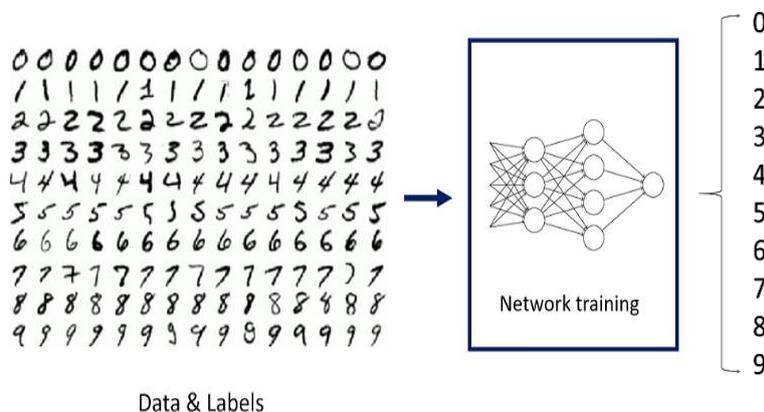


Figure 2.9 – Written numbers (from 1 to 9) recognition model with trained ANN

A famous example of how Artificial Neural Networks could possibly be employed, is demonstrated in the case of “Written Numbers Recognition”. Basically, there is a certain

quantity of pixels on a screen, for example 28 x 28. A well – trained ANN, with the adequate training datasets, should be able to recognize a number from 0 to 9 when a drawn number on the screen is given as an input. This example is schematised in Figure 2.9. Training a neural network is always required, in order to have the right output from given inputs. As a matter of fact, there is no way someone could build a well - functioning network starting from scratch without training it. First, number of hidden layers is pre – specified, along with the number of nodes. These are never fixed, network builder decides how many of them must be put in the network and see if they perform well after training. Said so, an initial random guess is made regarding weights imposed to the connections and nodes’ biases. It is almost impossible to predict immediately which weights and biases would confer the best properties to a model. On the other hand, activation functions computing inputs in nodes are not modifiable. Once their type is set, it must stay the same. Therefore, variable parameters which are going to be changed through training process are Weights and Biases. If number recognition had to be done without training, it would probably give wrong answers. How is a written number put into a network in form of numerical input from an image? This is done through pixels Brightness, going from 0 (maximum darkness) to 1 (maximum brightness). Once number of pixels is given, every pixel brightness becomes the input for the network. This should be able to produce an outcome from 0 to 1 for each number from 0 to 9 (so possible outputs quantity is imposed as 10). The number with the highest value of activation should be most probable answer. ANN which was never trained before produce a certain error in the output, meaning there is a difference between the expected output and the produced one. This error is used to modify weights and biases, going backwards from the output layer to the input layer. This process is called “Back Propagation”. Then the input is fed again to the network, the error is re – calculated and this cycle goes on and on iteratively. It stops either when a desired error threshold is reached or when a pre – imposed number of iterations is overcome. Back Propagation plays an important role in this thesis work, since it is used in ANFIS hybrid learning method modelling to adapt premise parameters on training datasets. Therefore, it is worth explaining its working principles. First, the definition of error is given in Equation 2.9.

$$e_j(n) = d_j(n) - y_j(n) \quad (2.9)$$

Where j is referred to an output node, n is the iteration number, d is the desired output and y is the produced output. Weights and biases have to be modified in order to minimise a certain “Cost Function”, which is obtained by computing the sum of all square errors coming from the output for every single data, then averaging it for the entire training dataset by summing them all and dividing by number of datasets (see Equations 2.10 – 2.11).

$$E(n) = \sum_j e_j^2(n) \quad (2.10)$$

$$\text{cost function} \rightarrow \epsilon_{av} = \frac{1}{N} \sum_{i=1}^N E_i(n) \quad (2.11)$$

Where i is the index referred to a certain input – output dataset and N is the total number of data set involved in the training process. It is important to say, even if already mentioned, that such cost function is a function of Weights and Biases. Hence, in order to minimise it, these parameters are modified during the learning process. This is achieved through iterative methods. In Back Propagation, the learning process follows a double pass iterative procedure (Tadeusiewicz, 1995), a forward and a backward passages. Forward consists in entering inputs in the system from the input layer, then letting the network do its calculations without changing anything. Next, once the outputs are produced, the cost function is obtained and the path is inverted, modifying weights and biases going backwards from the output layer to previous hidden layers. This is the reason why this is called Back Propagation. However, weights and biases are not modified casually. In fact, in this method they are selectively increased or decreased in value, following the *Gradient Descent* approach. In Figure 2.10 an example of gradient descent is shown in two dimensions, just to explain the concept graphically.

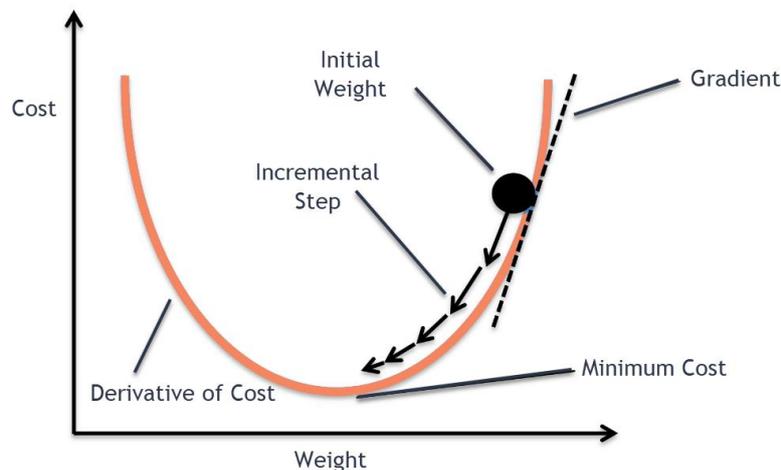


Figure 2.10 – Graphical illustration of Gradient Descent Method in minimum search for a Cost function (adapted from “Gradient Descent with Momentum”, 2019)

However, in ANN the gradient descent is performed over thousands of dimensions, equal to the number of weights and biases. Right after the forward pass, training system computes the gradient of the cost function, doing the derivatives respect to weights and biases. Substantially, modifications are applied to those parameters where the derivative has highest value towards the desired outcome in each dataset. The major drawback of

this method is that the gradient of the cost function is usually a vector of huge dimensions. Therefore, its manipulation can be computationally heavy. Methods such as Gradient Descent are required whenever non – linear problems must be solved. In fact, flow pattern determination is a non – linear problem (Fontenla-Romero, Erdogmus, Principe, Alonso-Betanzos, & Castillo, 2003).

It is not difficult to see, from this short review of Artificial Neural Networks, that they have great potential as Machine Learning method. This quality derives from the fact that it has many ways which can be followed to efficiently train it, given that premises for the model are sufficiently well – imposed. In addition to this, as Fuzzy Logic, ANN employs an encoded numerical space from 0 to 1 to let the internal parts of the structure communicate among themselves. A major drawback of ANN resides in the fact that hidden layers section acts as a “*Black Box*”, unable to interpret relationship between input and output.

2.2.3 ANFIS: Adaptive Neuro – Fuzzy Inference System

It is simple to imagine that this approach comprehends features from both Artificial Neural Networks and Fuzzy Inference Systems. Substantially, an ANFIS is mathematically speaking a FIS with a Sugeno – type inference system (or Tsukamoto, but in this text, it will not be considered). Nevertheless, it is not just a Sugeno FIS, it has a structure which allows it to be trained using powerful methods, applicable to ANN. In this way, this hybrid apparatus combines these two worlds of soft computing, obtaining as result a human – like reasoning style typical of fuzzy systems along with the learning and connective nature of artificial neural networks. Starting from these premises, it is quite understandable how these inference systems can be employed in decision making, signal processing, control and, as in this case, modelling. In addition to this, ANFIS is not the only type of neuro – fuzzy hybrid systems. In fact, other neuro – fuzzy architectures exist, for instance FALCON (*Fuzzy adaptive learning control method*) and GARIC (*Generalized Approximate Reasoning based Intelligence Control*) (Vieira, Dias, & Mota, 2004). In Figure 2.11, an example of an ANFIS structure for a Sugeno model is represented. It must be considered that, even if in this picture only two inputs with four membership functions and four rules are shown, these systems often have a considerable quantity of them. Therefore, this is shown in this way just for simplicity. In addition to an input layer and an output layer, which is obviously shown both by FIS and ANN, five layers are labelled in Figure 2.11. Since such system proceeds from input insert as it must be Feed Forward, layers are numbered from 1 to 5 starting from the left to the right.

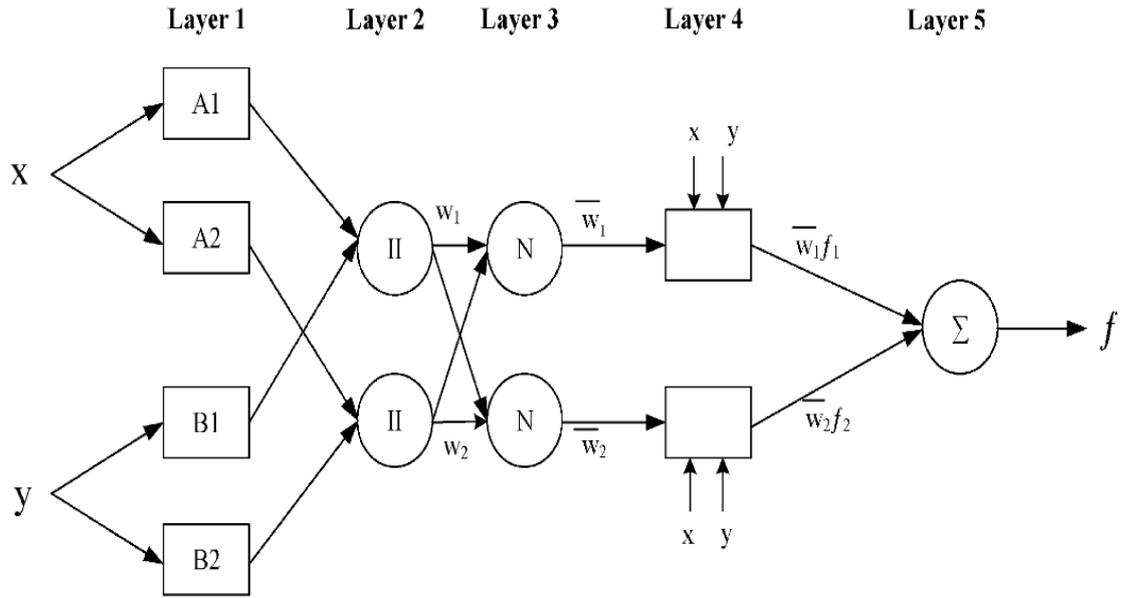


Figure 2.11 – Explicative example of an Adaptive Neuro – Fuzzy Inference System structure for a Sugeno type (adapted from “Optimization of EPB Shield Performance with Adaptive Neuro-Fuzzy Inference System and Genetic Algorithm”, K. Elbaz et al., 2019”).

Layer 1: this is the layer which is adopted for *input Fuzzification*. In fact, its nodes are Adaptive, since they are characterised by Membership Functions, thus having modifiable parameters. For example, Gaussian – type membership function has the form reported in Equation 2.12.

$$f(x) = a e^{-\frac{(x-b)^2}{2c^2}} \quad (2.12)$$

Where a, b and c are here defined as *Premise Parameters* (J. S. R. Jang, Sun, & Mizutani, 2005). It was already stated, at the end of last paragraph, that membership functions can be considered as activation functions, since they squish any input into a space of truth between 0 and 1. Usually, these nodes are connected each one to every input node, as it is usually seen for neural networks. Each node of this layer computes an outcome based on its membership functions. Considering inputs x and y, in notation this can be exploited as (Equations 2.13 – 2.14):

$$O_{1,i} = \mu_{A_i}(x) \quad \text{for } i = 1, 2 \quad (2.13)$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \quad \text{for } i = 3, 4 \quad (2.14)$$

First subscript 1 is referred to the first layer, the second index to node number, which is membership function number where A is referred to input x and B to input y.

Layer 2: considering that involved rules have AND connectives, in this case it is chosen to perform a product between incoming signals (inputs to layer 2 nodes). In fact, in Figure 2.11 it is noticed that these nodes circles have product signs Π in them. Therefore, in such layer there will be node function as Equation 2.15:

$$O_{2,i} = w_i = \mu_{A_i}(x) \mu_{B_i}(y) \quad i = 1, 2 \quad (2.15)$$

Sign w_i stands for weight. In ANFIS language, weights are usually called “*Firing Strengths*” of a rule. Since it is Sugeno system what is being talked about, weights are produced for next steps. Besides, in this case there are only two nodes for layer 2 because only two rules are considered to be present in this example.

Layer 3: in this layer, weights are normalized respect to all other weights. As previously mentioned, since these can be also denominated Normalized Firing Strengths. For two rules it would be (Equation 2.16):

$$O_{3,i} = \bar{w}_i = \frac{w_i}{W_1 + W_2} \quad i = 1, 2 \quad (2.16)$$

Layer 4: here nodes become adaptive again. In fact, each node has its own linear function (considering this Sugeno FIS as first – order). Therefore, there are linear parameters which are modifiable, specifically three for each function f_i : p_i , q_i and r_i . These are labelled as *Consequent Parameters*. In this case, since there is an output function for each rule, there are two linear functions. They are multiplied by their own normalized firing strengths (Equation 2.17):

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad \text{for } i = 1, 2 \quad (2.17)$$

Layer 5: there is only one node which computes a summation over all incoming signals from layer 4 nodes. Finally, it gives an outcome which is the final solution from the ANFIS (Equation 2.18):

$$\text{overall output} = O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad \text{for } i = 1, 2 \quad (2.18)$$

Hence, this is how any ANFIS produces an output from its fuzzy network by inserting inputs. However, what happens with membership functions of FIS and nodes in ANN, still applies for these inference systems: there is no specific way to tell which the best number of rules and membership functions is to use, in order to have a good approximator, without being too heavy in terms of computation. It is easily

understandable to say that if an ANFIS had a very high number of functions and rules, it would satisfyingly represent trained datasets. The determination of membership functions shapes and quantity should be left to the user, then the training will adjust their non – linear parameters. Initial fuzzy model can be derived systematically through a few techniques, in order to have a raw structure to start with. Most popular methods are *Grid Partitioning* and *Subtractive Clustering*.

2.2.3.1 Grid Partitioning

This technique substantially partitions input space in several fuzzy regions. This allows to create the antecedents which are then going to be positioned within rules of the inference system (Mrinal, 2008). It must be specified that this method turns out to be effective when inputs number is not too high. It is easier to understand how it works by considering an input space composed by only two inputs, as in Figure 2.12. In this way, fuzzy regions are visible. The outcome for the implementation of this method is a 3 x 3 rules system, with a total of 9 rules, each one defined by an output function.

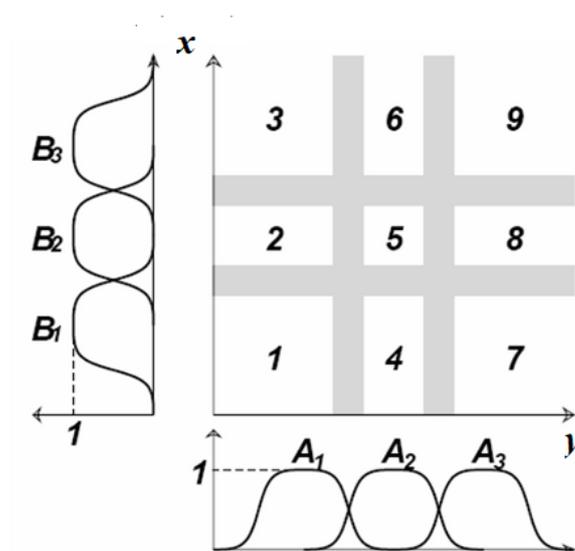


Figure 2.12 – Grid Partitioning for an input space composed by two inputs x and y , each one defined by three membership functions (adapted from “Estimating Development Time and Effort of Software Projects by using a Neuro Fuzzy Approach”, V. Marza et al., 2009)

In addition to this, grid partitioning does not create adapted regions in the input space from the start. They are homogeneously distributed. However, they are modified as training proceeds. This method has the disadvantage of being “Greedy”, that means creating a lot of rules and membership functions. This drawback will be discussed in Chapter 4.

2.2.3.2 Subtractive clustering

This method is more specific on hitting certain areas by analysing datasets, classifying them with clusters (Yeom & Kwak, 2018). Number of clusters is not initially specified when the method is applied. However, considering input space as a hypercube, the user must fix the dimensions of cluster by their radii. Number of fuzzy rules increase proportionally as cluster radius decreases. This type of clustering follows a certain path. First, density function (Equation 2.19) is implemented in order to individuate which points are most likely to become clusters centres. Cluster radius, r_a , is a constant in this expression, while between double brackets there is a Euclidean distance.

$$P_i = \sum_{j=1}^m \exp\left(-\frac{\|x_i - x_j\|}{\left(\frac{r_a}{2}\right)^2}\right) \quad (2.19)$$

Then, the data points having the highest density, found through this function, are grouped in the first cluster. This one is removed using Equation 2.20:

$$P_i = P_i - P_{c_1} \times \sum_{j=1}^m \exp\left(-\frac{\|x_i - x_j\|}{\left(\frac{r_b}{2}\right)^2}\right) \quad (2.20)$$

It is noticed that instead of r_a there is r_b . This is a constant too, it is the radius of the elimination function. These steps are recursively followed until density value becomes smaller than the set value. Figure 2.13 reports a graphical example of a two variables case, where three clusters are individuated in black, while other data points are illustrated in red.

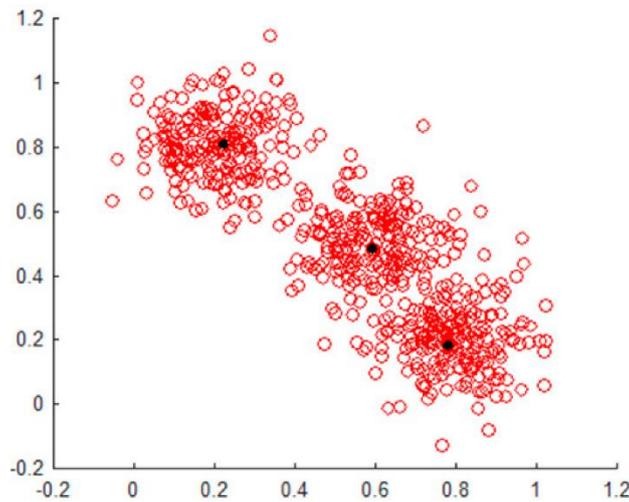


Figure 2.13 – Subtractive clustering for an input space composed by two inputs x and y (adapted from “Performance Comparison of ANFIS Models by Input Space Partitioning Methods”, Yeom et al., 2018)

2.2.3.3 ANFIS Hybrid learning method

In previous paragraph, Back Propagation training method was explained for ANN, with the purpose of anticipating it also for neuro – fuzzy system. First, it must be specified that Back Propagation can still be used for ANFIS. Nonetheless, most implemented method for such fuzzy networks turns out to be a combination between Back Propagation and LSE, least square estimator. As a matter of fact, compared to solely applying Back Propagation, this method returns a higher speed of convergence since the gradient descent has less dimensions where to find the minimum error (J. S. R. Jang et al., 2005). It is based on linearity of adaptive parameters present in the inference structure. Specifically, premise parameters such as Gaussian curve coefficients can be classified as non – linear. On the other hand, consequent parameters, which are constituted by output linear functions coefficients, are labelled as linear. Hybrid training algorithm is obtained by alternatively modifying premise and consequent parameters, while maintaining fixed the other one. This is done in a two passes recursive algorithm: a forward pass, where premise parameters are kept fixed and consequent parameters are modified through linear least square estimator. In the backward pass, Back Propagation of error is applied following gradient descent procedure, changing non – linear premise parameters while maintaining unvaried consequent parameters. In Table 2.1 this concept is resumed.

Table 2.1 – Synthesis of the procedure of ANFIS hybrid training method

	Forward pass	Backward pass
Premise parameters	Fixed	Gradient descent
Consequent parameters	Least squares estimator	Fixed
Signals	Node outputs	Error signals

A brief description of how LSE operates in this field is needed to have a basic comprehension of how hybrid training works. First, it is necessary to define why modifying consequent parameters can be considered a linear problem. Considering a FIS such as the one reported in previous paragraphs, carrying two output functions to be weighted, it would be:

$$\begin{aligned}
 f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \\
 &= \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_2 x + q_2 y + r_2) \\
 &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2
 \end{aligned}
 \tag{2.20}$$

This can be extended also to a higher number of output functions. It is clear that f is linear in consequent parameters p_i , q_i and r_i . In general, after fixing values of premise parameters, if a certain batch of datasets goes to substitute variables in final output function several times, letting linear coefficients to be determined, it becomes a system with the form of Eq. 2.21:

$$A\theta = y \quad (2.21)$$

Where A is a matrix containing all known terms constituted by products between averaged weights and values of input data, while y represents desired output, still dictated by data. On the other hand, θ represents consequent parameters vector, which are unknown at each epoch. This vector is called the “*Least Square Estimator*”, as it is used to minimize the error. Generally, problem consists in finding the minimum of:

$$E(\theta) = \sum_{i=1}^m (y_i - a_i^T \theta) = e^T e = (y - A\theta)^T (y - A\theta) \quad (2.22)$$

2.3 Correlation Indexes

This section gives a general perspective over some of the most used correlation indexes: Relative Root Mean Square Error (RRMSE), Index of Agreement (IA) and Coefficient of Determination (R^2). These will be employed after modelling procedures shown in next chapters, to understand how well different created models approximate data, besides comparing them. These evaluations will be shown on both training and validation data. Moreover, a “Fuzzy” combination of these three indicators will be adopted in order to sum up their effects in only one parameter. This will be called the “*Goodness Index*”. There is no strict way on how to determine goodness of fitting for a certain model. In fact, choice of correlation indexes is left to the user. Moreover, use of multiple indexes is advised in order to consider multiple aspects of modelling. Such indicators are defined in following paragraphs. O_i and O_m respectively represent i^{th} data point and medium of observed experimental values. Predicted values are indicated with P_i and P_m .

