Fuzzy logic approaches to Modelling, Identification and Control of non-linear systems

Thesis submitted to the University of Wales for the degree of

Doctor of Philosophy

By

Ioannis S Akkizidis, BEng, MSc
Mechatronics Research Centre
University of Wales College, Newport
October 2000

I dedicate this thesis to the memory of my Father

Στή μνήμη του πατέρα μου

Declarations

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... (candidate)

Date ...30. 11. 2000.........

STATEMENT 1

This thesis is a result of my own investigations, except where otherwise stated.

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... (candidate)

Date ...30. 11. 2000.........

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the tittle and summary to be made available to outside organisations.

Signed .... (candidate)

Date ...30. 11. 2000.........

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Geoff Roberts whose supervision and support throughout my research were very important for me to complete this thesis.

Special thanks are also due to Vivianne Bouchereau who was always there to support and help me in any way. Thank you for your love, understanding and patience during this tense period. Without you this thesis seemed impossible.

Thanks also to University of Girona, Spain, that kindly provided an underwater vehicle for me to design its controller. I would specially like to thank Prof. Joan Batlle for his support and hospitality during my stay in Girona. I also would like to thank Mark Carreras, Pere Ridao, Josep Cortada Hortala and all the people who helped me complete my experiments with GARBI underwater vehicle in lake Banyolas. It was a great pleasure to work with you guys. Also I would like to mention Dr. Joaquim Salvi from University of Girona for his useful advises and friendship.

Thanks to Dr. Hefin Rowlands who always responded to my request for help and spent his time to share his knowledge with me. Thanks also to Dr. Stammati Mavromati for his help on the earlier stages of my university studies in Athens, Hellas.

Thanks to all the staffs at the Allt-yr-yn Campus library who always helped me during my search of new publications and books and for being very patient when I delayed in giving back my books especially during the writing up of this thesis. I would specifically like to thank Dawne Leatherdale, Jenifer Coleman and Bronwen Stone.

Thanks to all the staffs from the University of Wales College Newport who helped me in any way during my stay in this university. Thanks to Mr Alan Lewis for his patient, waiting for me to finish my file saving every night before he locked the university doors. I would also like to thank my colleagues in the Mechatronics Research Centre for their advice and help during the last three years. Wish you every success in your future.

I would like to thank the University of Wales College Newport for the bursary that was provided to me during my three years PhD studies. Thanks also to the British Council for the financial support under the British/Spanish Acciones Integradas programme involving the application of my project. Finally a special thank you to the Hellenic Orthodox charity organisation for their financial support, which kindly helped me during a time that I really needed it.

Thank you to all my friends who supported me throughout this PhD and while I mention only a few, I would like to thank Peter Petrovic, Maja Kotzmuth, Alexandros Mouzakitis, Sotiris Siamndouras, Sofia Papida, Stratos Mastorakis, Irene Argire and Simone Xanthogiorgou for their moral support, precious advice and help. Your friendship means so much.

I want also to thank my family, my brother Nikos, my sister Athina, my brother in law Basilis, my niece Ageliki, my nephew Odyseas and especially my parents Savvas and Kiriaki for all their love, support and encouragement to undertake this PhD.

I specially would like to acknowledge my father who I recently lost. I know that you have always been proud of my achievements during my live and you still are from up there. Hope God is still looking after you and be sure that you are always in my heart and I will remember forever all the things that you use to say.

Summary

This thesis is concerned with the modelling of non-linear systems, identification of control strategies and designing fuzzy controllers for uncertain systems using fuzzy logic techniques assisted by other conventional methods. The application of the proposed approaches are tested and evaluated on an underwater vehicle, where limited knowledge of the vehicle's dynamic characteristics is available.

In particular the fuzzy and neuro-fuzzy techniques for modelling non-linear systems are reviewed. The combination of function approximation using neuro-fuzzy techniques and new approach to knowledge acquisition using a fuzzy supervised scheduling system is presented.

The identification and modelling of control strategies is also studied. Using fuzzy clustering techniques a systematic approach to identify and qualify the data in n-dimensional space that represents these strategies is presented and evaluated.

The design of robust and tuneable controllers for non-linear systems is also proposed and developed using a combination of the Taguchi design of experiments method and the fuzzy combined scheduling system approach.

The main contributions in this thesis are; the development of a hybrid fuzzy and neuro-fuzzy approach to model non-linear systems with the application to model the yaw dynamics of an underwater vehicle; an algorithmic methodology for identification and modelling of a complex system's control actions with application to construct control strategies for "avoid objects" task; an innovative contribution to the problem of determining the optimal parameters of fuzzy and fuzzy-like PD controllers in terms of robustness and tuning characteristics and the successful implementation of the fuzzy-like PD controller for course-changing and course-keeping in an underwater vehicle.

Table of Contents

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Appendix A

Appendix B

List of Tables

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Appendix A

1

Introduction to thesis

1.1 Introduction

In the first half of the third century B.C the first feedback control system was invented by the Greek engineer Ktesibios which was the float regulator mechanisms of the water clock (Drachmann, 1948) shown as in Figure 1.1. Since that time the development of the automatic control has been growing rapidly, playing a vital role in the advance of engineering and science. However, most of the control theories have been developed focussing on how to model and control linear or partly non-linear systems. A good methodology for solving control problems of non-linear systems is still an open problem. Nowadays, a large part of the research of the modern control technology is focused on techniques that are called Artificial Intelligent (AI) which includes mainly Neural Networks, Fuzzy Logic, Genetic Algorithms and their hybrid.

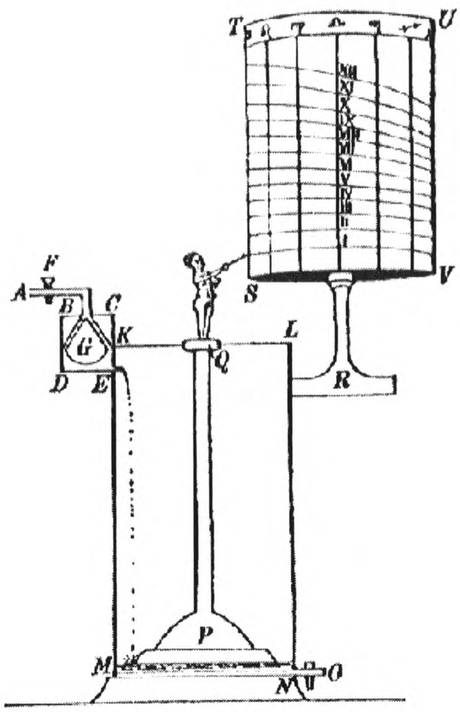


Figure 1.1 Ktesibios water clock float regulator mechanisms

A large part of the control research community believes that Fuzzy Logic Control technique can solve control and modelling problems for non-linear systems. This technique started its development almost 30 to 40 years ago. Despite all the research that has been carried out in all these years, the fuzzy controller design theories are far from being completed and even not close to being used at industrial level. In 1990 Lee stated in his survey (Lee, 1990a) that *"there is no systematic procedure for the design of fuzzy controller"*. During the last decades many approaches have been proposed, in most of the cases, even if the theoretical background are soundly based, the presented application results are mostly simulated using second or no more than third order systems. Therefore, it is unknown if these approaches are still able to be successful when applied to real systems. This is maybe one of the main reasons why many approaches that have been used in fuzzy logic control have not yet been developed in industry.

In this thesis the modelling and control of non-linear systems is proposed and developed using neuro-fuzzy, fuzzy clustering techniques and Taguchi design of experiments method combined with fuzzy set theory. The application of this work was an underwater vehicle. The modelling approaches that are proposed in this thesis use real experimental data resulting from the vehicle's underwater missions. Moreover, the proposed control techniques utilises real and simulated data and are tested in both a simulated and real environment.

The rest of this chapter is structured as follows: in Section § 1.2 a brief historical review of fuzzy system theory is presented. In Section § 1.3 the problems in fuzzy logic modelling and control are presented as well as how they have been approached in this Thesis. Finally Section § 1.4 gives an overview of the thesis presenting briefly the rest of the chapters.

1.2 Historical review of fuzzy systems

Aristotle and the philosophers who preceded him established the precision of mathematics. In their work to devise a concise theory of logic they proposed the "Laws of Thought" (Korner, 1967). One of these, the "Law of the Excluded Middle," states that every proposition must either be True or False. However, when Parminedes proposed the first version of this law (around 400 B.C.) there were strong and immediate objections: for example, Heraclitus proposed that things could be simultaneously *True* and *not True*.

It was Plato, in the fourth century B.C., that laid the foundation for what would become fuzzy logic, indicating that there was a third region (beyond True and False) where these opposites "tumbled about". Much later in the early 1900's Lukasiewicz proposed a systematic alternative to the bi-valued logic of Aristotle describing it as a three-valued logic, along with the mathematics to accompany it (Lejewski, 1967).

After more than half a century, in 1965 Prof. Lotfi A. Zadeh published his seminal work "Fuzzy Sets" (Zadeh, 1965); (Zadeh, 1968), which described the mathematics of fuzzy set theory and by extension fuzzy logic. He argued that more often than not, the classes of objects encountered in the real physical world do not have precisely defined criteria for membership. This proposed theory allows the membership function (or the values False and True) to operate over the range of real numbers [0.0, 1.0]. New operations for the calculus of logic were proposed and showed to be in principle at least a generalisation of classic logic.

Mamdani in 1974 (Mamdani, 1974) applied these theories in control systems with a number of successful applications (Mamdani and Assilian, 1975). Since then much research has been done in modelling and control of non-linear systems where their description is beyond this small historical review.

1.3 Problems in Fuzzy Logic Modelling and Control and how they have been approached in this Thesis

Fuzzy logic was developed for knowledge representation and symbolic/numeric interface, and its status is rather ambiguous in this respect. Fuzzy set theory has brought together researchers in AI (specially those that are dealing with NN), control engineers and experts who possessed little background in common and the temptation existed for each community to emphasise a narrow view of fuzzy logic that fits in with their own tradition.

This section discusses what are the main problems and drawbacks as well as where the new research tend to focus for solving modelling, identifying and controlling of non-linear systems using fuzzy logic techniques. The proposed approaches for these problems are also introduced.

1.3.1 Modelling

While modelling non-linear systems using fuzzy logic, it is mostly viewed as a problem of function approximation and not as a problem of knowledge acquisition. An approximate representation of functions should be general enough to capture a large class of functions, simple enough (especially the primitive objects, here the fuzzy rules) to achieve efficient computations and economical storage, and should be amenable to capabilities of learning from data. It is well known that a pure fuzzy system that is based only on expert's setting cannot always be successful in function approximation problems and thus it is very difficult for it to compete with the standard approximation methods such as Neural Nets, for example multi-layer perceptrons and radial basis functions.

However, to overcome the disability of fuzzy systems as function approximators, many approaches have considered the fuzzy rule base systems more as a standard, non-fuzzy universal approximator of functions (Buckley and Hayashi, 1993); (Castro, 1995); (Kosko, 1992a); (Wang, 1992) and less as a means of extracting control laws from heuristic knowledge. In some cases this approximator can be achieved by combining neural networks and fuzzy systems theory. As an example to approximate functions Jang, (1993) constructed adaptive networks that are functionally equivalent to fuzzy inference systems.

The above trend raises several observations in the majority of fuzzy logic approaches:

- If fuzzy logic competes with alternative methods in approximation theory, it faces a big challenge because approximation theory is a well-established field in which many successful results exist.

- In most cases, the number of fuzzy sets and rules becomes very large in universal approximation techniques using fuzzy systems. This number usually increases according to the complexity of the system under study.

- The modelling using fuzzy rule-base systems with neural nets or variants thereof (Kosko, 1992b); (Berenji and Khedar, 1992); (Jang, 1993), has created a lot of confusion for the researchers and engineers with respect to the actual contribution of fuzzy logic. To some extent it is not very clear that fuzzy logic based approximation methods for modelling and control need fuzzy set theory any longer.

- The knowledge representation of the fuzzy rules is meaningless using these (neuro-fuzzy) types of approximation techniques even if the initial settings of the rules may have some meaning. Therefore even if it is claimed that these types of approaches are "grey boxes", in most cases they are not.

In the early seventies Prof. Zadeh introduced fuzzy set theory as a tool to give ability to the experts to explore and use their knowledge (Zadeh, 1973). It is very surprising, however, that the incompatibility between high precision and linguistic meaningfulness drove the researchers far away from the initial purpose.

In this research, for the purpose of modelling non-linear systems, it is proposed to combine knowledge acquisition with function approximation approaches. The latter is used only to define the non-linear or linear parts of the system locally, whereas the former is used to define the linguistic interpretation in the rule base defined by the expert's knowledge together with some guidelines coming from fuzzy set theory. The proposed method is implemented to model an underwater vehicle, where experimental results that describe the vehicle's behaviour in a real environment are employed.

1.3.2 Identifying and modelling of control strategies

Identifying and Modelling Control Strategies is another important topic in fuzzy set theory and its applications. This is very useful in a real system where the operator is the main factor in the control loop. Control strategies can be considered as a fuzzy model system based on the operator's control experience and knowledge of a particular system. The problem in this research field is that the expert's control skills are difficult to be verbalised since the operators' control strategy is based on various control principles simultaneously and because the operator may not be able to explain why a particular control action is chosen. Different operators often contradict each other during the design of the rule base for knowledge acquisition (Babuska, 1998a).

The research therefore is intended to find ways to extract information from experimental or simulation data resulting from the expert's control actions and then to define the rules that describe the control strategy. One of the most successful ways to achieve this approach is to use fuzzy clustering method technique.

Fuzzy set theory in cluster analysis were firstly proposed in the work of Bellman *et al*, (1966), and Ruspini, (1969). These papers opened the door for research in fuzzy clustering. Despite that initially this technique was mainly used in image processing and pattern recognition fields (Bezdek *et al*, 1999); (Chi and Pham, 1996), nowadays fuzzy clustering is widely studied and applied in a variety of substantive areas such as modelling and identification in control systems.

As mentioned earlier, the collection of fuzzy rules that describe the control strategy results from an input-output mapping of the operators' control behaviour data. Using the cluster analysis, the estimation of the parameters that define the antecedent and consequent part of the fuzzy rule templates as well as the number of *If-Then* rules that constructs the knowledge base are achieved. The main idea is that by defining the prototypes of the clusters the centres of hyper-planes as well as the number of fuzzy sets are defined. Moreover, the projections of these hyper-planes, into n-dimensional input-output data space axes, determines the parameters of the extracted membership functions together with the partition of the input/output space.

Even after many successful applications of fuzzy clustering techniques in the image processing and modelling identification problems (Bezdek and Dumm, 1975); (Gustafson and Kessel *al* , 1979); (Yager and Filev, 1994a); (Bezdek *et al*, 1999); (Babuska, 1998a), problems with this techniques have been also identified. In this research after extensive review of fuzzy clustering algorithms it has been concluded that "*the tools are there*" but the appropriate way to use them is needed in terms of exploiting the advantages of each of them. Thus, in this thesis an algorithmic methodology to construct fuzzy control strategies based on the choice of different fuzzy clustering approaches to define initially the number and actual position of the prototypes and the variance of the clusters is proposed. Then the fuzzy sets are obtained by using projecting methods and by using merging techniques their number are reduced. As an optional step, some modification to the fuzzy system can be obtained to make some small improvements to the performance of the model identification. This can be achieved by applying techniques such as the gradient method.

An application for generating fuzzy rules for "avoid objects" control strategy using 3D input/output data space is used to investigate the capabilities of the proposed approach. The availability of the data stems from simulation results.

1.3.3 Control techniques using fuzzy logic

Whenever modelling is possible, classical control theory offers usually a safer approach, although a lot of work is sometimes necessary to bridge the gap for practical problems. However, fuzzy logic is very reasonable approach when modelling is difficult or costly, but knowledge is available in order to derive fuzzy rules. Thus, control engineering practice has benefited from the readability of fuzzy-rule base systems and fuzzy logic controllers have been one of the most successfully used controllers in a large number of complex and non-linear systems. Despite all the successful applications that has been reported using fuzzy logic controllers a systematic methodology of how to design their characteristic properties is still an open problem. This is because the method is highly dependent on expert's knowledge and sufficient knowledge of the operator(s), which may be problematic, since the human control skills can be difficult to verbalise or explain reasonably. Moreover, when the system is non-linear the parameters of the fuzzy logic control need to be robust and "tuneable".

In most approaches, trying to overcome the above difficulties has been to complement the benefits of fuzzy controllers and classical control theory. One of the most popular approaches is to combine fuzzy and conventional PID controllers. The fuzzy like PID (as well as like-PD, like-PI) controllers developed in this thesis stems from this idea. The parameters of the membership functions and the scaling factors for this type of controller are mostly selected on the basis of their influence on the fuzzy logic control surface, and the rules are formulated considering the control trajectory. So, the problem actually shifts towards how to optimise and tune these parameters and/or factors. Although several approaches have been proposed in this field (Jantzen, 1998); (Yager and Filev, 1994b); (De Silva, 1995); (Zheng, 1992); (Mudi and Pal, 1999), the most successful results are based on the combination of good experimental understanding of the controlled system and the use of the analogies between the FLC and PID controllers. Dealing with real systems however, the limitation of the experimental trials may become an important drawback in this type of analysis, as the extracted information may be very poor. Moreover, the qualitative and quantitative analysis of the interaction between the tuning parameters and factors is usually not considered during the design of these types of FLCs due to their complexity. Finally, the robustness of the system under study is not considered, in most of the proposed approaches, even if it is a key aspect in fuzzy logic type controllers.

In this thesis an innovative approach to determine the optimal parameters of control systems in terms of robustness and tuning characteristics is proposed. The number of experiments is minimised using Taguchi Design of Experiment method. The method also analyses the results of each performance criterion, investigates the significance of the parameters/factors of a system together with their optimal levels with regards to their robustness and combining all the performance criteria that are defined by the designer. The qualitative and quantitative analysis of the interaction between the tuning parameters such as the scaling factors and peaks of membership functions is also investigated. Finally, the parameters and/or factors are tuned using a proposed method called a "fuzzy combined scheduling system".

The proposed approach has been applied in the development of Fuzzy-like PD controllers to control the yaw and depth of an underwater vehicle. The design of these controllers was based on a minimum number of experimental trials. The information that was extracted from these trials was enough to optimise and tune the scaling factors of the controller resulting in very good performance.

1.3.4 A final Remark

Finally it must be noticed, that while in the beginning of fuzzy control, fuzzy rule-base systems were understood as relevant to artificial intelligence, it had been rejected as a non-orthodox approach that was not purely symbolic processing. To date, a big part of the fuzzy logic supporters tend to reject symbolic artificial intelligence as not capable of dealing with real complex systems analysis tasks. Recently *soft computing* has been proposed as an artificial intelligent approach to learning and machine intelligent. Mixing fuzzy rules/sets, neural networks and genetic algorithms may create some dangerous aspect in artificial intelligence research. First, the new steam of numerical modelling and controlling, while rejecting the methodology of symbolic AI and keeping the AI vocabulary result in a terminological confusion. Second, numerical methods and symbolic approaches are once again presented as competing while they are really complementary. It seems impossible to get rid of the language level when communicating with humans whilst overemphasising numerical modelling and control may result in cutting fuzzy logic from its roots.

This research attempts to keep fuzzy logic in its initial definition by proposing techniques to define the parameters of fuzzy sets and to assist the knowledge base definition and thus improving the performance of the fuzzy modelling and control. The main purpose of this research was to make bridges between fuzzy logic theory and other artificial intelligence techniques as well as other conventional techniques such as Taguchi design of experiments. It is believed, therefore, that fuzzy logic is an intelligent technique, that can perform independently as well as together with other techniques, however, it is not an "artificial" one, since the knowledge comes from the human experts.

1.4 Overview of the thesis

The rest of the thesis is constructed as follows:

- Chapter two describes fuzzy and neuro-fuzzy modelling techniques as well as the structure of the Takagi-Sugeno fuzzy models, a fuzzy supervisory schedule system and neuro-fuzzy modelling methods to approximate local models. A proposed hybrid fuzzy and neuro-fuzzy method for defining global models is described and applied to model the yaw rotation of an underwater vehicle.

- Chapter three explains how fuzzy clustering methods can be used as a technique for identifying and modelling control strategies. The most wildly used fuzzy cluster methods are overviewed defining their advantage and disadvantages. A method to determine the number of clusters as well as to project the cluster centres and variances is presented. An approach to generate fuzzy rules from *n*-dimensional input/output clustering data is proposed and developed.

- Chapter four describes the implementation of the proposed approach of chapter three to generate fuzzy rules for "avoid objects" control strategy of an underwater vehicle using 3-D input/output data space.

- Chapter five proposes an innovative approach for the design and tuning of robust control systems using Taguchi method and fuzzy logic. The steps of the proposed methodology are presented and discussed analytically. The proposed approach is developed and applied to the optimisation of the scaling factors and peaks of membership functions for tuning in fuzzy-like PD controllers as case studies.

- Chapter six continues with implementation of the proposed approach in chapter five for the steering and depth control of an underwater vehicle. Real experimental results are presented in this chapter to demonstrate the capabilities of the approach.

- Chapter seven completes the thesis by giving conclusions and future work which follows from the work presented in each of the above chapters

Figure 1.2 depicts a schematic outline of how the thesis is organised where the chapters that include the application for the underwater vehicle named GARBI points to GARBI's picture.

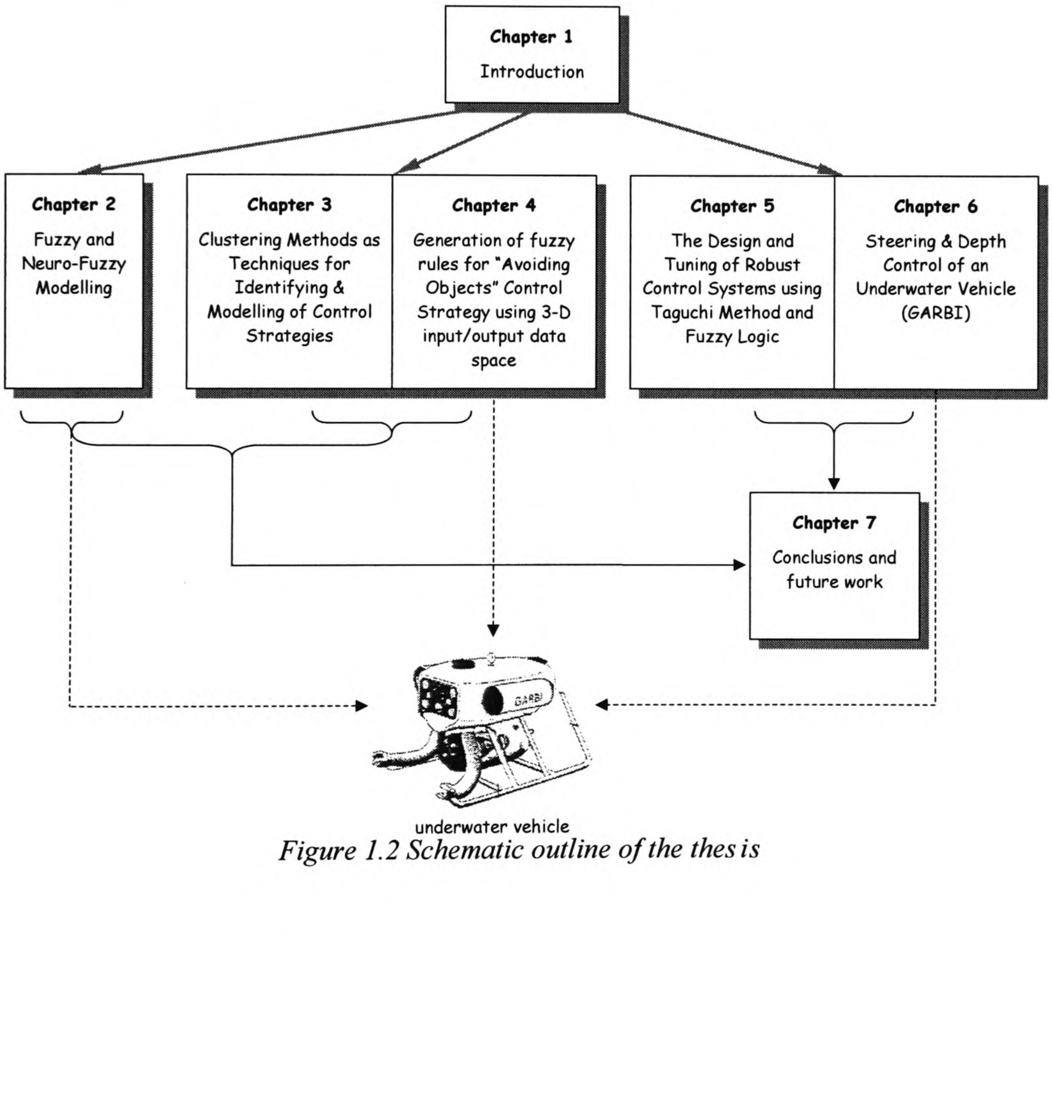


Figure 1.2 Schematic outline of the thesis

1.5 References

Babuska, R. 1998a. *Fuzzy Modelling for Control.* Boston: Kluwer Academic Publishers. 0792381548.

Bellman, R.,Kalaba, R., and Zadeh, L. A. 1966. Abstraction and Pattern Classification. *J. Math. Anal. anf Appl.,* **2**, pp. 581-586.

Berenji, H. R., and Khedar, P. 1992. Learning and Tuning Fuzzy Logic Controllers Through Reinforcements. *IEEE Trans. on Neural Networks,* **3** (5), pp. 724-740.

Bezdek, C. J.,Keller, J.,Krisnapuram, R., and Pal, N. R. 1999. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing.* Kluwer Academic Publishers.

Bezdek, J., and Dunn, J. 1975. Optimal Fuzzy Partition: a Heuristic for Estimating the Parameters in a Mixture of Normal Distributions. *IEEE Trans. Computers,* **C-24**, pp. 835-838.

Buckley, J., and Hayashi, Y. 1993. Fuzzy Input-Output Controllers are Universal approximators. *Fuzzy Sets and Systems,* **58**, pp. 273-278.

Castro, J. 1995. Fuzzy Logic Controllers are Universal Approximators. *IEEE Trans. on Systems, Man and Cybernetics,* **25**, pp. 629-635.

Chi, Z.,Yan, H., and Pham, T. 1996. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition.* Singapore: World Scientific Publishing Co. Pte. Ltd.

De Silva, C. W. 1995. *Intelligent Control : Fuzzy Logic Applications.*

Drachmann, A. G. 1948. *Ktesibios, Philon, and Heron, a Study in Ancient Pneumatics.*

Gustafson and Kessel 1979. *Fuzzy Clustering with a Fuzzy Covariance Matrix.* IEEE CDC . San Diego.

Jang, J-S. R. 1993. ANFIS: Adaptive - Network - Based Fuzzy Inference System. *IEEE Trans. on Systems, Man and Cybernetics,* **23** (3), pp. 665-685.

Jantzen, J., 15 May 1998. "Design of Fuzzy Controllers." Denmark: Technical University of Denmark.

Korner, S 1967. Laws of thought. *In: Encyclopædia of Philosophy.* NY: MacMillan, pp. 414-417.

Kosko, B. 1992a. *Fuzzy Systems as Universal Approximators.* Proc. IEEE Int. Conf. on Fuzzy Systems.

Kosko, B. 1992b. *Neural Networks and Fuzzy Systems, A Dynamic Systems Approach to Machine Intelligence.* Prentice Hall, Inc.

Lee, C. C. 1990a. Fuzzy Logic in Control systems: Fuzzy Logic Controller Part I. *IEEE Transactions on Systems, Man and Cybernetics,* **20**, pp. 404-418.

Lejewski, C. 1967. Jan Lukasiewicz. *In: Encyclopædia of Philosophy.* NY: MacMillan, pp. 104-107.

Mamdani, E. H. 1974. Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. *Proceedings of the Institution of Electrical Engineers,* **121**, pp. 1585-1588.

Mamdani, E. H., and Assilian, S. 1975. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *Int. J. Man-Machine Studies,* 7 (1), pp. 1-13.

Mudi, K P., and Pal, R. N. 1999. A Robust Self-Tuning Scheme for PI- and PD - Type Fuzzy Controllers. *IEEE Trans. on Fuzzy Systems,* 7 (1), pp. 2-16.

Ruspini, E. H. 1969. A New Approach to Clustering. *Information and Control,* **15**, pp. 22-32.

Wang, L. X. 1992. *Fuzzy Systems are Universal Approximators.* Proc. IEEE Int. Conf. on Fuzzy Systems. San Diego.

Yager, R. R., and Filev, D. P. 1994a. Approximate Clustering Via the Mountain Method. *IEEE Transactions on Systems, Man, and Cybernetics,* **24**, pp. 1279-1284.

Yager, R. R., and Filev, D. P. 1994b. *Essentials of Fuzzy Modelling and Control.* New York: John Wiley. 0471017612.

Zadeh, L. A. 1965. Fuzzy Sets. *Information and Control,* **8**, pp. 338-353.

Zadeh, L. A. 1968. Fuzzy Algorithms. *Information and Control,* **12**, pp. 94-102.

Zadeh, L. A. 1973. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man and Cybernetics,* **SMC-3**, pp. 28-44.

Zheng, L. 1992. A Practical Guide to Tune of Proportional and Integral (PI) Like Fuzzy Controllers. *Proc. FUZZ IEEE,* pp. 633-641.

2

Fuzzy & Neuro-Fuzzy Modelling

2.1 Introduction

Developing mathematical or other type of dynamic models of real systems is a central topic in control system theory and important steps in the design of control, supervision and fault-detection systems. Some of the modelling approaches for non-linear systems that been researched the last three decades are based on techniques called artificial intelligence such as neural network, fuzzy logic, neuro-fuzzy and genetic algorithms.

Fuzzy modelling is the method of describing the characteristics of a system using fuzzy rules. Compared to other "intelligent" modelling techniques, such as neural networks(Haykin, 1994) or radial basis function networks (Chen *et al*, 1991), fuzzy systems provide a more transparent representation of the non-linear systems under study, and can also be given a linguistic interpretation in the form of rules. Moreover, fuzzy sets serve as a smooth interface between qualitative variables involved in the rules and numerical domains of the inputs and outputs of the model. The rule-base nature of fuzzy models allows the use of information expressed in the form of natural language statements, and makes the models transparent to interpretation and analysis. At the same time, at the computational level, fuzzy models can be regarded as flexible mathematical structures, similar to neural networks or radial basis function networks, that can approximate a large class of non-linear system to a desired degree of accuracy (Wang, 1992), (Kosko, 1992), (Zeng and Singh, 1994). This duality allows qualitative knowledge to be combined with quantitative data. Finally, it can be said that the use of linguistic qualitative terms in the rules can be regarded as a kind of information quantisation. Thus, depending on the number of qualitative values considered, models at different levels of abstraction and accuracy can be developed for a given system. Each of the models may serve a different purpose such as prediction, controller design, monitoring.