2.3.1 RRMSE: Relative Root Mean Square Error

$$\text{RRMSE} = \frac{\left(\frac{1}{n} \sum_{i=1}^n (O_i - P_i)^2 \right)^{0,5}}{O_m} \quad (2.23)$$

It is the root square of the numerical mean of the square errors (difference between observed and predicted values), divided by the mean value of the observed data

(Equation 2.23). This last action was done in order to insert a mean of comparison, so to have such value in a dimensionless form. It ranges from 0 (no error produced by the model) to more than 1 (the model produces an error which is bigger than the mean observed value). If a single residual (error) is very large, the RRMSE could still be sufficiently small for the model to be considered good. In fact, if the number of data is very large, such deviation can be effectively distributed among these. Whereas the determination coefficient could be small instead, because it is majorly affected even by only one big error.

2.3.2 R²: Coefficient of Determination

$$R^2 = \frac{(\sum_{i=1}^n (O_i - O_m)(P_i - P_m))^2}{\sum_{i=1}^n (O_i - O_m)^2 \sum_{i=1}^n (P_i - P_m)^2} \quad (2.24)$$

It is given by ratio between regression deviance and total deviance (Equation 2.24) and it ranges from 0 (no correspondence) to 1 (perfect correspondence). Although the use of such coefficient is a common practice, it only gives a rough indication about model goodness. It is usually employed in linear regressions. In addition to this, it shows how much variation of a certain dependent variable, with respect to its medium value, can be well described by a certain model. Moreover, it is low when there is incapacity for the regression curve to keep close to observed data.

2.3.3 IA: Index of Agreement

$$IA = 1 - \frac{\sum_{i=1}^n (O_i - P_i)^2}{\sum_{i=1}^n (|O_i - O_m| + |P_i - O_m|)^2} \quad (2.25)$$

IA (Equation 2.25) ranges from 1 (perfect fit by this index) to 0 (useless fittings). It was invented by *Willmott* in 1985, starting from this generalized form (Equation 2.26):

$$\rho = 1 - \frac{\delta}{\mu} \quad (2.26)$$

Where δ is an average error magnitude, in this case the Mean Square Error, and μ is a basis of comparison (*Willmott, Robeson, & Matsuura, 2012*). μ is equal to the summation of the deviations of observed and predicted values from their mean values respectively. While RRMSE describes the average magnitude of the errors, *IA depicts the degree to which the errors approach the null set*. This form was recognized to overweight large

error because of the square, so it is preferable to use it with other model descriptors (RRMSE and R^2).

2.3.4 GI: Goodness Index

It is generally recognized that, although it is very common to talk about “Goodness” of a model, it would be more appropriate to say “Usefulness”. This regards the ability to provide the best projection of modelled data (Di Addario et al., 2017). Nevertheless, in this paragraph a newly created index is presented. The basic concept is to imagine a combination of certain correlation indexes, which could be obtained through a simple Mamdani fuzzy inference system. Substantially, previously explained correlation indexes are fed to the system as antecedents. Then, “*Goodness Index*” GI is obtained as consequent from a set of 15 different rules, in a range going from 0 to 1. First, it must be specified that this is not a trained model. In fact, there are no experimental evidences which could correlate these three indexes to this newly created one. It must be considered more as an “Expertise” situation, or even a system to be built from Intuition of the user. This model was created in PhD thesis “*Development of Fuzzylogic model to predict the effects of ZnO nanoparticles on methane production from simulated landfill*” of Di Addario M. and it is capable of giving an immediate comparison through the use of the first three indexes. It was achieved using *Fuzzy Logic Toolbox* of *MatLab*[®]. Crisp ranges were set from 0 to 1 for each index. For each antecedent, three equilateral triangular membership functions were used, with an equal partition of the crisp dominion. Triangular membership functions are employed for simplicity, since they are the easiest to be characterised by being constructed only by straight lines, with only one point with full membership degree. These are labelled, starting from the left, as “*Low*”, “*Medium*” and “*High*” and they refer to intervals along the crisp range of each antecedent.

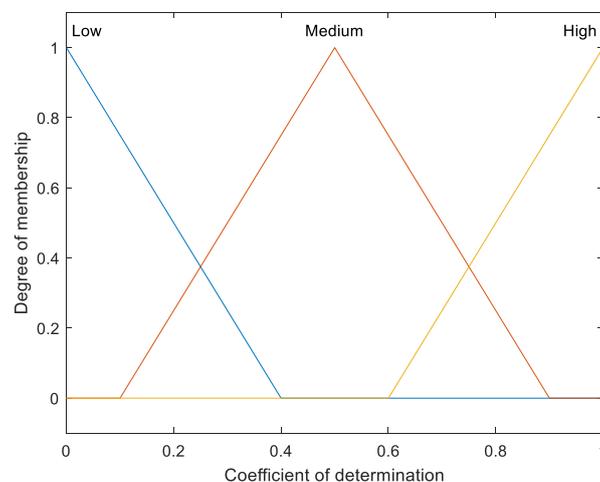


Figure 2.14 – Membership functions of Coefficient of determination antecedent

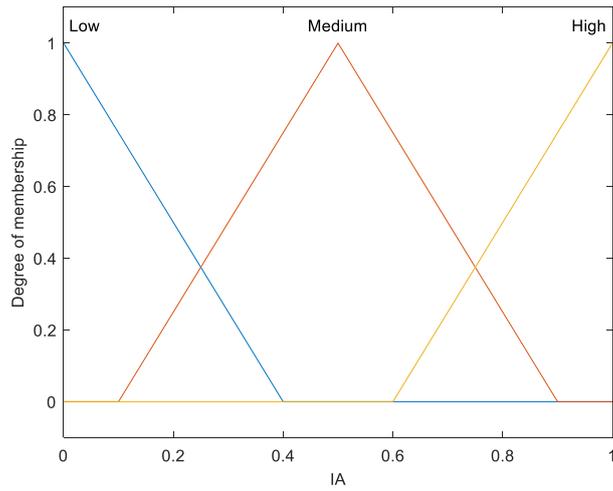


Figure 2.15 – Membership functions of Index of agreement antecedent

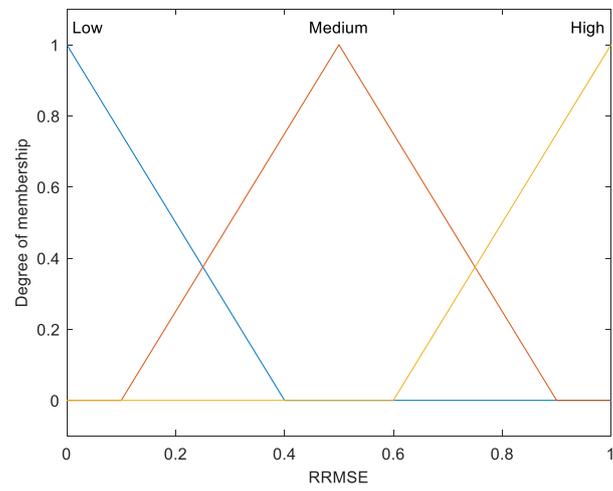


Figure 2.16 - Membership functions of RRMSE antecedent

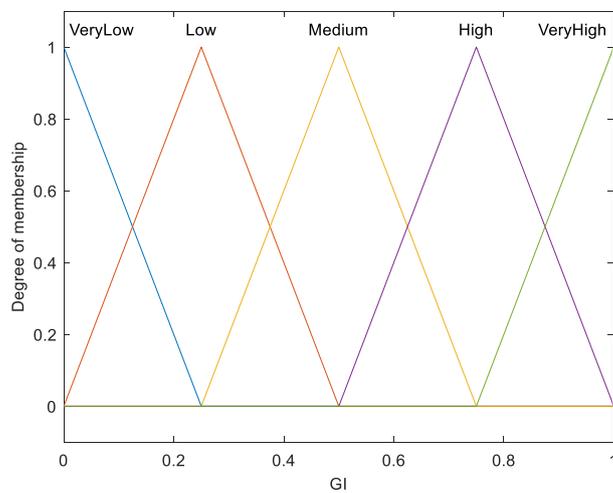


Figure 2.17 – Membership functions of Goodness Index consequent

From Figures 2.14 – 2.17, it is possible to see how these fuzzy sets interact with each other by their intersections. While for R^2 and IA goodness of correlation rises going from 0 to 1, thus “Low” to “High”, RRMSE works in the opposite way. Goodness Index value is located along with the other three indicators to have a more intuitive perception of the usefulness of constructed models. Five membership functions, still triangular – shaped, are employed for the consequent. These are necessarily given in a higher amount compared to the antecedents, in order to confer more adaptivity to the goodness index through rules. Implemented defuzzification method consists in conventional Centre of Gravity expression. Rules are given with AND connectives, performing minimum operators. Aggregation between these is obtained by a union of the figures separately obtained from different rules. In Table 2.2 the rules implied in the inference system are shown. It is evident that goodness of correlation by GI increases from 0 to 1, thus from “Very Low” to “Very High”.

Table 2.2 – Rules composing inference system in Fuzzy Goodness Index Model

IF	R²	IA	RRMSE	THEN	GI
	Low	Low	high		Very Low
	Medium	Medium	Medium		Medium
	High	High	Low		Very High
	Low	Low	Medium		Low
	High	Low	High		Medium
	Medium	Low	High		Low
	Low	High	High		Medium
	Low	Medium	High		Low
	Low	High	Low		Medium
	Medium	High	Low		High
	High	Low	Low		Medium
	High	High	High		Medium
	High	High	High		Medium

3. Experimental methods

In this chapter, modelled data format will be presented, along with the adopted software for the development of Fuzzy and ANFIS models.

3.1 Particle Image Velocimetry Experimental Data

Datasets which are going to be modelled, in the next chapter, were obtained as results of experimental observations made in the Master Thesis “*Influence of Impeller Type and Geometry on Particle Stress in a Stirred Fermenter by Means of Particle Image Velocimetry*”, written by Giulia Moretti. Experiments consisted in obtaining velocity vector field data through particle image velocimetry technique, for an agitated vessel. The apparatus was constituted by artificial lighting pointed on the vessel to illuminate the liquid within the small tank. Such illumination was provided by a laser. The liquid was filled with tracing particles, which allow to obtain velocity vector field by differential time scansion of their movement. However, in order to acquire such information, a camera was installed so that a great number of pictures could be taken. However, if one camera is required to perform bi – dimensional investigations, three – dimensional is only derived by means of two cameras. Therefore, two cameras were used when needed for 3D method. The lighting source and the camera were plugged with a synchronizer, so that they could function in parallel. A specific software had to be employed for this type of application, and it is *DaVis 8.4*, produced by *LaVision*[®]. This program can calculate velocity vector and its intensity on a wide range of points of the vessel.

The examined vessel is made of glass. It is cylindrical with a tori spherical bottom. The internal diameter is equal to 160 mm. Observed fluid is poured into the vessel up to a height of 160 mm, with a capacity of almost 3 L. Four baffles are placed inside, with a width of 12 mm, a length of 114 mm and a thickness of 2 mm. These are located at 5 mm from the vessel wall. Agitator is composed by a single impeller.

Even though there is a wide variety of configurations examined in such work, only a few of them take part in modelling. As a matter of fact, in order to correlate different values of a certain input, an adequate number of different values for its parameter must be adopted. Each data set regarding a given fluid, agitating one of the three liquids, with a certain impeller rotating with an exact power input, presents approximately 6000 velocity values, each one corresponding to a given position dictated by radial and axial coordinates. Therefore, if velocity is chosen as consequent, these two coordinates indicating a point in the vessel will certainly be interesting to set as antecedents. The third possible antecedent could be the Power Input, since five values of them are given

for few apparatus configurations. Specifically, these were observed in bi – dimensional camera configuration, regarding water as fluid and Ring Propeller as agitator module. A simple model will also be exposed on BiLoop impeller. This one produces a very different flow pattern, compared to the Ring Propeller. In Figure 3.1, both impellers are represented, along with a scheme of flow patterns developed by each one separately.

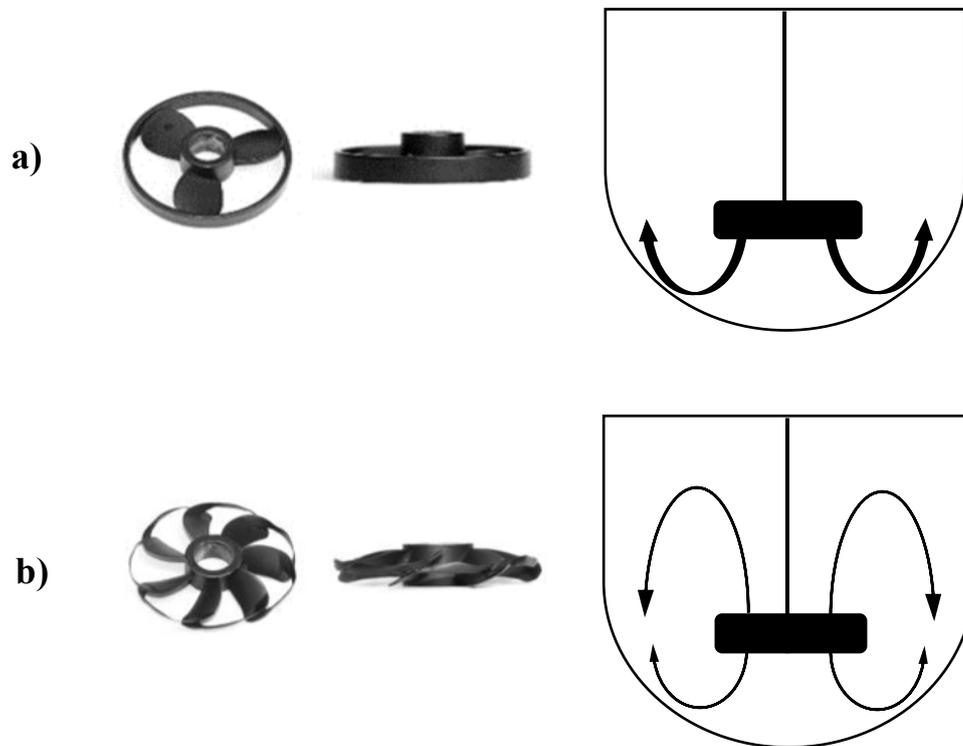


Figure 3.1 – Image and scheme of flow pattern for: a) Ring Propeller; b) BiLoop impeller

Flow schemes on the right show that, while Ring Propeller push the fluid mainly downwards, BiLoop impeller sends it both upwards and downwards. In Table 3.1 available Power inputs value are shown for observations made on Ring Propeller and water. These will be used to create a three antecedents’ model. These are shown coupled with relative tip velocity, which will be used to obtain dimensionless velocity by dividing velocity vector norm by tip velocity.

Table 3.1 – Available power input values regarding vessel filled with pure water, agitator equipped with a ring propeller, with a 2D camera disposition.

P/V (W/m ³)	20	50	100	200	500
Tip velocity (m/s)	1,168	1,585	1,997	2,516	3,415

Data sets were fed as training batches in the format reported in Table 3.2. Since original velocity data were gained as two components, one radial and one axial, these were condensed in one value represented by velocity vector norm. This is obtained by applying Euclidean norm of the two components (Equation 3.1).

$$v = \sqrt{V_r^2 + V_z^2} \quad (3.1)$$

Table 3.2 – Example of data set given to the ANFIS training algorithm

r / R	z / H	v / U_{tip}
0.122	0.492	0.013
0.703	0.350	0.046

Software which elaborated PIV data reported also a number addressed as “*IsValid*”, equal either to 0 or 1. Values with such parameter equal to 0 were not taken in consideration, as they give null velocity for being in non - observed position in the vessel. Another issue regards the fact that data were taken for an entire half of the vessel and the opposite half on the lower side. Thus, it was reasonable to pick the values of the whole half of the vessel, considering the other half to have same velocity values for symmetry. Coordinates data are indicated in mm. In those cases where power input is inserted as third antecedent, the first column reports its values repeatedly for the position and velocity values to be addressed to the right P/V. Every column of data, except for power input, was transformed in dimensionless respective parameter. This was done by making the ratio between original values and the maximum value of every parameter. Thus, radial coordinate is compared to the vessel radius, axial coordinate to vessel height and velocity field to impeller tip velocity. Nonetheless, some models were modelled through original data for various reasons which will be explained. Application of such ratios has the purpose of creating models applicable to any reactor geometry. In fact, by converting spatial coordinates into dimensionless values, produced model could refer to any radius or height. In other terms, this would mean trying to produce a model, eventually useful in scale – up applications. Data points obtained by experimental observations through the software implemented in the synchronizer, were elaborated in order to produce graphs like the one in Figure 3.2. These graphs will be later used as basis of comparison between observed and model – predicted values.

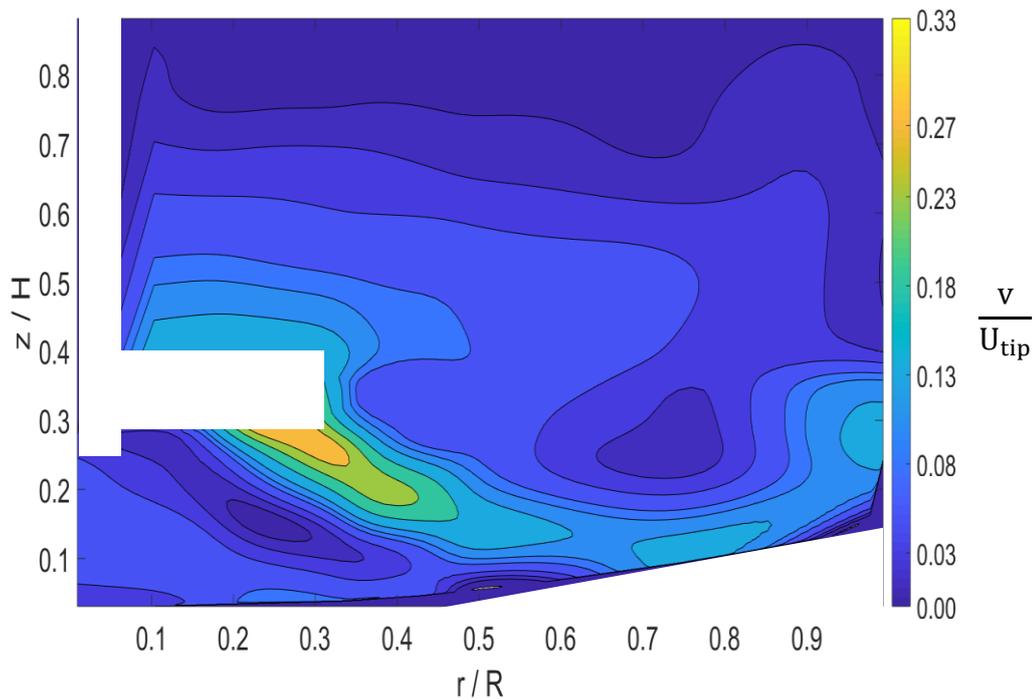


Figure 3.2 - Example of a graph obtained from PIV experimental data

3.2 ANFIS MatLab® Toolbox

This Neuro – Fuzzy Designer toolbox has several interesting features which were investigated during this thesis work. It was implemented in order to model multiple data batches, varying simulation parameters and characteristics from time to time to further understand the power of the training optimisation algorithm. First, an overall description of this soft computing modelling tool is given, then a description of the fixed constraints adopted during modelling simulations is defined. In Figure 3.3 the interface of the

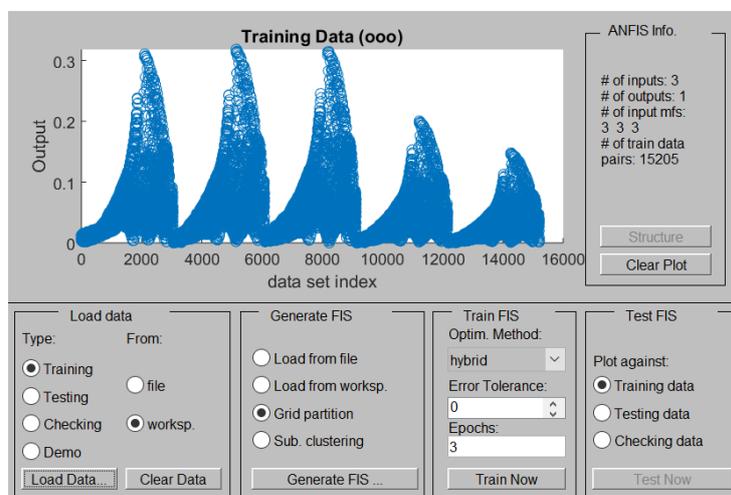


Figure 3.3 – Neuro Fuzzy designer MatLab toolbox interface

toolbox is represented. A plot is shown, presenting data set index as x – axis and Output (dimensionless velocity) as y – axis. A specific procedure has to be followed for each modelling situation. This is described through steps.

- 1) *Data organisation from file*: data are elaborated from text files, picking the ones with “*isValid*” equal to 1 and calculating velocity field norm for each position from its two components.
- 2) *Training and Validation matrixes*: once ready to be inserted in the modelling interface, data are split into two matrixes: Training Data and Validation Data. Their proportion varies along the next chapter, since also the number of employed training data is investigated in terms of correlation effectivity.
- 3) *Training data on the interface*: they are loaded from the workspace. As soon as they are loaded, they are graphically represented in terms of output value against its dataset index, as in Figure 3.3. The program automatically recognises the inputs as the first columns of the matrix and the output as the last one. It must be noticed that this algorithm only supports one output at a time.
- 4) *Sugeno FIS Generation*: once data are correctly loaded into the toolbox, an initial Fuzzy Inference System must be either loaded from file, if already existing, or created from scratch. In this work, inference systems were directly generated from this program. In Chapter 3, it was already mentioned that FIS in this context must respect some constraints. First, only Sugeno – type FIS can be used. Secondly, they must present only one output. Other features can be decided from the user. When it comes to generating a FIS from this interface, it can be achieved by adopting Grid

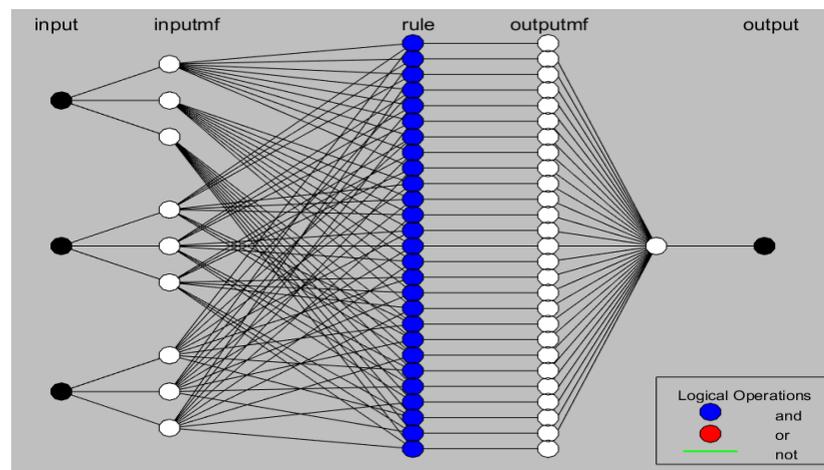


Figure 3.4 – Selection of number and type of membership functions

Partitioning or Subtractive Clustering. Grid partitioning was proved as more suitable for this case, since it gives lower processing time and more accuracy compared to clustering technique. Since data are homogeneously dispersed, it is not feasible to obtain a good classification using clusters. In addition to this, even though Grid Partitioning method is considered to be “Greedy” (Al-Mahasneh, Rababah, & Ma’Abreh, 2013), as it adopts many membership functions and rules to work properly, it was considered adequate since antecedents are only two to three in total.

Figure 3.5 – FIS structure generated from Grid Partitioning

A constraint regarding this method is met when type of membership function must be chosen. In fact, all inputs must be characterised by the same kind of MF. However, their number can be varied for each antecedent. Either zero or first order Sugeno FIS can be generated, so the output can be a constant or a linear function (Figure 3.4).

Immediately after choosing MF type and number, along with output MF type, the FIS automatically sets the rules. For each rule a beginning consequent membership function is yielded. FIS structure generated by grid partitioning is represented in Figure 3.5, where all five layers are visible. Finally, rules connective AND are implemented as product operator, while overall crisp output is gained with a Weighted Average of all rules’ outputs.

- 5) *Generated FIS Training:* after creating a new FIS, training settings must be imposed. First, optimisation method must be selected. In this module, one can choose between Hybrid method and Back Propagation, which were thoroughly illustrated in previous chapter. In this study, Hybrid method was always implemented, as it needs smaller processing time and it provides better consequent parameters changes using Least Square Estimator in its forward pass. In addition to this, iterative algorithm stop criteria must be decided, either through error tolerance or Epochs number. An epoch

corresponds all iterations which guarantees that all data set are presented to the FIS when trained before restarting the loop (J. S. R. Jang et al., 2005). In this work, error tolerance was always put equal to 0, whereas Epochs number was modified from model to model. When training is initiated, it proceeds showing graphically the root mean square error (RMSE) plotted against epochs number, as in Figure 3.6.

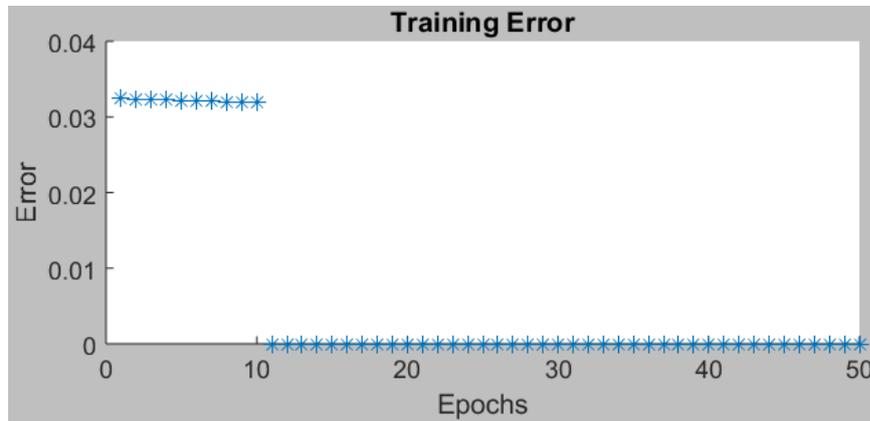


Figure 3.6 – Training error dynamically changing through data processing

Once desired epoch data is reached, iterative process stops and final error is presented both numerically and graphically, indicating in blue circles the observed values and in red asterisks the predicted values (Figure 3.7). This feature can serve the purpose of making a quick judgement over training process success. As a matter of fact, the more model outputs get to cover data sets, the better the trained model is. Nonetheless, accurate analysis on created models were done, through this study, using previously mentioned correlation indexes.

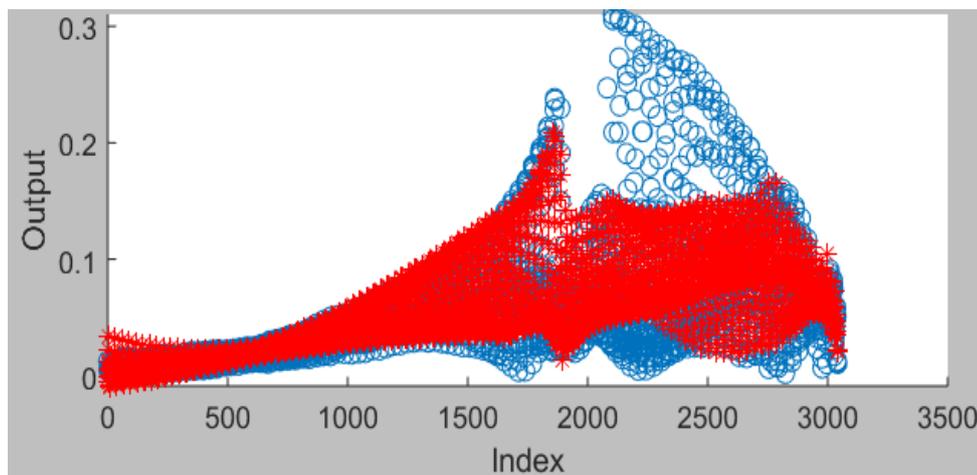


Figure 3.7 – Quick comparison between observed and predicted values from produced model

4. Velocity Field Modelling

In this section, different analysis will be shown towards modelling of data presented in the previous chapter. Graphical outputs will be reported in order to have a visual comparison between observed and predicted outcomes. ANFIS functionalities are investigated in its multiple features, in order to develop the best model possible which could correlate power input, radial coordinate and axial coordinate with velocity norm.

4.1 Comparison between Fuzzy and Neuro – Fuzzy Modelling with big data

In previous chapters, differences in terms of capability in manipulating certain moles of data were discussed. Fuzzy logic is more suitable for expertise – based inference systems, rather than correlation of great quantities of data. In this paragraph, a brief example is shown, in order to practically demonstrate Neuro – fuzzy power of adapting its parameters through its algorithms, respect to pure Fuzzy recursive procedures of correlation. In Table 4.1 characteristics of this comparison are shown.

Table 4.1 – Adopted conditions in a comparison between Fuzzy and Neuro – Fuzzy modelling performances

	Fuzzy	Neuro - Fuzzy
FIS	Mamdani	Sugeno
Input MF type	Triangular	Triangular
Output MF type	Triangular	Linear Functions
Input MF number	7	7
Output MF number	49	49
Number of Rules	49	49
Training Method	Particle Swarm	Hybrid
Epochs	100	100
RMSE	0.0333	0.0063

In order to produce a quick modelling estimate, antecedents' number was limited to two, as relative spatial coordinates r/R and z/H , fixing power input value at 100 W/m^3 . Such dataset was split in two to obtain a training set and a validation set, for a total of 3040 data points each. However, correlation estimate was only based on training data correlation, looking at the root mean square error. First, FIS was set as Mamdani for Fuzzy and Sugeno for Neuro – Fuzzy. Membership functions were created equal for both inference systems, as triangular – shaped. Both inputs have seven MF. Since ANFIS grid partitioning automatically yields several rules, thus output functions, equal to the product between input MF number, this was also given as specification for Fuzzy training algorithm. Epochs number was fixed at 100. In addition to this, while Neuro – Fuzzy employed hybrid technique (Back Propagation and least squares method), Fuzzy used particle swarm method. Differences were exploited in terms of root mean square error. Other correlation indexes did not take part in such analysis, since RMSE already underlines differences in terms of correlation performances. As expected, Neuro – Fuzzy outperformed Fuzzy, showing a RMSE one order of magnitude smaller. This is probably because while Fuzzy has few slow algorithms available, Neuro – Fuzzy can integrate efficient learning methods from Neural Networks. Moreover, while Neuro – Fuzzy shows one set of non – linear parameters (consequent), Fuzzy has both antecedents and consequents parameters a non – linear.

4.2 Comparison on performance between different types of MF

This section defines which membership function shape best suits velocity field datasets. As a matter of fact, there is no proof of which one them is the highest – performing during modelling. In Table 4.2, MF are shown in terms of dependence of membership degree from input data. Each one of them was implemented in a raw model. The simplest class of fuzzy functions is composed by these which are built with the use of only straight lines (J.-S. R. Jang & Gulley, 2015). The first one to be enunciated is the *Triangular*, obtained as a set of three dots, having three modifiable parameters a , b and c . If it is cut at the top, the *Trapezoidal* comes out as function, having no more just one point where membership degree equals one, but a range, with four modifiable parameters a , b , c and d . Another class of functions comprehends the ones constructed based on *Gaussian* distribution curve. Specifically, the conventional one, with two modifiable parameters c (curve centre) and σ (standard deviation). There is also a union of two different curves forming a two – sided composite one, with double the number of modifiable parameters of the single one, thus four. Next, there is an important type of membership function, represented by the *Generalised Bell*. These are quite common in reproducing fuzzy sets, as they are smooth and have a concise expression. They have three modifiable parameters a , b and c . Both Gaussian and Generalised Bell have the peculiarity of giving non null

outcomes on the whole input range. Then, *Sigmoidal* functions have the peculiarity of being able to be asymmetric and closed. In this case, difference (*D – sigmoidal*) and product (*P – sigmoidal*) of two sigmoidal curves are examined. Finally, *Pi* functions are spline – based, with a total of four modifiable parameters a , b , c and d .

Table 4.2 – Examined membership functions' shapes

Membership Function	Mathematical Expression
Triangle	$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$
Trapezoid	$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$
Gaussian	$\mu(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$
Double Gaussian	$\mu(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$
D – Sigmoidal	$\mu(x) = \frac{1}{1+e^{-a(x-c)}}$
P – Sigmoidal	$\mu(x) = \frac{1}{1+e^{-a(x-c)}}$
Generalised Bell	$\mu(x) = \frac{1}{1+\left \frac{x-c}{a}\right ^{2b}}$
Pi	$\mu(x) = \begin{cases} 0, & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1-2\left(\frac{x-a}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & b \leq x \leq c \\ 1-2\left(\frac{x-c}{d-c}\right)^2, & c \leq x \leq \frac{c+d}{2} \\ 2\left(\frac{x-d}{d-c}\right)^2, & \frac{c+d}{2} \leq x \leq d \\ 0, & x \geq d \end{cases}$

After giving a brief description of the function in use, definition of modelling circumstances is necessary. Produced models comprehend three antecedents: power input, relative radial and axial coordinates, presenting six membership functions for each spatial coordinate and five membership function for the power input. These yielded a total of 180 rules and consequently 180 output linear functions. Epochs number was maintained at 100, since convergence was always met at around 60 epochs. Datasets coming from each power input were condensed in one batch, containing more than 30000

data points. This was split in two parts by alternatively picking rows from this matrix, in order to obtain a training and a validation data matrix. Results are exposed in terms of three correlation indexes, defined in previous chapters, plus a newly created fuzzy index. These were measured for correlation of training data and validation data, in order to notice if difference between representation of these two different batches is remarkable. Once training data are well followed by trained model, this must also respect validation data trend, in order to be considered predictive towards data which were not presented during training process. Figure 4.1 – 4.4 present obtained results, with two columns indicating training and validation indexes outcome.

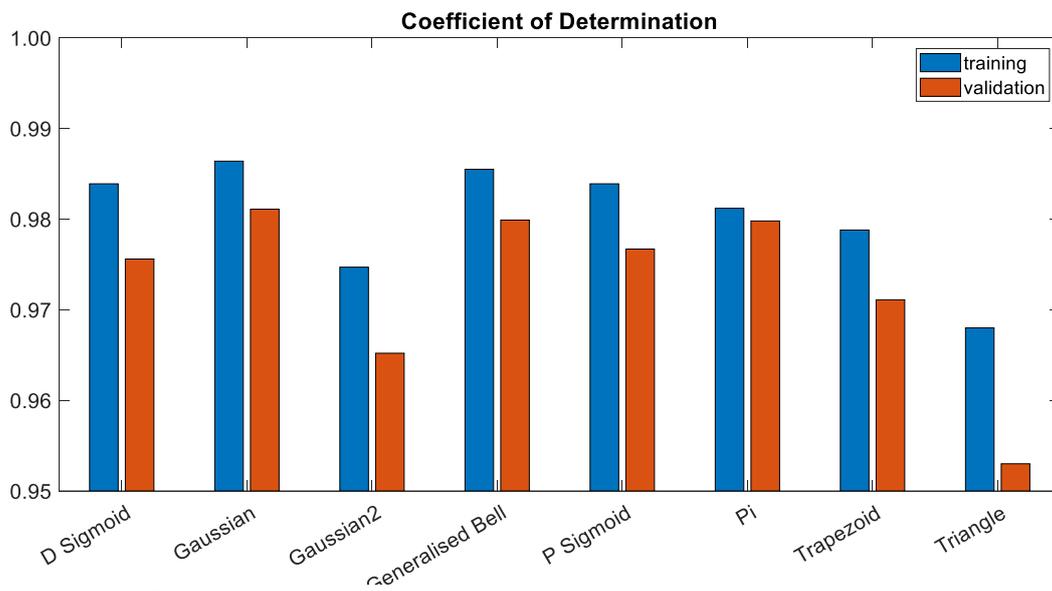


Figure 4.1 – Coefficient of determination of models presenting differently shaped membership functions

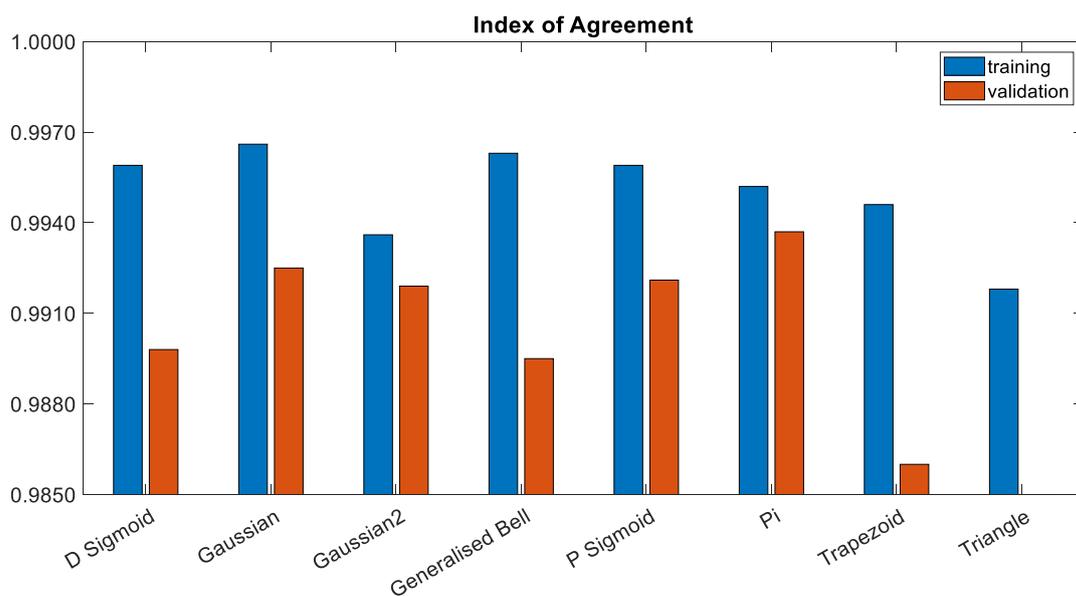


Figure 4.2 – Index of agreement of models presenting differently shaped membership functions

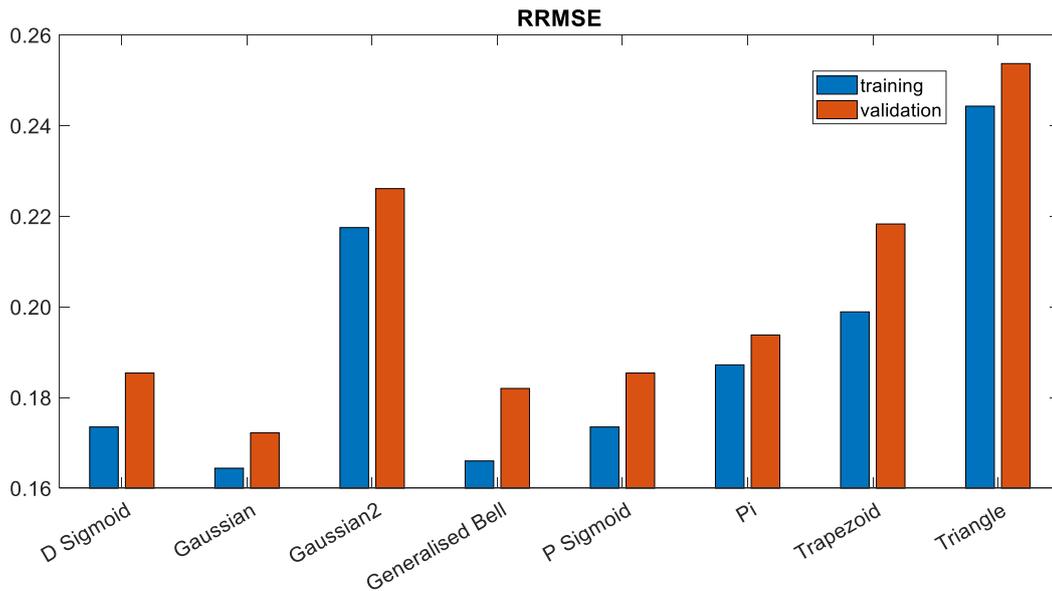


Figure 4.3 – RRMSE of models presenting differently shaped membership functions

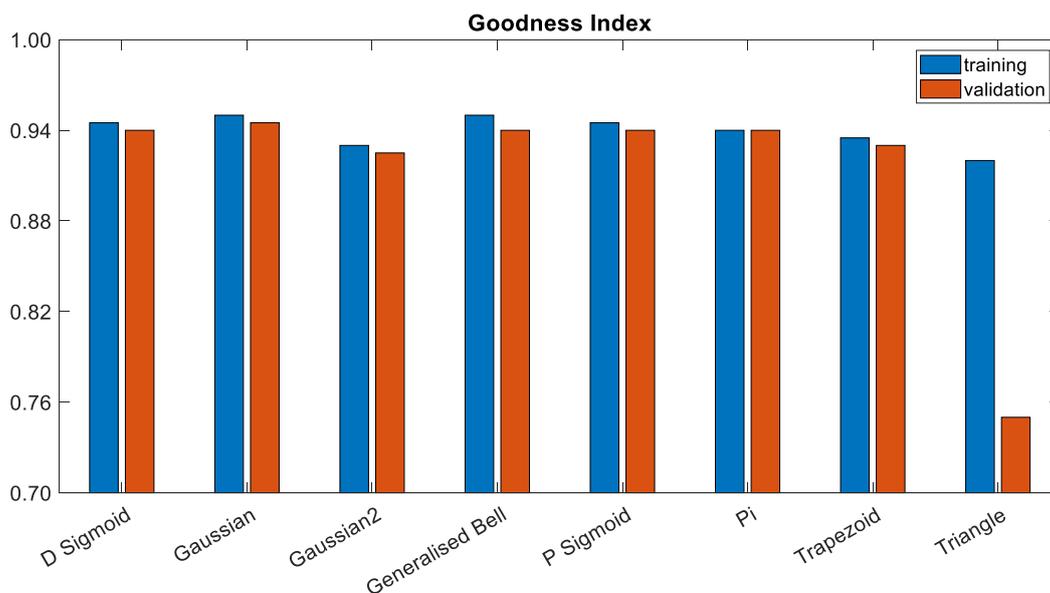


Figure 4.4 – Goodness index of models presenting differently shaped membership functions

All models show very high Index of Agreement (Figure 4.2) and Coefficient of Determination (Figure 4.1), hence data behaviour is very well followed in every case. Small differences can be noticed on RRMSE estimates (Figure 4.3). This fact indicates that, however data trend is well represented by every model, some membership functions allow to reduce the distance between observed and predicted value. Gaussian and generalised bell give best overall results. Goodness index sums up previous observations on such three indicators. Even though only triangular membership functions are remarkably worse than the others in terms of correlation, gaussian and

generalised bell seem to give highest correlating performances. Validation indexes are very similar to training ones. It is a common rule of thumb to use validation data as different as possible from training ones, in order to better understand the predictivity of the model towards unseen inputs (Ii & Ground, 1998). However, it looks like in this case training data are very similar to validation ones.