Modelling using fuzzy logic is a topic that has been studied extensively in recent years(Takagi and Sugeno, 1985) (Jang and Sun, 1995) mostly as a problem of function approximation instead of a problem of knowledge acquisition. A good fuzzy model, when is only based on expert's knowledge, is a trade off between accuracy and linguistic meaning. It is a well-known fact that if the accuracy is increased, the linguistic interpretation is reduced. If therefore the interest is to have a very accurate model some other techniques (like Neural Nets, multi-layer perceptrons and radial basis functions), are better because these techniques do not have the inconvenience that fuzzy systems have, which usually comes from its high dimensionality. However, fuzzy systems provide a more transparent representation of the non-linear systems under study, and can also be given a linguistic interpretation in the form of rules. In this way, systems data can be translated into a model and analysed in a manner of linguistic meanings.

In this work fuzzy modelling is approached by combining the function approximation together with knowledge acquisition approaches. The latter can be achieved by using a proposed Fuzzy logic Supervised Scheduling System (FSSS) where the former is solved by using Neuro-Fuzzy approach. A main feature of the proposed hybrid fuzzy and neuro-fuzzy model approach is that it approximates a non-linear system by a set of local models within defined fuzzy regions that describe the system globally (Palman *et al*, 1997).

The proposed method is implemented in a model of an underwater vehicle that is inherently non-linear and only some experimental results describing its behaviour in a real environment are available. As the neuro-fuzzy part of the model needs good data, in terms of representation of the dynamics of the system, and the fuzzy part needs good linguistic reasoning interpretation, the experiments were planned regarding these requirements.

This Chapter is structured as follows: in Section § 2.2 different modelling techniques are introduced. In Section § 2.3 the three main types of fuzzy modelling are presented. As the Takagi-Sugeno fuzzy model is used in this work, the criteria of how to select its parameters are discussed in Section § 2.4. Section § 2.5 describes a proposed fuzzy supervisory scheduling system method. Moreover, in Section § 2.6 it is discussed how a neuro-fuzzy modelling method defines local models. A proposed hybrid fuzzy and neuro-fuzzy model method that defines a global model is described in Section § 2.7. The proposed method is applied to model the yaw rotation of an underwater vehicle in Section § 2.8. The hydrodynamic forces and moments acting on the vehicle are presented in Section § 2.8.1. The experimental design and the trials for the yaw rotation in a real environment are discussed in Section § 2.8.2. From these results the local models are defined as described in Section § 2.8.3. Then, fuzzy models are constructed to define the relationship between fuzzy inputs and the local models as discussed in Section § 2.8.4. The global fuzzy model is designed as explained in Section § 2.8.5 and some simulation results are presented in Section § 2.8.6. The work presented in this chapter is discussed in Section § 2.9 and summarised in Section § 2.10.

2.2 Different modelling techniques based on fuzzy systems theory, neural networks and neuro-fuzzy approaches

Traditionally, modelling is seen as a conjunction of a thorough understanding of the system's nature and behaviour, and a suitable mathematical treatment that leads to a usable model. This approach is usually termed "white box" (physical, mechanistic, first-principle) modelling. In practice, however, when complex and poorly understood systems are considered, the requirement for a good understanding of the physical background of the system proves to be a severe limiting factor. The difficulties that can arise in conventional "white-box" modelling approaches appear from poor understanding of the underlying phenomena, inaccurate values of various process parameters, or from the complexity of the resulting model. A complete understanding of the underling mechanisms is virtually impossible for a majority of real systems. However, gathering an acceptable degree of knowledge needed for physical modelling may be very difficult, time-consuming and an expensive task. Even if the structure of the model is determined, a major problem of obtaining accurate values for the parameters remains. It is the task of system identification to estimate the parameters from data measured. Identification methods have been developed to a mature level, mostly, for linear systems. Most real systems are, however, non-linear and can be approximated by local models.

The accuracy of mathematical models is based on how good are the approximation of the mathematical functions that are used to describe the system's characteristics under study. If the model is not accurate enough, the subsequent steps of analysis, prediction and controller synthesis, cannot be successful. However, there is an obvious trade-off between the necessary accuracy of the model and its complexity. Models should provide information at the most relevant level of precision (abstraction), suppressing unnecessary details when appropriate. If the model is too simple, it cannot properly represent the characteristics of the system and does not serve its propose. However, the model should not be too complex if it is to be practically useful.

An other approach to identify a model of non-linear systems is to use some sufficiently general "black-box" structures, such as, Artificial Neural Networks (ANN) (Patterson, 1996), used as a general function approximator. The modelling problem is then that of obtaining an appropriate structure of the approximator, in order to correctly capture the dynamics and the non-linearity of the system. In "black-box" modelling, the structure of the model is hardly related to the structure of the real system. The identification problem consists of estimating the parameters in the model. If representative system data is available, "black-box" models usually can be developed quite easily, without requiring system-specific knowledge. A severe drawback of this approach is that the structure and parameters of these types of models usually do not have any physical significance. Additionally such models cannot be used for analysing the system's behaviour otherwise than by numerical simulation. Finally, it is neither possible to use prior knowledge to initialise the network, nor can its final state be interpreted in terms of rules.

The drawback of the conventional "white-box" and "black-box" techniques in modelling non-linear system is their trade-off between accuracy and knowledge acquisition as well as that they are based mostly on quantitative mathematical techniques. The weakness of the traditional quantitative techniques to adequately describe complex systems was summarised in the well-known principle of incompatibility, formulated by L. Zadeh, (1973). This principle states *that "as the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes, until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics".*

Fuzzy model identification is a technique that has been developed based on fuzzy set theory (Sugeno and Kang, 1988), (Yager and Filev, 1994b), (Harris *et al*, 1994), (Babuska, 1998a). Fuzzy models can be seen as logical models, which use *"If-Then"* rules and logical operators to establish qualitative relationships among the variables in the model. In many applications it is often desirable to combine qualitative information and numerical data with qualitative and heuristic knowledge. To achieve this, user-friendly methods are needed for efficient translation of the knowledge into a computer-manageable form, for interfacing qualitative information with numerical data and for appropriate validation of the models. The rule-base nature of fuzzy models allows the use of information expressed in the form of natural language statements, and makes the models transparent to interpretation and analysis. Modelling based on the use of fuzzy sets in combination with local regression techniques has a great potential to achieve these goals.

The specification of good linguistic rules depends on the knowledge of the expert, but the translation of these rules into fuzzy set theory depends on the choice of certain parameters, such as shape and degrees of membership functions, for which particular rules do not exist. Fuzzy models are unable to learn by their own experiences and unable to adapt to new conditions in cases of non-linear systems (of high order with uncertainties in parameter and structure).

The advantages of the fuzzy approach are mainly the disadvantages of the ANN approach, and vice versa. So the idea is naturally to combine neural networks and fuzzy systems to overcome their disadvantages, but to retain their advantages. The integration of these two fields has given birth to neuro-fuzzy systems. To develop a fuzzy neural structure, two ideas can be followed: the first is to "neuralise" existing fuzzy systems and the second one, is to "fuzzify" existing neural networks, that is to introduce either neural concepts into fuzzy systems or fuzzy concepts into neural networks.

Neuro-fuzzy systems or neuro-fuzzy models can be divided into *co-operative* and *hybrid models. Co-operative* approaches use neural networks to determine certain parameters (e.g. the fuzzy sets, or the fuzzy rules) of fuzzy controller which are then implemented without use of neural nets. H*ybrid* approaches create a new architecture using concepts from both paradigms and thus can be interpreted as a neural net and as a fuzzy controller. Besides this, there are *concurrent neural/fuzzy models* that use neural networks and fuzzy systems separately.

2.3 Fuzzy Modelling

Fuzzy *If-Then* rules are used not only to incorporate human knowledge in fuzzy expert systems and controllers, but also can be applied to modelling of non-linear dynamic systems. Thus, a fuzzy model describes the relationships between the system's input/output variables by means of *if-then* rules, such as:

If the engine power is HIGH ***then*** *the* speed *is* FAST

These rules establish local relations between the system's variables by relating qualitative values of one variable (power is high) to qualitative values of another variable (speed of the vehicle will increase fast). The qualitative values typically have a clear linguistic interpretation and are called linguistic terms. The meaning of the linguistic terms with regard to the input/output variables which may be numerical (engine power, speed) is defined by appropriate fuzzy sets. In this sense, fuzzy sets (or more precisely their membership functions) provide an interface between the input and output numerical variables and the linguistic qualitative values in the rules.

Depending on the particular structure of the consequent part of the "*If-then*" rule, three types of models are distinguished:

- *Linguistic fuzzy model* (Zadeh, 1973); (Mamdani, 1977), where a fuzzy rule can be written as *"if x is A then y is B"*, with the premise and consequence expressed in the form of fuzzy sets.

- *Fuzzy Relational model* (Pedrycz, 1984); (Yi and Chung, 1993), which can be regarded as a generalisation of the linguistic model, allowing one particular antecedent proposition to be associated with several different consequent propositions via fuzzy relation.

- *Takagi-Sugeno's (T-S) fuzzy model* (Takagi and Sugeno, 1985), where a fuzzy rule can be written as *"if x is A then $y = f(x)$ "*. In this case, the premise is a classical fuzzy set expression which indicates a fuzzy subspace and the consequence is a functional relation, usually a linear function or a singleton which indicates the input-output relationship in this fuzzy subspace.

In this research, the last structure is considered. There are two reasons for this choice: firstly, this type of fuzzy model is more powerful for representing the input/output behaviour of non-linear systems when dynamics depend on the operating conditions (Babuska and Verbruggen, 1997); secondly, because the consequence part of a T-S fuzzy model rule is a linear equation rather than a fuzzy set. The fuzzy model can be viewed as a collection of linear sub-models and the existing linear control theory can be applied to analyse the stability and for designing a model-based set of linear controllers (Tanaka and Sugeno, 1990), (Sugeno and Kang, 1988). The construction, however, of a rule-based fuzzy model requires:

1. identification of the antecedent and consequent structure of the membership functions for different operating regions and

2. estimation of the consequent regression parameters.

The second task is solved using functional or singleton consequences, defined as local linear regression models. However, the construction of the membership functions in the first task is a non-linear optimisation problem.

2.4 Structure of T-S fuzzy Model

The T-S fuzzy model is a special case of fuzzy systems and has been one of the most widely used structures in model identification techniques of non-linear systems. However, to construct this type of model is not an easy task due to the large number of degrees of freedom which include shape and number of membership functions and aggregation methods. This large number of degrees of freedom gives high flexibility to the fuzzy system but also demands more systematic criteria to choose these parameters. The following subsections outline the structure of the T-S model as well as the criteria to select some of its parameters.

2.4.1 The Fuzzy State Space

The Fuzzy State Space technique can express complex non-linear dynamic systems by linguistic *if-then* rules.

A typical Takagi-Sugeno fuzzy model has the form:

$$R_s^i \;\; if \;\; s = \Delta S^i \;\; then \;\; x_i = f_i(x,u)$$

where s is the operating point vector $s = \{s_1, s_2, \ldots, s_{n_s}\}$, f is a $n \times 1$ non-linear vector function, x is the $n \times 1$ state vector, and u is the $n \times 1$ input vector. The vector consists, in general, of state, input and output variables (n_s is its dimension). These variables comprise the *premises of implications*. ΔS^i is the *i*-th *fuzzy state vector* as in Equation 2.1

$$\Delta S^i = (\Delta S_1, \Delta S_2, \ldots, \Delta S_{n_s})^T \tag{2.1}$$

where ΔS_{n_s} is the fuzzy values of s with appropriate membership function or the membership function of the fuzzy sets in the premises, known as *premise parameters*.

For each fuzzy state vector ΔS^i there is a set of crisp state vectors {s*}, each one satisfying the given fuzzy state vector to a certain degree. Thus for each fuzzy state vector ΔS^i, it is possible to construct a fuzzy set defined on the domain of these crisp state vectors. This fuzzy set is such that the degree of membership of crisp state vector is equal to the degree of satisfaction of ΔS^i by the particular crisp state vector. This fuzzy set is called a *fuzzy region* and the number of fuzzy regions is equal to the number of fuzzy state vectors.

The centre of fuzzy region, $\Delta S^i = (\Delta S_1^i, \Delta S_2^i, ..., \Delta S_n^i)^T$, is defined as this crisp state vector $s^i = \{s_1^i, s_2^i, ..., s_{n_s}^i\} \in S^T$, where s_n^i are crisp values, such that

$$\mu_{LS_1}(x_1^i) = 1, \mu_{LS_2}(x_2^i) = 1, ..., \mu_{LS_n}(x_n^i) = 1 \tag{2.2}$$

where μ_{LS_n} is the degree of satisfaction of ΔS_n^i.

The *then*-part of this fuzzy rule defines a linear autonomous open loop model representing the system dynamics within the fuzzy region ΔS^i specified in the *if*-part of the same fuzzy rule. This model is of the form $x_i = f_i(x,u)$ where f_i is a linear or non-linear function normally obtained via an identification procedure. The parameters of x_i are called *consequence parameters*.

2.4.2 Outline of the basic Structure of the Fuzzy Inference System and fuzzy rule's parameters

Fuzzy Inference Systems (FIS) for modelling and identification can be regarded as flexible mathematical structures that can approximate a large class of linear and non-linear systems, called "universal approximators". However the FIS may approximate the system dynamics in different ways but not always give the most appropriate ones. The meaning of "appropriate" in this case is referring to the linguistic meaning. The alternatives involved in the structure selection of the FIS are very wide: shapes of membership functions, *and* and *or* operations, implications, defuzzification method are some of them. Some choices therefore have to be made otherwise the problem becomes too complex. Some general rules of how these parameters can be defined is presented as follows:

Premise Variables

- The premise variables are chosen out of possible input variables under consideration. Given a set of possible candidates, it is possible to find a set of input variables, which affect the output of the system. In most applications, this operation is not difficult since the input variables, which affect the outputs are known to the designers.

Premise Parameters Identification

- The choice of the variables in the premises implies that its space is divided. A general theoretical approach to identify the number of these divisions or fuzzy subspaces described by membership functions does not seem to be available. A general rule can be that the number of membership functions in every universe of discourse for the antecedent should be 7 ± 2. This number is related to the maximum number of linguistic quantifiers that a human being can handle. However, if the number of experiments that are held to collect data for identification purpose is limited, the number of the fuzzy sets should be chosen considering this number.

- The shape of the membership functions for the fuzzification process is triangular (Pedrycz, 1994) with overlap 50%. Thus every point in the universe of discourse belongs to at least two fuzzy sets.

Consequence Parameters Identification

- The consequences are singletons or linear/non-linear combinations of the inputs that represents local models as discussed in Section § 2.5.

Operational parameters of rules

- The operations "product" and the "bounded sum" are selected as *and* and *or* operations (Mendel, 1995).

Using the above suggestions a basic structure for modelling of non-linear systems is developed.

2.5 Fuzzy logic as a method to design a Fuzzy Supervised Scheduling System (FSSS)

The reason for adopting the fuzzy model described above is that most systems are non-linear and therefore cannot be described by a single linear model. Instead of constructing complicated non-linear models based on physical laws, an alternative approach can be used constructing local models. In this case, each local model approximates the original non-linear system around different operating points and the FSSS determines which of the local models suits the particular operating conditions. In this case the FSSS uses a discriminant function, f_d, of Equation 2.3.

$$w = f_d(s) \tag{2.3}$$

This function defines a weight vector $w = \{w_1, w_2, ..., w_k\}$, with $w_i \in [0,1]$, for each particular value of the operating point vector. This definition can be applied using *min* or *max* operation. For example, for the particular crisp value $s^* = \{s_1^*, s_2^*, ..., s_n^*\}$ of the operating point vector, that is defined from *and* operation in the fuzzy rules, the weight vector is as in Equation 2.4.

$$w = \min(\mu_{LS_1}(s_1^*), \mu_{LS_2}(s_2^*), \ldots, \mu_{LS_{n_s}}(s_{n_s}^*)) \tag{2.4}$$

where $\mu_{LS_{n_s}}$ is the degree of membership of the crisp value $s_{n_s}^*$ of s_{n_s}.

The overall output X of the composite model is calculated as the weighted mean of the outputs x_i of the local models, given as in Equation 2.5, that actually represents the non-linear expression for the global system.

$$X = \frac{\sum_{i=1}^{k} w_i \cdot x_i}{\sum_{i=1}^{k} w_i} \tag{2.5}$$

where k is the number of local models.

2.6 Neuro-Fuzzy Modelling Methods to approximate local models

The combination of FIS and ANN is expected to sum the positive features of these two systems. From ANN, the powerful learning capabilities enable these systems to learn from a set of system measurements, whereas the fuzzy presentation enables the extraction of learnt information in a form easily understandable even to a less experienced user. As the former property reduces the time required to create the model, the latter increases the usefulness of the model since there now exists at least some kind of explanation for the model outcome. Several fuzzy-neural approaches have been introduced and can be found in the literature (Jang *et al*, 1997), (Nauck and Kruse, 1996).

In this thesis a neuro-fuzzy architecture proposed by Jang, (1993) called the Adaptive Network-based Fuzzy Inference System (ANFIS) is applied. The architecture of this technique consists of adaptive networks that are functionally equivalent to fuzzy inference systems. The approach is widely used as it provides quite an efficient way to approximate non-linear systems with high accuracy. However, despite the fact that the method gives some transparency in the systems properties by investigating the parameters of the fuzzy sets, this does not always include the system's integrity. Moreover, even the initial setting of the fuzzy inference part of the ANFIS architecture is defined by a designer the ANFIS algorithm may change the architecture dramatically and thus the knowledge that is initially set by the expert may end up with minor or almost no significance. That is because the method inherently looks at the case as a function approximation. The explanation of how the architecture works is presented in Jang, (1993)

2.7 Design of a proposed Hybrid Fuzzy and Neuro-Fuzzy Model (HFNFM) architecture for defining the global model

The model identification method proposed in this thesis is a Hybrid between Fuzzy and Neuro-Fuzzy Model (HFNFM) architecture since it combines Fuzzy and Neuro-Fuzzy Model techniques as presented in this section.

The neuro-fuzzy (ANFIS) architecture is used to optimise the parameters of both antecedent and consequence parts of the fuzzy rules that define the local models of a non-linear system. This is a case where the definition of the local fuzzy models is referred to as a function approximation problem.

For the global model, the fuzzy model structure settings are defined by linguistic concepts from the experimental design that is used to extract information about the dynamics of the system under study. In this case the fuzzy model is considered as a problem of knowledge acquisition. This model operates as the FSSS of the local models as explained in Section § 2.5. Figure 2.1 shows schematically the combination of the two types of models from which the HFNFM architecture arises.

The approach is constructed by the following step procedure:

1. Define the dynamics that need to be modelled.

2. Define the input/output global "fuzzy" variables that excite the system's dynamics.

3. Construct the experiments for all combinations of the input/output variables.

4. Define the local models from the data and optimise their representative function by using neuro-fuzzy technique (ANFIS) as discussed in Section § 2.6.

5. Construct fuzzy models to describe the relationships between the fuzzy inputs defined in step 2 and the defined local models from step 4, as discussed in Section § 2.4.

6. Apply the FSSS (Section § 2.5) to combine the local models for each fuzzy region and thus define the global fuzzy model. Therefore, each local fuzzy model is weighted by the degree of satisfaction of the present operating fuzzy region and the degrees of satisfaction of its neighbouring fuzzy regions. In between the centres of neighbouring regions the overlapping of these fuzzy regions ensures the smoothness of trajectory between the local fuzzy models.

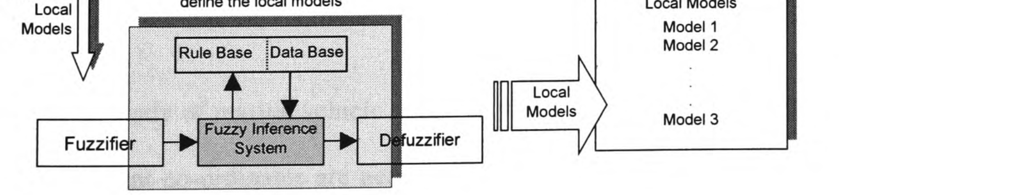


Figure 2.1 Hybrid Fuzzy and Neuro-Fuzzy Model approach

In the next section the HFNFM approach will be applied to model the yaw of an underwater vehicle.

2.8 Implementation of the HFNFM method to model the yaw of an underwater vehicle

This section presents the development of a yaw model of a low-cost Remotely Operate Vehicle (ROV) named GARBI developed at the Polytechnic of Barcelona and the University of Girona in Spain. The vehicle, which is illustrated in Figure 2.2, is used for underwater mission operations such as observations and inspections. An umbilical cable carries power and provides communication links to a surface ship or other operating platform (Amat *et al*, 1999). The length, width and height of the vehicle are 87cm, 72cm, 85cm respectively. The four propellers of the vehicle have 200W of power each performing at maximum speed of 3 knots.

Figure 2.2 GARBI underwater vehicle

2.8.1 The hydrodynamic Forces and Moments of GARBI

The motion study of marine vehicle involves *six Degrees Of Freedom* (DOF) (Table 2.1), since six independent co-ordinates are necessary to determine the position and orientation of a rigid body. The first three co-ordinates Surge, Sway and Heave and their time derivatives correspond to the position and translational motion along the *x-, y-, and z-axes.* The last three co-ordinates Roll, Pitch and Yaw (or heading angle) and their time derivatives are used to describe orientation and rotational motion. In GARBI the motions in x and z direction (Surge and Heave) are controlled by the horizontal propellers (T_1, T_2) and vertical propellers (T_3, T_4) respectively (Figure 2.3). Moreover, the horizontal propellers (T_1, T_2) control the rotation in z-axis (Yaw).

DOF		Forces	Linear Velocity	Positions
1	Motions in the x-direction (Surge)	X	u	x
2	Motions in the y-direction (Sway)	Y	v	y
3	Motions in the z-direction (Heave)	Z	w	z
		Moments	Angular Velocity	Euler angles
4	Rotation in the x-axis (roll)	K	P	φ
5	Rotation in the y- axis (Pitch)	M	Q	ψ
6	Rotation in the z- axis (Yaw)	N	R	θ

Table 2.1 Relationship between the DOF with the forces, linear speed and Position

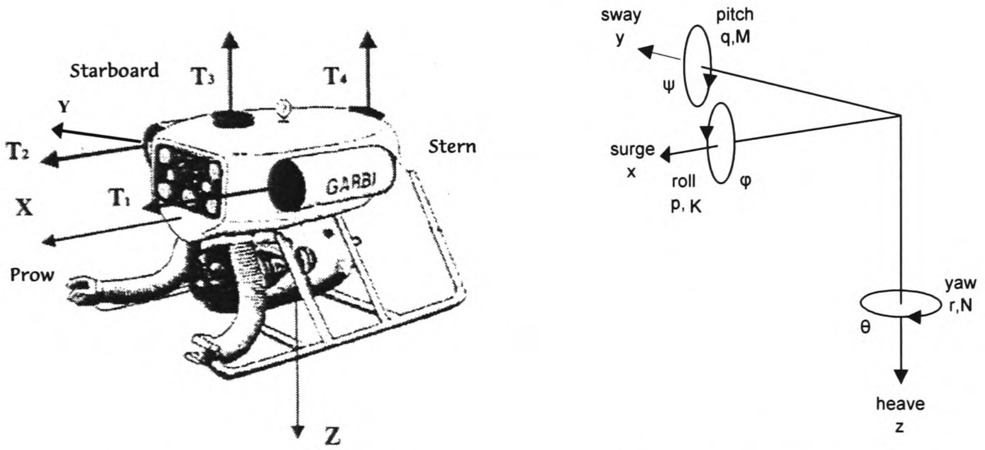


Figure 2.3 GARBI Body-fixed reference frames showing the six degrees of freedom.

The structure of GARBI is designed in such a way that Pitch and Roll cross-coupling is virtually non-existent. However, coupling appears between yaw control and surge in x-direction only when the vehicle has initial speed. Nevertheless, this coupling is expected and acceptable due to the navigation properties. No other major couplings are present.

In this research the yaw is used for investigating the modelling that is implemented using the proposed HFNFM method. The limitation of the study allows only the yaw to be modelled and is based on experimental data that were available only for GARBI's rotation about the z-axis.

2.8.2 Experimental design and the trials for the rotation in z-axis (yaw) in a real environment

As discussed in the previous subsection the rotation yaw is affected by the power of the horizontal propellers T_1 and T_2. The input variables are the set point of the yaw and the applied power (or difference of power) in propellers T_1 and T_2 to reach this point. The smallest voltage that can be applied to the propellers is 3V, whereas the maximum one is 10V. The output, therefore, is the actual yaw according to a sampling time *t*.

In October 1999 experimental trials in a real environment (Lake Banyolas, Spain) were held to test the yaw behaviour of the robot. Due to the limitation of the number of experiments (as they are time and money consuming), it was decided that three types of power and yaw angle should be applied with their values distributed equally in the universe of discourse that is defined by their minimum and maximum points. Hence, for the voltage the values that are applied are 3V, 6.66V and 10V. Moreover, for the yaw the set points of the angles are 30°, 60° and 90° for left and right turning. These are Zig-Zag type experiments and thus more types of turning can be investigated as will be explained in Section § 2.8.3.

The values of the above trials may be represented as fuzzy values with the peaks of the fuzzy sets equal to these actual values. Thus, for the three values of power and yaw angle the fuzzy sets can be labelled as Low Power, Medium Power, High Power and Low Angle, Medium Angle, High Angle respectively. The nine combinations of the power and the yaw angles are as illustrated in Table 2.2 and the results of these experiments are shown in Figure 2.4 to Figure 2.8. Note that these responses do not follow the linear change of the input set points (voltage and angle) and thus the non-linear behaviour of GARBI's dynamics is illustrated experimentally.

\ Power Angle θ	Low Power	Medium Power	High Power
Low Angle θ	Exp. 1,1	Exp. 1,2	Exp. 1,3
Medium Angle θ	Exp. 2,1	Exp. 2,2	Exp. 2,3
High Angle θ	Exp. 3,1	Exp. 3,2	Exp. 3,3

Table 2.2 The experiments for the 9 combinations of the power and the yaw angles

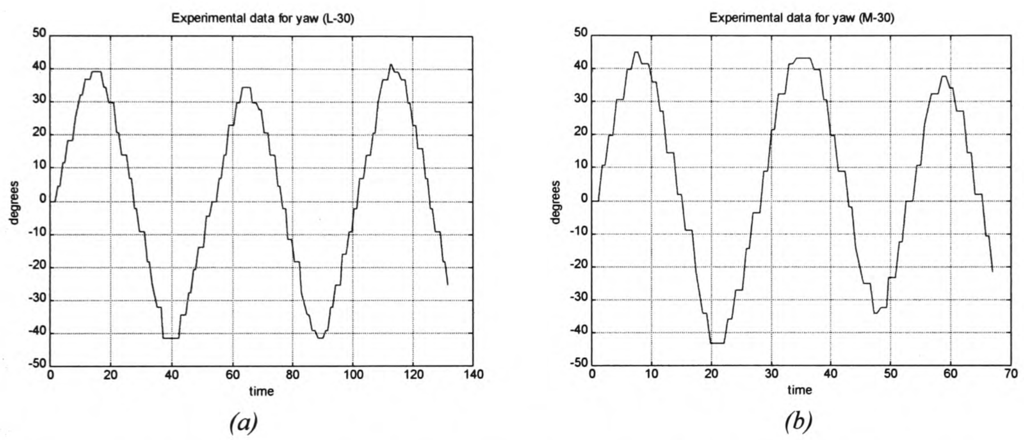


Figure 2.4 Zig-Zag trials when (a) Low Power and Low Angle is applied and (Exp. 1,1), and (b) Medium Power and Low Angle is applied (Exp. 1,2)

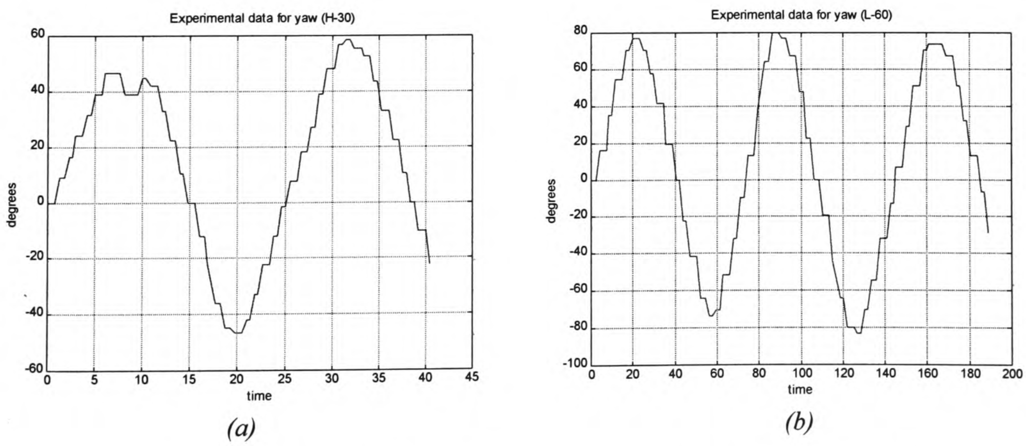
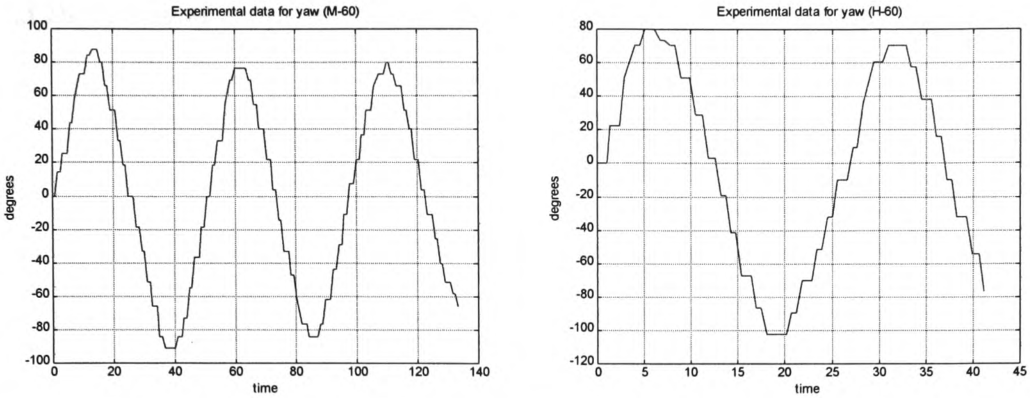
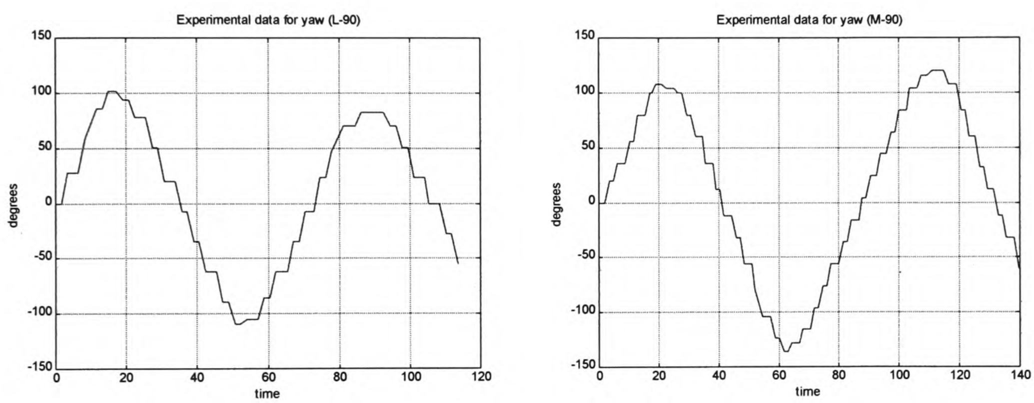


Figure 2.5 Zig-Zag trials when (a) High Power and Low Angle is applied and (Exp. 1,3), and (b) Low Power and Medium Angle is applied (Exp. 2,1)



(a) (b)

Figure 2.6 Zig-Zag trials when (a) Medium Power and Medium Angle is applied and (Exp. 2,2), and (b) High Power and Medium Angle is applied (Exp. 2,3)



(a) (b)

Figure 2.7 Zig-Zag trials when (a) Low power and High Angle is applied and (Exp. 3,1), and (b) Medium power and High Angle is applied (Exp. 3,2)

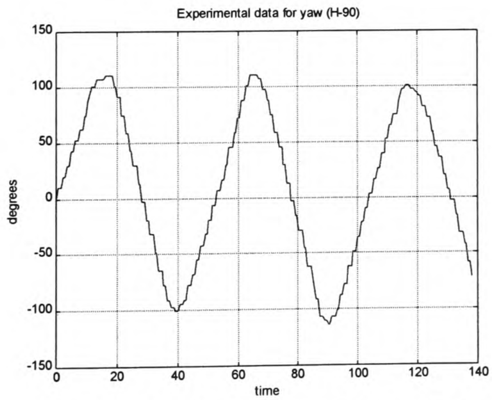


Figure 2.8 Zig-Zag trials when High Power and High Angle is applied (Exp. 3,3)

2.8.3 Definition of the local models from the experimental data

In Table 2.2, nine experiments have been defined to measure the response of the vehicle's yaw rotation for the nine combinations of applied power and set angle θ. The modelling of these yaw responses is a non-linear optimisation problem. This modelling problem is simplified by breaking the model up into a series of local models. Thus, the responses of the yaw motion defined in Figure 2.4 through Figure 2.8 are classified as one of three different types as follows:

- firstly, when the vehicle changes course from straight ahead to left and for Low, Medium and High Angle θ i.e., 30°, 60° and 90°, (it is also assumed that the case is almost the same for straight ahead to right),

- secondly, when the vehicle changes course from left to right and for Low, Medium and High Angle θ i.e., 60°, 120° and 180°.