4.3 Variation of quantity of training data

Since modelling showed satisfying results using an amount of training data equal to 50% of the total dataset, an analysis regarding the variation of such percentage was conducted. This was done in order to see how predictivity is affected by changing this parameter. In this case, membership functions were used only in the best performing shape, accordingly to previous paragraph results. Although Gaussian and Generalised Bell present almost equal estimates, Gaussian curve was chosen to be employed. In fact, it has the advantage of being computationally less heavy to manipulate because of the lack one modifiable parameter compared to the Generalised Bell. However, in certain case this characteristic allows the bell to adapt better to non – linear data. Theoretically, the more a model is fed with training data, the more it should be able to predict unseen input values. On the other hand, too heavy training batch could lead a model to be too “Rigid” and stick too much to it, thus reducing predictivity. Percentage of training data on total was varied from 5% to 50%. Validation set percentage was changed accordingly to training set, picking all data points which were not used in training. Figure 4.5 - 4.8 show the trend of correlation indexes applied both to training and validation data. Critical improvement of all correlation indexes is obtained by increasing training data percentage from 5 to 10%. Figure 4.5 shows that coefficient of determination reaches an asymptote at 30%, with training correlation collapsing with validation at 50%. Figure 4.6 indicates that index of agreement reaches satisfying results even at 20%, with training curve meeting validation curve at already 40%. Up to now, this is not the best index to make comparisons, since it is always very high for such models. On the other hand, Figure 4.7 shows that a sharp decrease is met from 5 to 10% for RRMSE, with curves going to coincide at 20%. Lastly, in Figure 4.8, Goodness index reaches an asymptote at circa 0.95 in correspondence of 10%. Such fuzzy index is not sensitive to small variations of its antecedents. Hence, training and validation curves are coincident. In addition to this, it tells the user that a 10% on total is enough to obtain good modelling. It is quite impressive that by selecting one data point over 10 at a time, ANFIS can build a well – predictive model. Reasons for this fact could either be the adaptivity of such neuro – fuzzy inference systems, or the similarity between data points among the grid. Generally, it could be stated that if a 20% of training data on total is employed during the training, it is enough to achieve a good modelling over these PIV velocity field data, with a good

predictivity over unseen coordinates. Moreover, since there are asymptotes for all correlation indexes on validation data, no overtraining occurred. In addition to this, the employment of few training data, which has poor correlation consequently, is explained by the changes happening to antecedents' membership functions during the training process.

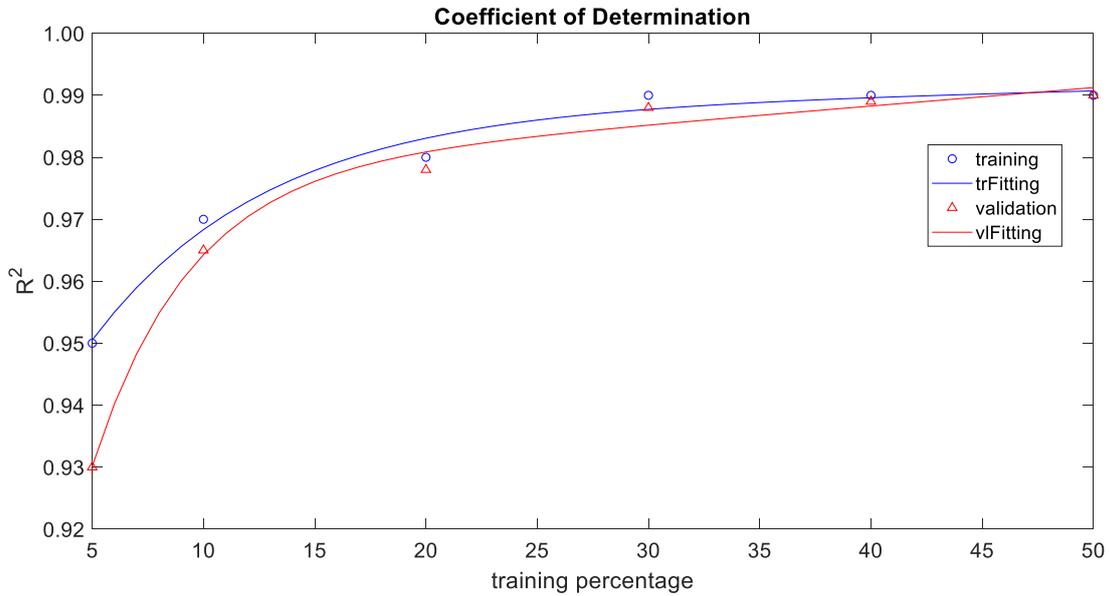


Figure 4.3 – Coefficient of determination of models with gaussian membership functions, trained with different percentages of data

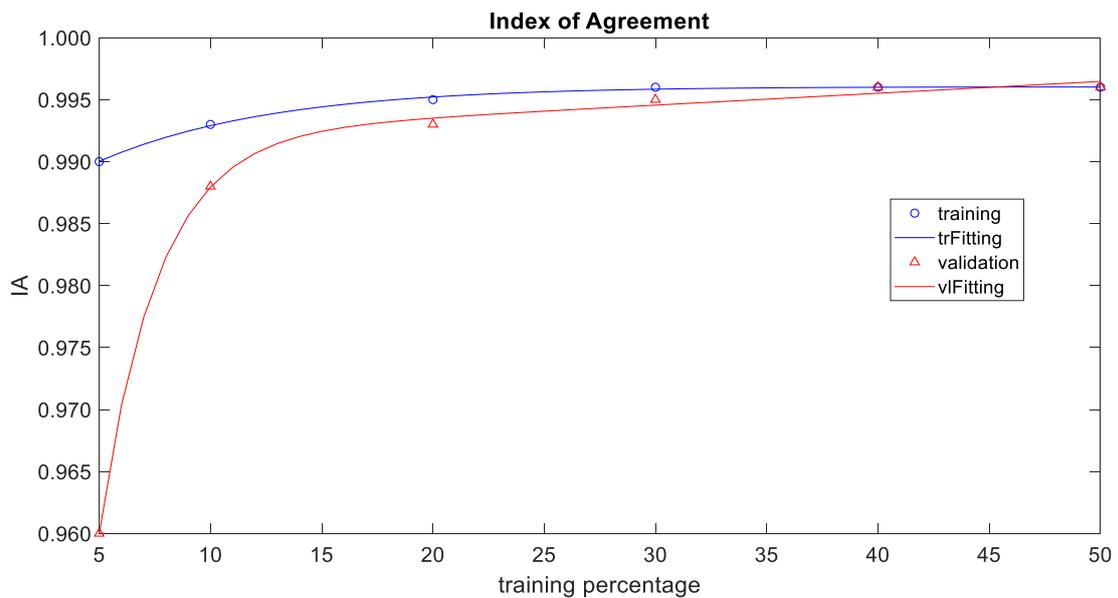


Figure 4.4 – Index of agreement of models with gaussian membership functions, trained with different percentages of data

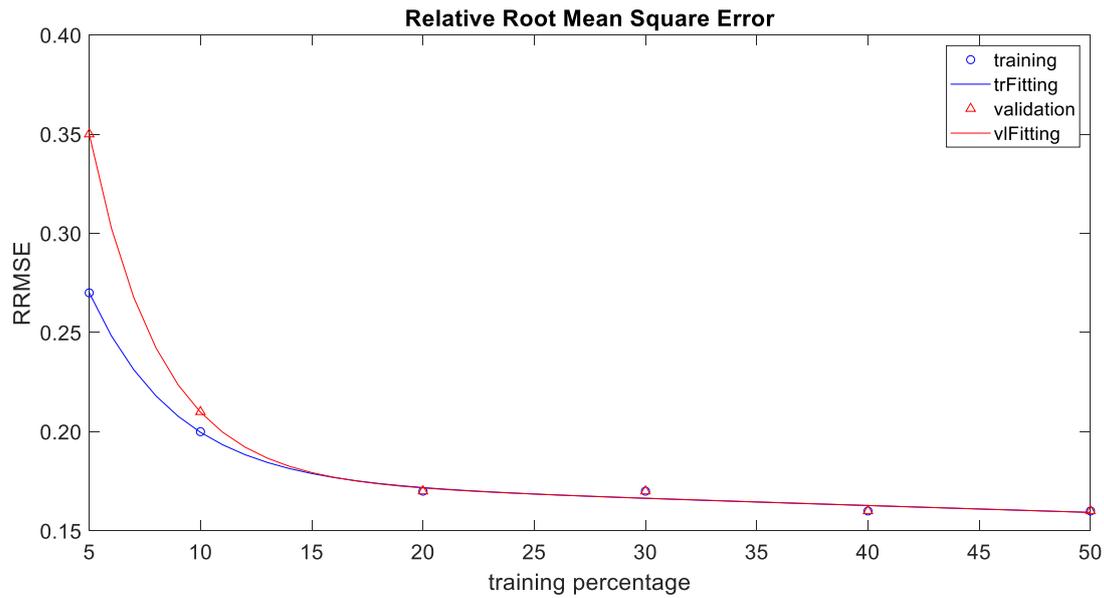


Figure 4.5 – RRMSE of models with gaussian membership functions, trained with different percentages of data

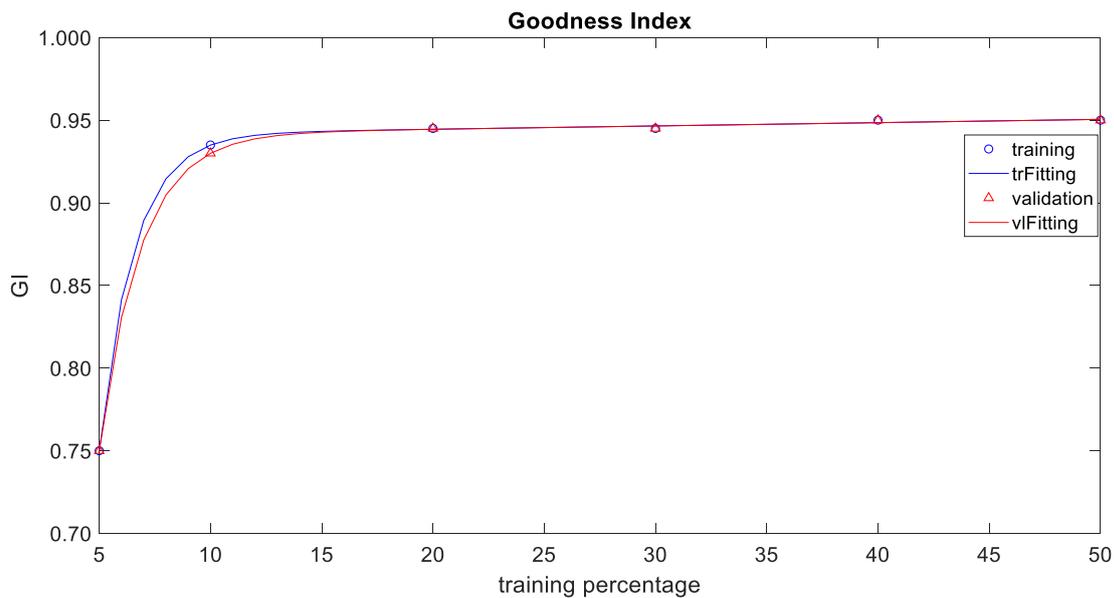


Figure 4.6 – Goodness Index of models with gaussian membership functions, trained with different percentages of data

In Figure 4.9, three antecedents' functions are reported for the raw model which was not subjected to training yet. These initial fuzzy sets are homogeneously distributed along the crisp range. This is because grid partitioning FIS generator creates equally spaced cells on the input planes. During training process, these regions adapt to training data in order to improve the representation of the model towards such data. In Figure 4.10, a

side by side comparison is presented between models obtained respectively by 5% and 30% training data batches processing. First, even after training, power input fuzzy sets remained unchanged for both models. Since power input data only refer to 5 values, these are not enough to significantly influence the grid structure. Next, for second and third antecedents, referring to the radial and axial coordinates, there are visible differences between 5% and 30% models. In fact, while 5% presents its membership functions almost completely unvaried, 30% shows remarkable swings.

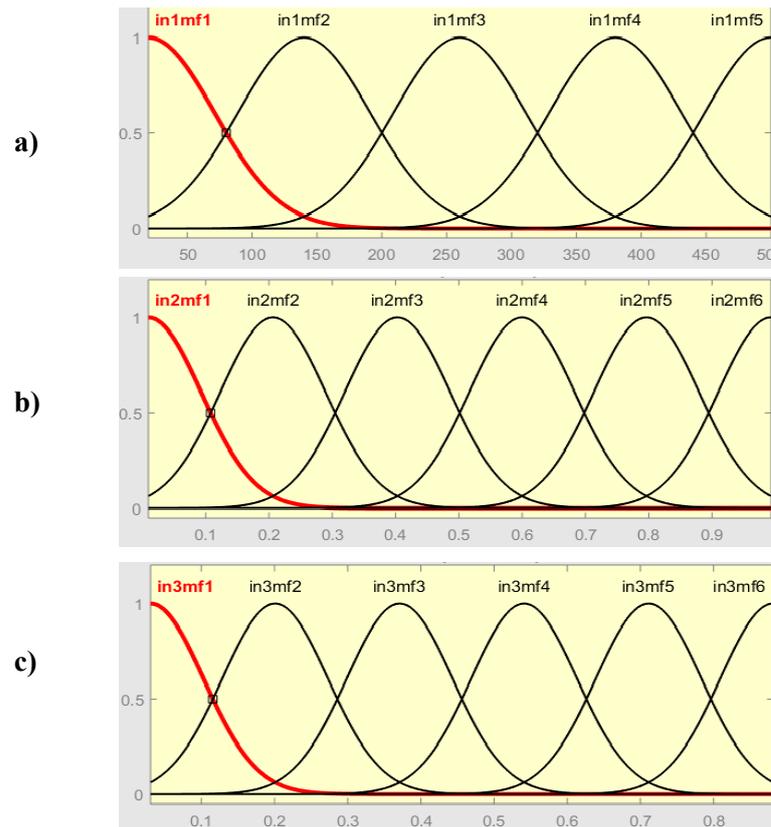


Figure 4.7 – Raw model antecedent membership functions: a) Power input; b) Radial coordinate; c) Axial coordinate.

Every membership function varied its domain and some of them even modified their shapes, especially second and third membership functions for both spatial antecedents. Radial antecedent sees a tightening of MF in the interval between 0.2 and 0.5, while axial antecedent between 0.2 and 0.4. It is evident how the algorithm has the ability of intensifying the grid where it is needed the most, which in this case is defined as the zone around the impeller. While other membership functions enlarge, the ones concerning such area become thinner. This is due to necessity of defining highly non – linear variations of velocity field which occur close to the agitator blades. Neuro – fuzzy models need to adapt to inputs trend in a certain manner in order to give proper outcomes, this can be achieved only with the correct amount of training data.

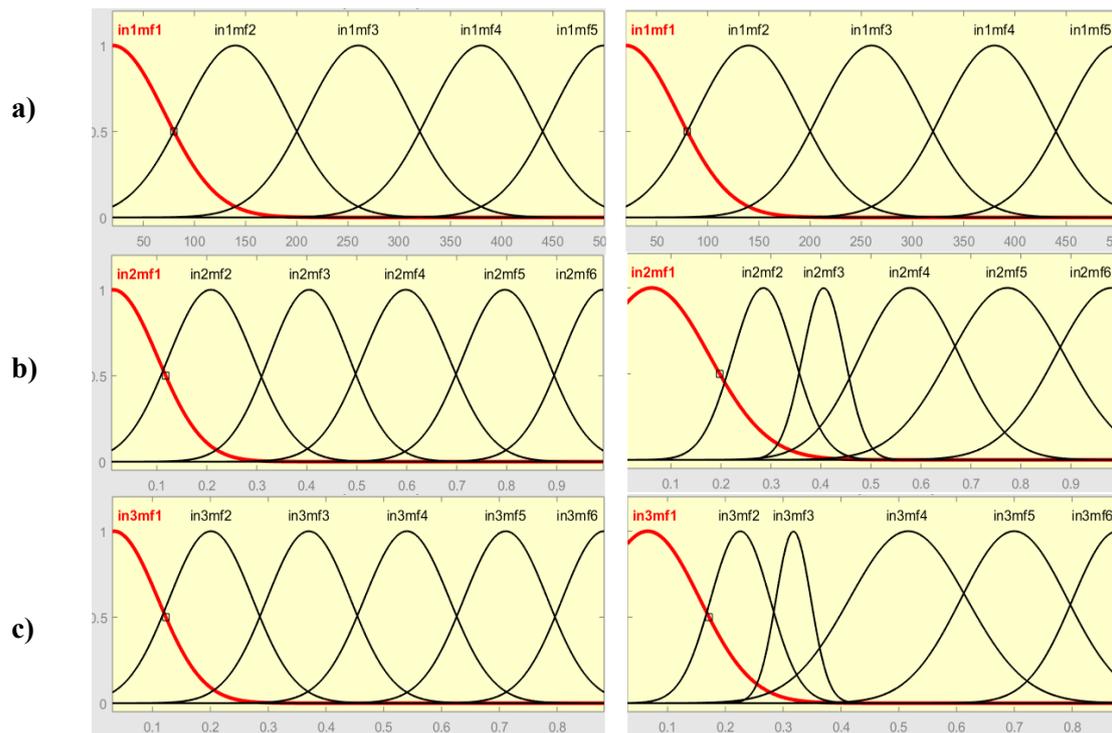


Figure 4.8 – Comparison between 5% (left) and 30% (right) trained models in terms of membership function adaptivity: a) Power input; b) Radial Coordinate; c) Axial Coordinate.

4.4 Alternative approach for modelling

In previous paragraphs, different models were built by directly training them with three antecedents. In this section another way to build a robust model is shown. This had the purpose to see if it is possible to obtain a more useful model, compared to the previous ones, with the same number of membership functions. However, in this case different types of training were performed, in order to explore also ANFIS characteristics. First, a model projecting only two inputs, r and z in their original dimension (mm), was produced, starting from data regarding only power input equal to 20 W/m^3 . In this case, seven gaussian membership functions were employed for each antecedent, instead of six. This means having 49 rules (and output functions) instead of 36. Training was made with 20% training data over the total of 20 W/m^3 data batch. However, after training process this model showed strange membership functions adaptation. In fact, one membership function of radial coordinate got to cover the whole range, while another one on the axial dominion almost erased itself from the graph. Therefore, these two were thought as unnecessary and were cut off. Along with these, also rules employing such functions were erased, going back to having a 6/6 membership functions inference system. Figure

4.11 shows the applied modification. On the left graphs, useless functions are highlighted in red and on the right, they are not present.

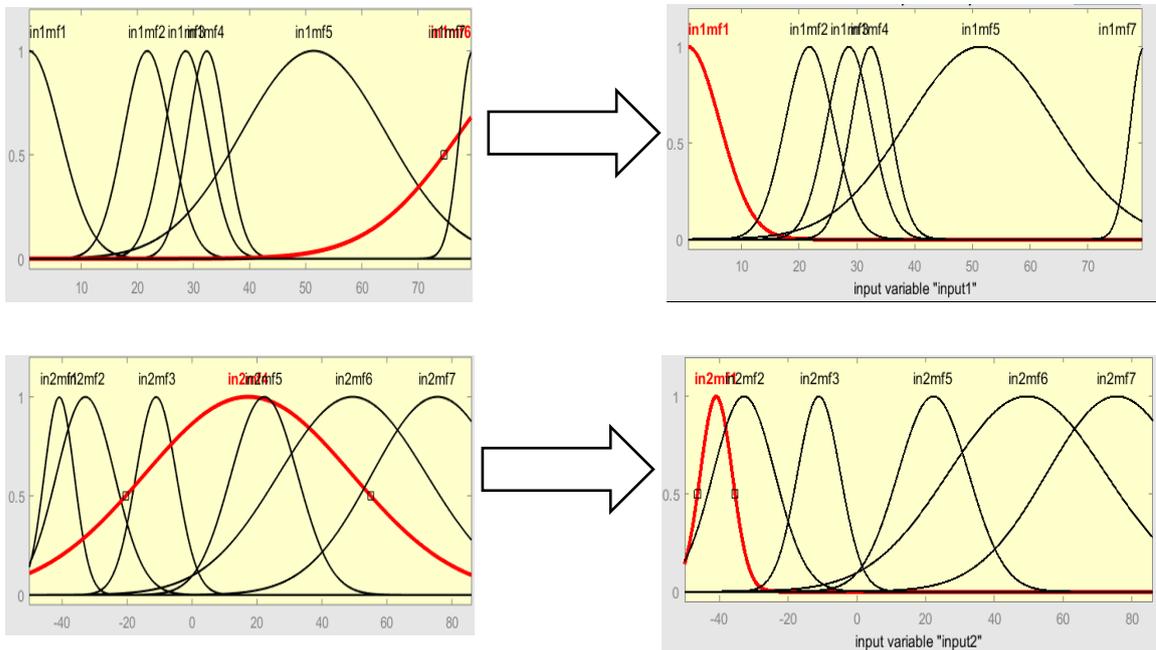


Figure 4.9 – Passage from 7/7 model (left) to 6/6 model (right) erasing membership functions highlighted in red on the left

Correlation performances of these two models are reported in Figures 4.12 – 4.15.