- thirdly, when the vehicle changes course from right to left and for Low, Medium and High Angle θ i.e., 60°, 120° and 180°.

Note that as discussed in Section § 2.8.2 in all cases the different levels of voltage that are applied to the vehicle's propellers is 3V, 6.66V and 10V.

As discussed in Section § 2.6 these cases can be described by a non-linear identification method. Here the ANFIS architecture is used to model these local non-linear subsystems.Figure A.1 to Figure A.36 in Appendix A shows the training data used for ANFIS and the generated fuzzy local models that are defined as a funtion of time t and angle θ. Thus the rules that construct these models take the form:

Rule i: *If* time t_i is PROPOSITION *then* angle $\theta_i = f(t_i) + c_i$

where i is the number of the rule and c_i is a constant number.

For the local model that represents the yaw of the vehicle when it turns from straight ahead to left or right and when it turns from left to right or right to left, the number of the membership functions that are used are two and three respectively. Different types of membership functions have been applied. However, using Gaussian membership functions, the approximation of these models has the most accuracy with less iterations of the back-propagation algorithm. Note that as the local models under approximation were not very complex, the maximum number of iterations in the ANFIS algorithm was not more than 100.

2.8.4 Constructing the Fuzzy Models to define the relationship between fuzzy inputs and the defined local models

As discussed in Section § 2.8.3 three types of yaw modelling have been defined and optimised according to *time, power and angle* θ variables using the ANFIS optimisation algorithm. These can be considered as the local neuro fuzzy models of the yaw response. Therefore, Table 2.2 can be redefined for each of the three different types of the yaw as shown in Table 2.3 to Table 2.5.

\ Power Left/Right Angle θ	Low Power	Medium Power	High Power
Low Angle θ (30)	Neuro-Fuzzy Local Model HLR 1,1	Neuro-Fuzzy Local Model HLR 1,2	Neuro-Fuzzy Local Model HLR 1,3
Medium Angle θ (60)	Neuro-Fuzzy Local Model HLR 2,1	Neuro-Fuzzy Local Model HLR 2,2	Neuro-Fuzzy Local Model HLR 2,3
High Angle θ (90)	Neuro-Fuzzy Local Model HLR 3,1	Neuro-Fuzzy Local Model HLR 3,2	Neuro-Fuzzy Local Model HLR 3,3

Table 2.3 The local neuro fuzzy models where the course changed from straight ahead to Left/Right

\ Power Right Angle θ	Low Power	Medium Power	High Power
Low Angle θ (60)	Neuro-Fuzzy Local Model LR 1,1	Neuro-Fuzzy Local Model LR 1,2	Neuro-Fuzzy Local Model LR 1,3
Medium Angle θ (120)	Neuro-Fuzzy Local Model LR 2,1	Neuro-Fuzzy Local Model LR 2,2	Neuro-Fuzzy Local Model LR 2,3
High Angle θ (180)	Neuro-Fuzzy Local Model LR 3,1	Neuro-Fuzzy Local Model LR 3,2	Neuro-Fuzzy Local Model LR 3,3

Table 2.4 The local neuro fuzzy models where the vehicle changed course from left to right

\ Power Left Angle θ	Low Power	Medium Power	High Power
Low Angle θ (60)	Neuro-Fuzzy Local Model RL 1,1	Neuro-Fuzzy Local Model RL 1,2	Neuro-Fuzzy Local Model RL 1,3
Medium Angle θ (120)	Neuro-Fuzzy Local Model RL 2,1	Neuro-Fuzzy Local Model RL 2,2	Neuro-Fuzzy Local Model RL 2,3
High Angle θ (180)	Neuro-Fuzzy Local Model RL 3,1	Neuro-Fuzzy Local Model RL 3,2	Neuro-Fuzzy Local Model RL 3,3

Table 2.5 The local neuro fuzzy models where the vehicle changed course from right to left

2.8.5 Design the global fuzzy model

Table 2.3 to Table 2.5 define all the fuzzy rules that can be used to identify the global model of the vehicle in terms of yaw response. Applying the FSSS as described in Section § 2.5, all the local models for each defined fuzzy region are supervised and thus the global fuzzy model is determined. The fuzzy sets of these models are as illustrated in Figure 2.9 and Figure 2.10.

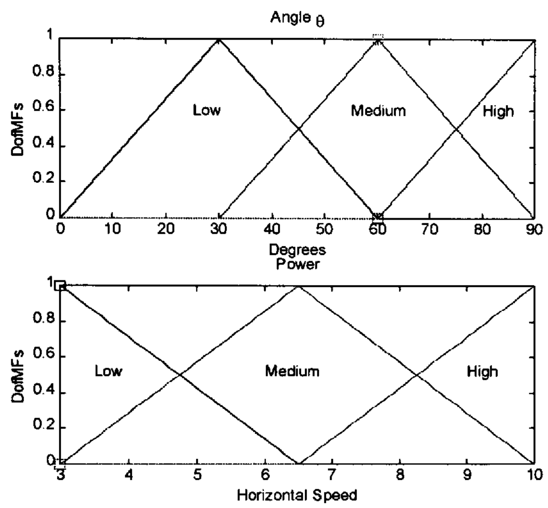


Figure 2.9 Fuzzy sets for the Power and Angle θ when the vehicle changes its course from straight ahead to Left/Right direction

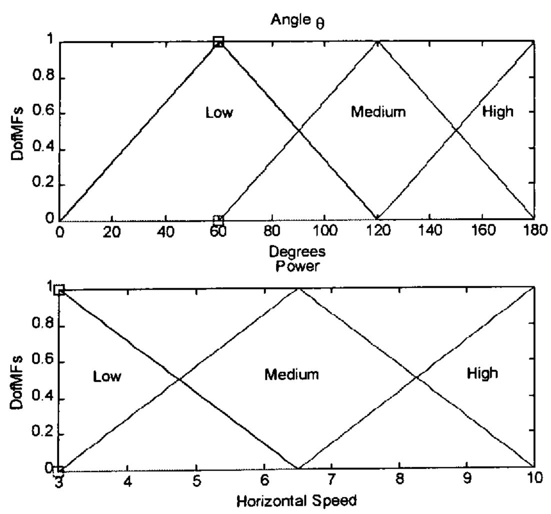


Figure 2.10 Fuzzy sets for the Power and Angle θ when the vehicle changes its course from Left to Right and Right to Left direction

The triangular shape of the membership functions is chosen for the fuzzification process as discussed in Section § 2.4.2.

The rule base of the global fuzzy model is constructed from rules with the form:

$$\textit{Rule } i\text{: } \textit{If} \; \underline{\text{Power}} \; P_i \text{ is } \begin{cases} \textit{Low} \\ \textit{Medium} \\ \textit{High} \end{cases} \textit{and} \; \underline{\text{Set Angle}} \; \theta_i \text{ is } \begin{cases} \textit{Low} \\ \textit{Medium} \\ \textit{High} \end{cases} \textit{then} \; \underline{\text{Overall Angle is}} \; \Theta$$

where i is the number of the rule.

As the *"and"* operation is used in the fuzzy rules the weights w (or degrees of the membership functions) are defined by using *min* operation (Zadeh, 1965).

The overall output of the composite model of the FSSS is calculated as the weighted mean of the outputs θ_i of the local neuro fuzzy models (defined in Appendix A), given as in Equation 2.6, which actually represents the non-linear expression for the global model.

$$\Theta = \frac{\sum_{i=1}^{k} w_i \cdot \theta_i}{\sum_{i=1}^{k} w_i} \tag{2.6}$$

where k is the number of local models.

Remark

- A decision about which of the groups of the local fuzzy model should be fired has to be done. This can be simply achieved by checking the previous and the new setting commands and/or positions in terms of θ. Note also that in the case of turning the vehicle in the same rotation i.e., from left to left or from right to right, the same group of rules are fired initialising only the time variable.

2.8.6 Simulation results

The Hybrid Fuzzy and Neuro-Fuzzy model were developed, using MATLAB/SIMULINK computer program, to observe its capabilities. In Appendix B the basic SIMULINK block diagrams are presented. Two are the set points that are applied for the simulation studies. These are the desired yaw angle and the power of the propellers. Figure 2.11 illustrates some of the simulation results that come from the settings shown in Table 2.6.

No of Exp.	First Turn Left	Applied Power	Second Turn Left	Applied Power	Third Turn Left	Applied Power
(i)	30	5	90	5	60	5
(ii)	30	5	90	7.5	60	7.5
(iii)	60	4	90	4	60	4
(iv)	80	5	120	4	40	4
(v)	30	8	90	10	60	5
(vi)	45	7	30	5	45	6
(vii)	15	6	45	8	30	7.5
(viii)	20	10	70	10	50	5

Table 2.6 Navigation properties of simulation trials

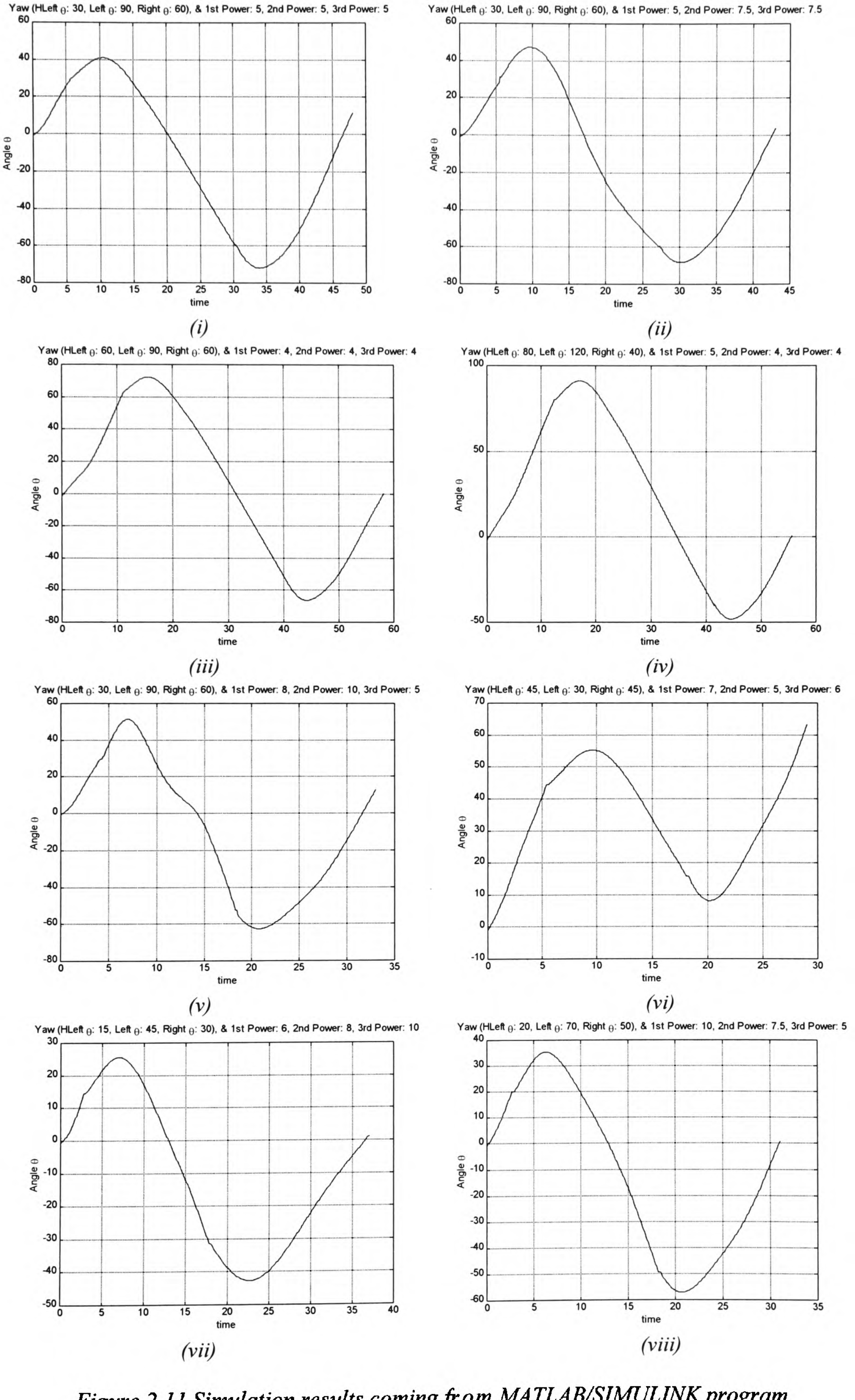


Figure 2.11 Simulation results coming from MATLAB/SIMULINK program

2.9 Discussion

As discussed in Section § 2.2, there are different ways to identify and model non-linear systems including fuzzy logic, neural networks and neuro fuzzy techniques. It is well known however that using fuzzy logic technique for modelling is an approach that mostly depends on good knowledge of the system. Neural network approaches are a very successful, well-known and widely used technique in modelling and identification problems of non-linear systems. Nevertheless, their "black box" structure is their main drawback. Moreover, neuro-fuzzy technique try to combine the above two techniques and to make the model more transparent (or grey). However, the problem in this case is approached as a function approximation instead of knowledge acquisition.

The aim, therefore, was to approach the problem of modelling non-linear systems combining the knowledge acquisition of fuzzy modelling technique with the function approximation of the neuro-fuzzy one. The fuzzy part defines the global model using the proposed FSSS approach to determine which of the defined models that describe the system locally suits the particular operating conditions. By using neuro-fuzzy technique these local models can be approximated. This is the main idea of the proposed HFNFM method that is outlined in Section § 2.7 and implemented for modelling an underwater vehicle in terms of its yaw behaviour as described in Section § 2.8.

The inputs that excite and thus operate the yaw of the vehicle have to be defined. These are the powers that are applied into two horizontal propellers of the vehicle. This definition comes from the design and dynamics properties of the system as discussed in Section § 2.8.1.

To define the vehicle's behaviour in terms of yaw, experiments have been held in a real environment according to the combination of the above two inputs as shown in Table 2.2 (Section § 2.8.2). The results are shown in Figure 2.4 to Figure 2.8 and can be used individually to define the yaw behaviour of the vehicle. However the approximation of these types of manoeuvring is a very complicated task and thus approximation techniques with high degree of non-linearity have to be applied. Thus, it was decided that an approximation of each of the angle turning θ, defined in the trials, should be investigated individually considered as a local model. The ANFIS architecture was used to develop Neuro-Fuzzy models that describe these angles locally as discussed in Sections § 2.6 and 2.8.3. Note that as mentioned in Section § 2.8.3 these local models are defined as a function of time t and the angle θ. Thus these are variables that construct the two dimensional space in the resulting figures i.e., Figure 2.4 to Figure 2.8. In Appendix A Figure A.I to Figure A.36 shows that its local yaw rotation is used as the training data for the ANFIS technique. The local neuro-fuzzy models are also illustrated. Also in these figures, it can be seen that the accuracy of the approximation is very high, as the outputs of the local models follow the training data. Note that, this accuracy is achieved using a small number of iterations (less than 100) of the ANFIS algorithmic operation.

The next step of the modelling analysis was to define the relationship between the fuzzy inputs i.e. power and angles, and the extracted local models as discussed in Section § 2.8.4. The local models were grouped according to vehicles course changing i.e., from straight ahead to left or right, from left to right, from right to left (see Table 2.3 to Table 2.5). With this grouping definition only one set of local models is fired in each of the course changing manoeuvre. Thus, the computation time during the modelling identification is minimised.

When all the local models are available the construction of the global fuzzy model can be defined as in Section § 2.8.5. In this case the FSSS defines the degree of belonging of the local models and thus determining which of these models suits the particular operating conditions. The fuzzy sets (see Figure 2.9 and Figure 2.10) that construct this global model are used to shift smoothly the modelling trajectory from one local model to the other. It is important to note also that if the number of the experimental trials increases the number of local models and fuzzy sets increase. Thus, the (global) model can be more accurate as the local space of the models is smaller and thus the trajectories between them smoother.

The hybrid model method was implemented in MATLAB/SIMULINK. The program is available to the Computer Vision and Robotics Group[1] for further testing and improvement using new experimental trials and data. Some simulation results are presented in Figure 2.11.

Note finally, that the identification of GARBI's model could be achieved based on conventional procedures by analysing the model structure of the vehicle, the type of available sensors and the actuator dynamics (Indiveri, 1998). Nevertheless, these types of approaches for model identification problem were beyond the scope of this research. In this research, the combination of fuzzy and neuro-fuzzy approaches was under investigation specifically how they can deal with model identification problems for non-linear systems.

2.10 Summary

In this chapter a brief introduction to fuzzy and neuro-fuzzy techniques is presented, discussing some of their advantages and disadvantages. The linguistic, fuzzy relational and Takagi-Sugeno fuzzy models are defined. The latter is emphasised since it is utilised in this work and thus the structure properties of this type of model are discussed. A proposed fuzzy supervisory scheduling system is also presented. How the neuro-fuzzy modelling method can approximate local models is also outlined. A hybrid fuzzy and neuro-fuzzy method is also proposed to define the global model of a non-linear system. The method is constructed using six main steps and is implemented to model the yaw rotation of an underwater vehicle. The hydrodynamic forces and moments of the vehicle are presented. Furthermore, the design of the experiments and the actual results of the trials in a real environment that are used to extract information for the modelling are presented. The local models are then defined using the ANFIS optimisation technique and the global model is finally constructed. Simulation results of the model are presented to show

---

[1] Institute of Informatics and Applications Girona, SPAIN

the capabilities of the proposed HFNFM method. The simulation program is also used for a series of trials of avoidance object action. The data have been collected and used for modelling control strategies in this term. In the next chapter an approach is proposed to solve this problem based on fuzzy clustering methods.

2.11 References

Amat, J.,Monferre, A.,Batlle, J., and Eufi, X. 1999. GARBI: a Low-cost Underwater Vehicle. *Microprocessors and Microsystems,* **23** (2), pp. 61-67.

Babuska, R. 1998a. *Fuzzy Modelling for Control.* Boston: Kluwer Academic Publishers. 0792381548 (acid-free paper).

Babuska, R., and Verbruggen, H. B. 1997. Fuzzy Modelling and Model-Base Control for Non-linear Systems. *In:* M. Jamshidi,L. Zadeh,A. Titli, and S. Boverie ed. *Applications of Fuzzy logic towards High Machine Intelligence Quotient Systems.* Upper Saddle River, N.J: Prentice Hall PTR, pp. 49-74.

Chen, S. Cowan C., and Grant, P. 1991. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. **2** (2), pp. 302-309.

Harris, C. J.,Moore, C. G., and Brown, M. 1994. Intelligent Control Aspects of Fuzzy Logic and Neural Nets. *In:* Word Scientific Publishing Co. Pte. Ltd., pp. 37-134.

Haykin, S. 1994. *Neural Networks.* New York: Macmillan Maxwell Int.

Indiveri, G., 1998. *Modelling and Identification of Underwater Robotic Systems.* Ph.D. Thesis. Spain: University of Genova.

Jang, J-S. R. 1993. ANFIS: Adaptive - Network - Based Fuzzy Inference System. *IEEE Trans. on Systems, Man and Cybernetics,* **23** (3), pp. 665-685.

Jang, J-S. R., and Sun, C-T. 1995. Neuro-Fuzzy Modelling and Control. *Proc. IEEE,* **83**, pp. 378-405.

Jang, J. S. R.,Sun, C. T., and Mizutani, E. 1997. *Neuro-Fuzzy and Soft Computing.* Upper Saddle River, NJ: Prentce Hall. 0-13-261066-3.

Kosko, B. 1992. *Neural Networks and Fuzzy Systems, A Dynamic Systems Approach to Machine Intelligence.* Prentice Hall, Inc.

Mamdani, E. 1977. Application of Fuzzy Logic to Approximate Reasoning using Linguistic Systems. *Fuzzy Sets and Systems,* **26**, pp. 1182-1191.

Mendel, J. M. 1995. Fuzzy Logic Systems for Engineering: A tutorial. *Proceedings of the IEEE,* **83** (3), pp. 345-377.

Nauck, D, and Kruse, R. 1996. Designing Neuro-Fuzzy Systems Trough Backpropagation. *In:* W. PEDRYCZ ed. *Fuzzy Modelling Paradigms and Practice.* Kluwer Academic Publishers, pp. 203-228.

Palman, R.,Driankov, D., and Hellndroom, H. 1997. *Model Based Fuzzy Control.* Germany: Springer-Verlag Berlin Heidelberg. 3-540-61471-0.

Patterson, D. W. 1996. *Artificial Neural Networks Theory and Applications.* Prentice Hall. 0-13-295353-6.

Pedrycz, W. 1984. An Identification Algorithm in Fuzzy Relational Systems. *Fuzzy Sets and Systems,* **13**, pp. 153-167.

Pedrycz, W. 1994. Why Triangular membership Functions? *Fuzzy Sets and Systems,* **64**, pp. 21-30.

Sugeno, M., and Kang, G. T. 1988. Structure Identification of Fuzzy Model. *Fuzzy Sets and Systems,* **28**, pp. 15-33.

Takagi, T., and Sugeno, M. 1985. Fuzzy Identification of Systems and its Applications to Modelling and Control. *IEEE Trans. on Systems, Man and Cybernetics,* **SMC-15**, pp. 116-132.

Tanaka, K., and Sugeno, M. 1990. *Stability of Analysis of Fuzzy Systems using Lyapunov's Direct Method and Construction Procedure for Lyapunov Functions.* Proc. 6th Fuzzy Systems Symposium. pp. 353-356, Tokyo.

Wang, L. X. 1992. *Fuzzy Systems are Universal Approximators.* Proc. IEEE Int. Conf. on Fuzzy Systems. San Diego.

Yager, R. R., and Filev, D. P. 1994b. *Essentials of Fuzzy Modelling and Control.* New York: John Wiley. 0471017612 (acid-free).

Yi, S., and Chung, M. 1993. Identification of Fuzzy Relational Model and its Application to Control. *Fuzzy Sets and Systems,* **59**, pp. 25-23.

Zadeh, L. A. 1965. Fuzzy Sets. *Information and Control,* **8**, pp. 338-353.

Zadeh, L. A. 1973. Outline of a New Approach to the Analysis of Complex Systems and Desision Processes. *IEEE Transactions on Systems, Man and Cybernetics,* **SMC-3**, pp. 28-44.

Zeng, X., and Singh, M. 1994. Approximation Theory of Fuzzy Systems . *IEEE,* pp. 916-1921.

3

Fuzzy Clustering Methods as Techniques for Identifying and Modelling of Control Strategies

3.1 Introduction

Identification and modelling of a complex non-linear system's control action can be viewed as determining a description between the input and the output of the controller under consideration. The task becomes much harder when there is only little *a priori* knowledge on the type of non-linearities present in the system. In this case, fuzzy reasoning provides an effective formalism through which large classes of non-linear systems can be represented without a precise mathematical description of the non-linear elements.

As been discussed in chapter one, fuzzy control is an important topic in fuzzy set theory and applications. A fuzzy (behaviour) controller can be considered as a fuzzy model system based on the operator's control experience and knowledge for a particular system. The expert's control skills, however, are difficult to describe in words since the operators' control strategy is based on various control principles simultaneously, combining feedforward, feedback and predictive strategies in a complex, time-varying fashion, and because the operator may not be able to explain why a particular control action is chosen. Experience from knowledge acquisition also shows that the rules provided by different operators are often contradictory.

One way to solve these types of problems is based on the assumption that expert information is readily available. This information actually comes from experimental or simulation data resulting from the expert's control action according to the particular applied inputs. Thus, estimation of the parameters that define the antecedent and consequent templates is achieved as well as the number of *If-Then* rules that constructs the knowledge base. The premise and consequence, indicates *structure identification* and *parameter estimation.* The collection of rules for fuzzy sets comes from an input-output mapping of the operators' control behaviour data. These tasks cannot be performed separately. Usually, the case where the input variables have been selected *a priori* can be considered so that what remains to be done is (Sugeno and Yasukawa, 1993):

- to determine the number of fuzzy rules and the partition of the input/output space into fuzzy sets (structure identification);
- to estimate the best values for the parameters of the membership functions (parameter estimation).

The above can be achieved using fuzzy clustering methods, where the prototypes of the clusters are usually the centres of hyper-planes. Their numbers define the number of the fuzzy sets and their projections, into n-dimensional input-output data space axes, determines the parameters of the extracted membership functions together with the partition of the input/output space.

In this work an algorithmic methodology is proposed to construct fuzzy control strategies based on the choice of different fuzzy clustering algorithms to define the number and actual position of the prototypes and the variance of the clusters. Then the generation of the fuzzy sets is obtained by using projecting methods that are varied according to the chosen type of fuzzy system. Moreover, merging of membership functions methods are used to reduce the number of fuzzy sets and guidelines of when to do so are also proposed. Finally, improvements and small modification to the fuzzy system are introduced using techniques such as gradient method.

This chapter is organised as follows: Sections § 3.2 to 3.4 introduce the basic theory of cluster analysis. Sections § 3.5 to § 3.6 discusses some of the most commonly used fuzzy cluster algorithms that are applied in identification problems based on input/output mapping data discussing their advantages and disadvantages. Section § 3.7 shows techniques to determine the number of clusters. The problem of how and when the normalisation of data should be applied is discussed in Section § 3.8. The projection of cluster centres and variances is introduced in Section § 3.9. In Section § 3.10, methods that are used for merging the membership functions are reviewed as well as some guidelines of when to use them are proposed. A proposed methodology to construct fuzzy control strategies generating the fuzzy rules from n-dimensional input/output clustering data is presented in Section § 3.11. Section § 3.12 discusses the proposed algorithm. A brief summary of this chapter is finally presented in Section § 3.13.

3.2 Cluster analysis

Cluster analysis is a technique that is used to seek out natural clusters corresponding to natural classes in the data, dividing all objects (samples) into smaller subgroups, classifying them according to the similarities among them. It therefore, reconstructs the probability of data density from the samples and thus extracts the information that the data are carrying by extrapolating class membership to unlabeled samples or simply to better understand the system and/or process from which the data arises.

Clustering analysis comprises of three main problems (Bezdek *et al*, 1999):

1. Investigation of a tendency of clustering data or in other words the substructure(s) in the data.
2. Choice of clustering method, which measures the mathematical similarity that captures a data structure in the sense that a human might perceive it.
3. Definition of the cluster's validity.

Clustering techniques belong to the class of *unsupervised* (learning) methods, since they do not use prior class identifiers. Most clustering algorithms also do not rely on assumptions common to conventional statistical methods, such as the underlying statistical distribution of data, and therefore they are useful in situations where little prior knowledge exists.

3.3 Definitions and Notations in Cluster Analysis

Before identifying the cluster methods that can be used in identification problems and control strategies, some definitions should be denoted.

3.3.1 The data used in clustering analysis

One of the important advantages of clustering techniques is that they can be applied to data that is *quantitative* (numerical), *qualitative* (categoric), or a *mixture* of both. In this work, the clustering of quantitative data is considered. The data are typically observations and/or records of some physical process of a real system and/or control action. Each observation and/or record consists of n measured variables, grouped into an n-dimensional Euclidean space $\Re^n$ column vector $\mathbf{z}_k = [z_{1k}, z_{2k}, ..., z_{nk},]^T$, $z_k \in \Re^n$. A set of N observations and/or records is denoted by $Z = \{\mathbf{z}_k \mid k = 1,2,...,N\}$, and is represented as an $n \times N$ matrix:

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1N} \\ z_{21} & z_{22} & \cdots & z_{2N} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ z_{n1} & z_{n2} & \cdots & z_{nN} \end{bmatrix} \tag{3.1}$$

In pattern recognition terminology, the *columns* of the matrix are called *patterns* or *objects*, the *rows* are called the *features* or *attributes*, and Z is called the *pattern* or *data matrix*. The meaning of the *columns* and *rows* of Z depends on the context of the classification problem.

3.3.2 Definition of clusters

Various definitions of a cluster can be formulated, depending of the objective of clustering. Bezdek, (1981) defined a cluster as a group of homogeneous classes or objects that are more similar to one another than to members of other clusters. The term "similarity measure" has an important effect on the clustering results since it indicates which mathematical properties of the data set, i.e. distance, connectivity, and intensity, should be used and in what way in order to identify the clusters. Distance can be measured among the data vectors themselves, or as a distance from a data vector to some prototypical object of the cluster. The prototypes (which are usually the centres of the clusters) or centroids are usually not known beforehand, and are sought by the clustering algorithms simultaneously with the partitioning of the data. The prototypes may be vectors of the same dimension as the data objects, but they can also be defined as "high level" geometrical objects, such as linear or non-linear subspaces or functions (Babuska, 1998a).

3.4 Fuzzy clustering

The objective of clustering methods is to perform a partition of the collection of elements in Equation 3.1, into c data sets with respect to a given criterion, where c is defined (given) as the number of clusters. The criterion is usually to optimise an *objective function* that acts as a performance index of clustering.

In classical non-fuzzy "hard" clustering analysis, the data are distributed into partitions such that the degree of their association is strong within blocks of the partition and weak in different blocks. In other words, each data sample is assigned to only one cluster and all clusters are regarded as a disjoint gathering of the data set. In practice however, there are many cases in which the clusters are not completely disjointed and data could be classified as belonging to one cluster almost as well as to another. A crisp classification process cannot cater for such situations, therefore, the separation of the clusters becomes a fuzzy notion, and the representations of real data structures can then be more accurately handled by fuzzy clustering methods. In these cases, it is necessary to describe the data structure in terms of fuzzy clusters. Thus, the end result of the fuzzy clustering is the *partition matrix* U as in Equation 3.2.

$$U = [\mu_{ik}]_{i=1...c,\ \ k=1...N} \tag{3.2}$$

where μ_{ik} is a numerical value in [0,1] i.e. $\mu_{ik} \in \{0,1\}$, and expresses the degree to which the element $\mathbf{z}_k$ belongs to the i cluster. However, there are two additional constraints on the value of μ_{ik}. First, a total membership of the element $\mathbf{z}_k \in Z$ in all classes is equal to 1.0; that is,

$$\sum_{i=1}^{c} \mu_{ik} = 1 \qquad \text{for all } k = 1,2,...,N \tag{3.3}$$

Second, every constructed cluster is non-empty and different from the entire set; that is,

$$0 < \sum_{k=1}^{N} \mu_{ik} < N \qquad \text{for all } i = 1,2,\ldots,c \tag{3.4}$$

A general form of the objective function is

$$J(\mu_{ik}, \mathbf{v}_i) = \sum_{i=1}^{c} \sum_{k=1}^{N} \sum_{i=1}^{c} g[w(z_i), \mu_{ik}] \cdot d(\mathbf{z}_k, \mathbf{v}_i) \tag{3.5}$$

where $w(z_i)$ is the *a priori* weight for each $\mathbf{z}_k$ and $d(\mathbf{z}_k, \mathbf{v}_i)$ is the degree of dissimilarity between the data $\mathbf{z}_k$ and the supplemental element $\mathbf{v}_i$, which can be considered the central vector of the ith cluster. The degree of dissimilarity is defined as a measure that satisfies two axioms:

$$d(\mathbf{z}_k, \mathbf{v}_i) \geq 0, \qquad d(\mathbf{z}_k, \mathbf{v}_i) = d(\mathbf{v}_i, \mathbf{z}_k) \tag{3.6}$$

and thus it is a weaker concept than distance measures.

With the above settings, fuzzy clustering can be precisely formulated as an optimisation problem:

$$\text{Minimise} \quad J(\mu_{ik}, \mathbf{v}_i), \qquad i = 1,2,\ldots,c\,;\; k = 1,2,\ldots,N \tag{3.7}$$

Figure 3.1 illustrates these types of clusters, with cluster centres $\mathrm{V} = [\mathrm{v}_1, \mathrm{v}_2]$ and *partition matrix* U as in Equation 3.8.

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \end{bmatrix} \tag{3.8}$$

In the literature numerous clustering approaches have been developed on iterative minimisation of the criterion function (Equation 3.7). In this Chapter some of these approaches are introduced in terms of their significance for the identification and modelling control strategies problem

(Babuska, 1998a) . One of the widely used clustering methods based on Equation 3.7 is the fuzzy c-means (FCM) algorithm developed by Bezdek, (1981) and is presented in the following Section § 3.5.

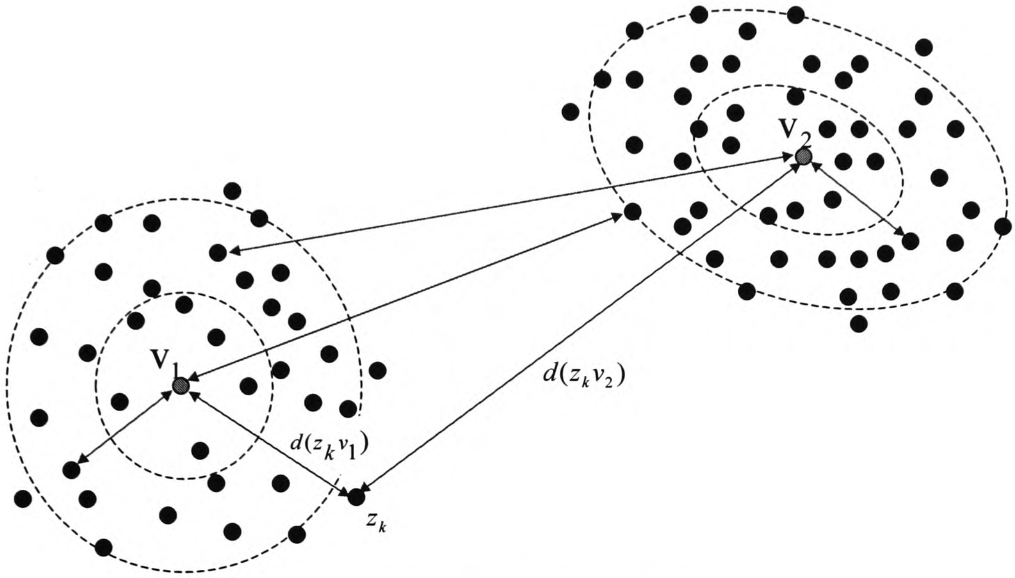


Figure 3.1 Fuzzy clustering representation

3.5 Fuzzy c-means clustering approach

The objective (or cost) function of the FCM algorithm takes the form of

$$J(\mu_{ik},\mathbf{v}_i)=\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m}\cdot\left\|\mathbf{z}_k,\mathbf{v}_i\right\|^2 \qquad m>1 \tag{3.9}$$

where m is called the *exponential weight* which determines the fuzziness of the resulting cluster. To solve this minimisation problem, the objective function in Equation 3.9 is differentiated with respect to $\mathbf{v}_i$ (for fixed μ_{ik}, $i=1,2,...,c$; $k=1,2,...,N$) and to μ_{ik} (for fixed $\mathbf{v}_i$, $i=1,2,...,c$) and the condition of Equation 3.3 is applied

$$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^{N} (\mu_{ik}^{l-1})^m \cdot \mathbf{z}_k}{\sum_{k=1}^{N} (\mu_{ik}^{l-1})^m}, \qquad i = 1,2,...,c \tag{3.10}$$

$$\mu_{ik}^{l} = \frac{1}{\sum_{j=1}^{c} (D_{ik\mathrm{A}} / D_{jk\mathrm{A}})^{2/(m-1)}}, \qquad i = 1,2,...,c \quad k = 1,2,...,N \tag{3.11}$$

where l is the number of repeats and

$$D_{ik\mathrm{A}}^2 = \|\mathbf{z}_k - \mathbf{v}_i\|_{\mathrm{A}}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{z}_k - \mathbf{v}_i) \tag{3.12}$$

is a square inner product distance norm.

The algorithm of the FCM approach is simply an iteration through the preceding three steps which are summarised as follows:

Algorithm FCM:

Select the data set Z, allocate a number of clusters c ($2 \le c \le N$), the exponential weight ($1 < m < \infty$), the termination tolerance $\varepsilon > 0$ and the norm-inducting matrix $\mathbf{A}$. Choose an initial partition matrix $\mathbf{U}^{(0)}$.

Repeat for $l = 1,2,...$

Step 1: Compute the fuzzy cluster prototypes -centres- (means) $\{\mathbf{v}_i^{(l)} \mid i = 1,2,...,c\}$ by using $\mathbf{U}^l$ and Equation 3.10.

Step 2: Compute the distance using Equation 3.12.

Step 3: Update the partition matrix $\mathbf{U}^{l+1}$ by using $\{\mathbf{v}_i^{(l)} \mid i = 1,2,...,c\}$ and Equation 3.11.

until $\|\mathbf{U}^l - \mathbf{U}^{l+1}\| < \varepsilon$

The hard c-means (HCM) clustering algorithms can be considered as a special case of the fuzzy c-means clustering algorithms. In Equation 3.9, if μ_{ik} is 1 for only one class and zero for all other classes, then the criterion function $J(\mu_{ik}, \mathbf{v}_i)$ used in the FCM is the same as in the hard c-means clustering algorithm. The use of membership values in the FCM method provides more flexibility and formulates the clustering results in a more useful form in practical applications. However, both algorithms are iterative, and therefore there are no guarantees that they will converge to an optimum solution. The performance depends on the selection of the *initial positions of the cluster centres,* thereby another fast algorithm can be use to determine the initial clusters or to run HCM/FCM algorithms several times, each starting with a different set of initial cluster centres. Moreover, an important parameter that has also to be selected in applying these algorithms is *the number of clusters*. The number of clusters should ideally correspond to the number of sub-structures naturally present in the data.

3.5.1 Inner-product Norms

The shape of the clusters is determined by the choice of the matrix $\mathbf{A}$ in the distance measured (Equation 3.12). One choice of $\mathbf{A}$ is $\mathbf{A} = \mathbf{I}$. This induces the standard Euclidean norm:

$$D_{ik}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T (\mathbf{z}_k - \mathbf{v}_i) \tag{3.13}$$

which actually stimulates hyperspherical clusters, i.e., clusters whose surfaces of constant membership are hyperspheres.

A $n \times n$ diagonal matrix that accounts for different variances in the directions of the co-ordinate axes of Z can also be as another choice of $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} (1/\sigma_1)^2 & 0 & \dots & 0 \\ 0 & (1/\sigma_2)^2 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & (1/\sigma_n)^2 \end{bmatrix} \tag{3.14}$$

This matrix induces a diagonal norm on $\Re^n$.

Finally, $\mathbf{A}$ can be defined as the inverse of the $n \times n$ sample covariance matrix of Z: $\mathbf{A} = \mathbf{R}^{-1}$, where:

$$\mathbf{R} = \frac{1}{N}\sum_{\mathbf{k}=1}^{\mathbf{N}} (\mathbf{z}_\mathbf{k} - \overline{\mathbf{z}})(\mathbf{z}_\mathbf{k} - \overline{\mathbf{z}})^T \tag{3.15}$$

Here $\overline{\mathbf{z}}$ denotes the sample mean of the data. In this case, $\mathbf{A}$ induces the Mahalanobis norm on $\Re^n$ (Bezdek, 1981).

Both the diagonal and Mahalanobis norms generate hyperellipsoidal clusters, the only difference is that with the diagonal norm, the axes of the hyperellipsoids are parallel to the co-ordinate axes while with the Mahalanobis norm the orientation of the hyperellipsoid is arbitrary.

The choice of which norm is the most appropriate depends on the data themselves. Euclidean distance is the overwhelming favourite as it is considered to be closer to what human observations are like. Mahalanobis distance is useful when there are large disparities in the ranges of the measured features because it rotates the basis $\Re^n$ so that the data are scaled equally and are pair-wise uncorrelated.

A common limitation of cluster algorithms based on a fixed distance norm is that such a norm induces a fixed topological structure on $\Re^n$ and forces the objective function to prefer clusters of that shape even if they are not present. However, a method using the norm-inducing matrix

A can be adapted to the local topological structure of the data. This method can be used to estimate the dependence of the data in each cluster as presented in the next Section.

3.6 Extensions of the Fuzzy *c*-means algorithm using Fuzzy Covariance Matrix

Several algorithms can be derived from the basic FCM scheme by adapting the inner-product norm (Equation 3.12). The most used and successful clustering algorithms are the Gustafson and Kessel (G-K) proposed by Gustafson and Kessel, (1979) and Fuzzy Maximum Likelihood Estimates (FMLE) proposed by Bezdek and Dunn, (1975). Both algorithms recognise the fact that different clusters in the same data set Z may have different geometrical shapes and in order to detect them, the standard fuzzy c-mean algorithm is extended, by employing an adaptive distance norm. The analytical description of these algorithms is outside the scope of this thesis, however, some important aspects should be noted:

The fuzzy covariance matrix defined for the i_{th} cluster is for the GK method as in Equation 3.16 denoted as $\mathbf{F}_i$, whereas for the FMLE method as in Equation 3.17 denoted as $\mathbf{\Sigma}_i$.

$$\mathbf{F}_i = \frac{\sum_{k=1}^{\mathrm{N}} (\mu_{ik})^m (\mathbf{z}_k - \mathbf{v}_i)(\mathbf{z}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{\mathrm{N}} (\mu_{ik})^m} \tag{3.16}$$

$$\mathbf{\Sigma}_i = \frac{\sum_{k=1}^{\mathrm{N}} \mu_{ik} (\mathbf{z}_k - \mathbf{v}_i)(\mathbf{z}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{\mathrm{N}} \mu_{ik}} \tag{3.17}$$

The difference between the fuzzy covariance matrixes $\mathbf{F}_i$ and $\mathbf{\Sigma}_i$ is that the latter does not include the weighting exponent m. This is simply because the two weighted covariance matrices arise as generalisations of the classical covariance from two different concepts. Note that the choice of *weighting exponent* m is as in the case of FCM algorithm i.e. $1 < m < \infty$.

Each cluster has its own inner product distance norm. The first method is as in Equation 3.18 and for the second as in Equation 3.19.

$$D_{ik\mathbf{A}_i}^2 = (\mathbf{z}_k - \mathbf{v}_i)^T \left[\rho_i \det(\mathbf{F}_i)^{1/n} \mathbf{F}_i^{-1}\right](\mathbf{z}_k - \mathbf{v}_i) \tag{3.18}$$

$$D_{ik\mathbf{A}_i}^2 = \frac{\left[\det \Sigma_i\right]^{1/2}}{P_i} \exp\left[\frac{1}{2}(\mathbf{z}_k - \mathbf{v}_i)^T \Sigma_i^{-1} (\mathbf{z}_k - \mathbf{v}_i)\right] \tag{3.19}$$

In Equation 3.19 P_i is the prior probability of selecting cluster i where the membership degrees μ_{ik} are interpreted as the posterior probability as described in Equation 3.20. In Equation 3.18 the *cluster volumes* ρ_i are simply fixed at 1 for each cluster if no *prior knowledge* is available.

$$P_i = \frac{1}{N}\sum_{k=1}^{N} \mu_{ik} \tag{3.20}$$

Note that Equation 3.19 involves an exponential term and thus $D_{ik\mathbf{A}_i}^2$ decreases faster for a given change in distance measure than the inner product norm in Equation 3.18. Therefore the clusters are not constrained in volume as may happen when using the GK algorithm. Thus, the algorithm is able to detect clusters of varying shapes, sizes and distances. However, FMLE needs good initialisations i.e., close to the optimal, as due to the exponential distance norm, it tends to converge to a nearby local optimum.

3.6.1 Definition of the hyperellipsoids from the covariance matrix

Using Covariance Matrix algorithms, each cluster can be approximated as hyperellipsoid. The eigenstructure of the cluster covariance matrix provides information about the shape and

3.6.1 Definition of the hyperellipsoids from the covariance matrix

Using Covariance Matrix algorithms, each cluster can be approximated as hyperellipsoid. The eigenstructure of the cluster covariance matrix provides information about the shape and orientation of the clusters in the *n*-dimensional input-output state space (Strang, 1980). Therefore, the directions ϕ_i of the axes are given by the eigenvectors of $\mathbf{F}_i$ and $\mathbf{\Sigma}_i$. Moreover, the ratio of the square roots of the eigenvalues of $\mathbf{F}_i$ and $\mathbf{\Sigma}_i$ defines the ratio of the l lengths of the cluster's hyperellipsoid axes $L_l = \alpha\sqrt{\lambda_{in}}$, $(l \leq n)$. For instance Figure 3.2 illustrates a covariance hyperellipsoid defined from the locus of all $\mathbf{z}$ that satisfy Equation 3.21.

$$(\mathbf{z}-\mathbf{v}_i)^T \mathbf{F}_i^{-1}(\mathbf{z}-\mathbf{v}_i) = \alpha^2 \tag{3.21}$$

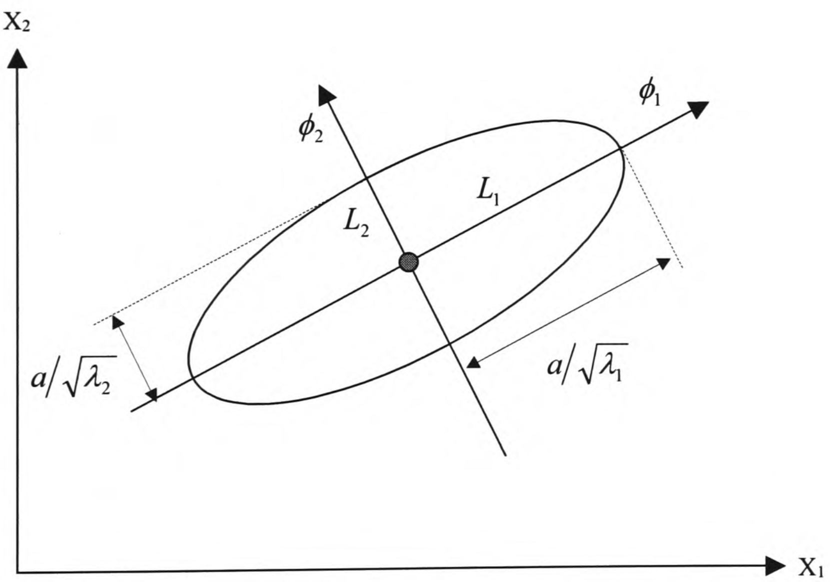


Figure 3.2 The lengths $\sqrt{\lambda_1}$, $\sqrt{\lambda_2}$ and the directions ϕ_1, ϕ_2 for the each axes of a hyperellipsoid

3.7 Determination of the number of clusters

The main disadvantage of the algorithms described in Sections § 3.5 and § 3.6 is that the number c of clusters has to be known in advance. In many applications, this knowledge is not available. However, the determination of the number of "natural" groups in the data is important for the successful application of fuzzy clustering methods. A number of methods have been proposed to determine the relevant number of clusters in the clustering problem.

External cluster *validity measures* is one of the techniques that are used to determine the goodness and therefore the number of clusters (Xie and Beni, 1991). These methods assess the validity of a given partition with a specific number of clusters considering criteria like the *compactness* of the clusters and the *distance* between the clusters. A computational drawback of cluster validity methods is the need for reparative clustering of the data using different number of clusters each time.

Another approach, proposed by Krishnapuram and Freg, (1992), is called *Compatible Clustering Merging*. The method determines the number of clusters based on starting with a larger number of clusters than required, and then iterative merging of similar (compatible) clusters until some threshold is reached and no more clusters can be merged (Kaymak and Babuska, 1995). The key elements of this method are the criteria which measure the degree of compatibility between clusters. This degree is actually determined on the basis of the geometrical properties of the clusters, by analysing the eigenvalues and the unit eigenvectors of the cluster covariance matrix. This method offers a more automated and computationally less expensive way of determining the right partition by applying essentially a similarity measure between clusters. However, the method realistically can be applied only in two or maybe three-dimensional cluster space data, as it is hard to encounter geometrically similar clusters in a high dimensional data space.

A method called *Mountain clustering* is used in the work described here. The method defines the number of clusters together with their relatively significance based on validity measures (density) of clusters. The prototypes are defined from the peaks of the mountains resulting from the mountain function. An analytical description of the method is discussed in the following Section § 3.7.1.

3.7.1 Mountain clustering meth od

The mountain clustering method, proposed by Yager and Filev, (1994a); Yager and Filev, (1994b), is a relatively simple and effective approach to approximate estimation of cluster centres on the basis of *density* measure called the *mountain function*. The method is divided into four steps:

- The first step involves forming a grid on the data space, where the intersections (nodes) of the grid lines constitute the candidates for cluster centres, denoted as a set V .
- The second step uses the observed data to construct the mountain function. The *height* h of the *mountain function* at a node $\mathbf{v} \in V$ is as in Equation 3.22.

$$h_1(\mathbf{v}) = \sum_{i=1}^{N} \exp\left(-\frac{\sum_{k}^{n} \left\| \mathbf{v}_k - \mathbf{z}_{ki} \right\|^2}{2\sigma^2} \right) \tag{3.22}$$

where $\mathbf{z}_{ki}$ is the ith data point in kth dimensional data space and σ is an application-specific constant and determines the height as well as the smoothness of the resultant mountain function.

The closer a data point is to the node the more it contributes to the score at the node. The higher the mountain function value at a node, the higher is its potential to be a cluster centre. Therefore, the value of the mountain function is related to the potential ability of a grid point to be a cluster centre and it can be used as an indicator of the clusters.

- The third step is to use the mountain function to define the cluster centres. The first centre c_1 is the node, of the candidate centres V, and has the greatest value for the mountain function i.e., $c = \max(m(v))$. Obtaining the next cluster centre requires eliminating the effect of the just-identified centre, which is typically surrounded by a number of nodes that also have high-density scores. This can be done by revising the mountain function as in Equation 3.23.

$$h_q(\mathbf{v}) = h_{q-1}(\mathbf{v}) - h(c_q)\exp\left(-\frac{\sum_k^n \left\|\mathbf{v}_k - c_q\right\|^2}{2\beta^2}\right), \qquad q > 1 \qquad (3.23)$$

where q is the number of the cluster and β is a positive constant similar to the parameter σ.

The subtracted exponential part of Equation 3.23 is a Gaussian function inversely proportional to the distance between $\mathbf{v}$ and c_q, as well as being proportional to the height $h(c_q)$ at the centre. Note that after subtraction, the new mountain function $h_q(\mathbf{v})$ reduces to zero at $\mathbf{v} = c_1$.

After subtraction, the q cluster centre is again selected as the node in V that has the largest value for the new mountain function. This process of revising the mountain function and finding the next cluster continues until a sufficient number of cluster centres is attained.

The main advantage of the mountain function method is that it does not require a predefined number of clusters. It determines the *number* and the *approximate position* of the first c cluster centres that satisfy the stopping rule (the height of the mountain), starting from the highest ones, which is characterised with a maximal value of the mountain function at nodes V. It can therefore be used to obtain initial cluster centres that are required by more sophisticated and accurate cluster algorithms such as FCM, G-K, FMLE introduced in Sections §3.5 and § 3.6. It can also be used as a measurement method to define the importance of the subspace clusters. Moreover, the number of the most significant mountains can be used in decisions related to the cluster centres and/or membership functions merging as is discussed in Section § 3.10 and therefore to define the appropriate number of the generated rules.

Remarks

- A finer gridding increases the number of potential clustering centres, but it also increases the computations required. The gridding is generally evenly spaced, but it is not a requirement. If *a priori* knowledge of data distribution is available, an unevenly spaced gridding can be used.

- It is evident from the construction of the mountain function that its values are approximations of the *density* of the data points in the vicinity of each node.

- The method is actually based on what a human does in visual forming clusters of data set. High density in the subspace of a control action means high importance.

- The parameters σ and β are chosen by the operator on a trial-and-error basis. However, when the method is applied to high dimensional data space this may become a difficult task for the operator and as a result some approaches have been introduced to set these parameters (Yager and Filev, 1994a); (Lori and Costa Branco, 1995).

3.8 Data Normalisation

In the literature on pattern recognition, it is often suggested that the data should be appropriately normalised before clustering (Jain and Dubes, 1988). Thus, considering Equation 3.1, the *k*th pattern is denoted by the column vector and the *j*th feature value for the *k*th pattern is denoted by z_{jk}. The simplest type of normalisation is the subtraction of the feature average $\bar{z}_j$ as in Equation 3.24. This normalisation makes the feature values invariant to rigid displacements of the co-ordinates. The asterisk denotes the "raw" or unnormalised (unscaled) data. Another type of normalisation which is the most used, translates and scales the axes so that all the features have zero mean and unit variance as described in Equation 3.25.

$$z_{jk} = z^*_{jk} - \bar{z}_j \tag{3.24}$$

$$z_{jk} = \frac{z^*_{jk} - \bar{z}_j}{\sigma_j} \tag{3.25}$$

The *j*th feature average, $\bar{z}_j$, and the *j*th feature variance, σ_j^2, are defined as the sample mean and the sample variance for the *j*th feature (Equation 3.26 and Equation 3.27).

$$\bar{z}_j = (1/n)\sum_{i=1}^{n} z^*_{jk} \tag{3.26}$$

$$\sigma_j^2 = (1/n)\sum_{i=1}^{n} (z^*_{jk} - \bar{z}_j)^2 \tag{3.27}$$

Normalisation, however, is not always desirable, as it may influence the result of clustering when the separation between clusters is altered. Distance norms are sensitive to variations in the numerical ranges of the different features. For instance, the Euclidean distance assigns more weighting to features with wide ranges than to those with narrow ranges. The result of clustering can thus be negatively influenced by, for instance, choosing different measurement units. Nevertheless, clustering algorithms that are based on adaptive distance measure are less sensitive to data scaling, since the adaptation of the distance measure automatically compensates for the distance in scale.

3.9 Projection of cluster centres and variances

The clusters that are defined using the methods discussed in the previous sections may incorporate linguistic meanings for the input/output relationships of a system. These clusters can actually be described as membership functions that are often assigned by linguistic labels and thus gives a transparency into a system's analysis, i.e. easy to read and interpreted by humans. In one-dimensional domains labels such as "low", "medium", "high" etc are frequently used. However, it is often difficult to specify meaningful labels for membership functions with higher n-dimensional domains. The projection, therefore, of these memberships onto the n measurement variable axis can produce membership functions with corresponding linguistic terms. There are different types of projection that mainly depends on which clustering method is being used to identify the cluster centres and variances. The most used projecting methods recommended in literature are described in the following:

1. If only the c cluster centres are considered, i.e. using FCM method (Section § 3.5), their projections on each dimension n define the peak values p_i^j $\{i=1,...,c, j=1,...,n\}$ of the projected membership functions. Moreover, sorting these values on each dimension as in Equation 3.28, the membership functions such as triangular are constructed with $1/2$ overlap using Equation 3.29.

$$p_i^j \le p_{i+1}^j \qquad \forall j \tag{3.28}$$

$$\mu_i^j(v^j) = \max\left[0, \min\left(\frac{v^j - p_{i-1}^j}{p_i^j - p_{i-1}^j}, \frac{v^j - p_{i+1}^j}{p_i^j - p_{i+1}^j}\right)\right] \tag{3.29}$$

2. In cases where approximation of clusters are hyperellipsoids, as described in Section § 3.6.1, their projections onto each domain axis will produce symmetric membership functions with the peak point p_i^j $\{i = 1,...,c, j = 1,...,n\}$ being the corresponding component of each cluster centre. The left and right fraction of the membership functions defined from the eigenstructure of the cluster covariance matrix for each hyperellipsoid is as in Equation 3.30 and Equation 3.31.

$$\delta_{1right} = \delta_{1left} = L_1 |\cos\phi_1| + L_2 |\cos\phi_2| \tag{3.30}$$

$$\delta_{2high} = \delta_{2low} = L_1 \left|\cos(\frac{\pi}{2} - \phi_1)\right| + L_2 \left|\cos(\frac{\pi}{2} - \phi_2)\right| \tag{3.31}$$

Figure 3.3 illustrates schematically the projection of an ellipsoid on axes X_1 and X_2.

Note that in both cases it is recommended to add two modal values into the extremes of each universe of discourse to ensure full coverage of the input space. Thus,

$$p_0^j = \min_{k=1,...,N} u_k^j \tag{3.32}$$

$$p_{c+1}^j = \max_{k=1,...,N} u_k^j \tag{3.33}$$

where u_k^j is the kth cluster data in jth dimension space and the trapezoidal membership function at the extremes of each universe of discourse is as in Equation 3.34 and Equation 3.35.

$$\mu_0^j(v^j) = \max\left[0, \min\left(\frac{v^j - p_1^j}{p_0^j - p_1^j}, 1\right)\right] \tag{3.34}$$

$$\mu_{c+1}^j(v^j) = \max\left[0, \min\left(\frac{v^j - p_c^j}{p_{c+1}^j - p_c^j}, 1\right)\right] \tag{3.35}$$

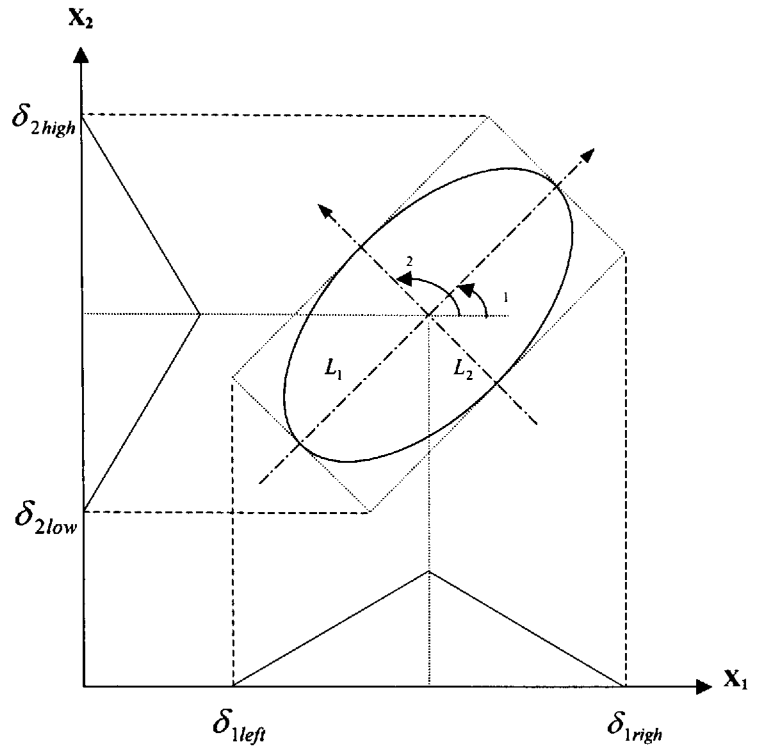


Figure 3.3 Schematic demonstration of projection of an ellipsoid on axes X_1 and X_2

3.10 Merging Membership Functions Method

By reducing the number of projected membership functions in the input/output domain, the number of fuzzy rules the are produced is reduced. One way to do this is to merge the neighbouring fuzzy sets with regards to the similarity between them. In this context, similarity between fuzzy sets is defined as the degree to which the fuzzy set are *equal* in terms of distinguishability or in terms of compatibility (Satnes *et al*, 1998). Therefore, the similarity measures for fuzzy sets are divided into two main groups (Zwick *et al*, 1987):

i) *geometric similarity measures* that are suited for measuring similarity (or dissimilarity) among distinct fuzzy sets,

ii) *set-theoretic similarity measures* that are most suitable for capturing similarity among overlapping fuzzy sets (a cross-point level) that describe almost the same region in the input/output domain variable.