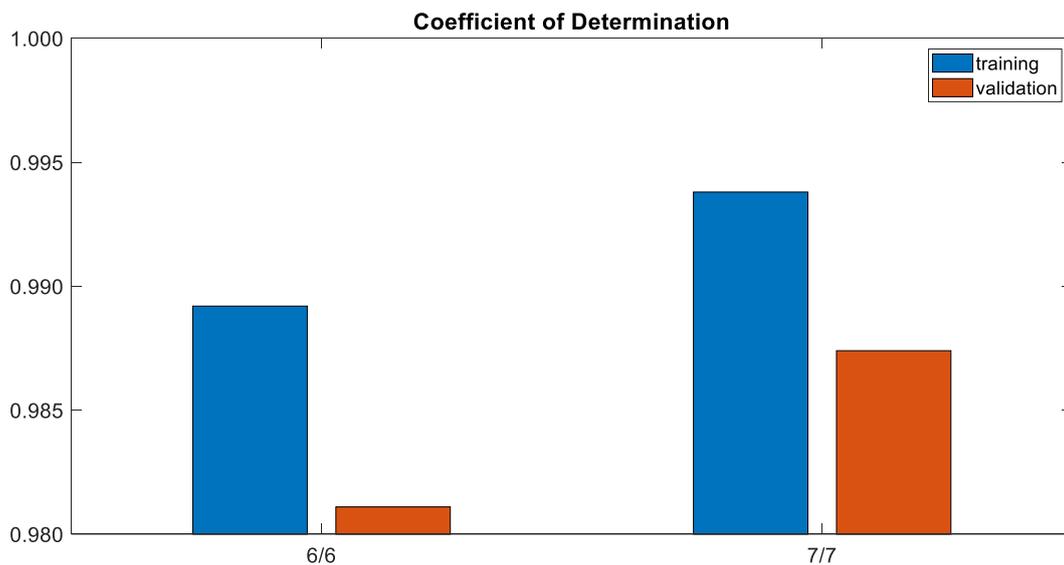


Figure 4.10 – Coefficient of determination of 6/6 model (left) and 7/7 model (right)

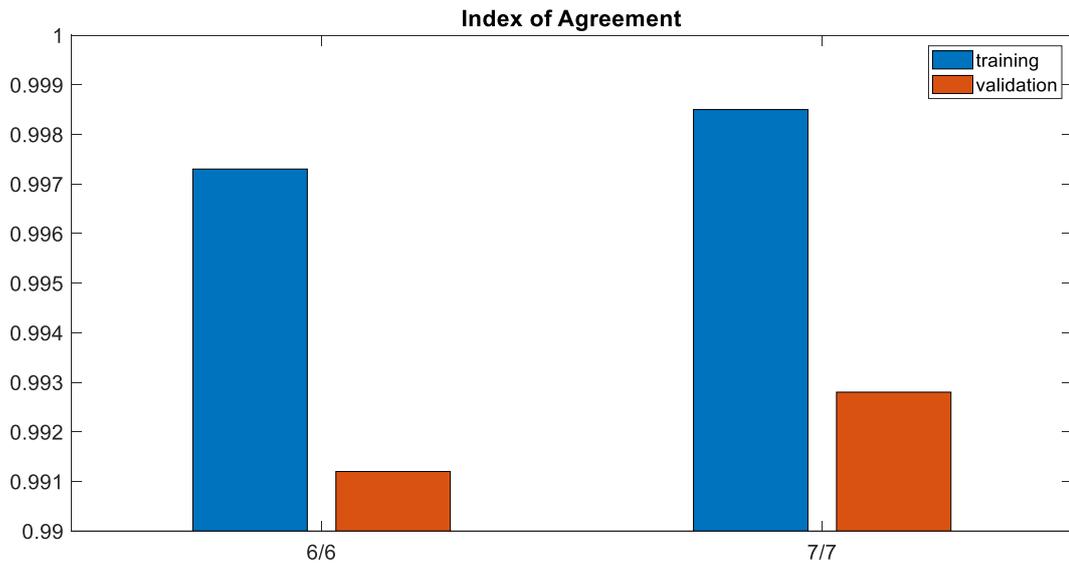


Figure 4.11 – Index of Agreement of 6/6 model (left) and 7/7 model (right)

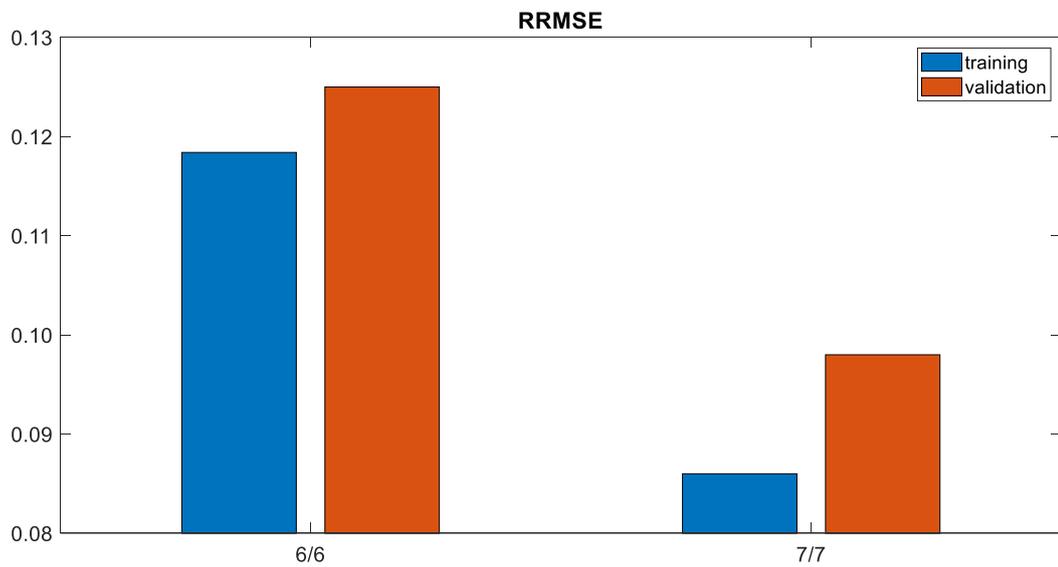


Figure 4.12 – RRMSE of 6/6 model (left) and 7/7 model (right)

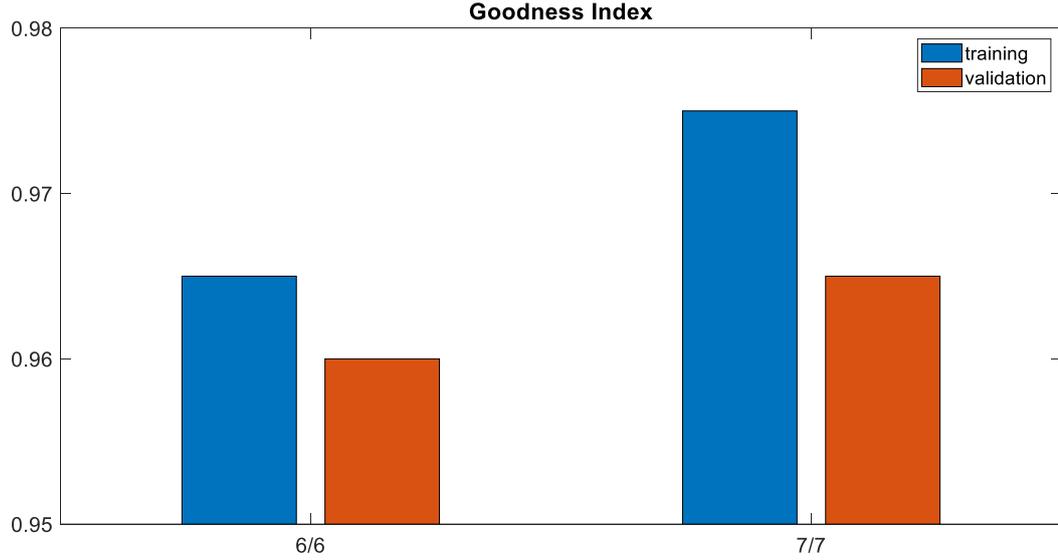


Figure 4.13 – Goodness index of 6/6 model (left) and 7/7 model (right)

As expected, correlation usefulness did not decrease considerably. In fact, only RRMSE shows a small worsening from 0.9 to approximately 0.12. Furthermore, rules were taken to 36 from 49 (with them also output functions). This could be the proof that more than 6 membership functions per position input are not necessary. Moreover, computational cost is reduced when using less membership functions, speeding up convergence during training process. Since previous models were directly trained with dimensionless radial and axial antecedents, it was thought as interesting to see what happens if this procedure is applied starting from dimensioned position antecedents and velocity consequent. After this, it was thought of converting them again into dimensionless values (Equation 4.1).

$$\begin{cases} \tilde{r} = \frac{r}{80} \\ \tilde{z} = \frac{z + 55}{160} \\ \tilde{v} = \frac{v}{u_{tip}} \end{cases} \quad (4.1)$$

These three parameters are made dimensionless by making them respectively relative to vessel radius, vessel height and tip velocity. Besides, axial coordinate was converted in positive values, since originally data takes height 0 as the impeller position. The exposed procedure taking from seven to six membership functions was repeated with these modified data. Surprisingly, results in terms of correlation goodness turned out to be worse. This is due to the fact that, as occurring with neural networks, neuro – fuzzy modelling solutions depend a lot from the starting guess of modifiable parameters (J. S. R. Jang et al., 2005). Next, it was discovered that, by converting antecedents' parameter

accordingly to the change of range of each antecedent (Equation 4.2), correlation efficiencies could get back to previous higher values.

$$\text{radial} \rightarrow \begin{cases} \tilde{\sigma} = \frac{\sigma}{80} \\ \tilde{c} = \frac{c}{80} \end{cases} ; \quad \text{axial} \rightarrow \begin{cases} \tilde{\sigma} = \frac{\sigma}{160} \\ \tilde{c} = \frac{c}{160} \end{cases} \quad (4.2)$$

All correlation efficiencies are very promising for these two inputs models. In fact, as Figure 4.16 shows, produced model follows experimental values very well. Four graphs are shown, each one indicating four different fixed radial position, with velocity field (x – axis) trend with axial position (y – axis).

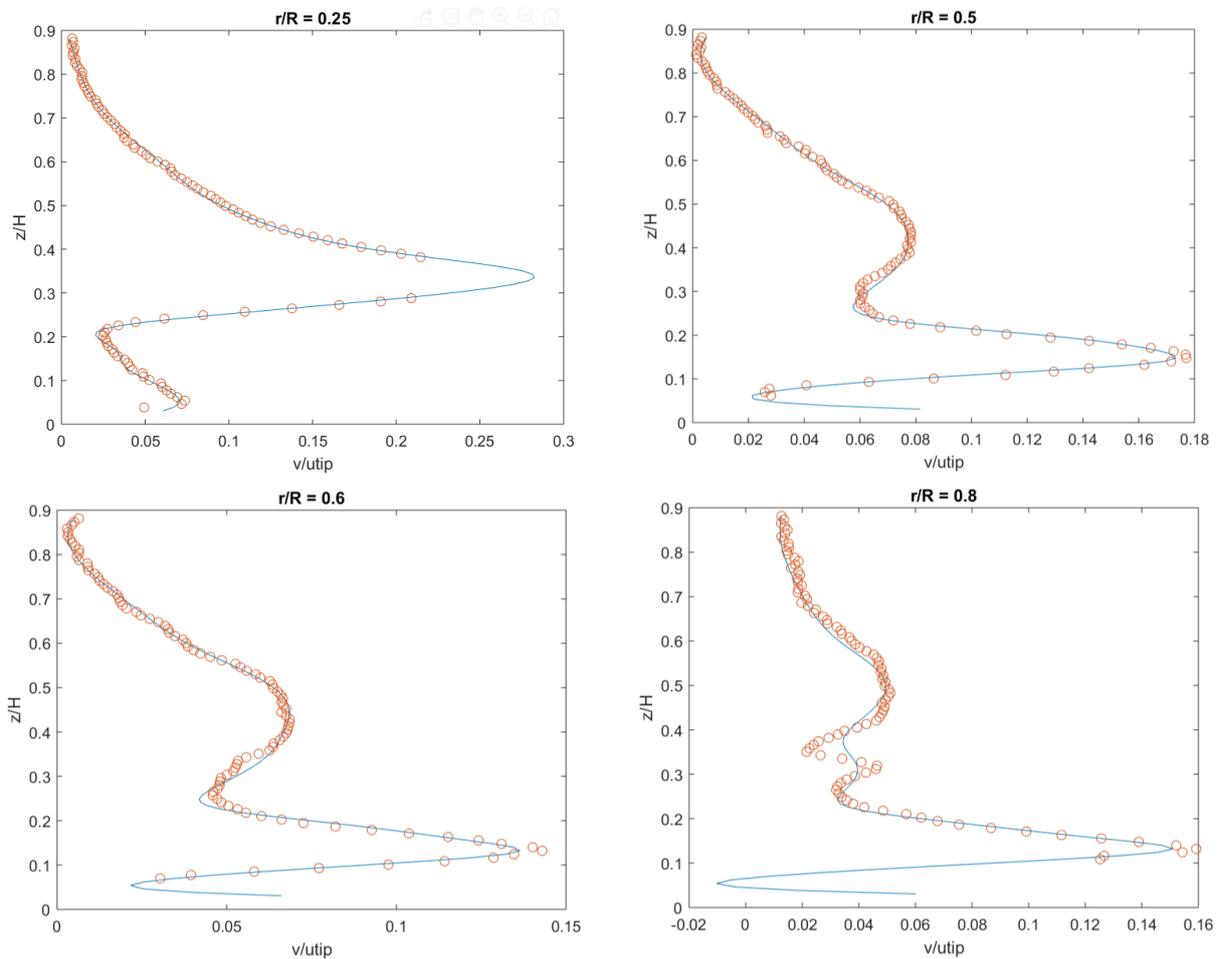


Figure 4.14 – Comparison between experimental values and two antecedents model predicted values

After having produced such well – predictive model, next step consisted in adding power input as third antecedent. The aim of this pass was based on trying to maintain such achieved accuracy but forming a more complex system as the ones seen in previous paragraph. Thus, the antecedents regarding spatial coordinates were translated to the new

model as they were. Power input antecedent was implemented with five membership functions. Finally, training was done in order to see if starting from such premises could improve final model characteristics. Training batch was constituted by 10% of the total number of data. Hence, comparisons are presented against the three antecedents – model of previous section, trained with the same amount of data. However, after 100 epochs of training process, results did not prove to be as promising as expected. Figures 4.17 – 4.20 show comparisons in terms of correlation indexes between mentioned models. The model directly trained as a three antecedents’ model, indicated as Model 3, has the same training and validation correlation indexes of the other one, indicated as Model 2 – 3. This could be because, even if the starting point given to the hybrid algorithm is different, both raw models converge to the same minimum after training.

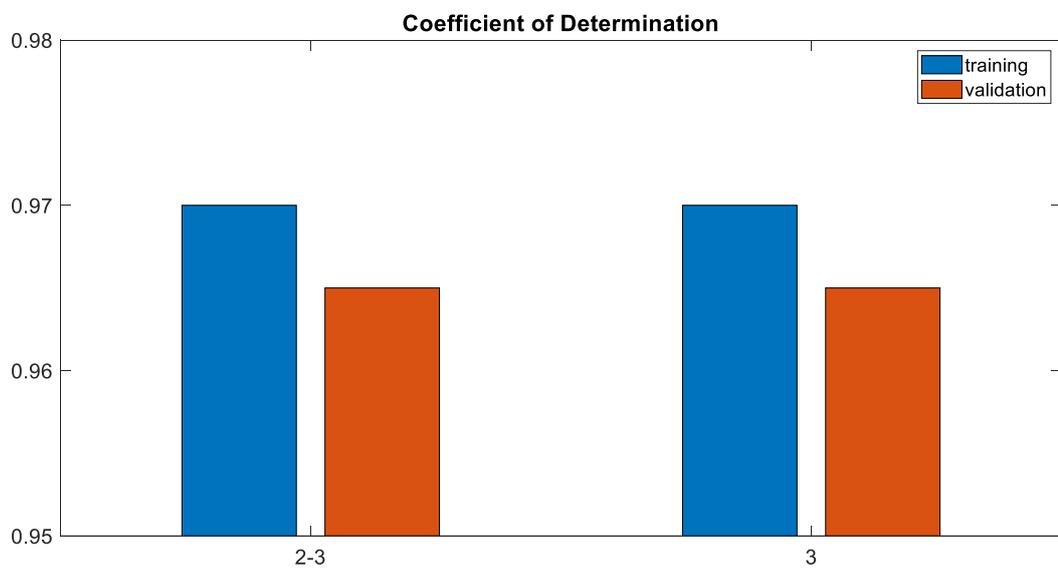


Figure 4.15 – Coefficient of determination of 2 -3 model (left) and 3 model (right)

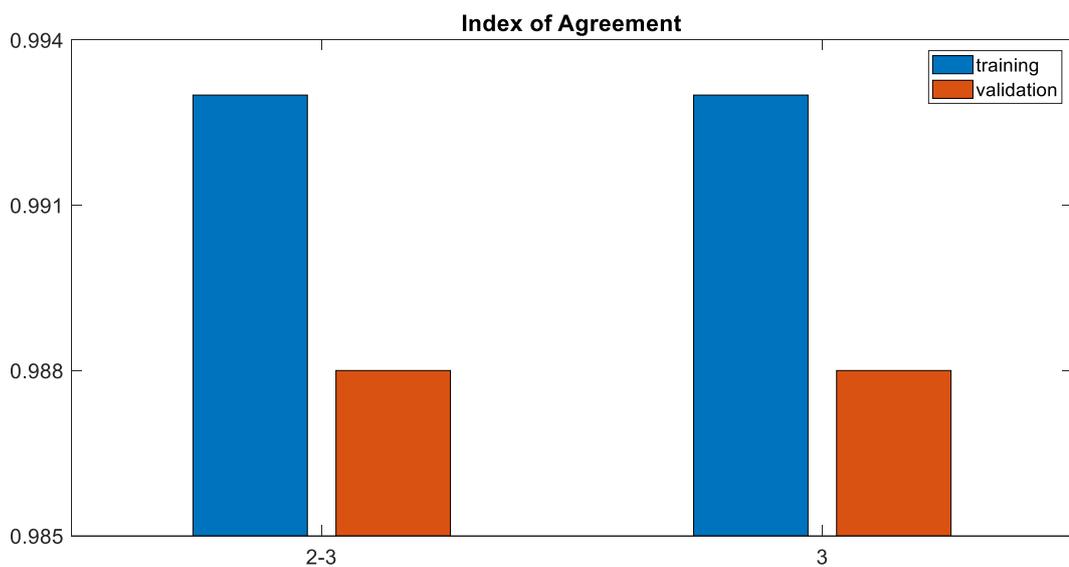


Figure 4.16 – Index of agreement of 2 -3 model (left) and 3 model (right)

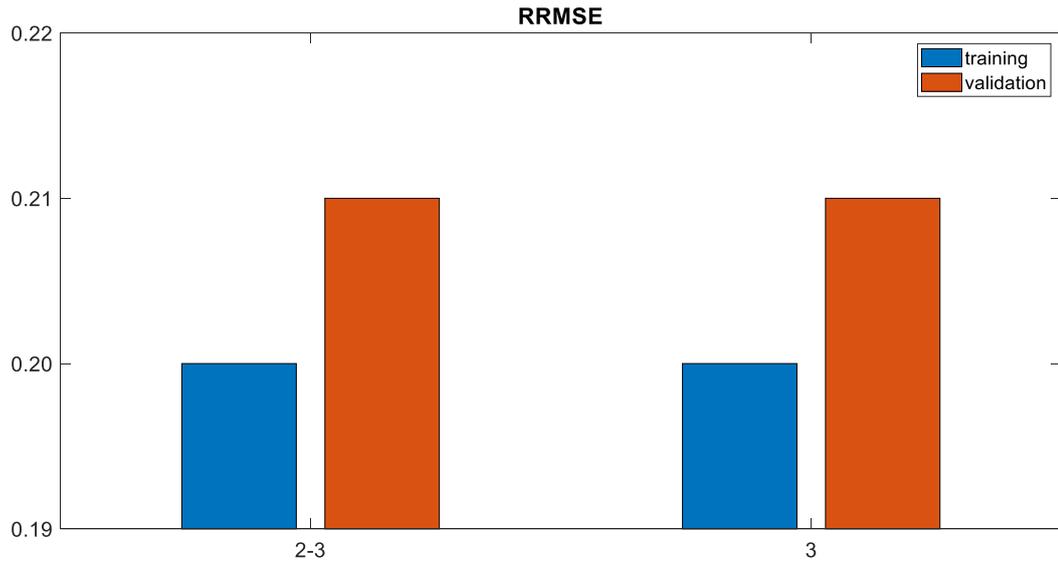


Figure 4.19 – RRMSE of 2 -3 model (left) and 3 model (right)