The geometric similarity measures represent similarity as proximity of fuzzy set and not as a measure of equality. The interpretation of similarity as "approximate equality" can be better represented by the set-theoretic similarity measures based on operations such as union and intersection. They also have the advantage above geometrical measures that scaling and ordering of domain does not influence them.

Several merging methods have been proposed in the literature using set-theoretic similarity measures preserving, however, the *distinguishability* and the *justifiable number of elements* on each input domain to guarantee the semantic integrity. Chi *et al*, (1996) merged triangular neighbouring membership functions that resulted in trapezoidal. Espinosa and Vandewalle, (1999) use a merging algorithm called FuZion that considers the closeness between the peaks of the membership functions.

In this work it is proposed that the merging between neighbour membership functions should be achieved according to:

a) *their closeness,*

b) *the number of clusters that results from the "highest" mountain functions and*

c) *the linguistic interpretation of the generated fuzzy sets.*

Close membership functions can be considered as those that have high degree of overlapping γ. Thus, if $\gamma > \rho$ then merge the modal values $m_{c_{n-1}}$ and m_{c_n} (see Appendix C). where ρ is set by the designer and is $1 > \rho > 0.5$. Figure 3.4 illustrates a merging between membership function with high overlapping. The number of final fuzzy sets (after merging) should not be more than the number of the proposed cluster centres measured by using the height h of the mountain function (Equation 3.22, Equation 3.23) as mentioned in Section § 3.7.1. Finally, a group of fuzzy sets may imply the same linguistic interpretation and their merging is made according to this observation.

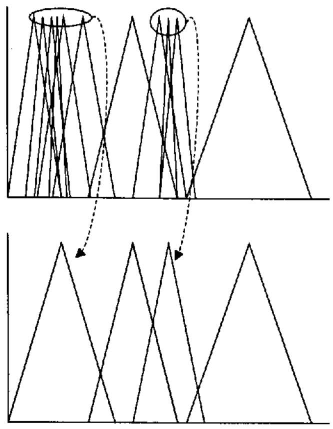


Figure 3.4 Membership functions before and after merging

3.11 Generate fuzzy rules from n-dimensional input/output clustering data

In this section a systematic methodology is proposed to identify a control strategy using data (experimental or not) resulting from multidimensional input/output space. The method uses a numerical data to approximate linguistic fuzzy rules to investigate the interpretation between input/output relationships. The main steps of the proposed method are as follows:

1. Define the multidimensional input/output space with $\xi - input$ and $\psi - output$ variables. Thus, for the inputs

$$U = \begin{bmatrix} u_1 \\ . \\ . \\ . \\ u_\xi \end{bmatrix} \tag{3.36}$$

where $U \in \Re^{\xi}$

For the outputs

$$Y = \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_\psi \end{bmatrix} \tag{3.37}$$

where $Y \in \Re^{\psi}$

2. Collect the number of points of the observations N for each input variable

$$u_k = [u_{k1}, u_{k2}, ..., u_{kN}],\ k = \{1, ..., \xi\} \tag{3.38}$$

and for each output variable

$$y_\lambda = [y_{\lambda 1}, y_{\lambda 2}, ..., y_{\lambda N}],\ \lambda = \{1, ..., \psi\} \tag{3.39}$$

For each output y_λ and all input variables, the patterns or data matrix $\mathbf{Z}$ is constructed as in Equation 3.40.

$$Z_\lambda = \begin{bmatrix} u_{11} & \cdots & u_{1N} \\ . & \cdots & . \\ . & \cdots & . \\ . & \cdots & . \\ u_{\xi 1} & \cdots & u_{\xi N} \\ y_{\lambda 1} & \cdots & y_{\lambda N} \end{bmatrix} \tag{3.40}$$

where $Z_\lambda \in \Re^{\xi+1}$

This matrix is actually a spatial representation of the data samples of the $\xi + 1$ dimensional space.

3. Clustering is the first step in the analysis of the data classification presented in Equation 3.40. There are no specific rules about which cluster analysis method(s) to be used. Different algorithms produce different partitions of the data, and it is difficult to decide

which is the most suitable. It mostly, depends on the *"quality"* and the *"quantity"* of the available data and what actually is the objective of the analysis. Table 3.1 presents the most used clustering algorithms highlighting some of their most important characteristics. Furthermore, a general guideline to which type of cluster algorithms and their analysis for identification and modelling of control strategies is proposed:

a) Using Mountain Clustering Method (Section § 3.7.1) the number of clusters and their approximate places of prototypes can be defined. These are two very important parameters that other more accurate and reliable cluster analysis algorithms need to start with. With this method the weight of each cluster is also defined and used in the cases of merging the cluster prototypes and/or membership functions as discussed in Section § 3.10.

b) FCM clustering algorithm is employed to define the prototypes of the clusters as introduced in Section § 3.5. However, using Fuzzy Covariance Matrix such as Gustafson-Kessel's and/or FMLE, the *centres* $\mathbf{V}c$ are detected together with the *variance* of clusters among different geometrical shapes as discussed in Section § 3.6. These *centres* and *variances* reflect the actual data distribution in the input/output space. The initialisation of the partition matrix and the number of clusters are defined from the resulting Mountain Clustering Method. Thus,

$$\mathbf{V}c = \begin{bmatrix} v_1^1 & v_2^1 & . & . & . & v_c^1 \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ v_1^{\xi+1} & v_2^{\xi+1} & . & . & . & v_c^{\xi+1} \end{bmatrix} \qquad (3.41)$$

where $\mathbf{V}c \in \Re^{\xi+1\times c}$

4. Projecting the *cluster centres* and *variances*, into the input/output spaces, the membership functions for each input are generated as discussed in Section § 3.9. The peak values p_i^j are equal to the cluster centres in the input spaces as in Equation 3.42.

$$p_i^j = v_i^j, \qquad i = 1,\dots,c \qquad j = 1,\dots,\xi \tag{3.42}$$

and in to the output space as in Equation 3.43

$$p_i^j = v_i^j, \qquad i = 1,\dots,c \qquad j = \xi + 1 \tag{3.43}$$

For the projections in the output dimensional spaces two different cases should be considered:

a) If the projections define membership functions, then their peaks are in Mamdani type fuzzy systems as in Equation 3.42, Equation 3.43 and in the antecedent part of the T-S fuzzy systems as in Equation 3.42.

b) If the projections define singleton values $\bar{s}_i^{\xi+1}$, as in Takagi-Sugeno (T-S) fuzzy systems, then their values are as in Equation 3.43 and therefore $\bar{s}_i^{\xi+1} = p_i^{\xi+1}$.

5. Merging neighbouring membership functions method is used to reduce the number of membership functions and therefore the number of rules as described in Section § 3.10.

6. The resulting membership functions and/or singletons in all dimensional spaces associated by linguistic labels (i.e. SMALL, MEDIUM, BIG, etc).

7. The rule base is constructed using all the possible combinations between the input membership functions for the antecedent part. This guarantees the completeness of the rules and full coverage of the working space. Thus, the rules R_r take the form:

$$R_r: \textbf{\textit{If}}\ u_{1l}\ \text{is}\ \mu_l^1\ \textbf{\textit{and}}\ u_{2l}\ \text{is}\ \mu_l^2\ \textbf{\textit{and} ... \textit{and}}\ u_{\xi l}\ \textbf{\textit{is}}\ \mu_l^{\xi}\ \textbf{\textit{then}}\ \hat{y}_r = \begin{cases} \overline{\mu}_l^{\xi+1} \\ s_l^{\xi+1} \end{cases}$$

where l is the observer point in the input/output variable ξ and $\xi+1$ respectively and r is the number of rules.

Note that μ have been defined in Equation 3.29, Equation 3.34 and Equation 3.35.

The evaluation of antecedents of each rule can be expressed in terms of the *min, max* and *product* operation as in Equation 3.44, Equation 3.45 and Equation 3.46 correspondingly.

$$\mu_l(u_l) = \min\{(\mu_l^1(u_{1l}), \mu_l^2(u_{2l}), ..., \mu_l^{\xi}(u_{\xi l})\} \tag{3.44}$$

$$\mu_l(u_l) = \max\{(\mu_l^1(u_{1l}), \mu_l^2(u_{2l}), ..., \mu_l^{\xi}(u_{\xi l})\} \tag{3.45}$$

$$\mu_l(u_l) = (\mu_l^1(u_{1l}) \cdot \mu_l^2(u_{2l}) \cdot ... \cdot \mu_l^{\xi}(u_{\xi l}) \tag{3.46}$$

8. If the fuzzy system that is been considered in the construction of the fuzzy rules is defined by the Mamdani model, various defuzzification methods could be used depending mostly of the *priori* knowledge of the system (Passino and Yurkovich, 1998). However, if the fuzzy systems is defined by the T-S model, the singletons used in the consequence part of the rules and the output of the fuzzy system is calculated as in Equation 3.47.

$$f(u_l) = \frac{\sum_{l=1}^{r} \mu_l(u_l)\overline{s}_l^{\xi+1}}{\sum_{l=1}^{r} \mu_l(u_l)} \tag{3.47}$$

where $\mu_l(u_l)$ is defined in Equation 3.44 to Equation 3.46.

9. *(An optional step).* The generated membership functions and/or singletons, using clustering methods, can be tuned to improve their performance by using the data Z_λ in Equation 3.40 as training data. The most widely used techniques in these types of problems are gradient methods where the position of the modal values are trying to be optimised (Driankov *et al*, 1993), (Chi *et al*, 1996). Hence the gradient optimisation method tries to pick the parameters φ to construct the fuzzy system $f(u_l \mid \varphi)$ that gives the best approximation (i.e. make $f(u_l \mid \varphi)$ as close to Z_λ as possible).

Considering the error between the output of the fuzzy system $f(u_l \mid \varphi)$ and m trained data y the gradient method seeks to minimise e_m (Equation 3.48) by choosing the parameters φ.

$$e_m = \frac{1}{2}\left[f(u^m \mid \varphi) - y^m\right]^2 \tag{3.48}$$

In a fuzzy system these parameters can be the singletons and/or the parameters of the membership functions. The update law for both membership functions and singletons are described analytically in Appendix D. Another approach would be to minimise the sum of such error values for a subset of the data Z_λ or all the data in Z_λ; however, with this approach computational requirements increase and algorithm performance may not have significant improvement (Passino and Yurkovich, 1998).

Algorithm	Prototypes Initialisation	Advantages	Disadvantages	Inventor(s)
Hard c-means	Unknown It can be random		Unrealistic and thus inapplicable in real problems	Dunn (1974)
Fuzzy c-means	Unknown It can be random	Data can be member of more than one cluster with different degree. More flexible than "hard" and more useful in practical applications	Detects clusters of fix shapes	Defined by Dunn (1974) Generated by Bezdek (1981)
Mountain Method	Not needed	Detects cluster prototypes and their approximate position	Not very accurate in definition of the cluster prototypes	Yager & Fiver (1992)
Gustafson-Kessel	Unknown It can be random	Detects clusters of varying shapes and sizes. However, the size of the clusters is limited. Not sensitive to scaling of the data.	Slow for a large data dimension n and large number of clusters, Not applicable to linear problems, Find clusters for approximately equal volumes	Gustafson-Kessel (1979)
Fuzzy Maximum Likelihood Estimates	Unknown Needs to be good	Detect clusters of varying shapes, sizes and densities (volumesρ)	Generates almost crisp partitions, Sensitive to the initial conditions	Bezdek & Dunn (1975)

Table 3.1 Clustering Methods

3.12 Discussion

As in any optimisation method, the proposed algorithmic method depends of the availability of "good" data. "Good" data implies data that has been extracted from full "control excitation" of the observed system. Good data are also those that result from well distributed input data. Note that the clustering methods optimise the cluster prototypes and variances correctly in a linguistic sense only if the input data measurements are well distributed, otherwise the cluster algorithms may be biased into the subspace (cluster) constructed from the input data that are most frequently measured. It is a good idea therefore to collect as much data as possible from the expert's control action and then use them only after equal distribution. Data that are close to each other can be merged or only one can be considered.

The mountain method is used in this proposed method as it gives the ability to observe the possible number of the prototypes using the graphical representation of the mountains. Moreover, the approximate position of the cluster centres is obtained. The accuracy of this is highly dependent on the density of the grid. It is however not critical as the Gustafson-Kessel and FCM algorithms can define the prototypes with high accuracy. Gustafson-Kessel and FCM algorithms are used, as they are well known and the most widely used clustering algorithm. Some approaches that improve the above algorithms can be found in literature and may be used instead for specific problems. However, the proposed method uses Gustafson-Kessel and FCM and so does not lose its generality.

The merging method as proposed is actually a combination of the mathematical/geometrical similarity of the fuzzy sets together with their linguistic interpretation. It is believed therefore that what should be considered most (mathematical/geometrical similarity or linguistic interpretation) is depended on what is available most. Note also that the merging method is one of the best ways to minimise the number of fuzzy sets, which results in the reduction of the number of the rules of the constructed fuzzy system, and thus its complexity.

The use of gradient descent methods guarantees convergence to a "local minimum" making the optimal solution close to the initial solution and hence there are no guarantees that will succeed in achieving the best approximation. This is the reason to indicate this step as optional, because the expected improvement in the solution will not be very significant for many applications, especially if there is more interest in the linguistic description of the rules.

Note finally that using optimisation methods such as gradient methods to define the modal values of the antecedents/ consequences means that the resulting model of the control system is highly dependent on the available training data.

3.13 Summary

In this chapter a small introduction to the basic theory of cluster analysis is presented. Investigation of the most basic and widely used fuzzy cluster algorithms such as "hard" clustering, fuzzy c-means, Gustafson-Kessel and fuzzy maximum likelihood estimates is undertaken. Their advantages and disadvantages are discussed as well as their possible applications in identification problems based on input/output mapping data. The effect of normalisation to the clustering methods is also pointed out and therefore its use or not in different types of cluster analysis algorithms is discussed. A well-known method called "mountain" is briefly presented and proposed as the one to determine and define the number and the approximate positions of the cluster prototypes. The generation of the fuzzy sets out of the projection of the cluster centres and variance is also presented. Merging technique to minimise the number of generated membership functions is proposed. A systematic methodology to construct an identification method of fuzzy control strategies based on availability of input/output mapping data is also proposed. Some points about the proposed method are finally discussed. In the next chapter the proposed method is applied to identify an underwater vehicle's control strategy in terms of avoiding objects.

3.14 References

Babuska, R. 1998a. *Fuzzy Modelling for Control.* Boston: Kluwer Academic Publishers. 0792381548 (acid-free paper).

Bezdek, C. J.,Keller, J.,Krisnapuram, R., and Pal, N. R. 1999. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing.* Kluwer Academic Publishers.

Bezdek, J., and Dunn, J. 1975. Optimal Fuzzy Partition: a Heuristic for Estimating the Parameters in a Mixture of Normal Distributions. *IEEE Trans. Computers,* **C-24**, pp. 835-838.

Bezdek, J. C. 1981. *Patern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press. 0-306-40671-3.

Chi, Z., Yan, H., and Pham, T. 1996. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition.* Singapore: World Scientific Publishing Co. Pte. Ltd.

Driankov, D.,Hellendoorn, H., and Reinfrank, M. 1993. *An introduction to fuzzy control.* Berlin, New York: Springer-Verlag. 3540563628 (Berlin). 0387563628 (New York).

Espinosa, J., and Vandewalle, J 1999. Linguistic Integrity: A Framework for Fuzzy Modelling - AFRELI. *In.*

Gustafson, D., and Kessel, W. 1979. *Fuzzy Clustering with a Fuzzy Covariance Matrix.* IEEE CDC . San Diego.

Jain, A., and Dubes, R. 1988. *Algorithms for Clustering Data.* Prentice Hall.

Kaymak, U., and Babuska, R. 1995. *Compatible Cluster Merging for Fuzzy Modelling.* FUUZ-IEEE/IFES'95. pp. 897-904,

Krishnapuram, R., and Freg, C.-P. 1992. Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recognition* , **25** ((4)), pp. 385-400.

Lori, N., and Costa Branco, P. J. 1995. *Autonomous Mountain-Clustering method applied to fuzzy system modelling.* ANNIE'95.

Passino, K M., and Yurkovich, S. 1998. *Fuzzy control.* Menlo Park, Calif: Addison-Wesley. 020118074X.

Satnes, M.,Babuska, R.,Kaymak, U., and Van Nauta Lemle, H. R. 1998. Similarity Measures in Fuzzy Rule Base Simplification. *IEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics,* **28** (3), pp. 376-386.

Strang, G. 1980. *Linear Algebra and its Applications.* 2nd end. Academic Press.

Sugeno, M., and Yasukawa, T. 1993. A Fuzzy Logic Based Approach to Qualitative Modelling. *IEEE Trans. on Fuzzy Systems,* **1** (1), pp. 7-31.

Xie, X. L., and Beni, G. A. 1991. Validly measure for fuzzy clustering. *IEEE Trans. on Pattern Anal. and Machine Intell.,* **8** (8), pp. 841-846.

Yager, R. R., and Filev, D. P. 1994a. Approximate Clustering Via the Mountain Method. *IEEE Transactions on Systems, Man, and Cybernetics,* **24**, pp. 1279-1284.

Yager, R. R., and Filev, D. P. 1994b. *Essentials of Fuzzy Modelling and Control.* New York: John Wiley. 0471017612 (acid-free).

Zwick, R.,Carlstein, E., and Budescu, D. V. 1987. Measures of Similarity among Fuzzy Concepts: A comparative analysis. *Int. Journal Approximate Reasoning,* **1**, pp. 221-242.

4

Generation of fuzzy rules for "Avoiding Objects" Control Strategy using 3-D input/output data space

4.1 Introduction

In Chapter 3 Section § 3.11 an algorithmic methodology is proposed to identify fuzzy control strategies using any n-dimensional input-output space. In this Chapter the proposed methodical steps, are applied to identify an underwater robots' (GARBI) control strategy to avoid objects (Figure 4.1). This application involves one of the most common cases in robot navigation problems. The objective, therefore, is to identify the fuzzy rules that the operator uses to construct the control surface.

The chapter is constructed as follows: in Section § 4.2 the algorithm is applied. Analytically, in Section § 4.2.1 the data samples and construction of the control surface is defined. In Section § 4.2.2 the mountain clustering method is applied to define the number and the positions of the cluster's prototypes. Both Gustafson-Kessel method and Fuzzy C-Means method are applied in Section § 4.2.3 and Section § 4.2.4 respectively. Two cases are investigated, in the first case the initial number of prototypes is considered as five, whereas in the second case the algorithm is applied for five and nine initial prototypes (Section § 4.2.4.1 Section § 4.2.4.2). In Section § 4.3 the results are discussed and finally a brief summary of the chapter is presented in Section § 4.4.

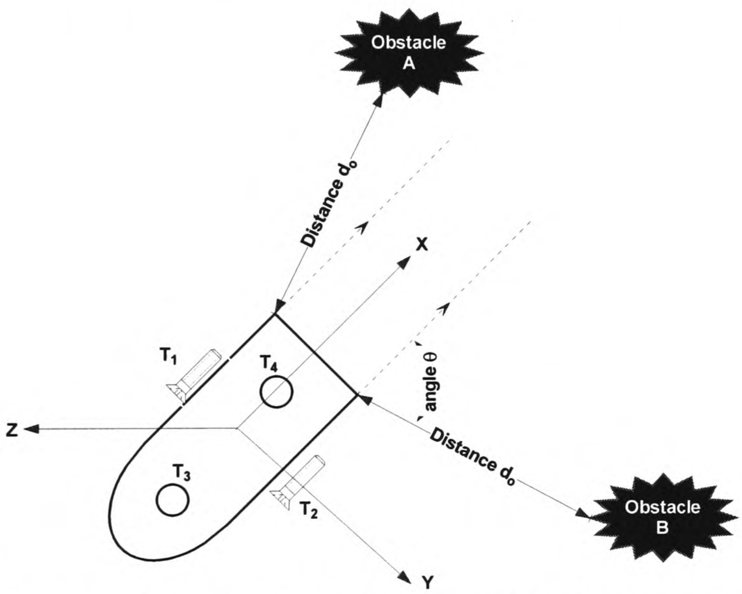


Figure 4.1 Distance d_o and angle θ as the input variables for obstacle avoidance

4.2 Identifying a control strategy for object avoidance

In this Section the identification of the control strategy for object avoidance of GARBI is presented. This case study considers only the XY surface of the robots navigation. The steps that are applied according to the algorithmic methodology discussed in Section § 3.11 are as follows:

4.2.1 Definition of data samples and construction of the control surface

The strategy of the output control action is based on two inputs: distance d_o and angle $\theta°$ between the robot and the objects. Therefore, the multidimensional input/output space is constructed by two-inputs and one-output variables. Thus from Equation 3.36, the input matrix is defined as in Equation 4.1

$$U = \begin{bmatrix} d_o \\ \theta \end{bmatrix} \tag{4.1}$$

where $U \in \Re^2$

and for Equation 3.37 and 3.39 the output vector is defined as in Equation 4.2

$$Y = [CtrlAction] = \lfloor y_{(3,1)}, y_{(3,2)}, \ldots, y_{(3,400)} \rfloor \tag{4.2}$$

For the distance d_o and angle $\theta°$ input variables, the vectors for 400 observations are constructed for Equation 3.38 as in Equation 4.3

$$d_o = \lfloor d_{o_{(1,1)}}, d_{o_{(1,2)}}, \ldots, d_{o_{(1,400)}} \rfloor, \qquad \theta = \lfloor \theta_{(2,1)}, \theta_{(2,2)}, \ldots, \theta_{(2,400)} \rfloor \tag{4.3}$$

and therefore the data matrix Z of Equation 3.40 is constructed as in Equation 4.4.

$$\mathbf{Z} = \begin{bmatrix} d_{o_{(1,1)}} & d_{o_{(1,2)}} & . & . & . & d_{o_{(1,400)}} \\ \theta_{(2,1)} & \theta_{(2,2)} & . & . & . & \theta_{(2,400)} \\ y_{(3,1)} & y_{(3,2)} & . & . & . & y_{(3,400)} \end{bmatrix} \tag{4.4}$$

Simulating the operator's control actions using the yaw model of GARBI, developed in Chapter 2, generates the data. Therefore the data is distributed in 3-dimensional space as illustrated in Figure 4.2. The co-ordinates are defined by the distance d_o the angle $\theta°$ and the control action. The domain of the distance d_o is within 10m, for the port side of the robot the angle $\theta°$ is within $[0°, -108°]$ and for the starboard side it is within $[0°, 108°]$ and the control action is a percentage of the power of the propellers within $[-100\%,100\%]$. Note that the minus sign for the power means that the propeller(s) is turning in reverse. Thus, the sign of the propellers T_1 and T_2 is the same when the robot has to move forward or backwards, minus for T_1 and plus for T2 when the robot has to turn left, plus for T_1 and minus for T_2 when the robot has to turn right. Using the generated data the control surface for the propeller T_1 is constructed as illustrated in Figure 4.3. The input co-ordinates are defined by equally distributing distance d_o and the angle $\theta°$ axis in 20 points for each.

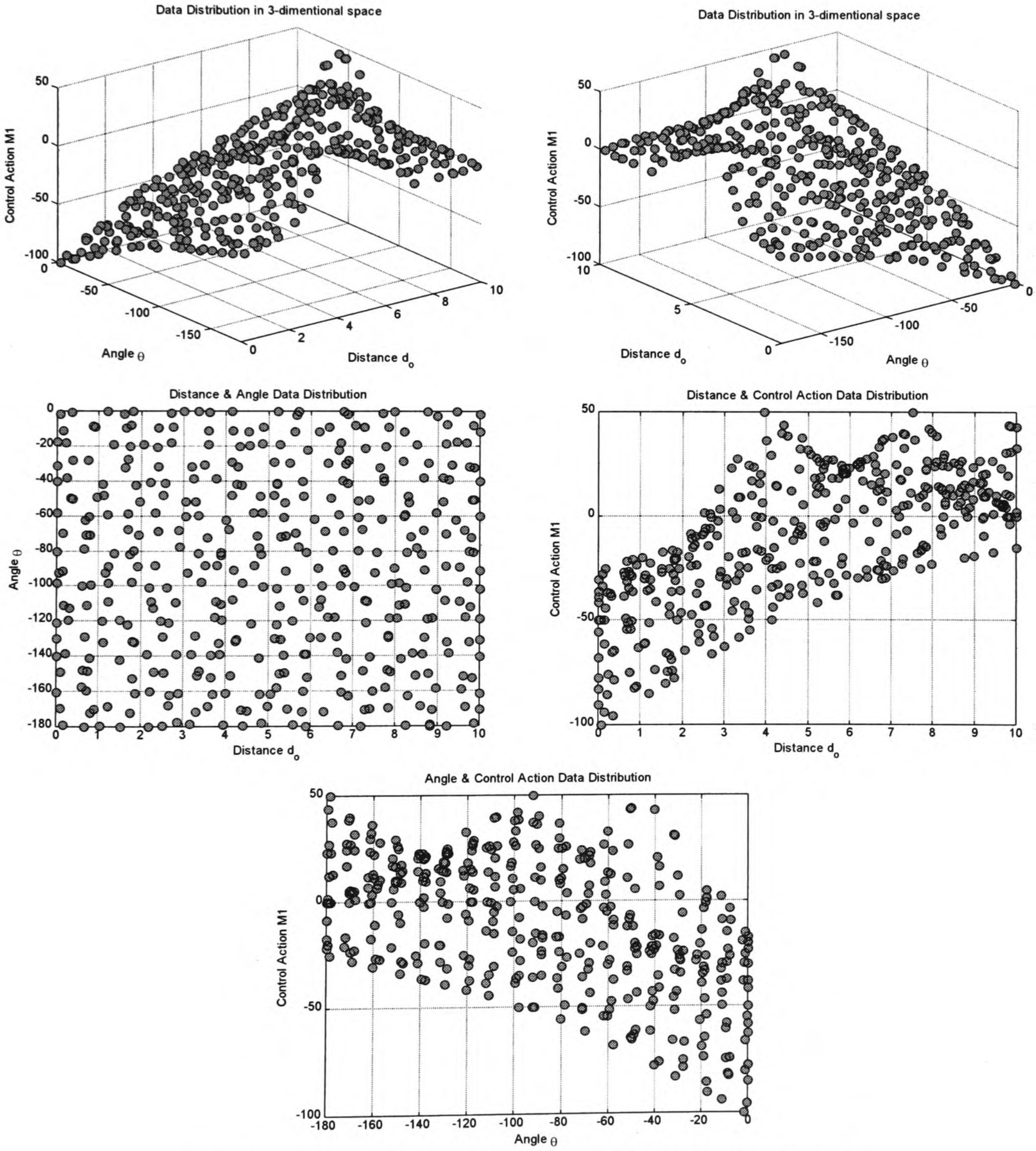


Figure 4.2 Data distribution in 3 and 2 dimensional space

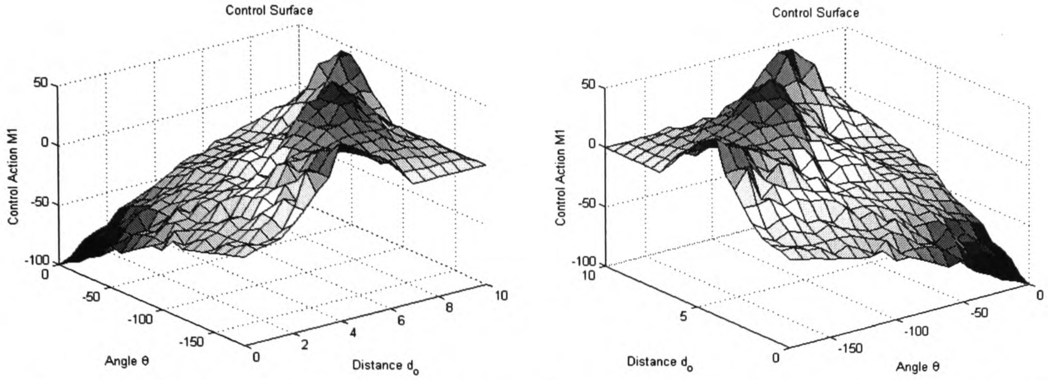


Figure 4.3 Control surface of the control action according to distance d_o and angle $\theta°$ measurements

4.2.2 Mountain Clustering method to define the number and the positions of prototypes

Applying the Mountain Clustering Method as explained in Section § 3.7.1, three groups of mountains (according to their height) can be observed from Figure 4.4 to Figure 4.6 i.e. very high mountains {119.40, 98.88}, high mountains {48.18, 42.50,40.87} and low mountains {28.05, 16.66, 14.92, 9.46}. Therefore, the proposed number of candidate prototypes could be either two (considering only the group with very high mountains) or five (considering both groups with very high and high mountains). As two clusters have been tested and found to be insufficient to describe the non-linear input-output mapping of the data, five cluster centres are selected to be used as the initial number for the Gustafson-Kessel and FCM algorithms. These are the two case studies described in Section § 4.2.3 and 4.2.4.1. The five prototypes resulting from the mountain method are illustrated in Figure 4.7 and are used as the initial ones for the FCM algorithm. Moreover, as another case study the third group of the mountains is considered and therefore the nine cluster centres illustrated in Figure 4.8 are used as the initial prototypes for the FCM algorithm as described in Section § 4.2.4.2. Note finally that the grid lines used in the mountain method appear every $1/6^{th}$ of each dimensional scale and therefore the number of intersections is $1.7 \times 30 \times 33.4 = 1603$ which means very high density.

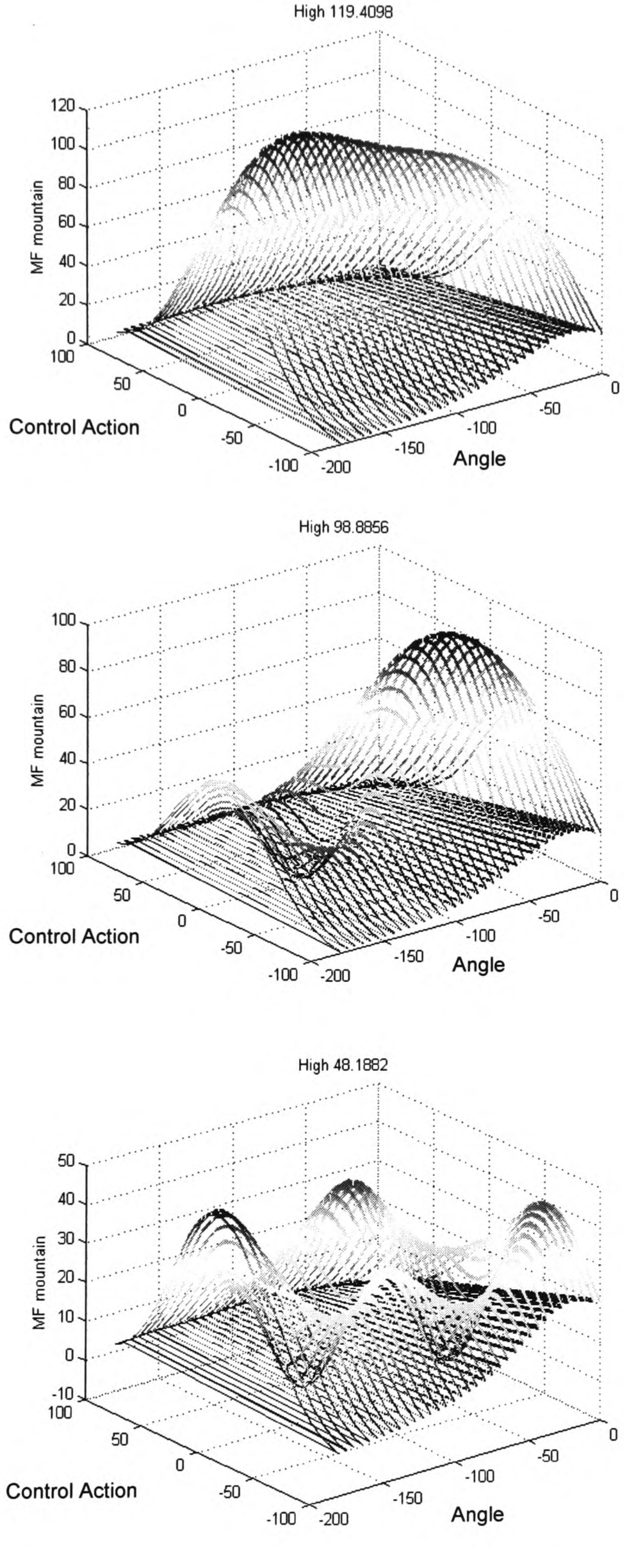


Figure 4.4 Graphical representation of first to third mountains (prototypes) together with their Heights

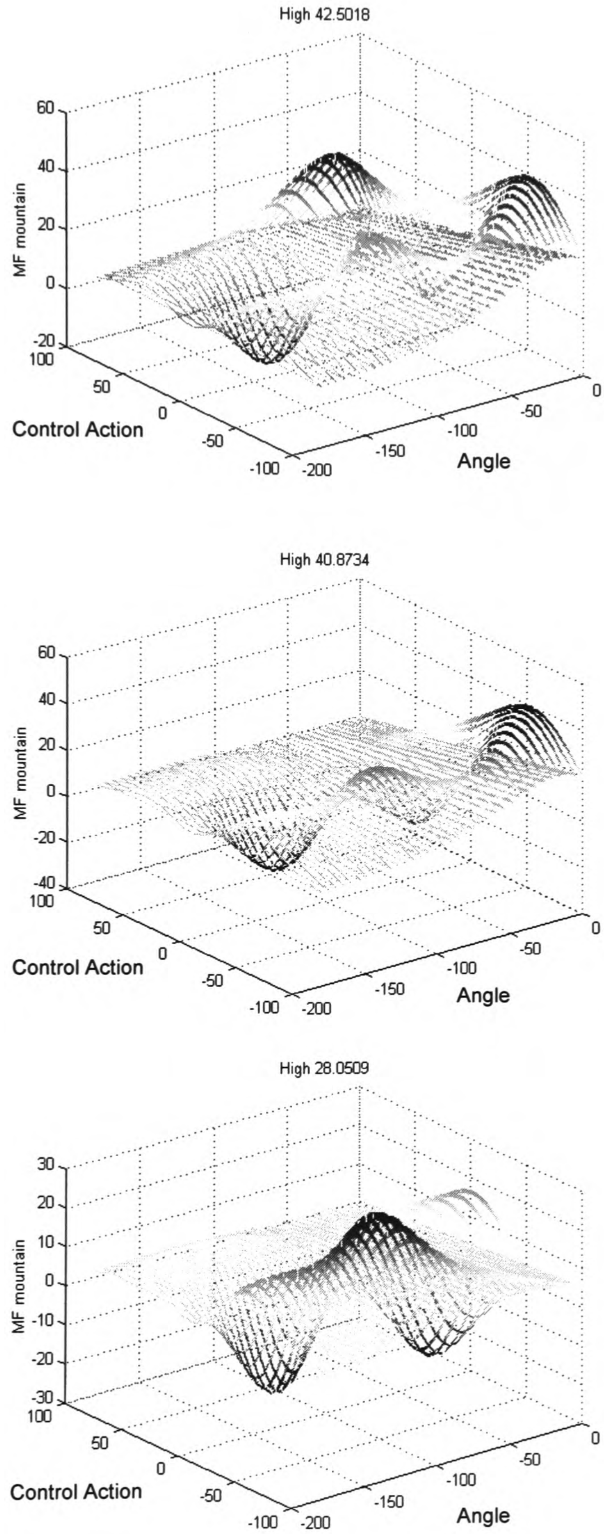


Figure 4.5 Graphical representation of fourth to sixth mountains (prototypes) together with their Heights

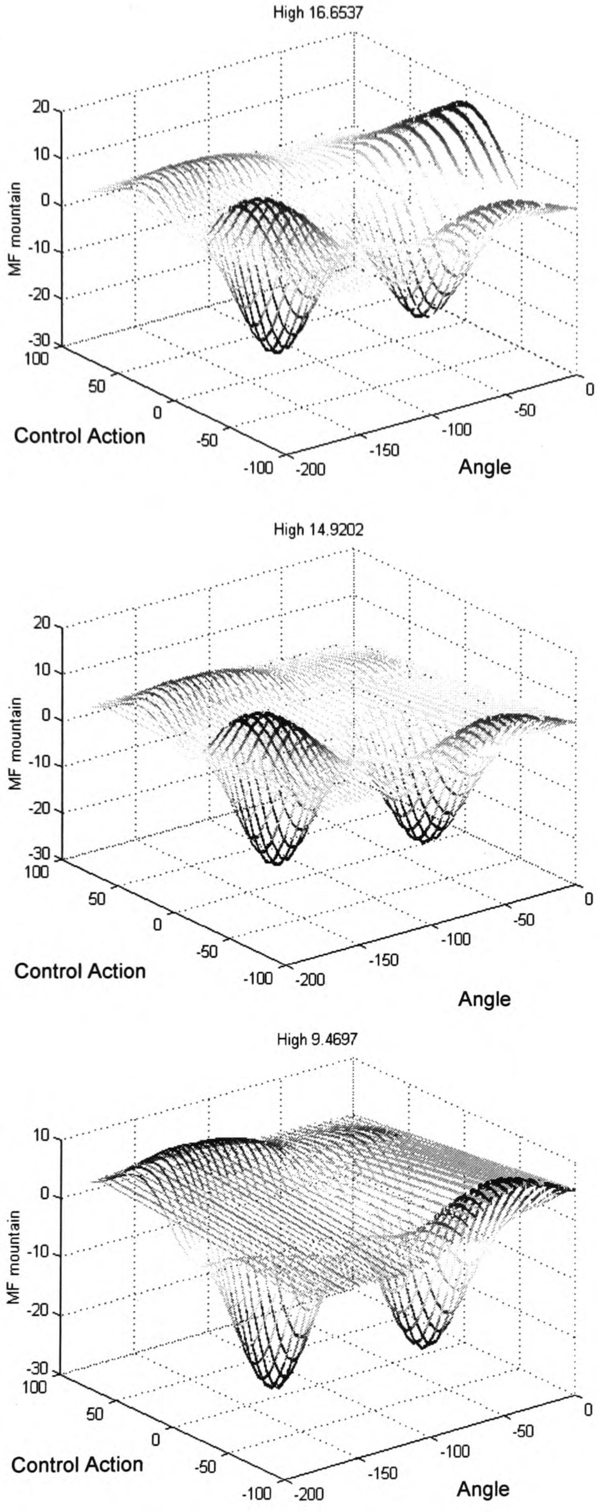
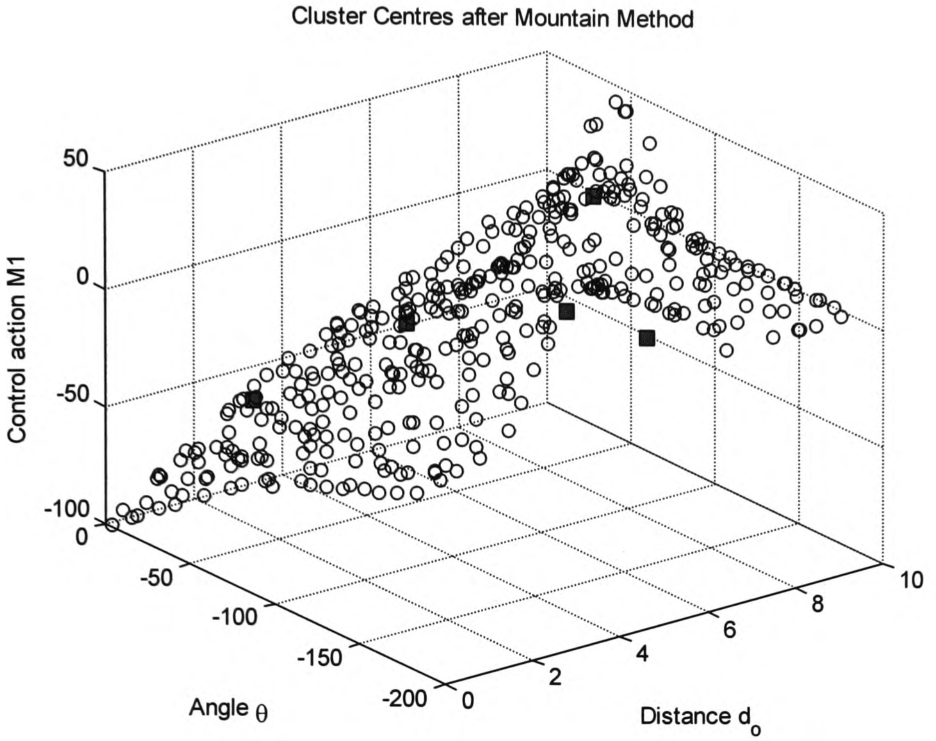
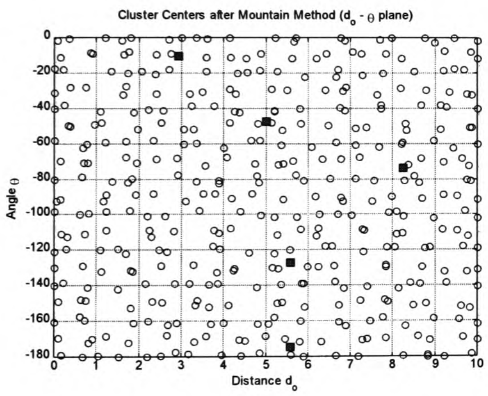


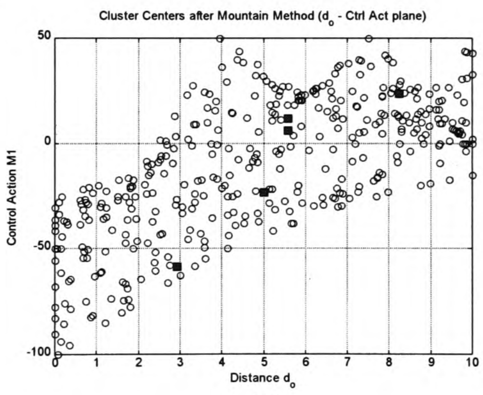
Figure 4.6 Graphical representation of seventh to ninth mountains (prototypes) together with their Heights



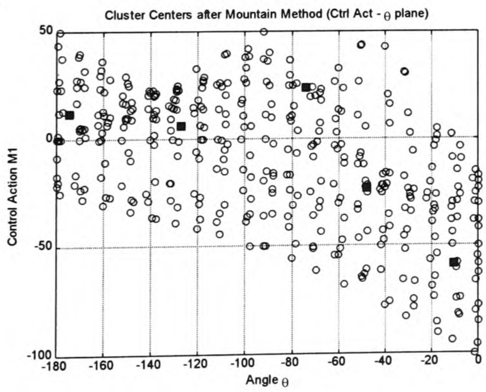
(a)



(b)



(c)



(d)

Figure 4.7 The five prototypes (marked by ■) identification using the Mountain Cluster Method in (a) 3-dimensional space (b), (c) and (d) 2-dimensional space

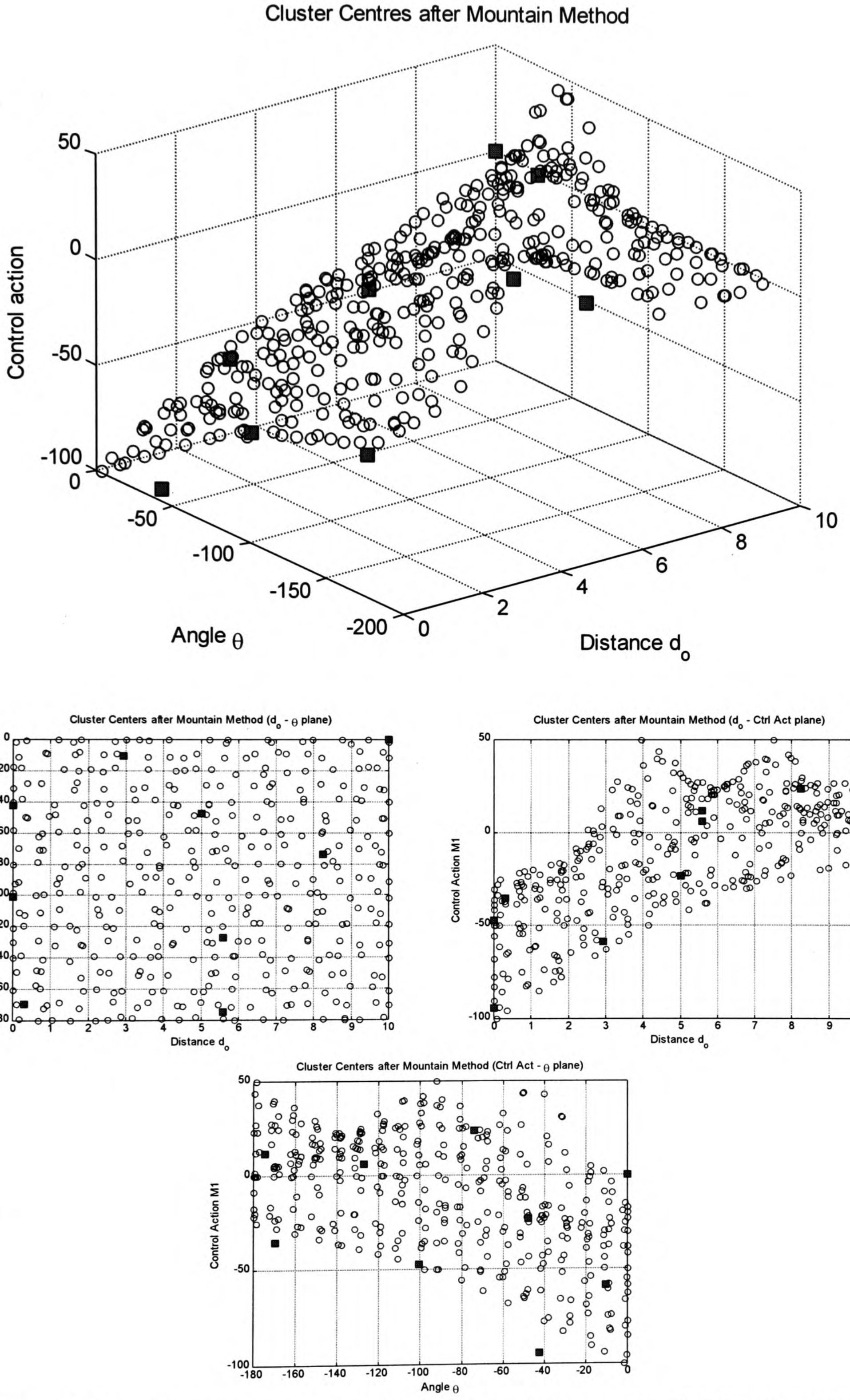


Figure 4.8 The nine prototypes (marked by ■) identification using the Mountain Cluster Method in (a) 3-dimensional space (b), (c) and (d) 2-dimensional space

4.2.3 Applying Gustafson-Kess el method and conducting the rest of the steps of the proposed algorithmic approach

The Gustafson-Kessel algorithm is applied to detect the cluster centres of the data as illustrated in Figure 4.9. The data have been normalised using Equation 3.25. The number of prototypes is defined as five as discussed in Section § 4.2.2. As explained in Section § 3.6.1 the covariance of the pattern classes in the data can define ellipsoidal patches. The eigenvectors and eigenvalues of the covariance matrix define the ellipsoids in the 3-dimensional input-output space as in Equation 3.21. These ellipsoids are shown in Figure 4.10 to Figure 4.12 in all combinations of 2-dimensional planes i.e. d_o-$\theta°$, d_o-*Ctrl action*, $\theta°$-*Ctrl action* respectively. The ellipsoidal are projected into the input (distance d_o and angle $\theta°$) and output (Ctrl action) axes to form the fuzzy sets. These projections create symmetric membership functions as explained in Section § 3.9 and shown in Figure 4.13*(i)* and Figure 4.14*(i)* with the peak point of the cluster centres and the left and right fractions defined as in Equation 3.30 and Equation 3.31. Note that to obtain full coverage of the input-output space, two modal values are added to the extremes of each universe of the discourse.

The membership functions in Figure 4.13*(i)* are merged considering the closeness between them, regarding the number of the prototypes that are defined using the mountain clustering method and observing their linguistic interpretation. Thus inFigure 4.13*(a,ii)* five fuzzy sets are generated which linguistically implies five different distances such as Very Far (VF), Far (F), Close (C), Near (N), Very Near (VN). In Figure 4.13*(b,ii)* the generated fuzzy sets have four different angles zones i.e. Front Left Zone1 (FL Zone1), Front Left Zone2 (FL Zone2), Left (L), Back Left (BL). Note that the two triangular membership functions that are applied for high values of the angle universe of discourse do not have very high degree of overlapping (approximately 70%). However, it was decided that they should be merged as in the Back Left zone the control action is not so significant for the controller to avoid objects as the vehicle mainly moves in the forward direction.

For the output control actions, singletons generated from the peak values of the membership functions were constructed after merging the initial fuzzy sets resulting from the projected cluster prototypes and ellipsoids as illustrated in Figure 4.14. It was decided that singletons would be used as it is recommended in fuzzy control systems producing faster processing time (Espinosa and Vandewalle, 1997), (Jantzen, 1998). Moreover, it makes it easier to improve the performance of the fuzzy system using optimisation methods such as Gradient method as introduced in Section § 3.11.

Defining the linguistic meaning of the generated fuzzy sets, the rule table can be constructed as in Table 4.1. The singleton value -100 represent the control actions that drives the vehicle Very Fast Backwards (VFB) and Turning the vehicle Very Fast Right (TVFR). The next three singletons value -52.19 , -23.61 and -9.35 are used for Turning the vehicle Fast, Medium and Slightly Right (TFR), (TMR), (TSR) respectively. The positive values of the singletons i.e. 50 and 9.27, define the action to Go Forward with Fast (GFF) and low (GF) speed respectively.

The diagonal equality of singletons is a standard way in design rule base table in fuzzy control systems. The table is constructed setting the linguistic labels in order. Thus, as can be observed from Table 4.1 the input referring to the distance is positioning from VN to VF and the input referring to angle is positioned in order from FL to BL.

To define the control action of the propellers, the Takagi-Sugeno (T-S) fuzzy system is used where the defuzzification obtained using Equation 3.47. Therefore the resulting control surface and control output data distribution (Figure 4.15) produced by the Fuzzy System uses the membership functions and singletons generated by the G-K method as well as the rule table

(Table 4.1). The input values of the distance d_o and the angle $\theta°$ are the same as in Equation 4.1.

The average percentage error (APE) is employed as a performance index (Jang *et al*, 1997) defined as in Equation 4.5.

$$APE = \frac{1}{P}\sum_{i=1}^{P}\frac{|T(i) - O(i)|}{T(i)} \times 100\% \tag{4.5}$$

where P is the number of the input/output data, $T(i)$ is the desired output data and $O(i)$ is the predicted output data. The APE is therefore 0.0586% that means that the data of resulting fuzzy control action Figure 4.15 is very close to the original ones as defined in Equation 4.1 and illustrated in Figure 4.2.

To improve the performance of the constructed Fuzzy Control System the gradient method is applied to re-optimise the singletons and the parameters of the membership functions as explained in the 9^{th} step of Section § 3.11 and Appendix D.

When the aim is to optimise only the singletons, the initial values of the singletons are those defined in Table 4.1. The final optimal values after 120 iterations are as in Table 4.2. The resulting control surface and control output data when the singletons are optimised are illustrated in Figure 4.16. The APE in this case is 0.0519% which means that some small improvement is obtained using gradient method.

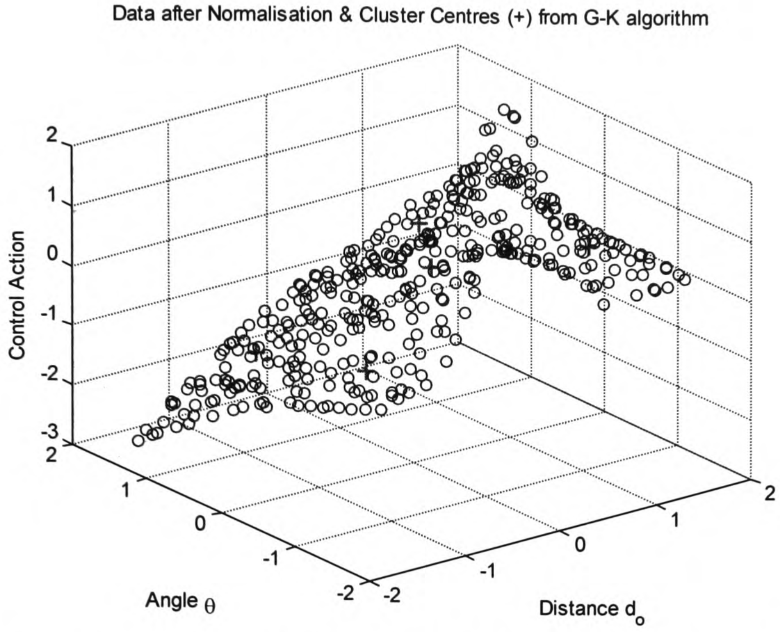


Figure 4.9 The prototypes (marked with +) identification using Gustafson-Kessel clustering algorithm in 3-dimensional space

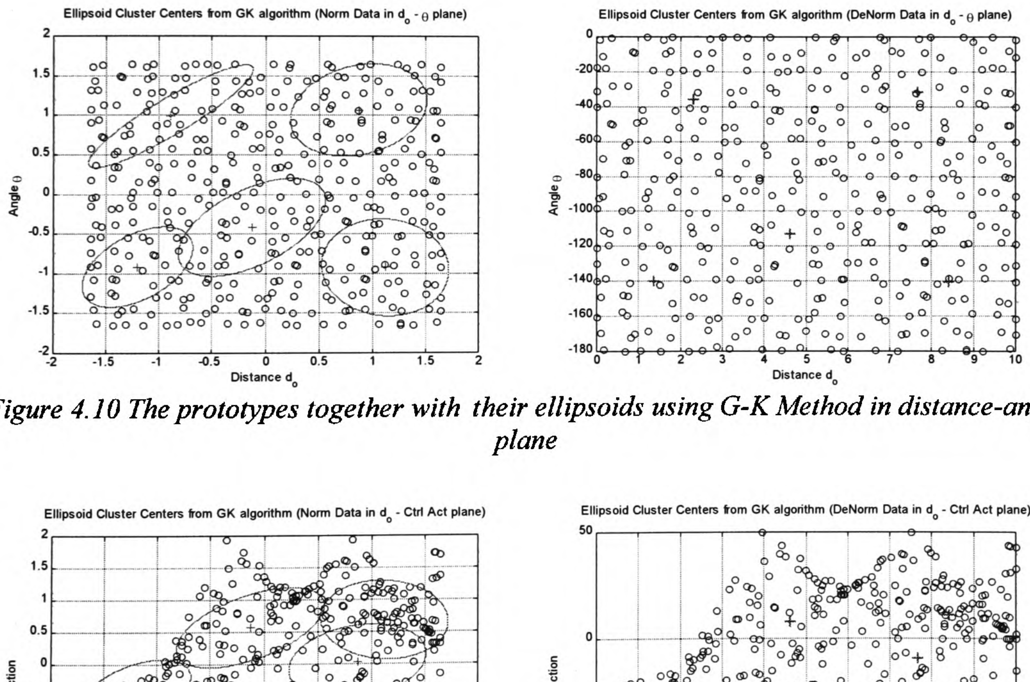


Figure 4.10 The prototypes together with their ellipsoids using G-K Method in distance-angle plane

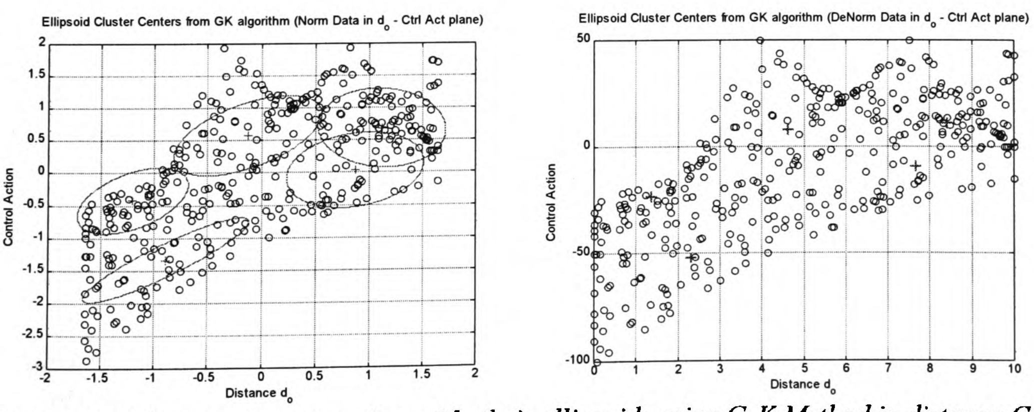


Figure 4.11 The prototypes together with their ellipsoids using G-K Method in distance-Ctrl action plane

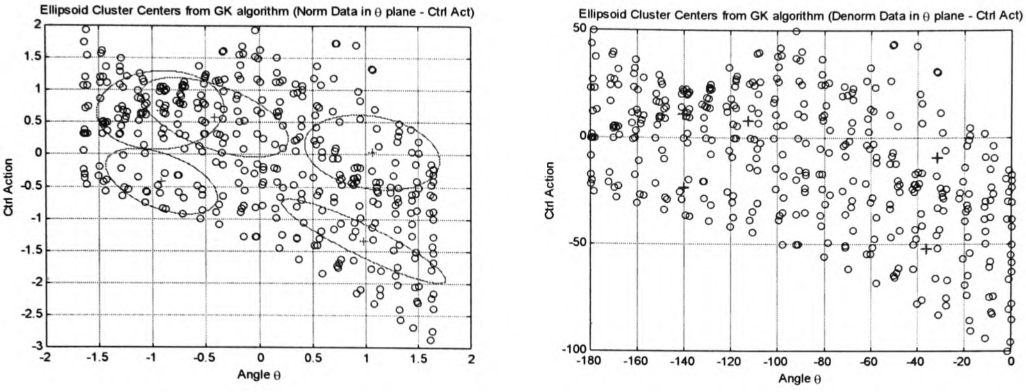
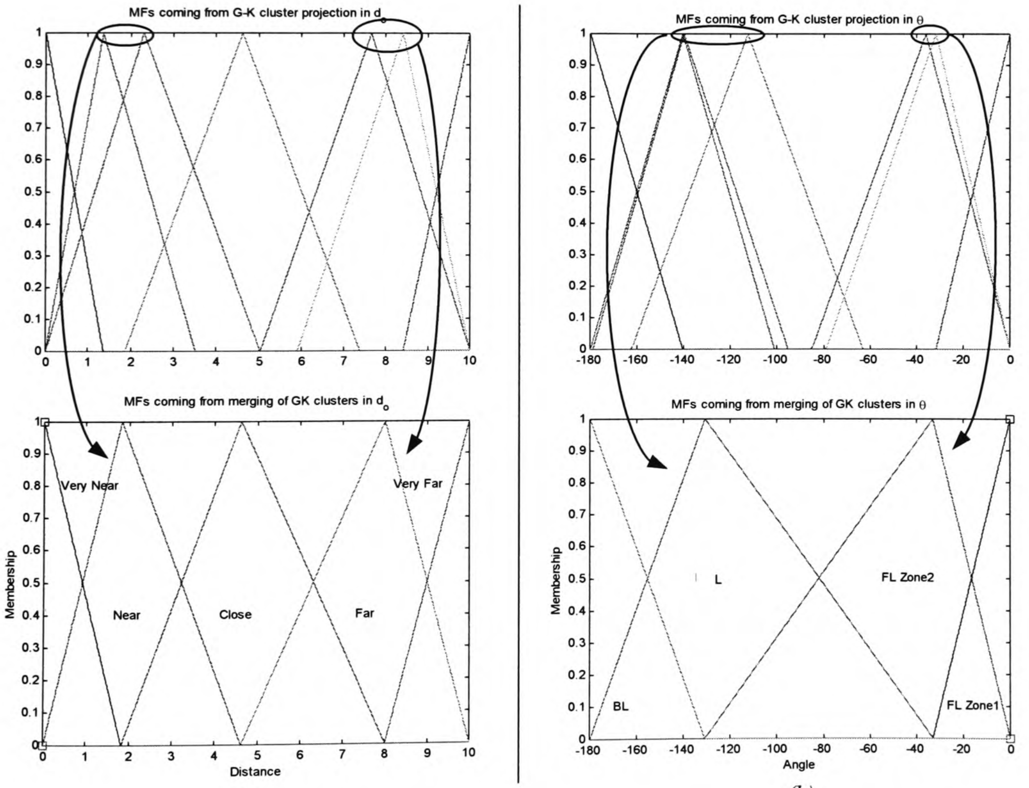