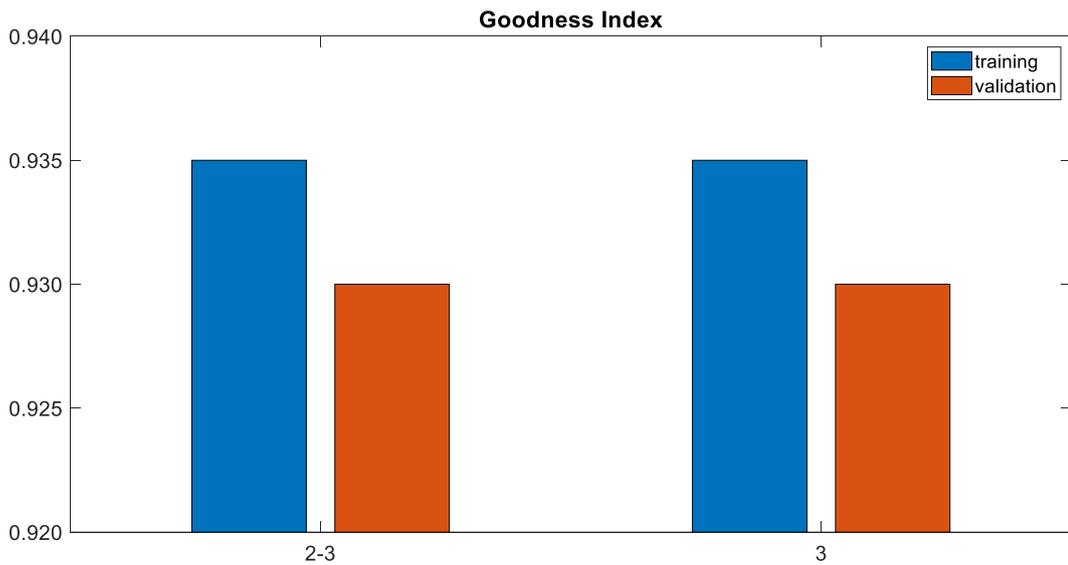


Figure 4.20 – Goodness Index of 2 -3 model (left) and 3 model (right)

4.5 Analysis on correlation and computational performances with variation on antecedent membership functions number

The aim of this section is to show how correlation and computational characteristics of models, during training and testing processes, are influenced by number of membership functions employed to define spatial antecedents. Fuzzy sets were fixed at 5 for power input antecedent, while they were increased equally for radial and axial antecedents from

3 to 7. This has the purpose of seeing if 6 membership functions are the best choice for such model inputs. In this case, not only correlation indexes are evaluated. In fact, considerations on training algorithms are made in terms of convergence and processing time. Furthermore, increasing complexity of the inference systems are discussed through the number of rules (thus output functions, since grid partitioning is used to create raw models). Figures 4.21 – 4.24 show correlation indexes behaviour with fuzzy sets increase, respectively R^2 in Figure 4.21, IA in Figure 4.22, RRMSE in Figure 4.23 and GI in Figure 4.24. Training and validation curves seem to keep their differences constant, as if increasing number of functions did not reduce such distance.

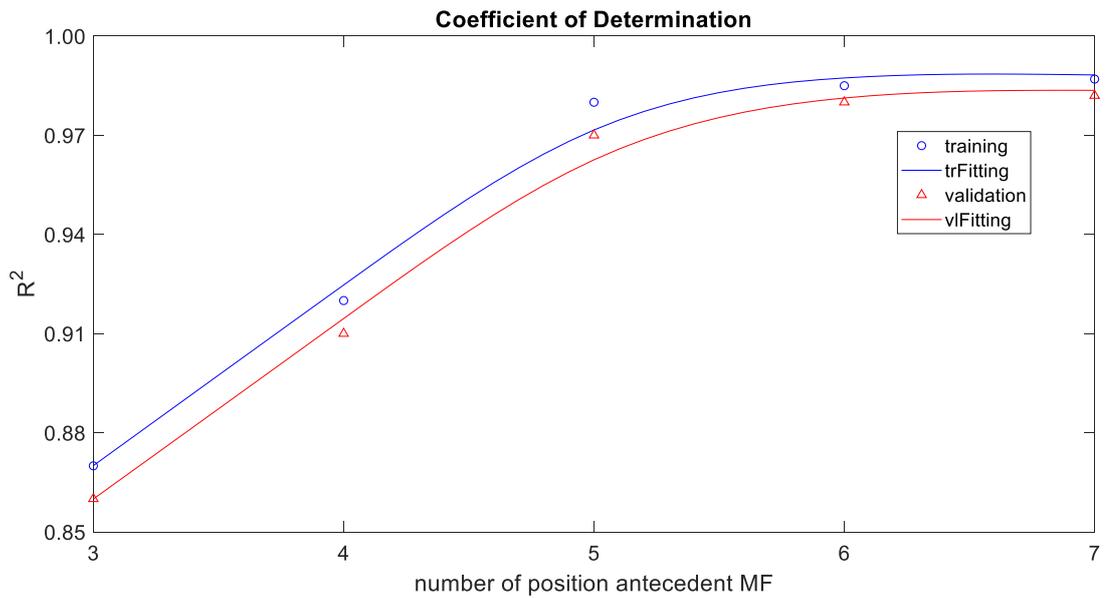


Figure 4.21 – Coefficient of determination of models with varying number of antecedents' membership functions

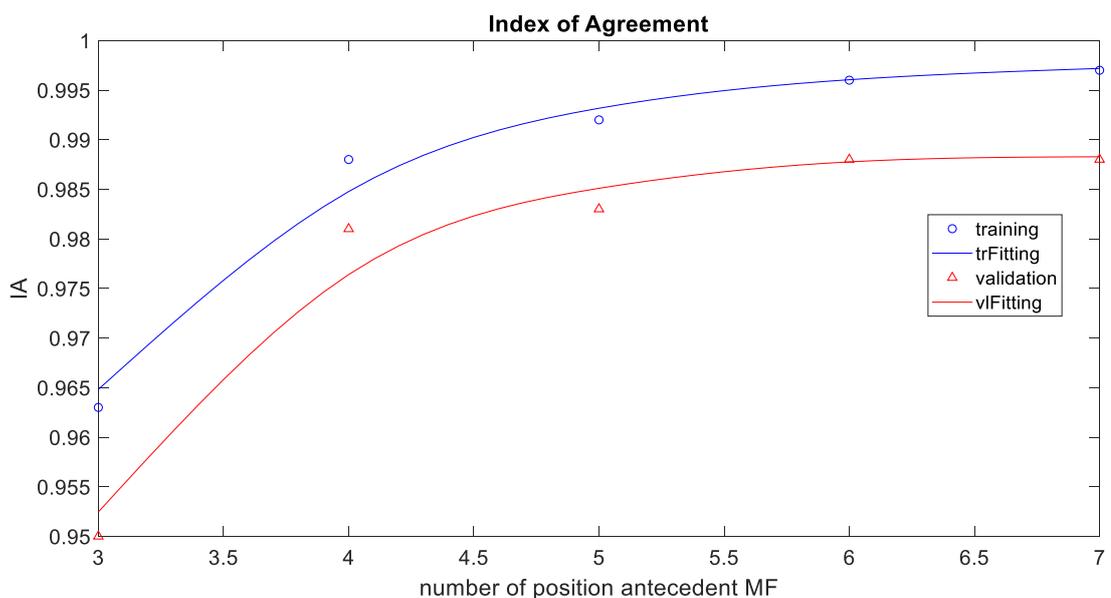


Figure 4.22 – Index of agreement of models with varying number of antecedents' membership functions

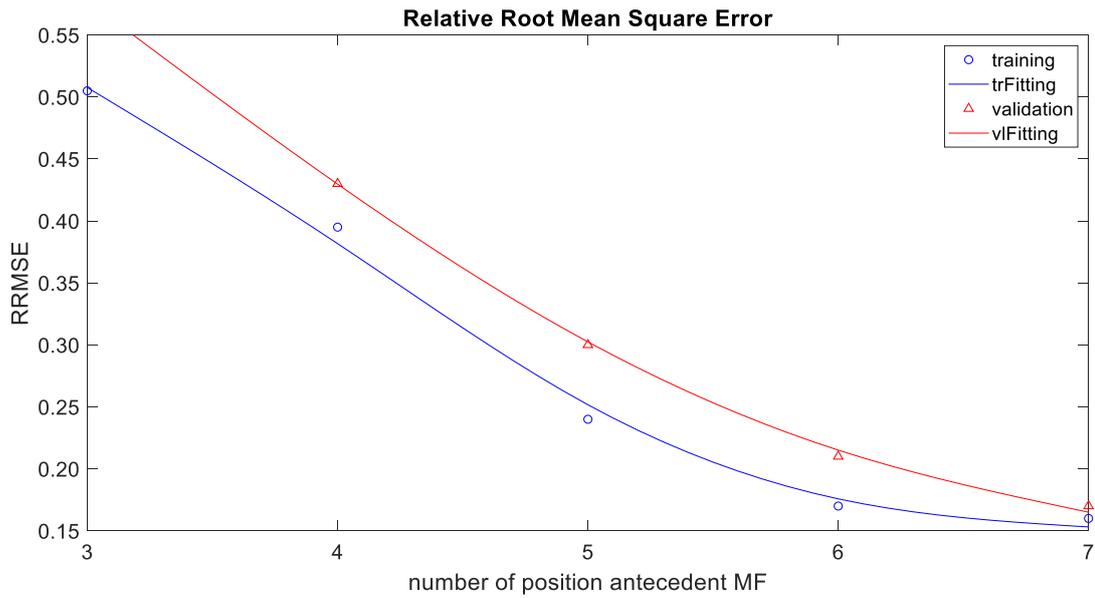


Figure 4.23 – RRMSE of models with varying number of antecedents' membership functions

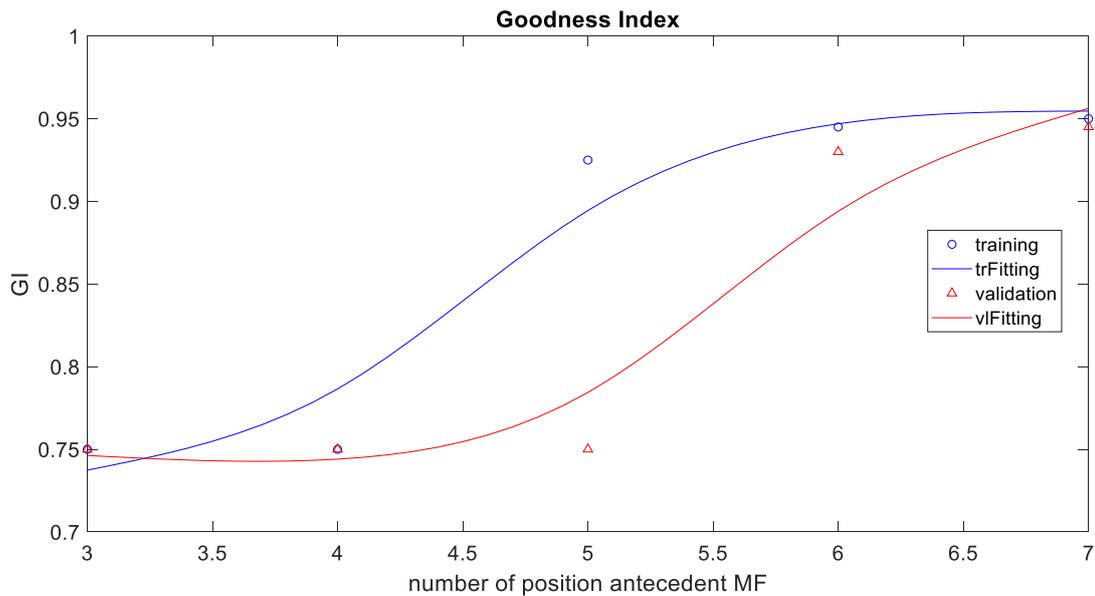


Figure 4.24 – Goodness index of models with varying number of antecedents' membership functions

In addition to this, there are asymptotic trends for all correlation indexes. As a matter of fact, they demonstrate a considerable improvement from 3 to 6 membership functions. On the other hand, the employment of 7 fuzzy sets for each of the two coordinates proves to be not so promising, as every correlation index did not improve considerably. These evidences reveal that 6 membership functions are already enough to give a good correlation of experimental values of PIV velocity fields. As highlighted in previous sections, it is always best to deal with fewer membership functions, in order to have a

more intuitive model. However, it is also desired to maintain correlation indexes at high standards. It would be remarkably useful to project mentioned data with only 3 membership functions. However, in this case they are not enough to build a good model. Such asymptotic behaviour is probably a reflection of functions to adapt their parameters over a certain crisp range. Modelling usefulness has a critical increase if fuzzy sets can be modified over their dominion. For this case in particular, seven membership functions are too many to be efficiently spread over crisp ranges. Moreover, training algorithm works better on consistently modifying premise parameters if their amount is not excessive. This would force the hybrid iterative process to change solely the consequent parameters, referred to the linear output functions. Another aspect which is wanted to be examined in this paragraph regards influence of such number of membership functions on computational aspects. These are reported in Figures 4.25 – 4.27. First, it was already mentioned that increasing antecedents' functions leads to a heavier and slower model. This is underlined in Figure 4.25. It is evident how by slightly increasing functions number, number of rules is sharply increased. If to use 3 fuzzy sets only 49 rules are needed, 245 rules are computed for 7 membership functions. Therefore, this is the first real cost of increasing number of antecedents' functions. In addition to all the issues discussed above, using too many rules would dramatically increase computation time for the inference system. This would lead to relevant problems in applications such as process control, where a prompt action is needed (Berardi, Chiaberge, Miranda, & Reyneri, 1996).

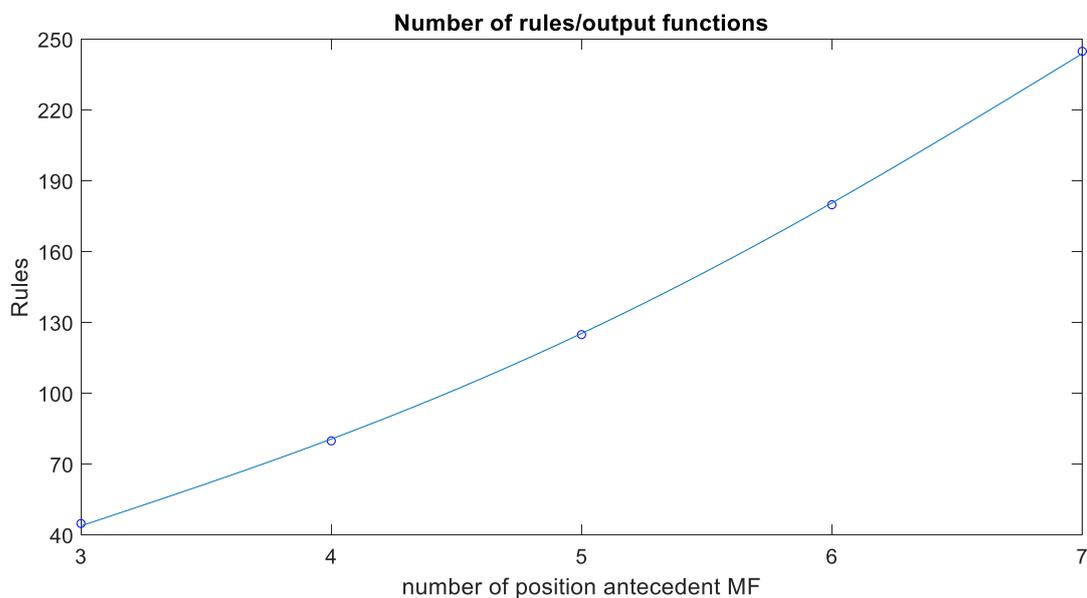


Figure 4.17 – Number of rules and output functions of models with varying number of antecedents' membership functions

Next, Figure 4.26 pictures the time needed for each epoch during training process, plotted against number of functions in position antecedents. Such duration rises almost exponentially at the increase of such amount, underlining again the major advantage of using 6 membership functions instead of 7. A much higher duration of epochs is clearly

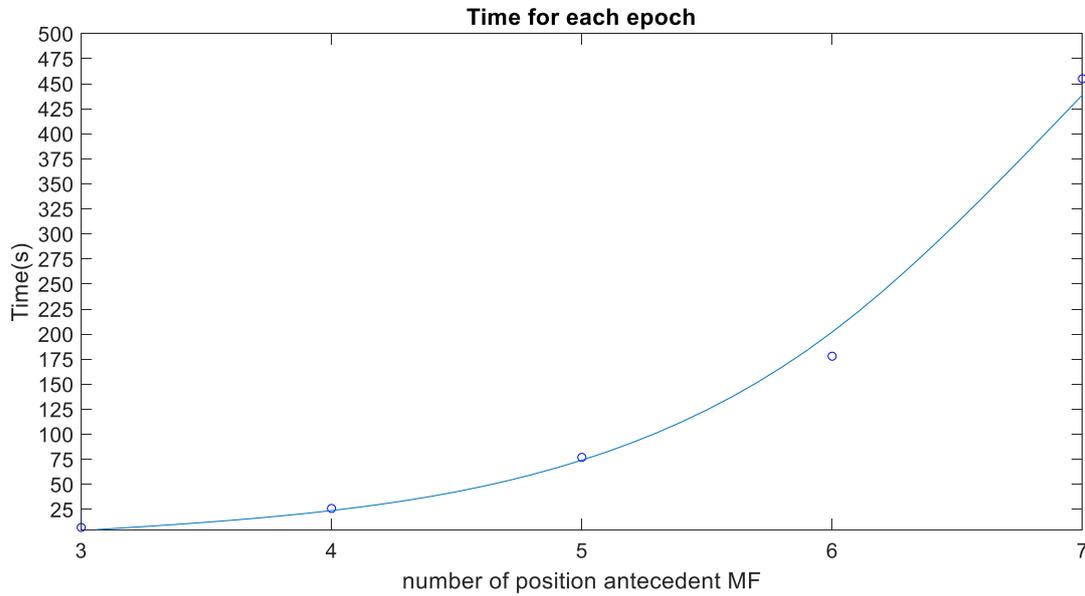


Figure 4.19 – Time for each epoch during training of models with varying number of antecedents' membership functions

connected to the amount of premise and consequent parameters which are iteratively modified during training process. In this case, time required for each epoch is related to adopted computer characteristics: 8.00 GB RAM and operative system at 64 bits.

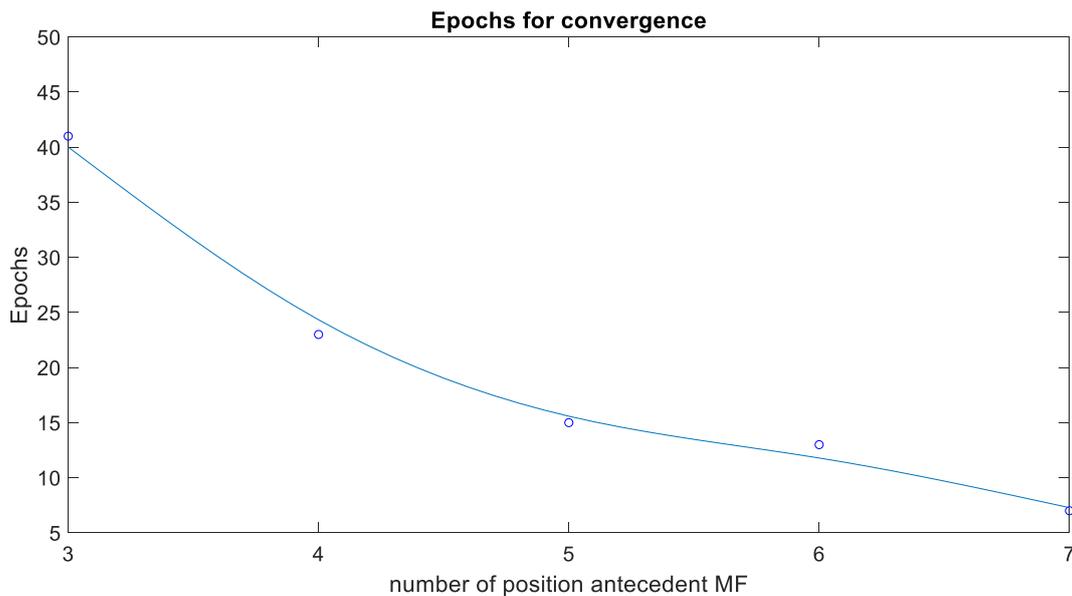


Figure 4.18 – Epoch for convergence during training of models with varying number of antecedents' membership functions

Then, Figure 4.27 results must be discussed due to their apparently controversial nature. Intuitively, one would think of having to iterate for more epochs in order to lead to convergence a system presenting more membership functions, since there are more parameters to be adapted. However, this graph shows completely opposite outcomes. In fact, convergence is reached at earlier stages using 7 membership functions with respect to 3. This controversial aspect can be explained through the variation of the error presented by raw model and trained model. The model implementing 3 membership functions can reduce its error even of a 30% after training, compared to a 5% in a model using 7 functions. Therefore, more epochs would be required for the simpler model to reach its minimum, simply because there is more space of improvement. Adopting less membership functions means increasing their efficiency, thus they are used at their best.