Figure 4.12 The prototypes together with their ellipsoids using G-K Method in angle-Ctrl action plane



(i)

(ii)

(a) (b)

Figure 4.13 Membership functions in the input axes, generated initially (i) from the projection of cluster prototypes and their ellipsoids, and finally after the merging method (ii)

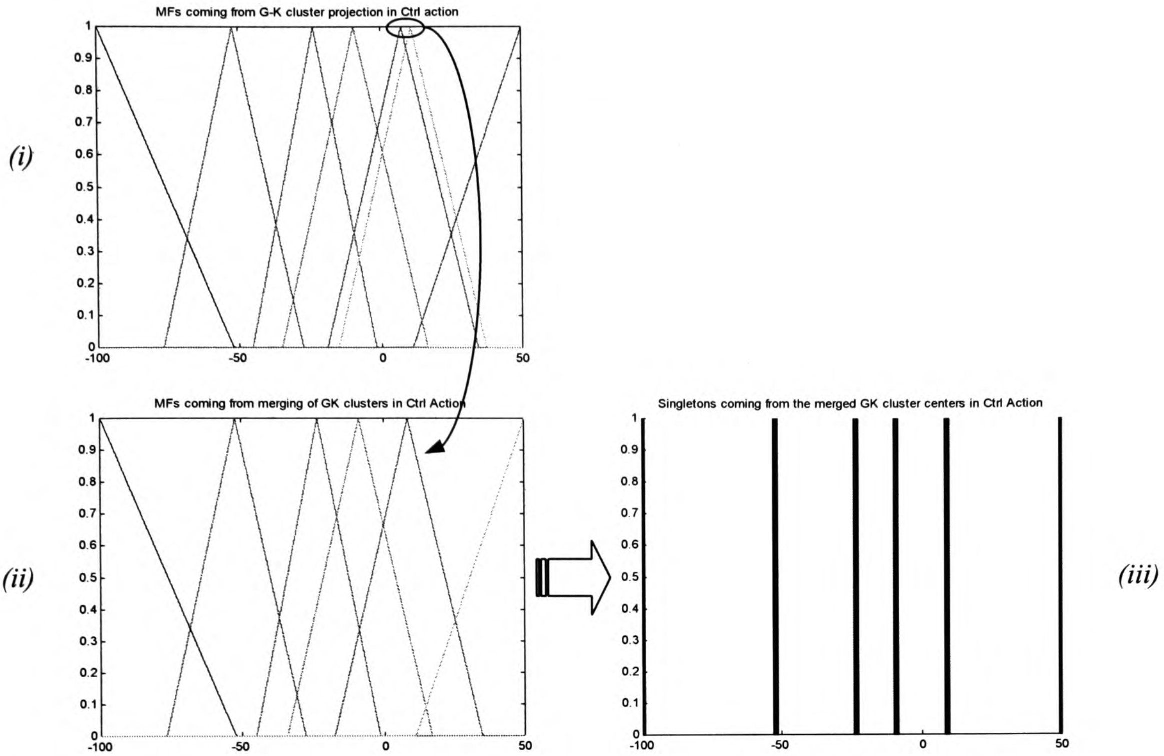


Figure 4.14 The projected cluster prototypes and ellipsoids (i) that merged (ii) and generate the singletons in the output axis (iii)

d_o \ θ	Front Left Zone 1	Front Left Zone 2	Left	Back Left
Very Near	VFB -100	TVFR -100	TFR -52.19	TMR -23.61
Near	TVFR -100	TFR -52.19	TMR -23.61	TSR -9.35
Close	TFR -52.19	TMR -23.61	TSR -9.35	GFF 50
Far	TMR -23.61	TSR -9.35	GFF 50	GF 9.27
Very Far	TSR -9.35	GFF 50	GF 9.27	GF 9.27

Table 4.1 The Rule Base of the generated FLC in tabular form

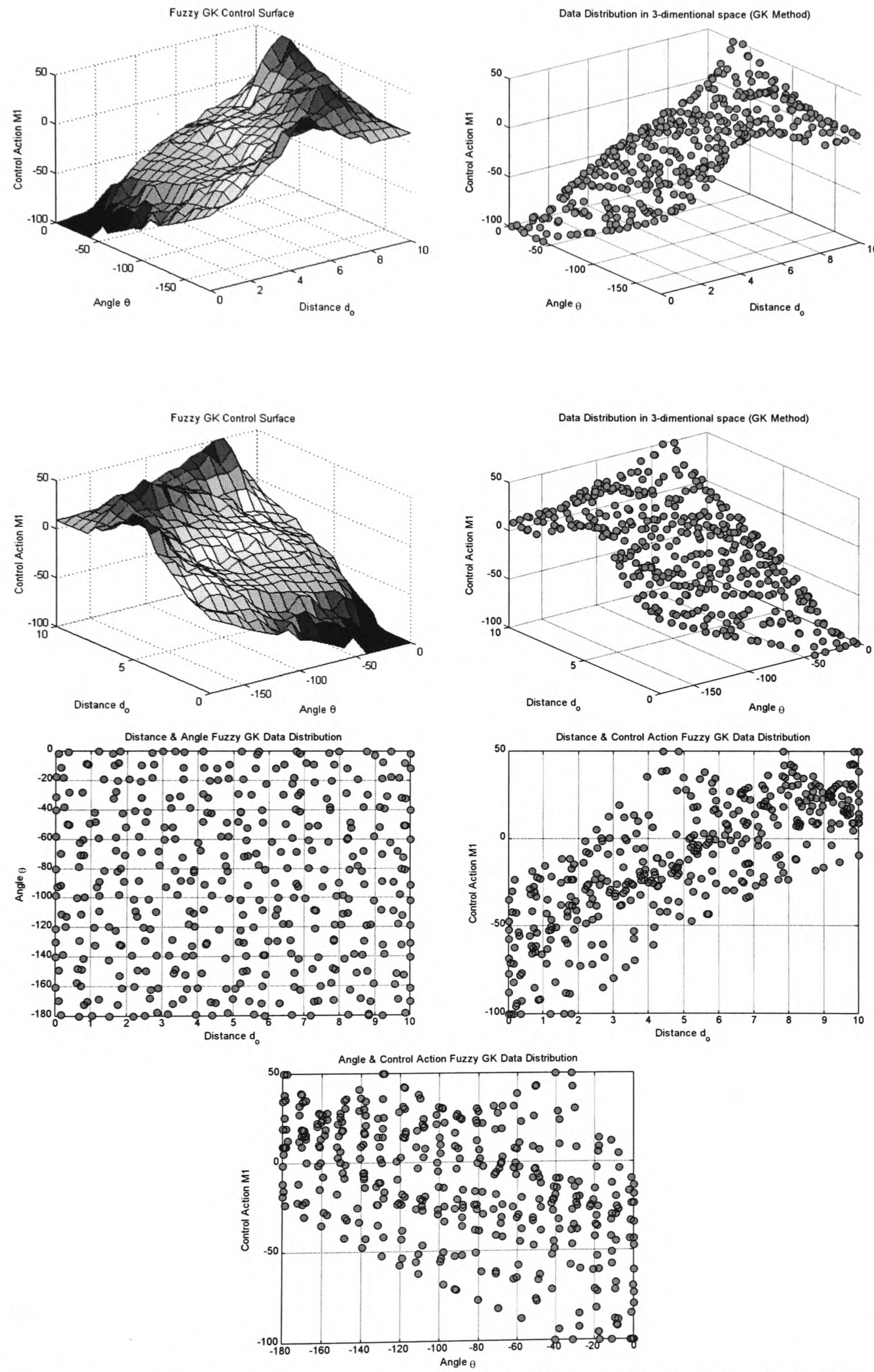


Figure 4.15 Surface and data distribution generated by the Fuzzy System using the G-K method

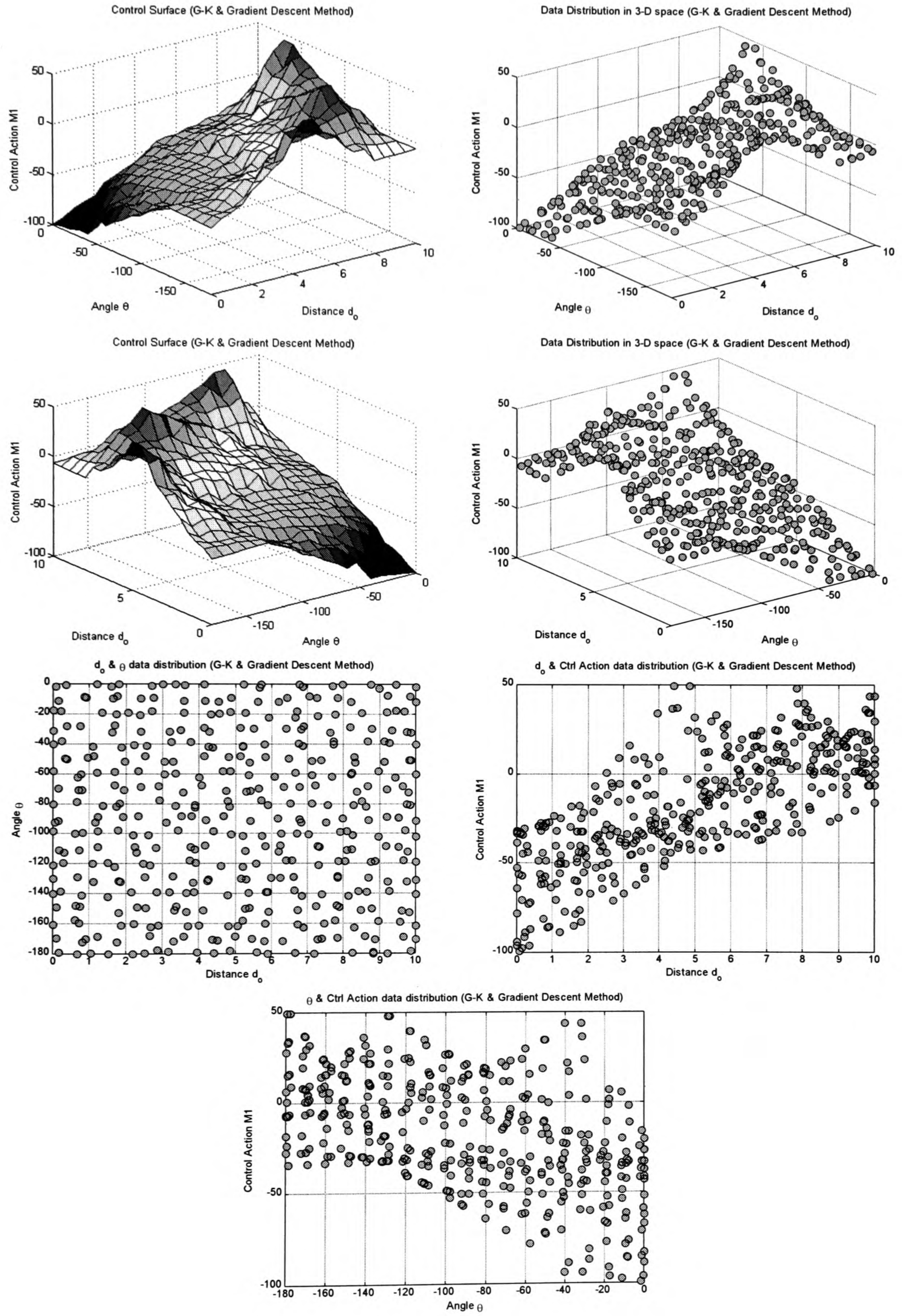


Figure 4.16 Surface and data distribution generated by the Fuzzy System using G-K method and modified in the output singletons by Gradient Descent algorithm

d_o \ θ	Front Left Zone 1	Front Left Zone 2	Left	Back Left
Very Near	VFB -99.16	TVFR -93.66	TFR -31.78	TMR -23.61
Near	TVFR -82.81	TFR -55.23	TMR -32.19	TSR -15.22
Close	TFR -45.12	TMR -35.06	TSR -16.05	GFF 49.67
Far	TMR -32.65	TSR -15.98	GFF 48.26	GF 7.51
Very Far	TSR -15.83	GFF 43.69	GF 6.15	GF 6.50

Table 4.2 Optimised Singletons using Gradient method

4.2.4 Applying Fuzzy C-Means method and conducting the rest of the steps of the proposed algorithmic approach

Two different cases are investigated in this section regarding the definition of the cluster's number and their initial values using the mountain method. The first case uses five prototypes and the second nine respectively. Note that in both cases the fuzzy sets are formed by projecting the cluster centres into the input (distance d_o and angle $\theta°$) and output (Ctrl action) spaces. The resulting membership functions are triangular with the peak point as the cluster centre and the left and right fraction defined from the neighbouring cluster centres as described in Section § 3.9. The merging method for the membership functions is applied as discussed in Section § 3.10 i.e. considering their proximity, the definition (using the mountain method) of the number of prototypes and the linguistic interpretation. The use of singletons and the design of the rule base construction are as discussed in Section § 4.2.3. The Takagi-Sugeno model as in Equation 3.47 is also applied to define the control action of the propellers. The input data i.e. distances d_o and angles $\theta°$ are used as in Equation 4.1 in the application of the fuzzy system that generates the output of the control action and thus the control surface.

The identification results are presented for the first case in Section § 4.2.4.1 and for the second case in Section § 4.2.4.2.

4.2.4.1 Five prototypes

In this case the five prototypes resulting from the mountain method (see Figure 4.7) are used as the initial ones for the FCM algorithm. The resulting cluster centres from this algorithm are as illustrated in Figure 4.17. Figure 4.18 *(i), (ii) and (iii)* illustrate the projected clusters constructing the membership functions in distance d_o, angle $\theta°$ and Ctrl action axis accordingly. After applying the merging method, the final membership functions are four for both inputs and five for the output (see Figure 4.18 *(a), (b) and (c)*. Thus the linguistic interpretation of the input membership functions for the distance d_o are named as Far (F), Close (C), Near (N) and Very Near (VN) and for the angle $\theta°$ Front Left Zone1 (FL Zone1), Front Left Zone2 (FL Zone2), Left (L), Back Left (BL). For the output, the resulting singletons come from the peaks of the output membership functions. Thus their linguistic meaning are set as Very Fast Backward (VFB) (value -100), Turn the vehicle Very Fast Right, Fast and Medium Right (TVFR), (TFR), (TMR) (values -64.671, -64.671, -22.52 respectively) and Go Forward Fast (GFF) and low (GF) speed (values 50 and 5). The rules therefore are constructed as in Table 4.3. Simulating the control action for the set inputs the control output data and the control surface of the (T-S) Fuzzy System is defined as illustrated in Figure 4.19. From Equation 4.5 the average percentage error (APE) is $APE = 0.0789\%$. This APE is very small and implies that the data of the fuzzy control action (Figure 4.19) is very close to the original ones (Equation 4.1).

d_o \ θ	Front Left Zone 1	Front Left Zone 2	Left	Back Left
Very Near	VFB -100	TVFR -64.67	TFR -64.67	TMR -22.52
Near	TVFR -64.67	TFR -64.67	TMR -22.52	GVFF 50
Close	TFR -64.67	TMR -22.52	GVFF 50	GF 5
Far	TMR -22.52	GVFF 50	GF 5	GF 5

Table 4.3 The Rule Base of the generated FLC in tabular form

4.2.4.2 Nine prototypes

When all the groups of the candidate prototypes are considered, nine initial cluster centres (see Figure 4.8) are applied in FCM algorithm. Thus applying this algorithm the cluster centres are redefined as illustrated in Figure 4.20. The generated membership functions in distance d_o, angle $\theta°$ and Ctrl action axis are as illustrated in Figure 4.21 *(i), (ii) and (iii)* respectively. Applying the merging method the resulting membership functions are five for the distance named as Very Far (VF), Far (F), Close (C), Near (N) and Very Near (VN) as shown in Figure 4.21 *(a)*. Moreover, for the angle $\theta°$ there are four resulting membership functions named Front Left Zone1 (FL Zone1), Front Left Zone2 (FL Zone2), Left (L), Back Left (BL) as depicted in Figure 4.21 *(b)*. Note that in Figure 4.21 *(ii)* although the four membership functions do not have a very high degree of overlap they present the same linguistic meaning and thus were merged i.e. FL Zone2 (Figure 4.21 *(b)*). For the output the number of the resulting membership functions after merging are six (Figure 4.21 *(c)*) where their peaks define the singletons as illustrated in Figure 4.21 *(d)*. Their linguistic meanings for the vehicle's control action, however, are set as: Very Fast Backward (VFB) (value -100), Turning the vehicle Very Fast, Fast, Medium and Slightly Right (TVFR), (TFR), (TMR), and (TSR) (values -73.91, -48.20, -20.78, -20.78 respectively) and Go Forward with Fast (GFF) and low (GF) speed (values 50, 17.33). Thus the rules are constructed as in Table 4.4. The resulting control surface and control output data distribution resulting from this (T-S) Fuzzy System is as illustrated inFigure 4.22. The average percentage error (APE) (defined in Equation 4.5) is $APE = 0.0817\%$ that means that the data of the obtained fuzzy system (Figure 4.22) is very close to the original ones (Equation 4.1).

θ / d_o	Front Left Zone 1	Front Left Zone 2	Left	Back Left
Very Near	VFB -100	TVFR -73.91	TFR -48.20	TMR -20.78
Near	TVFR -73.91	TFR -48.20	TMR -20.78	TSR -20.78
Close	TFR -48.20	TMR -20.78	TSR -20.78	GVFF 50
Far	TMR -20.78	TSR -20.78	GVFF 50	GF 17.33
Very Far	TSR -20.78	GVFF 50	GF 17.33	GF 17.33

Table 4.4 The Rule Base of the generated FLC in tabular form

Note that the values of TVFR, TFR in Table 4.3 are the same and the values of TMR, TSR Table 4.4 are also the same. This is due to the limited number of generated singletons. However, the values of TFR (Table 4.3) and/or TMR (Table 4.4) could form another two choices a) equal to TMR and/or TFR b) a value between TVFR and TMR and/or TFR and TSR respectively. All options have been tested with the applied ones (Table 4.3 and Table 4.4) appearing as the most significant.

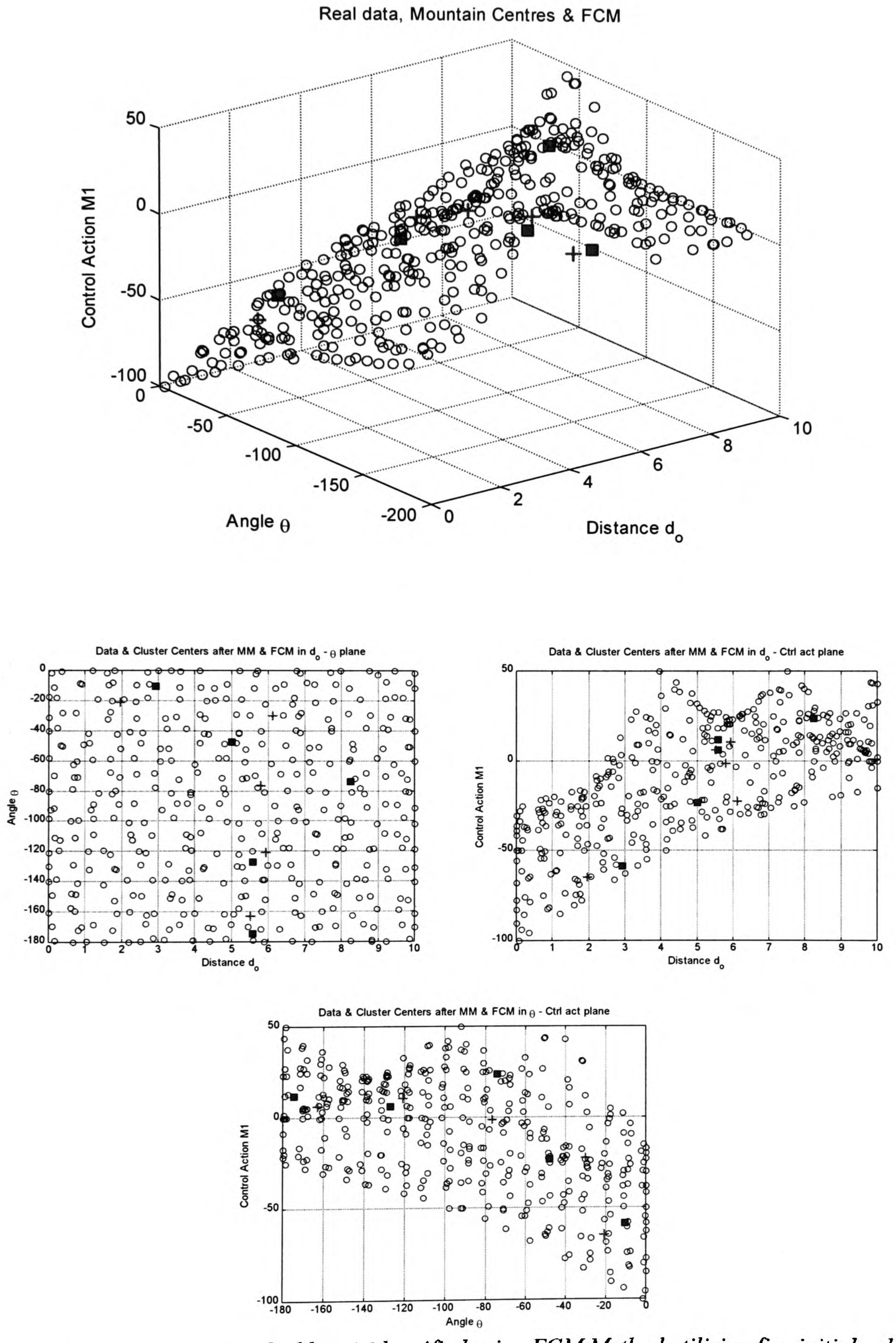


Figure 4.17 Prototypes (marked by +) identified using FCM Method utilising five initial values resulting from the mountain method (marked by ■)

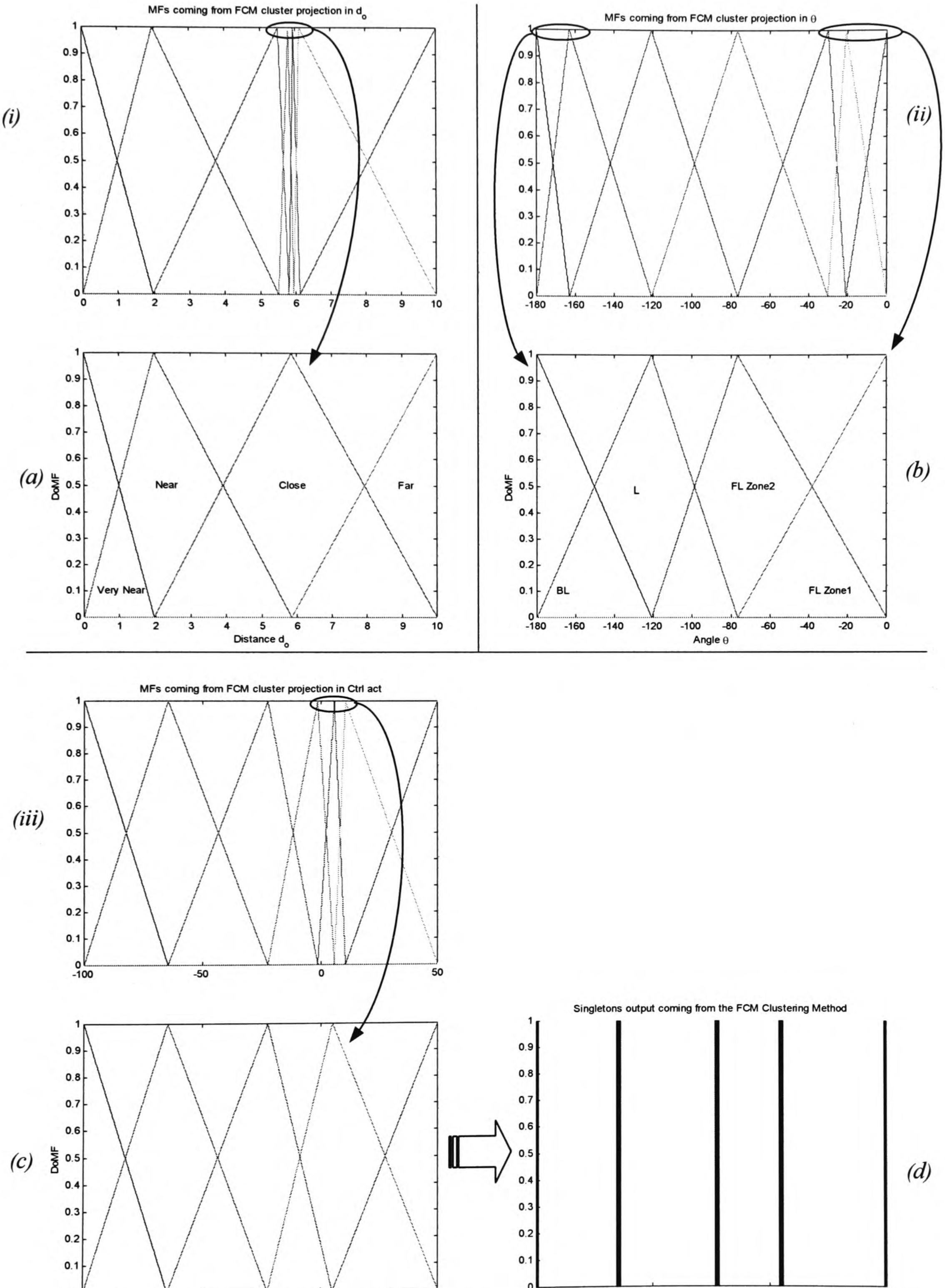


Figure 4.18 Membership functions and Singletons generated initially from the projection of cluster prototypes and their ellipsoids, and finally after merging method

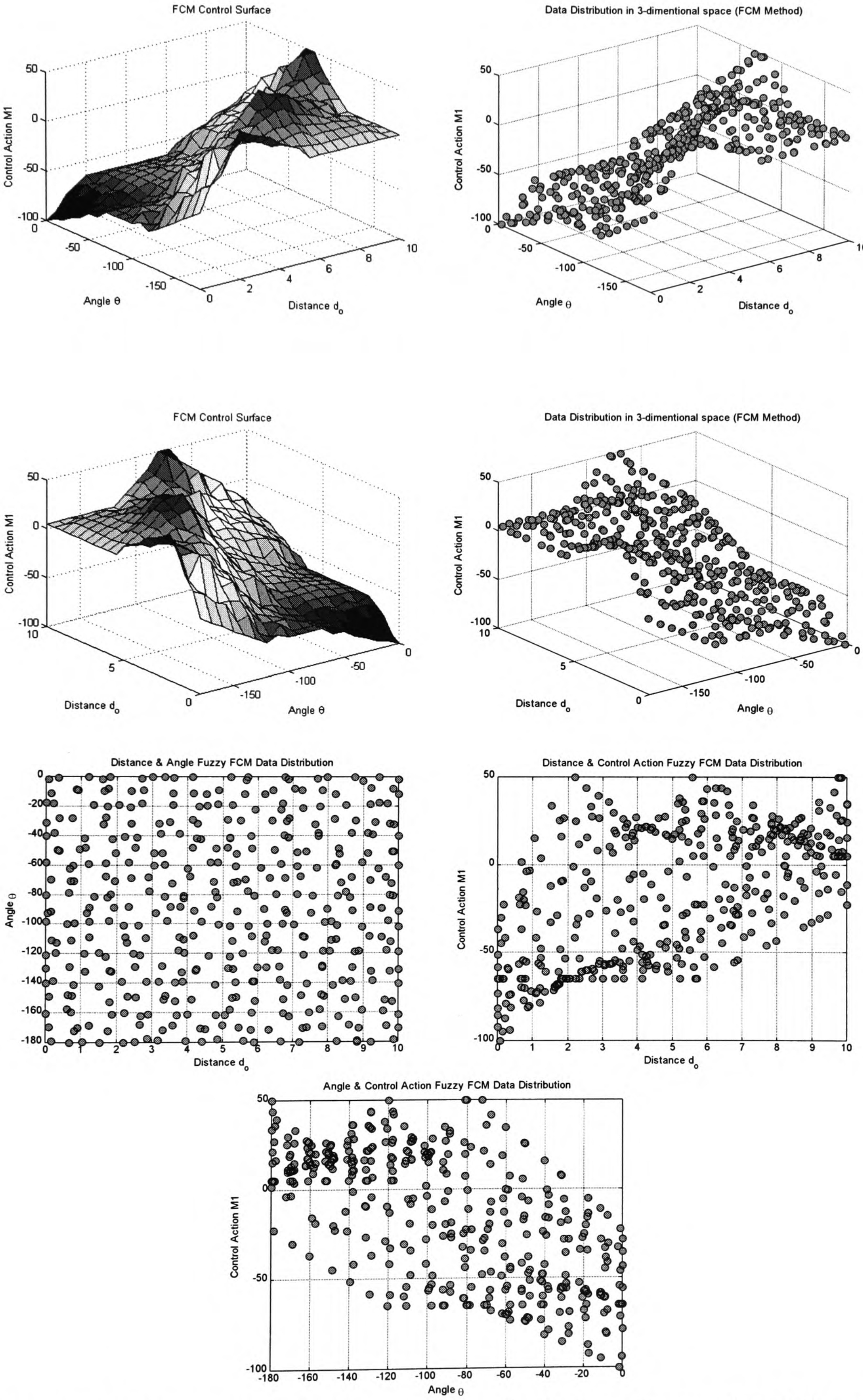


Figure 4.19 Surface and data distribution generated by the Fuzzy System using FCM method