4.6 Investigation on applicability of ANFIS modelling on BiLoop impeller velocity field

Results of neuro – fuzzy modelling of velocity fields produced by ring propeller are quite promising. This section aims at presenting a quick investigation about the application of an ANFIS modelling procedure on a different velocity field. This means trying to create a model capable to represent the output obtained from a BiLoop impeller. As a matter of fact, this type of agitator is characterised by a very different distribution of power input along a stirred tank. Figure 28 – 29 show velocity field as contour plots versus dimensionless radius and height. These concern PIV experiments. They both refer to

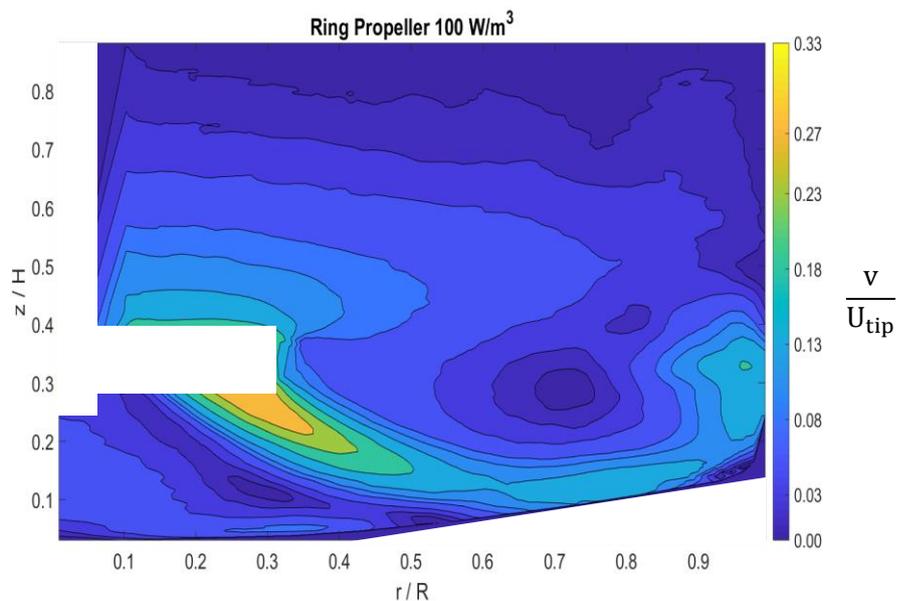


Figure 4.20 – PIV Velocity field for Ring Propeller at 100 W/m³

power input equal to 100 W/m³, respectively for Ring Propeller and BiLoop. It is possible to see how Ring Propeller directs liquid flow downwards with more intensity, while

BiLoop pushes it both upwards and downwards, but with less intensity. Consequently, higher peaks of dimensionless velocity are visible on Figure 28, while in Figure 29 it is much more distributed along the contour graph.

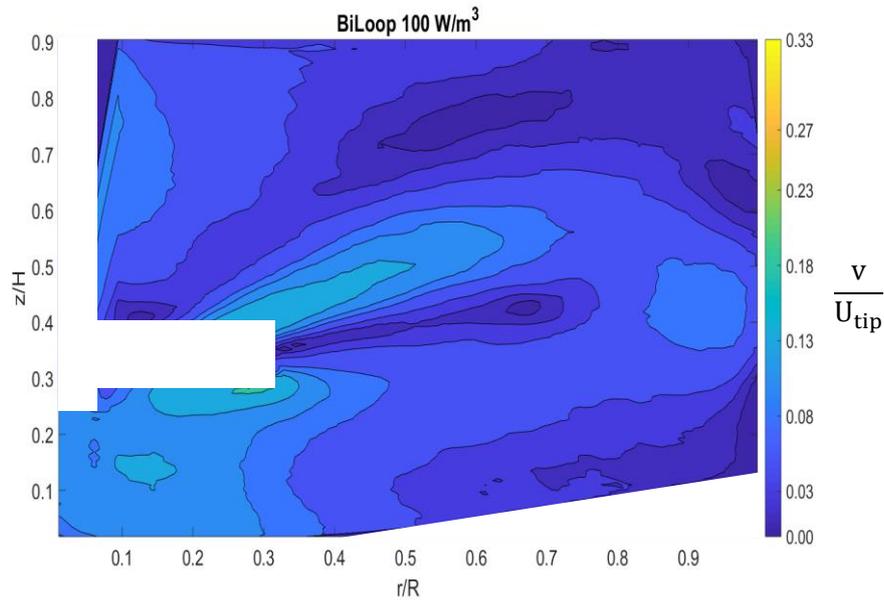


Figure 4.21 – PIV Velocity field for BiLoop at 100 W/m^3

A two antecedents – model (normed spatial coordinates) was developed. Training data were taken from the set regarding water as fluid, power input equal to 100 W/m^3 and BiLoop as impeller. As previously done, grid partition was used in order to create the raw model, using 6 gaussian membership functions for each normed spatial coordinate,

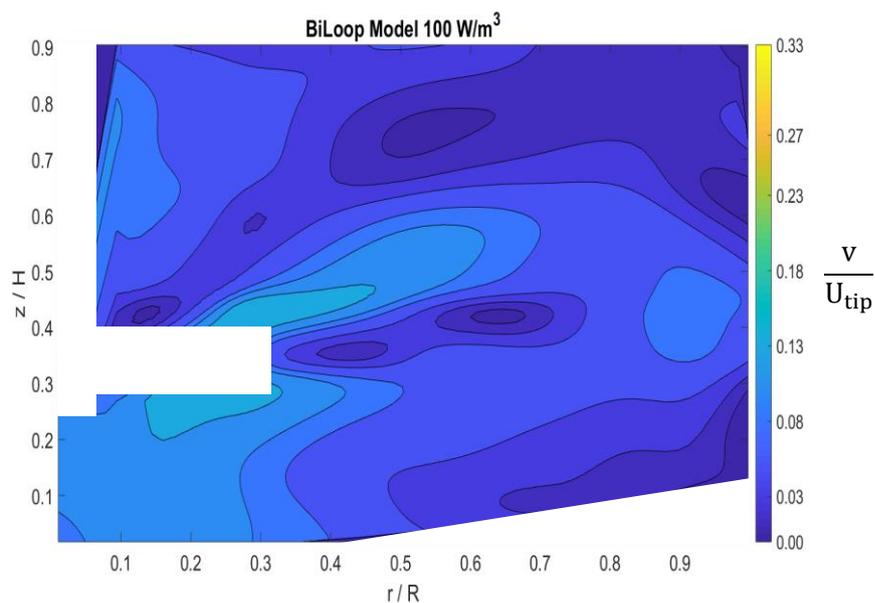


Figure 4.22 – Two antecedents' ANFIS model output as velocity field for BiLoop at 100 W/m^3

with a total of 36 rules and 36 output functions. 100 epochs were enough to train the designated model, since it reached convergence at circa 30 epochs. Half of the overall chosen dataset was involved in training process. Hence, remaining data points were used as validation set. Results for correlation indexes gave $R^2 = 0.967$, $IA = 0.992$, $RRMSE = 0.111$ and $GI = 0.95$ for validation dataset. Therefore, very good correlation was obtained. Figure 4.30 shows velocity field produced from such model output. BiLoop does not produce the same flow pattern nor velocity field, compared to Ring Propeller. Nevertheless, it is kind of able to “Smooth” original data asperities, creating a more intuitive way of graphically interpret mentioned velocity fields.

4.7 Contour plots processing in verification of obtained model correlation power

In this section, an utter method to test model with three antecedents, regarding ring propeller, is exposed. Previously, all considerations were based upon certain correlation indexes. However, there is another way to test general correlation properties of generated models. As a matter of fact, by comparing contour plots obtained by experimental data and model output, it is possible to state if model follows data behaviour. In this case, such aim is achieved by processing with ImageJ® an ensemble of contour plots obtained through MatLab® coding. Specifically, compared plots are produced for each observed power input (20, 50, 100, 200, 500 W/m^3). For each couple of plots at equal power input, two parameters are discussed. First one represents an ideal “Low mixing zone”, with relative velocity lower than 0.03. Second one depicts an ideal “High mixing zone”, with relative velocity higher than 0.13. Such areas are obtained in number of pixels, then they are normed with total pixel number, in order to have percentages over total image. This

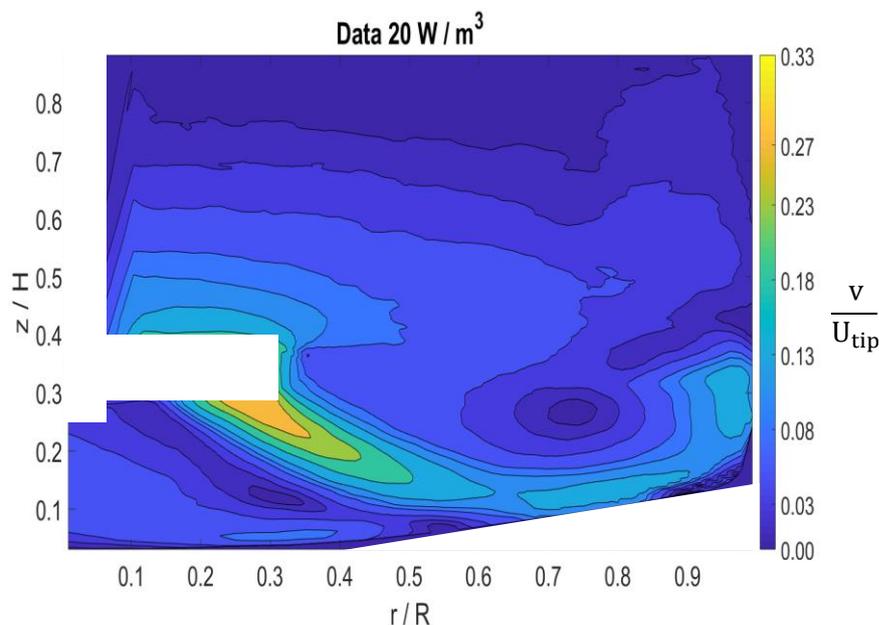


Figure 4.23 -- Experimental data of PIV measurements with power input equal to 20 W/m^3 for a Ring Propeller

is done because it is very difficult to tell whether a model approximates data well or not by just looking at contour graphs.

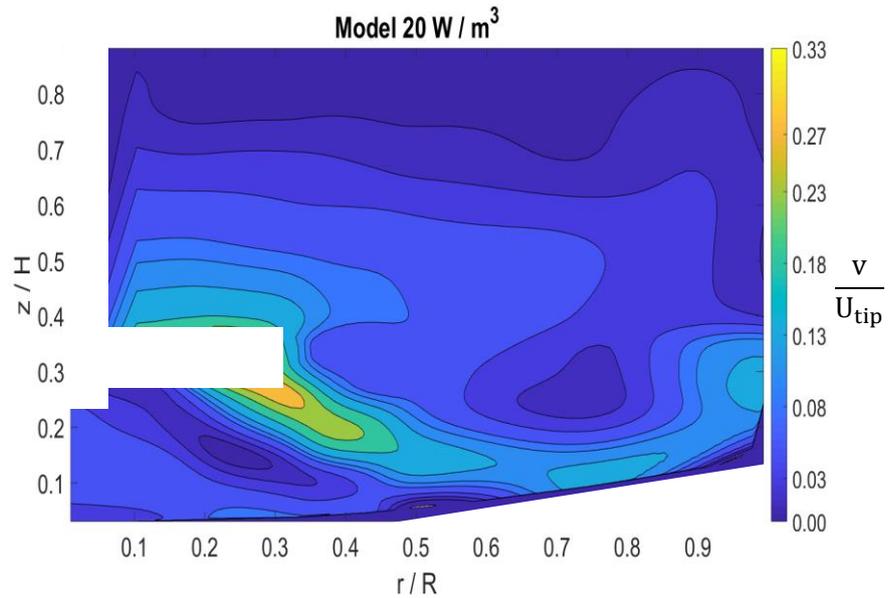


Figure 4.24 – Output of ANFIS model with power input equal to 20 W/m^3 for a Ring Propeller

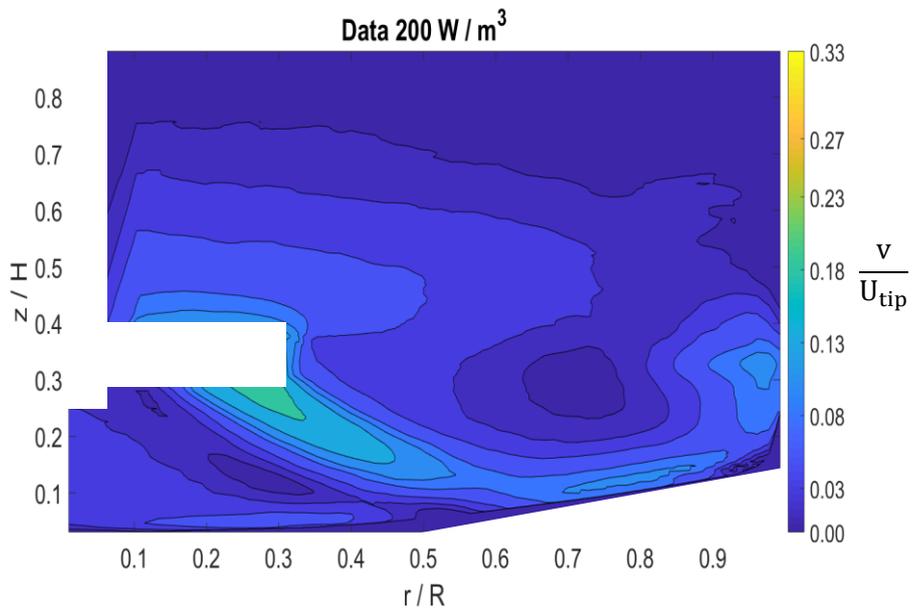


Figure 4.25 – Experimental data of PIV measurements with power input equal to 200 W/m^3 for a Ring Propeller

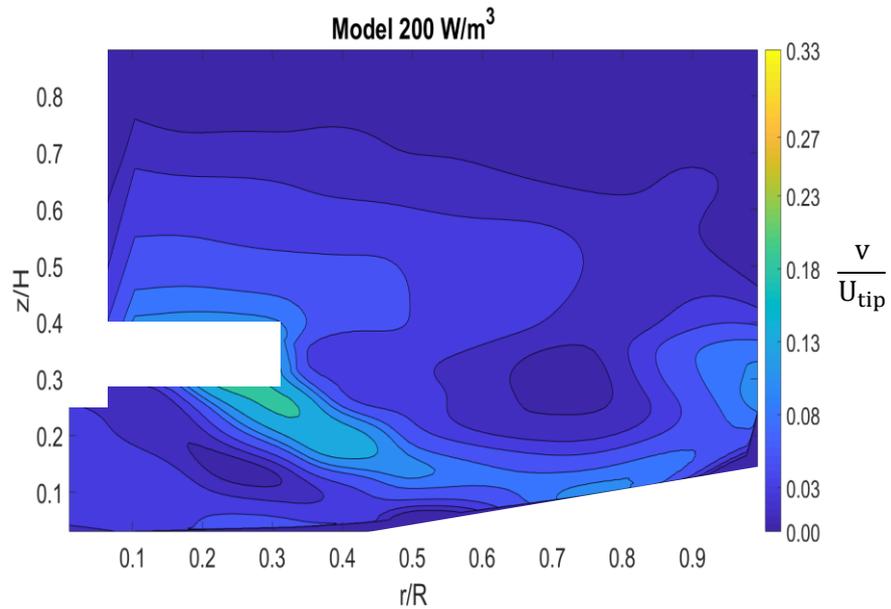


Figure 4.26 – Output of ANFIS model with power input equal to 200 W/m^3 for a Ring Propeller

Chosen inference system for production of modelled output regards three antecedents – model, with Gaussian membership functions, trained with half of total data. Two examples of comparisons are shown in Figures 4.31 and 4.32 for 20 W/m^3 and in Figures 4.33 and 4.34 for 200 W/m^3 . By just looking at these images, it is quite clear that model strongly resembles experimental data. Even though by visually comparing graphs for experimental data and model output the images could be defined very similar, further analysis results are reported below. Low and high mixing area percentages were evaluated for all five power inputs for both observed data and model output. Their differences are pictured in Figures 4.35 and 4.36.

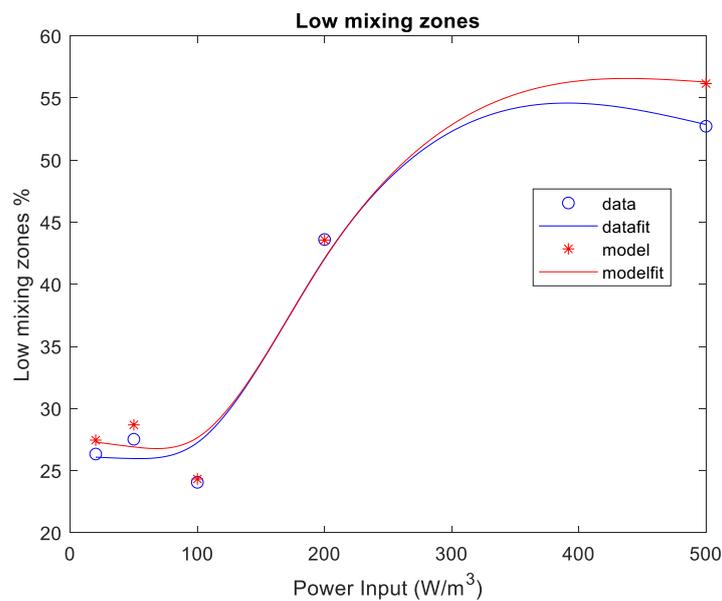


Figure 4.27 – Low mixing zones percentages for experimental data and model output, evaluated at five power inputs

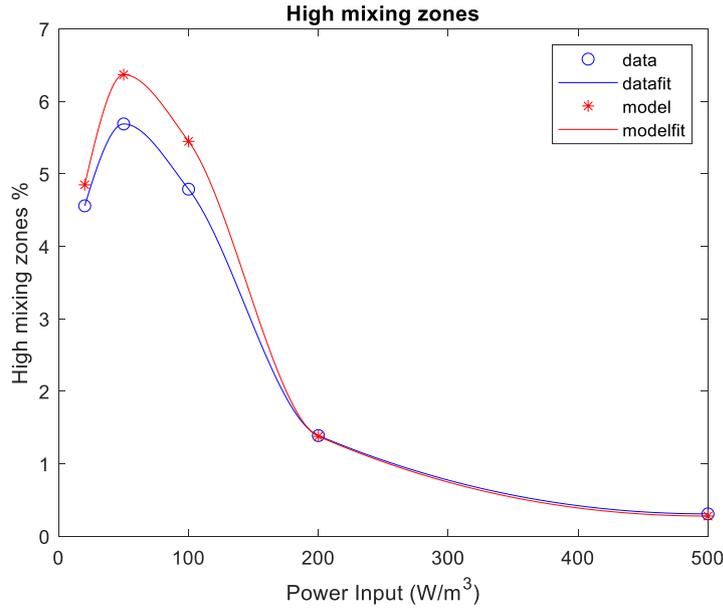


Figure 4.28 – High mixing zones percentages for experimental data and model output, evaluated at five power inputs

It is evident that, also on these evaluations, model correctly follows data behaviour. Correlation indexes turn out useful also in this section. All of them were calculated for both low and high mixing zones estimates. They are reported in Table 4.3.

Table 4.3 – Correlation indexes regarding Low and High mixing zones projection through ANFIS model

	Low Mixing	High Mixing
R ²	0.9945	0.9942
IA	0.9964	0.9920
RRMSE	0.0617	0.0972
GI	0.98	0.97

This paragraph has also the objective of exploring the possibility of extracting velocity fields, in usual space dominion, for non – experimented power input values. Two evaluations were made for 35 W/m³ and 350 W/m³. In order to consider such velocity fields well produced, low and high mixing zones were used. In fact, they should follow curves' trends represented in Figures 4.35 and 4.36, in the intervals where they are located. In other words, 35 W/m³ low mixing zones should be equal to 27%, since for 20 W/m³ they amount to 26.34% while for 50 W/m³ they are equal to 27.50%. The same reasoning is applied for high mixing zones, where they should be approximately between

4.5 and 7%. For 350 W/m³, low and high mixing zones percentages should be respectively between 43.61% and 52.73% and 1.39% and 0.28%, taking 200 W/m³ and 500 W/m³ as references for such boundaries. It is expected to have a better outcome on 35 W/m³ rather than 350 W/m³. This is due to larger distance of the second one from closest experimented values, compared to the first one. Results are shown in Table 4.4.

Table 4.4 – Low and high mixing zones percentages for two non – experimented power input values

Power Input W/m ³	Low Mixing %	High Mixing %
35	24.26	6.45
350	24.31	6.47

These values are very similar to each other. However, it is certain that 35 W/m³ outcome is much closer to its desired range, compared to 350 W/m³. Figures 4.37 and 4.38 represent model output for these two experimentally unseen power inputs. It is noticeable that, while the one obtained for 35 W/m³ is visually similar to the one regarding 20 W/m³ in Figure 4.31, the other power input does not give good results, looking at the one for 200 W/m³ in Figure 4.33. Therefore, results for 200 W/m³ are completely wrong.

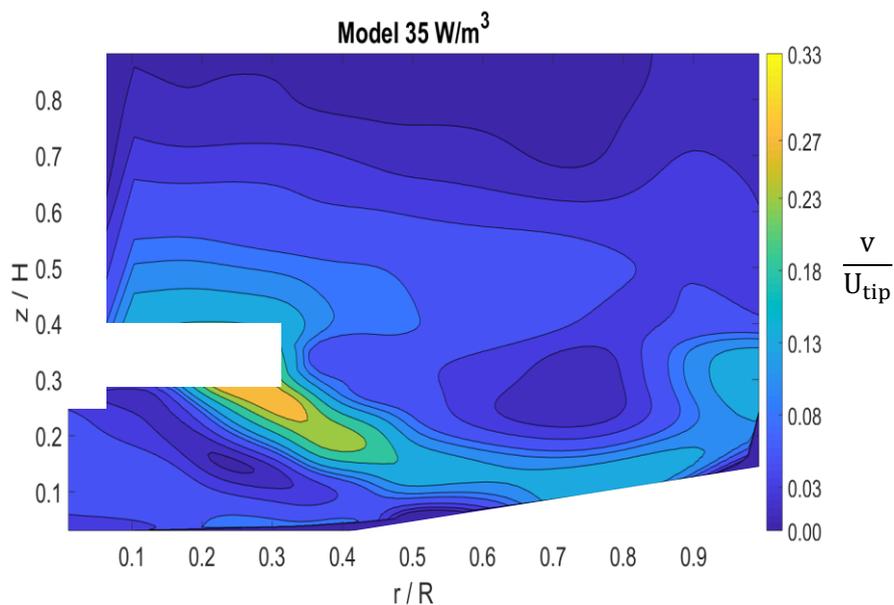


Figure 4.29 – Output of ANFIS model with non -experimented power input equal to 35 W/m³ for a Ring Propeller

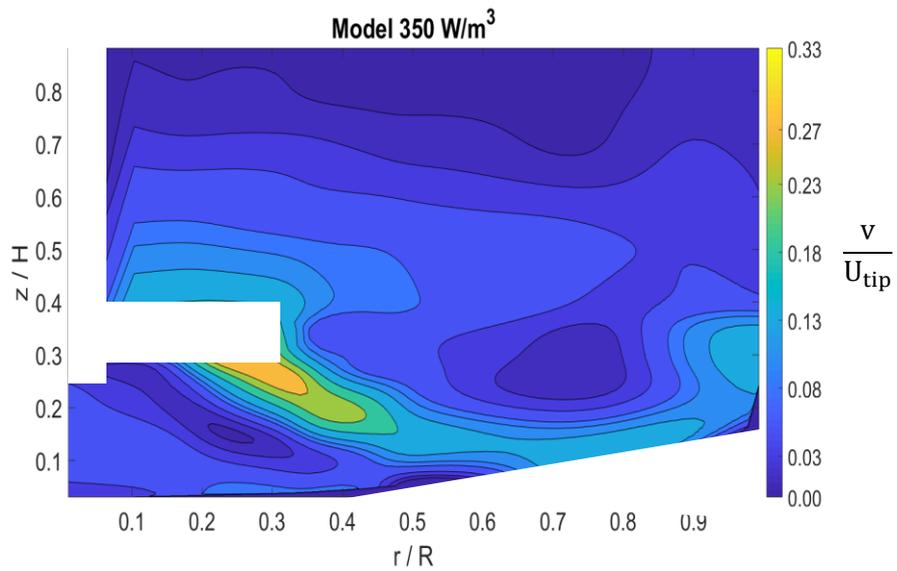


Figure 4.30 – Output of ANFIS model with non -experimented power input equal to 350 W/m³ for a Ring Propeller

Velocity profiles are clearly too similar, even though they regard completely different power inputs. In this case, ANFIS is not able to predict non - experimented values such as 350 W/m³. More power inputs should be fed to the raw model during training process in order to build a more predictive inference system.

5. Conclusions and Future Outlook

In this work, an overview was conducted on Fuzzy Logic, Neural Networks and, principally, Neuro – fuzzy inference systems. Adaptive inference systems type was investigated. Since Fuzzy Logic is not suitable for the realization of models portraying high amounts of data sets, ANFIS were employed to try to store experiments memory in them. Some obtained models manage to follow velocity field values very accurately. Proof of this is given by the exposed correlation indexes and contour plots at the end of Chapter 4. Newly created fuzzy index, even if not very predictable in value, is useful for an immediate comparison between different models. In fact, it can be adopted to make a quick judgement on the combination of the three original indexes. In addition to this, neural networks learning applications with fuzzy logic were illustrated. Gaussian membership functions proved to be the best among all kinds, possibly because they have better ability to reproduce data behaviour produced by exposed Particle Image Velocimetry velocity field. Furthermore, best combination of membership functions is probably the one reporting five membership functions describing Power Input antecedent and six differently distributed membership functions for each dimensionless spatial coordinate antecedent, radial and axial. Training with half of total data amount still seems the best choice among analysed percentages. Even though lower percentages showed good performances, this proved to be the best one on validation data, thus presenting best predictivity towards unseen data. Moreover, 50% as training batch is usually considered standard quantity to be taken (Al-Mahasneh et al., 2013).

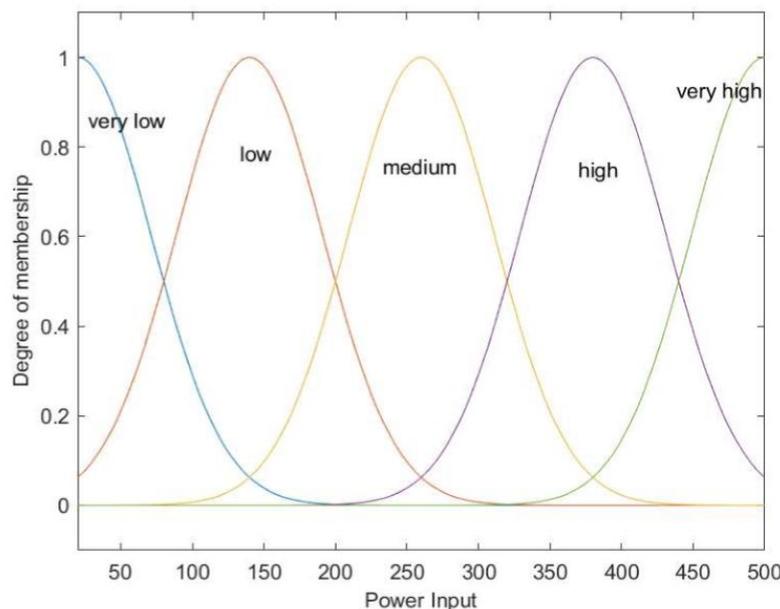


Figure 5.1 – Membership functions for power input antecedent of best obtained model

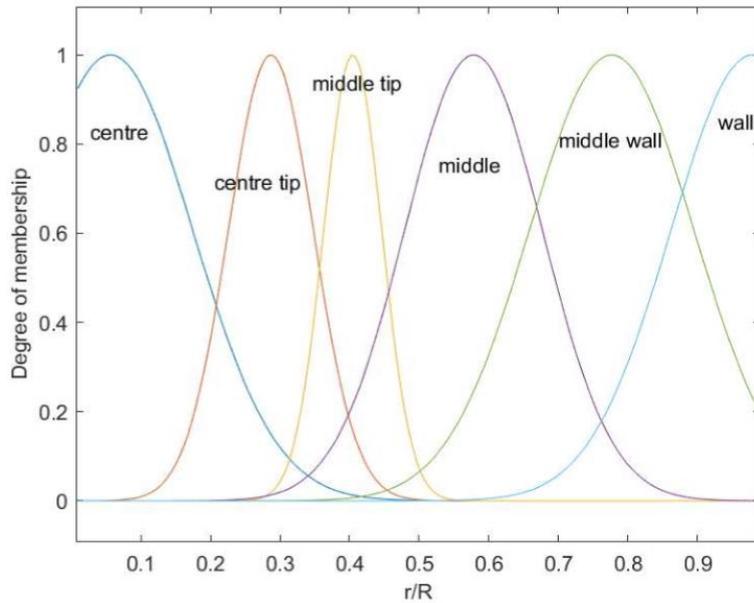


Figure 5.2 – Membership functions for dimensionless radial coordinate antecedent of best obtained model

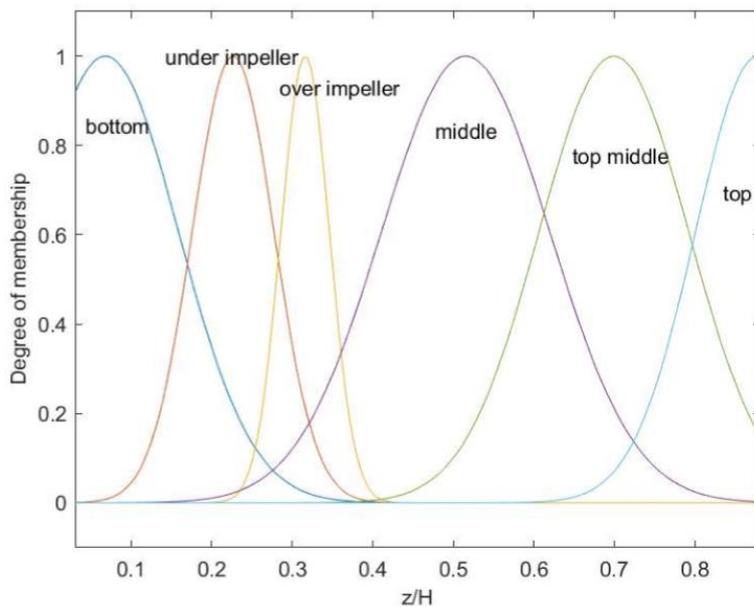


Figure 5.3 – Membership functions for dimensionless axial coordinate antecedent of best obtained model

In addition to this, training directly from three antecedent seems to be the best way to achieve final model. Figures 5.1 – 5.3 presents antecedents membership functions owned by this model. Each membership function was named accordingly to value set reference. A major drawback of this model resides in the fact there are too many rules. Because of grid partitioning disposal method “Greediness”, rules dramatically increase with membership functions quantity increase. The model which was exposed in this section

presents 180 rules and 180 output functions. This makes very difficult the comprehension and recognition of repeated patterns in small zones. A sample of 10 rules is defined in Table 5.1.

Table 5.1 – Sample of randomly picked 10 rules from a total of 180

IF	Power Input	r/R	z/H	THEN	f (P/V, r/R, z/H)
	Very low	Centre	Bottom		$a_1 \frac{P}{V} + b_1 \frac{r}{R} + c_1 \frac{z}{H} + d_1$
	Low	Wall	Wall		$a_{48} \frac{P}{V} + b_{48} \frac{r}{R} + c_{48} \frac{z}{H} + d_{48}$
	Medium	Middle	Under impeller		$a_{92} \frac{P}{V} + b_{92} \frac{r}{R} + c_{92} \frac{z}{H} + d_{92}$
	High	Centre tip	Over impeller		$a_{117} \frac{P}{V} + b_{117} \frac{r}{R} + c_{117} \frac{z}{H} + d_{117}$
	High	Middle	Bottom		$a_{127} \frac{P}{V} + b_{127} \frac{r}{R} + c_{127} \frac{z}{H} + d_{127}$
	Very high	Wall	Top		$a_{180} \frac{P}{V} + b_{180} \frac{r}{R} + c_{180} \frac{z}{H} + d_{180}$
	Medium	Middle tip	Top middle		$a_{89} \frac{P}{V} + b_{89} \frac{r}{R} + c_{89} \frac{z}{H} + d_{89}$
	Very low	Middle Wall	Over impeller		$a_{63} \frac{P}{V} + b_{63} \frac{r}{R} + c_{63} \frac{z}{H} + d_{63}$
	Very high	Centre	Top		$a_{150} \frac{P}{V} + b_{150} \frac{r}{R} + c_{150} \frac{z}{H} + d_{150}$
	Low	Middle	Middle		$a_{58} \frac{P}{V} + b_{58} \frac{r}{R} + c_{58} \frac{z}{H} + d_{58}$

It is known that adaptive neuro fuzzy inference systems are suitable not only for modelling realisation. Neuro – fuzzy systems are already implemented in some controlling applications (Berardi et al., 1996). However, this is not the only field in which such mathematical logic is applied. In fact, it is even implemented in decision making matters in multiple types of problems, such as Foreign Exchange Trading (Gradojevic, Yang, & Gravelle, 2002), Paediatrics (Sridevi & Nirmala, 2016) and many more. Future researches regarding this machine learning branch should be about trying to find a way to improve grid partitioning algorithms. This concerns the efficient reduction of output functions, along with the maintenance of correlation usefulness, in order to couple modelling with patterns recognition in small scale. Furthermore, particle image velocimetry coupling with machine learning methods, such as the inference system presented in this work, could be further enriched by combining data knowledge with expertise knowledge. This could possibly lead to the accomplishment of a model with predictivity of local velocity fields on non – experimented fluids. It would be of large interest to find a way to produce a rule based intuitive system returning the results the stress produced by mixing. As a matter of fact, knowing this parameter locally for biological broths could prevent cells damage. Other interesting solutions obtainable by

future developments on these models could be accomplished if more power input values are experimentally observed. This would mean having the possibility of having a more powerful interpolation over this parameter, allowing to achieve the visual of velocity fields produced by unseen power inputs. Since in this case only 5 values were interpolated, produced model does not have predictivity over power input magnitudes between the interpolated ones. It is possible to demonstrate, as in Chapter 4, that between 20 and 50 W/m³ or 50 and 100 W/m³ velocity profiles start to get closer to real ones, because interpolated values are less distant one from another. On the other hand, a velocity profile at 350 W/m³ is completely wrong, because 200 and 500 W/m³ are too different from each other.

Bibliography

- Al-Mahasneh, M. A., Rababah, T. M., & Ma'Abreh, A. S. (2013). Evaluating the combined effect of temperature, shear rate and water content on wild-flower honey viscosity using adaptive neural fuzzy inference system and artificial neural networks. *Journal of Food Process Engineering*. <https://doi.org/10.1111/jfpe.12014>
- Berardi, F., Chiaberge, M., Miranda, E., & Reyneri, L. M. (1996). Neuro-fuzzy systems for control applications. In *International Symposium on Neuro-Fuzzy Systems, Proceedings*.
- Di Addario, M., Temizel, I., Edes, N., Onay, T. T., Demirel, B., Coptu, N. K., & Ruggeri, B. (2017). Development of Fuzzylogic model to predict the effects of ZnO nanoparticles on methane production from simulated landfill. *Journal of Environmental Chemical Engineering*, 5(6), 5944–5953. <https://doi.org/10.1016/j.jece.2017.10.033>
- Egaji, O. A., Griffiths, A., Hasan, M. S., & Yu, H. N. (2015). A comparison of Mamdani and Sugeno fuzzy based packet scheduler for MANET with a realistic wireless propagation model. *International Journal of Automation and Computing*. <https://doi.org/10.1007/s11633-014-0861-y>
- Fontenla-Romero, O., Erdogmus, D., Principe, J. C., Alonso-Betanzos, A., & Castillo, E. (2003). Linear least-squares based methods for neural networks learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- Gradojevic, N., Yang, Y., & Gravelle, T. (2002). Neuro-Fuzzy Decision-Making in Foreign Exchange Trading and Other Applications. *CEA Conference 2002, 36th Annual Meeting of the Canadian Economics Association*.
- Ii, R. J. H., & Ground, A. P. (1998). LEARNING FUZZY RULES FROM DATA Thomas Sudkamp Wright State University, (April), 20–22.
- Jang, J.-S. R., & Gulley, N. (2015). Fuzzy logic toolbox functions. *User's Guide*, 208. Retrieved from <http://www.mathworks.com/help/fuzzy/functionlist.html>
- Jang, J. S. R., Sun, C. T., & Mizutani, E. (2005). Neuro-Fuzzy and Soft Computing-A Computational Approach to Learning and Machine Intelligence [Book Review]. *IEEE Transactions on Automatic Control*. <https://doi.org/10.1109/tac.1997.633847>
- Li, G., Li, Z., Gao, Z., Wang, J., Bao, Y., & Derksen, J. J. (2018). Particle image velocimetry experiments and direct numerical simulations of solids suspension in transitional stirred tank flow. *Chemical Engineering Science*. <https://doi.org/10.1016/j.ces.2018.06.073>

- Mrinal, B. (2008). Adaptive Network based Fuzzy Inference System (ANFIS) as a Tool for System Identification with Special Emphasis on Training Data Minimization, 141.
- Patterson, J., & Gibson, A. (2008). *Deep Learning A practitioner's approach. Clinical Cancer Research*. <https://doi.org/10.1038/nature14539>
- Prasad, A. K. (2000). Particle image velocimetry. *Current Science*.
- Ross, T. J. (2010). *Fuzzy Logic with Engineering Applications: Third Edition. Fuzzy Logic with Engineering Applications: Third Edition*. <https://doi.org/10.1002/9781119994374>
- Sagiroglu, S., & Sinanc, D. (2013). *Big data: A review*. <https://doi.org/10.1109/CTS.2013.6567202>
- Sheng, J., Meng, H., & Fox, R. O. (1998). Validation of CFD simulations of a stirred tank using particle image velocimetry data. *Canadian Journal of Chemical Engineering*. <https://doi.org/10.1002/cjce.5450760333>
- Singla, J. (2015). Comparative study of Mamdani-type and Sugeno-type fuzzy inference systems for diagnosis of diabetes. In *Conference Proceeding - 2015 International Conference on Advances in Computer Engineering and Applications, ICACEA 2015*. <https://doi.org/10.1109/ICACEA.2015.7164799>
- Sridevi, S., & Nirmala, S. (2016). ANFIS based decision support system for prenatal detection of Truncus Arteriosus congenital heart defect. *Applied Soft Computing Journal*. <https://doi.org/10.1016/j.asoc.2015.09.002>
- Tadeusiewicz, R. (1995). Neural networks: A comprehensive foundation. *Control Engineering Practice*. [https://doi.org/10.1016/0967-0661\(95\)90080-2](https://doi.org/10.1016/0967-0661(95)90080-2)
- Vieira, J., Dias, F., & Mota, A. (2004). Neuro-fuzzy systems: a survey. ... *on Neural Networks and Applications, Udine*
- Willmott, C. J., Robeson, S. M., & Matsuura, K. (2012). Short Communication A refined index of model performance. *International Journal of Climatology*, 32(13), 2088–2094. <https://doi.org/10.1002/joc.2419>
- Yeom, C. U., & Kwak, K. C. (2018). Performance comparison of ANFIS models by input space partitioning methods. *Symmetry*. <https://doi.org/10.3390/sym10120700>

List of symbols

μ	Fuzzy membership degree
w_i	Weight
b_i	Bias
σ	Activation function
e_j	Error in ANN training single iteration
d_j	Desired output in ANN training single iteration
y_j	Produced output in ANN training single iteration
E	Sum of square errors
\mathcal{E}_{av}	Cost function
O	ANFIS layer's node output
f_i	ANFIS output function
r_a	Cluster radius
r_b	Radius of elimination function for subtractive clustering
A	Known terms' matrix in subtractive clustering
θ	Consequent parameters' vector
R^2	Coefficient of determination
RRMSE	Relative Root Mean Square Error
IA	Willmotts' Index of Agreement
GI	Goodness Index
P/V	Power Input (W/m^3)
v	Velocity vector norm
V_r	Velocity radial component
V_z	Velocity axial component
R	Vessel radius
H	Vessel height

U_{tip}	Tip velocity
RMSE	Root Mean Square Error
c	Gaussian curve centre
σ	Gaussian curve standard deviation
r	Radial coordinate
z	Axial coordinate
\tilde{r}	Dimensionless radial coordinate
\tilde{z}	Dimensionless axial coordinate
\tilde{v}	Dimensionless velocity vector norm
$\tilde{\sigma}$	Modified Gaussian curve standard deviation
\tilde{c}	Modified Gaussian curve centre

Abbreviations and Acronyms

FIS	Fuzzy Inference Systems
ANN	Artificial Neural Networks
ANFIS	Adaptive Neuro Fuzzy Inference Systems
FALCON	Fuzzy Adaptive Learning Control Method
GARIC	Generalised Approximate Reasoning Based Intelligence Control
PIV	Particle Image Velocimetry
MF	Membership Function

Acknowledgements

Before beginning my academic path, I didn't know who I was. I used to find difficult defining my personality. I wanted to add something special and unique to my life, something which could forge my character in all its shades. Therefore, I decided to take part in marvellous world of Chemical Engineering. When I started, I didn't think I could be such a hard worker, always trying to make the best out of every single day. For this, I want to thank my parents and my brother. They taught me to be patient and dedicated. I've never been an easy – going person but studying hard gave me the possibility to learn how to be humble. Several times I complained about my situation, about sacrifices I had to make to proceed through these years of study. However, I enjoyed being a student. I was always surrounded by good friends, which accompanied me in this long way, which suddenly has come to an end. I'm glad I made myself more persistent. For this I want to thank all professors I met in these years. Even the strictest one had added fundamental fragments to my self – confidence. Without them, I wouldn't be the person I am right now. Nonetheless, many difficulties still have to come. However, I'm more than ever ready to face them.