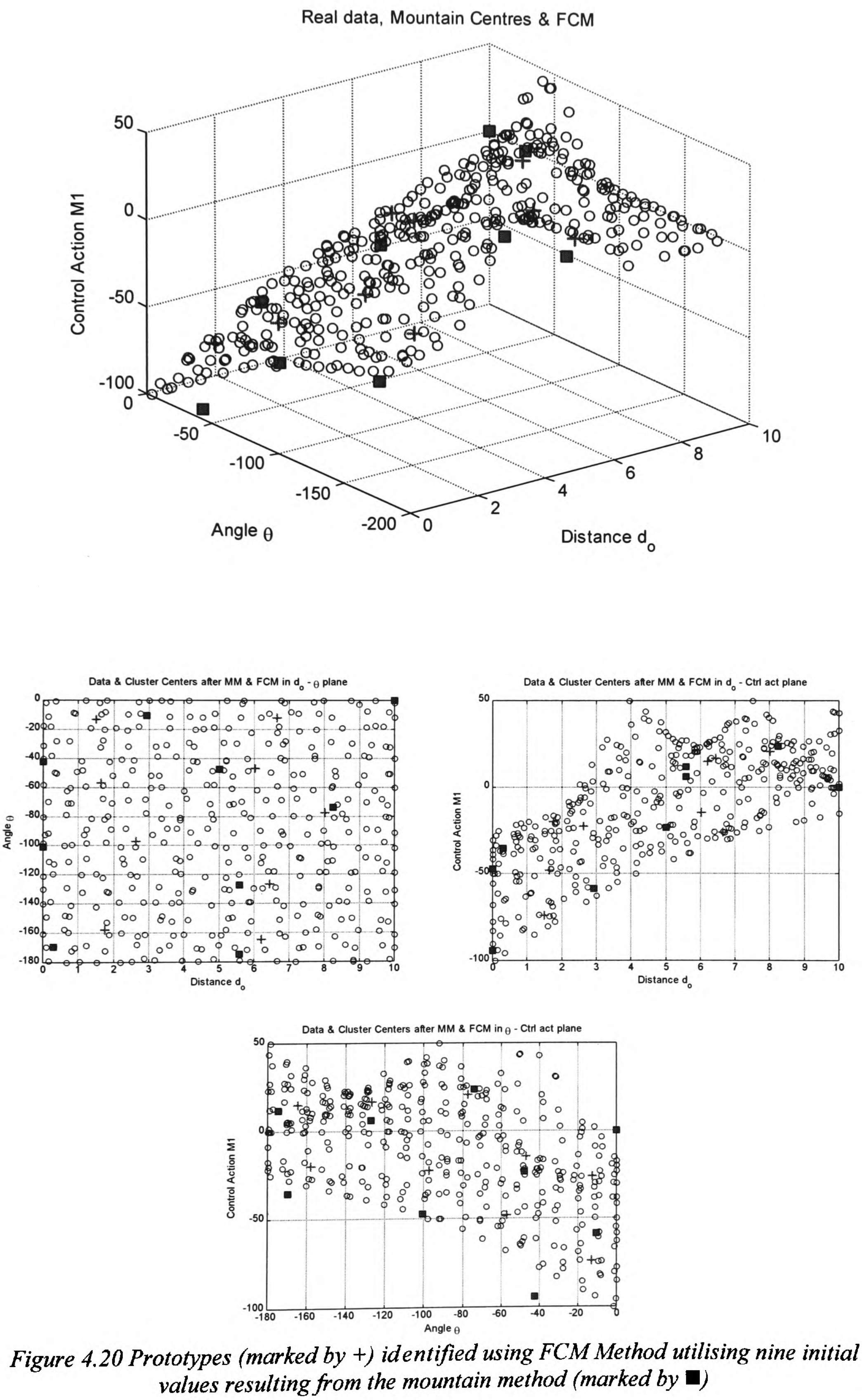


Figure 4.20 Prototypes (marked by +) identified using FCM Method utilising nine initial values resulting from the mountain method (marked by ■)

(i) (ii)

(a) (b)

(iii)

(c) (d)

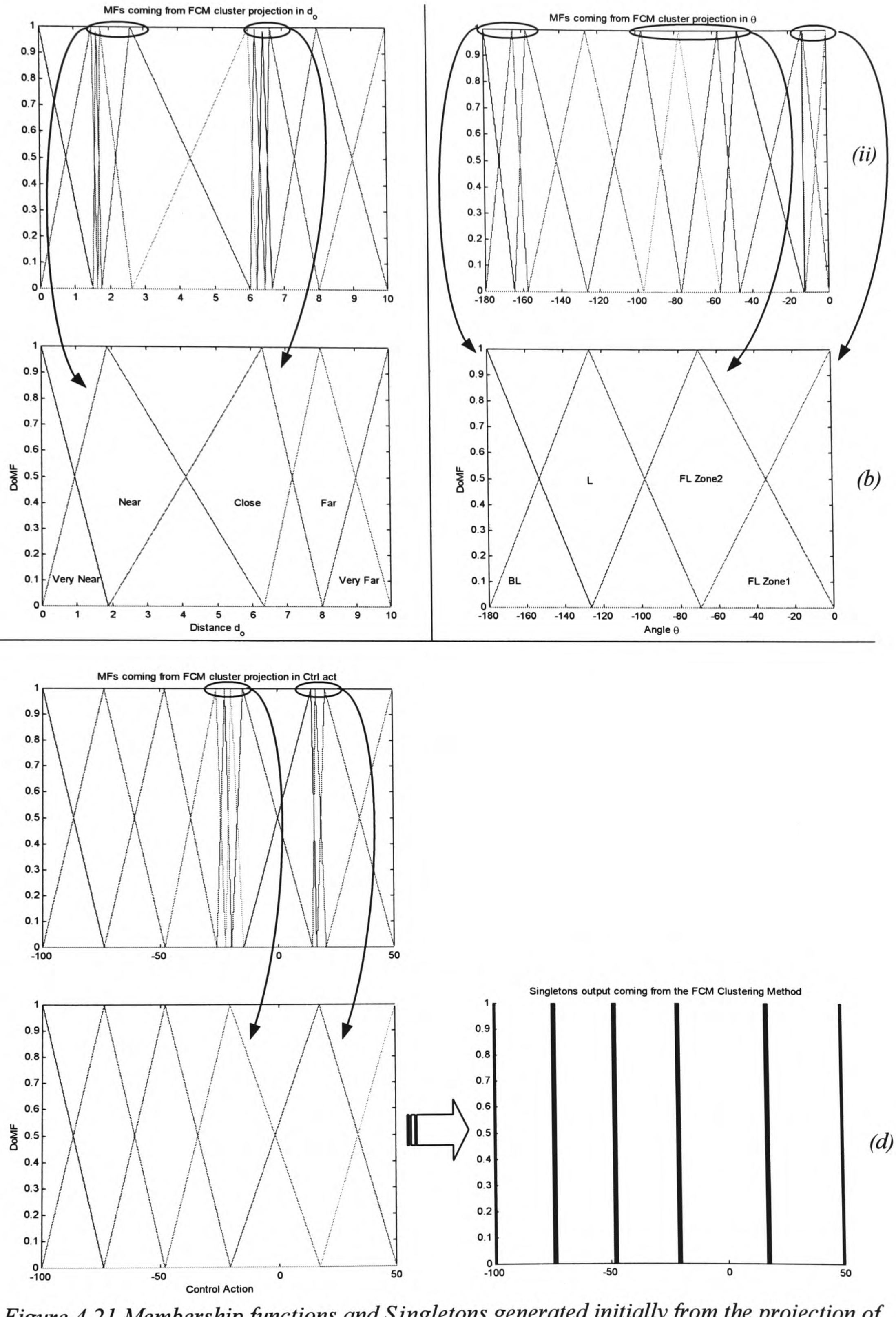


Figure 4.21 Membership functions and Singletons generated initially from the projection of cluster prototypes and their ellipsoids, and finally after merging method

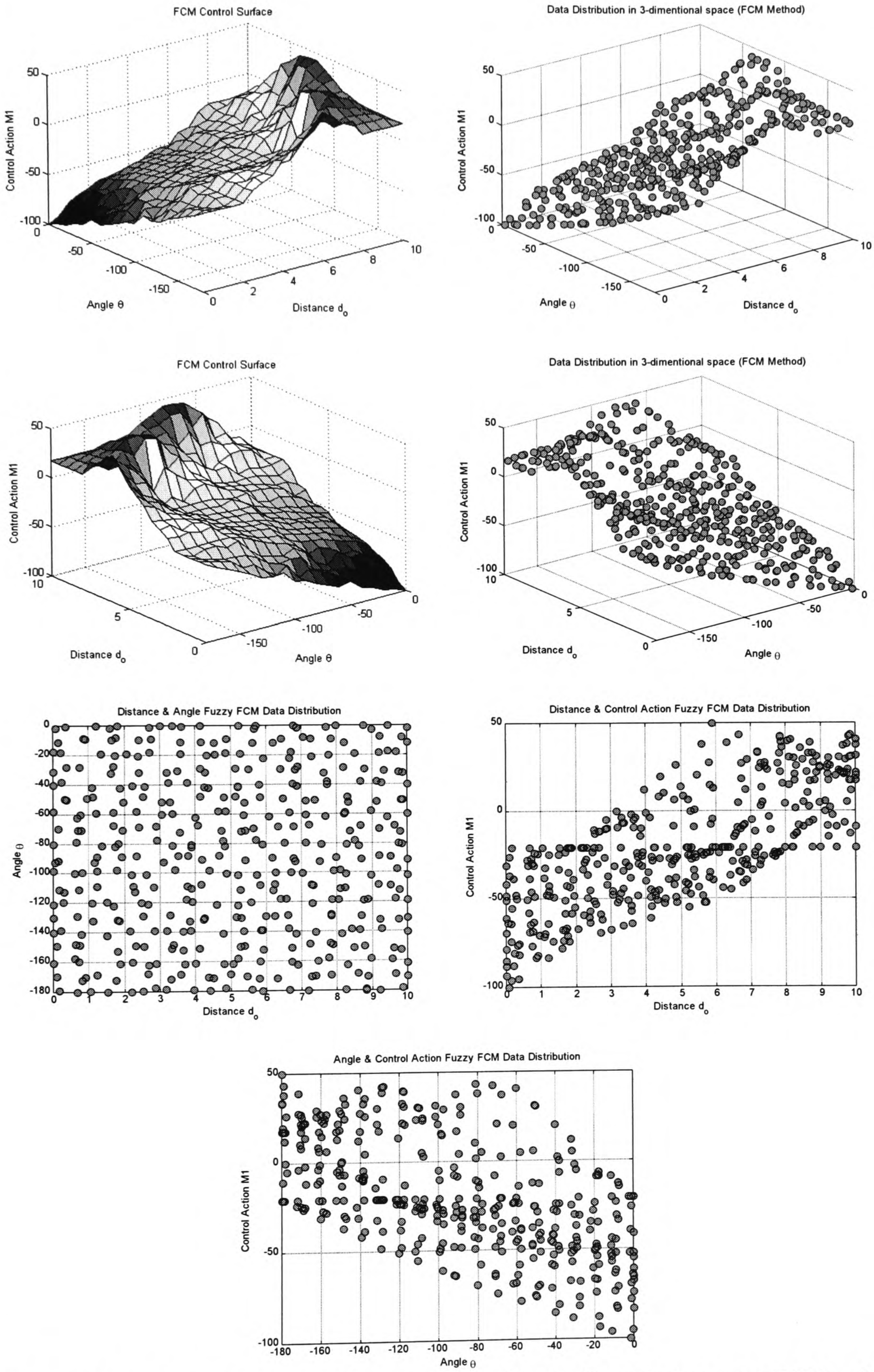


Figure 4.22 Surface and data distribution generated by the Fuzzy System using FCM method

4.3 Discussion

As mentioned in the introduction of this chapter (Section § 4.1) the case study is to identify a control strategy to avoid objects for GARBI underwater vehicle based on the operator's control actions data. In other words the objective of this case study is to identify fuzzy rules that the operator could use to construct the control action as in Figure 4.2 using the proposed methodology as discussed in Section § 4.2. Discussion of some of the applied steps of the method as well as the results is presented in this Section.

The graphical representation of the mountains using the mountain function assists the designer to make decisions about how many clusters should be considered for further analysis. Thus, executing the algorithm, a large number of clusters were defined initially but only the first nine were considered as the rest of them had very small and thus insignificant heights.

The proposed method splits after the 3rd (a) step, of the proposed methodology described in Section § 3.11, in three different cases i.e. applying a) Gustafson-Kessel method (Section § 4.2.3), b) FCM considering five cluster prototypes (Section § 4.2.4.1) and c) nine cluster prototypes (Section § 4.2.4.2). The resulting control surfaces in the first (see Figure 4.15), second (see Figure 4.19) and third (see Figure 4.22) case are very satisfactory as they are very close to the original ones in Figure 4.3 with $APE = 0.0586\%$, $APE = 0.0789\%$ and $APE = 0.0817\%$ respectively. The most satisfactory result is that which results from using the Gustafson-Kessel method, due to its ability to optimise clusters with non-fixed topological structure on n-dimensional space. Note that the representation of non-linear data distributed as in Figure 4.2 is closer to ellipsoid than spherical shapes. However, even when the FCM was used as the cluster method, the result was still satisfactory in both cases of 4×4 (Table 4.3) or 5×4 (Table 4.4) rule tables. Therefore, some flexibility exists in terms of the choice of the used methods listed in the proposed methodology.

By applying the gradient method in the output singletons, the performance is improved with $APE = 0.0519\%$. However, this improvement it is not considered very high regarding the time consumed for running the algorithm as well as the risk of the singletons losing their linguistic meaning. Moreover, the gradient method algorithm was applied also to the input sets and to both input and output sets. In both cases the degree of modifications to some of the membership functions is very high losing therefore their initial linguistic meanings. Furthermore, the resulting control surface is not significantly improved.

From the above analysis of the results, therefore, it can be remarked that by using clustering methods the generated membership functions are close to the "original" ones that the system could originate from. Additional optimisation improvements such as gradient method should be applied only in the consequent part of the rules and only if the APE improvement is significantly big.

The data was normalised when the G-K algorithm was applied since it uses an adaptive distance measure method and thus does not influence the result of clustering. However, when applying the FCM algorithm the data was not normalised due to the drawback of using Euclidean distance as discussed in Section § 3.8.

Finally note that the control surface defined in Figure 4.3 is an approximate one as the input output data are not exactly distributed equally (see Figure 4.2). Positioning the input data into equal points and the output data accordingly therefore construct the surface (Figure 4.3). However, the only use of this surface is just to compare visually the generated surfaces (Figure 4.15, Figure 4.16, Figure 4.19, Figure 4.22) with the initial one (Figure 4.3).

4.4 Summary

In this chapter the proposed systematic methodology to construct an identification method of fuzzy control strategies based on availability of input/output mapping data is applied in the "avoid object" problem for an underwater vehicle. The steps of the method are consequently a) collection of the simulating data of the control action, b) definition of the number and the initial approximate positions of the cluster prototypes using mountain method, c) used of Gustafson-Kessel and FCM method to obtain the actual positions of the cluster prototypes as well as their variance using a case of five and nine clusters, d) projection and merging of the generated fuzzy sets, e) construction of the rule table and finally f) use of gradient method to improve the performance of the resulting fuzzy control system (only in the case of the singletons resulting from G-K method). The results are presented analytically and discussed analysing the success and applicability of the proposed method in modelling of non-linear operating control strategies.

4.5 References

Espinosa, J., and Vandewalle, J 1997. *Fuzzy Modelling and Identification, A guide for the user.* The IEEE International Symposium on Control Theory and Applications. Singapore.

Jang, J. S. R.,Sun, C. T., and Mizutani, E. 1997. *Neuro-Fuzzy and Soft Computing.* Upper Saddle River, NJ: Prentce Hall. 0-13-261066-3.

Jantzen, J., 15 May 1998. "Design of Fuzzy Controllers." Denmark: Technical University of Denmark.

5

The Design & Tuning of Control Systems using Fuzzy Logic & Taguchi Method

5.1 Introduction

This chapter presents an innovative approach to determine the optimal parameters of control systems in terms of robustness and tuning characteristics. The approach is based on the Taguchi Design of Experiment method that uses the minimum number of experiments, analyses the results of each performance criterion, investigates the significance of the parameters/factors of a system as well as factor interactions and their optimal levels with regards to their robustness. A design method of how to construct the fuzzy rules to tune the factor levels is also proposed. A method called a FCSS is then used to define the weights of the factor levels, which are related to the performance characteristics of the system. As an example the proposed approach is applied to the optimisation of parameter tuning used in fuzzy-like PD controllers. The qualitative and quantitative analysis of the interaction between the tuning parameters is usually not considered during the design of FLCs due to their complexity. This chapter introduces methods to overcome this difficulty. The results are satisfactory in terms of improvement in IAE and ITAE performance indexes. The proposed synergetic approach can be used as a general approach for any parameters of the FLCs, which may be of interest to be studied as tuning factors.

Fuzzy logic controllers (FLCs) have been one of the most successfully used controllers in a large number of complex and non-linear systems (Sugeno, 1985). The design of FLCs is based on the idea of mimicking the control actions of the human operator, which are implemented as a collection of *If-Then* control rules that are directly encoded into a control algorithm. In this case, *a priori* knowledge from experts is used and the final controller performs nearly as well as the operator. The performance of this type of controller is successful where the system is non-linear, partly known and difficult to describe by a "white box" model. However, the method is highly dependent on expert knowledge and sufficient knowledge of the operator(s), which may be problematic, since the human control skills can be difficult to verbalise or explain reasonably. Moreover, when the system is non-linear the parameters of the FLC need to be robust and tuneable. Therefore, a systematic way to design and/or modify FLCs needs to be applied. An approach is proposed in this thesis based on design of experiments and their analysis, including studying the results of the set performance criteria. This analysis is used to assist an analytical methodology to optimise the tuning parameters of a FLC with respect to their significance, their interacting effects and their robustness characteristics.

Robustness is a key aspect in fuzzy logic type controllers. Realising that exact knowledge of non-linear systems is almost never available, robust control methods take into account the uncertainty and disturbances of the system, measured by the performance description and try to make the controller less sensitive to parametric variations. In a FLC system, robustness reflects the ability to resist disturbances or abrupt chances in a fuzzy input. A number of factors may affect this property, including *the control resolution, the partition of the input space that is related to the Scaling Factors (SFs), the number of control rules, and the membership functions of the fuzzy sets*. Since the control resolution and the number of control rules are often determined by other design objectives, they can be difficult to use for improvement in robustness. This leaves modification of the *partition of the input space* of the *SFs* and of the *fuzzy membership functions'* factor levels as the most readily accessible approaches (Yan *et al*, 1994).

Tuning a FLC system is a fundamental problem due to its design structure, which involves a large number of degrees of freedom. Fuzzy and/or non-fuzzy logic techniques can be used for the tuning of fuzzy controller's (He and Xu, 1993). Due to the difficulties in building a tuning controller for non-linear and/or high-order systems, a robust tuning scheme for FLC's is needed, which would be applicable irrespective of the nature of the processes and the structure of the FLCs. Its overall design may include two levels of controller tuning. The first level of tuning is the determination of the shape and the number of membership functions, T and S norms, aggregation methods. The second level is the tuning of equivalent gain parameters. This includes the input and output SF's and other gains used in building the structure. Therefore, FLCs have no fixed design structures like conventional PI, PD, and PID controllers. However, its large number of degrees of freedom gives high flexibility to choose the tuning factors for the fuzzy system, and also demands systematic criteria to define the most suitable tuning parameters.

The proposed approach presented in this thesis is based on the synergy between the *Taguchi Design of Experiments method* together with the *Fuzzy Combined Scheduling System* approach. The technique of laying out the design of experiments involving multiple factors/parameters was first proposed by Ronald (Fisher, 1935). The method is popularly known as *factorial design of experiments*. A full factorial design will identify all possible combinations for a given set of factors. When a significant number of factors is used, a full factorial design results in a large number of experiments. This is a time consuming procedure and in order to reduce the number of experiments to a practical level, only a small set from all the possibilities is selected. The method of selecting a limited number of experiments, which produces the most information, is known as a *fractional factorial experiment*. Although this method is well known, there are no general guidelines for its application or the analysis of the results obtained by performing the experiments. Taguchi envisaged a new method of conducting the design of experiments, which are based on well-defined guidelines. This method uses a special set of arrays called orthogonal array. These standard arrays stipulate the way of conducting the minimal number of experiments, which will give information on the significance and interaction of inhomogeneous factor levels (or tuning parameters of a FLC) that affect the performance characteristics of the system. The importance of each factor can be evaluated and defined according to the measured responses of the system's performance characteristics. The crux of the orthogonal array method lies in choosing the level combinations of the input design parameters for each experiment. A *fuzzy combined scheduling system* is then used to define the weights of the factor levels in relation to the performance characteristics of the system.

An investigation of how to optimise and tune the factors/parameters of fuzzy-like PD controllers is used as a case study in this work. A large number of fuzzy-like PD controllers have been developed (Mann and Gosine, 1999). These controller designs are based on the combination of Fuzzy Logic flexible systems and the well-known conventional PD structure (Astrom and Hagglund, 1995). There are no general methods for optimising and tuning the parameters of Fuzzy-like PD controllers, although several approaches have been proposed. Jantzen, (1998) uses the idea of starting with a tuned conventional PD controller and replaces it with an equivalent linear fuzzy controller. Some proposals are based mainly on the conventional PD controllers tuning criteria (Yager and Filev, 1994b), (De Silva, 1995). (Zheng, 1992) introduces a practical guide to tune SF's, peak values, width values and rules. Mudi and Pal, (1999) uses only the output SF for tuning. There are some basic rules for tuning, derived by

Procyk and Mamdani, (1979). The most successful results are based on the combination of good experimental understanding of the controlled system and the use of the analogies between the FLC and PID controllers.

The rest of the chapter is divided into four Sections. In Section § 5.2, the proposed design and analysis of the FLC's parameters is presented. Analytically, in Sections § 5.2.1 and 5.2.2 the performance criteria as well as the independent parameters/factors are defined respectively. In Section § 5.2.3 the selection of the interactions of the factors that may influence the performance characteristics of the systems under study is discussed. The possible tuning control factor levels are defined in Section § 5.2.4. The number of experiments and the selection of the appropriate orthogonal array are defined in Section § 5.2.5. Moreover, in Section § 5.2.6 how to assign the factors to columns of the orthogonal arrays and the locating interaction columns is discussed. In Sections § 5.2.7 and § 5.2.8 the conduction and analysis of the experimental results is reviewed. In Section § 5.2.9 the optimal factor levels are verified. The way to construct the fuzzy rules to tune the factor levels is proposed in Section § 5.2.10. The fuzzy combined scheduling system approach is discussed in Section § 5.2.11. In Section § 5.3 some simulation studies are used to define the application of the proposed method. The results are discussed analytically in Section § 5.4. Finally the summary of the chapter is presented in Section § 5.5.

5.2 Steps in the design and analysis of the FLC's parameters (Factors) identification

A proposed systematic methodology for the identification of the tuning parameters (factors) of a FLC system is presented in this section based on a *systematic analysis of experimental results* together with the *Fuzzy Combined Scheduling System* approach. The proposed method involves the following steps:

1. *Definition of the performance criteria*
2. *Definition of the independent parameters/factors*
3. *Selection of interactions that may influence the performance characteristics of the systems*
4. *Definition of the possible tuning control factor levels*
5. *Definition of number of experiments and selection of an orthogonal array*
6. *Assigning the factors to columns of the orthogonal arrays and locating interaction columns*
7. *Conducting the experiment - replications as described by the trials in the orthogonal arrays*
8. *Analysis of the resulting data to determine the optimal factor levels*
9. *Verification (confirmation) of the optimal factor levels*
10. *Constructing fuzzy rules to tune the factor levels*
11. *Applying the Fuzzy Combined Scheduling System approach to define the weights of the factor levels*

The above steps are discussed in detail in the following sections:

5.2.1 Definition of the performance criteria

In order to apply robust and/or tuning controller, its control strategy should be able to assess its own performance. There are two ways to measure these performances based on either local or global criteria. Local criterion measures the performance over a small set of plant states whereas the global one measures the overall performance. The problem in using a global criterion is firstly to choose an appropriate figure of merit (criterion) and secondly to relate a change in this figure of merit to a set of control actions that caused it. To overcome this, local criteria are chosen which make the problem of assigning credit to individual control actions easier and in the hope that an improvement in the local performance will also improve some global criterion. Thus the way that the performance is measured is critical in producing a successful implementation. The performance state vector $p_S = \{p_{S_1}, p_{S_2}, ..., p_{S_m}\}$ may be defined by direct measurements or by indirectly estimating its value via other related measurements(Mudi & Pal, 1999). Finally the designer may define the space or the boundaries of values that the optimal performance criteria vector $p_O = \{p_{O_1}, p_{O_2}, ..., p_{O_m}\}$ is required to be within.

5.2.2 Definition of the independent parameters/factors

Before conducting the experiments, knowledge of the system under investigation is of prime importance for identifying the factors' vector $f = \{f_1, f_2, ..., f_j\}$ likely to influence the outcome. These are the controllable factors that excite the performance criteria and can be set by the expert. For example, the influence in terms of step response characteristics in FLCs comes mostly from the SFs gains and/or the peak values of Membership Functions (MFs) which are known as the tuning factors (Yager and Filev, 1994b). Moreover, uncontrollable (or noise) factors may occur, which are sources of variations often associated with the operational environment. Overall performance should, ideally, be insensitive to their variation. One of the objectives of Taguchi's approach is to identify these controllable factors that minimise the variation in systems response while keeping the mean response on target (see Sections § 5.2.7 and § 5.2.8).

5.2.3 Selection of interactions that may influence the performance characteristics of the systems

When the effect of one factor depends on the effect of another factor an interaction is said to exist (Fowlkes, 1995). In other words, an interaction occurs when the collective effect of two (or more) factors taken together is different from the sum of each of the factors taken individually. When such an effect occurs, it becomes difficult to predict the effect of a factor (and its level) selection. However, using orthogonal arrays, the interactions can be dealt with.

Moreover, if no decision can be made about which interaction to study, then all of them (in pairs of the factors) should ideally be investigated. However, including interactions in the design requires more experiments and makes the analysis more complicated.

5.2.4 Definition of the possible tuning control factor levels

Once the choice of the factors that may be used to tune the controller is made, the number of possible tuning levels for each of these factors has to be defined. The selection of this depends on how a performance parameter is affected due to those different level settings. If the performance parameter is a linear function of these Possible Tuning Factor Levels (PTFLs), then the number of levels should be two. However, if it is not linearly related, a higher number of the PTFLs is needed. The latter is the most common case in real systems due to inherent non-linearity. These level values form the vector $l_{f_{pt}} = \{l_{f_{pt}1}, l_{f_{pt}2}, \ldots, l_{f_{pt}n}\}$, which is defined by a collection of local PTFLs in a manner where each of the collected levels approximates locally the optimal levels around different performance states. The PTFLs are usually defined as *min, max* and some *intermediate* values that excite the response of the system $l_f \in [f_{\min}, f_{\max}]$. When the number of these local levels increases, the neighbouring levels are closer and the shifting trajectory between them is faster and smoother. The determinations of which PTFLs are the optimal ones to be tuned come from the analysis of the experimental data for different performance criteria using the Taguchi method as will be explained in Section §5.2.8. Note that with this analysis, the factor levels may not be defined in terms of their actual optimal values but their direction to the subspace that they belong.

The current factor level vector $l_{f_c} = \{l_{f_c1}, l_{f_c2}, \ldots, l_{f_cn}\}$ is comprised of the present levels' values. The *fuzzy combined scheduling system* method defines the weights of the levels related to the performance criteria characteristics of the system. This will be explained extensively in Section §5.2.11.

5.2.5 Definition of number of experiments and selection of an orthogonal array

Orthogonal arrays are used in the Taguchi Method for designing efficient experiments and analysing experimental data (Roy, 1990). A primary advantage of orthogonal arrays is that for each level of any one factor, all levels of the other factors occur an equal number of times. Another advantage of orthogonal arrays is their cost efficiency since the design of an orthogonal array does not require that all combinations of all factors be tested. So, the experimental matrix can be smaller without losing any vital information.

To design an experiment is to select the most suitable orthogonal array, assign the factors to the appropriate columns, and finally, describe the combinations of the individual experiments. To select the most appropriate orthogonal array the number of experiment has to be defined. This number has to be less than or equal to the number of experiments, set by one of the standard Taguchi orthogonal arrays (Roy, 1990). If no interaction between the factors need to be analysed, then the minimum number of experiments is calculated based on the formula in Equation 5.1.

$$N_{Exp} = 1 + \sum_{i=1}^{j} (l_i - 1) \tag{5.1}$$

where N_{Exp} is the number of experiments, j is the number of factors, and l_i is the number of levels for each factor f. $l_i - 1$ measures the *degree of freedom (DOF)*, which is the number of independent measurements available to estimate sources of information. The number of degrees of freedom indicates the number of independent comparisons that exists within a set of data.

However, if interaction between two or more factors need to be analysed, then the number of experiments may increase. If the number of columns are inappropriate, then a larger orthogonal array with more experiments (rows) and columns has to be chosen. It is not necessary to fill all the columns of the orthogonal array due to its balancing properties and orthogonality. Thus the product of the degrees of freedom gives the number of experiments for each of the interacting factors. Therefore Equation 5.1 is expanded to include the interactions as in Equation 5.2.

$$N_{Exp} = 1 + \sum_{i=1}^{j} (l_i - 1) + \sum_{j=1}^{I_n} \prod_{1}^{fi_n} (l_j - 1) \tag{5.2}$$

where I_n is the number of interaction that needs to be analysed, fi_n indicates how many factors are analysed for each interaction. Note finally that the unit in both formulas (Equation 5.1 and Equation 5.2) is the degree of freedom associated with the overall mean regardless of the number of control factors to be studied.

When the number of experiment has been defined, the appropriate orthogonal array, described by the symbol L_{row}, can be obtained from experimental design textbooks (Phadke, 1989), (Fowlkes, 1995).

5.2.6 Assigning the factors to columns of the orthogonal arrays and locating interaction columns

As illustrated in Figure 5.1, the orthogonal array has *r* rows and *c* columns. Each row represents a trial condition with factor levels indicated by the numbers in the row. The columns correspond to the factors specified in the study. The maximum number of factors that can be studied in any one experiment is equal to the number of columns in the orthogonal array used. The order in which the factors are assigned to the column is very important (Taguchi, 1987).

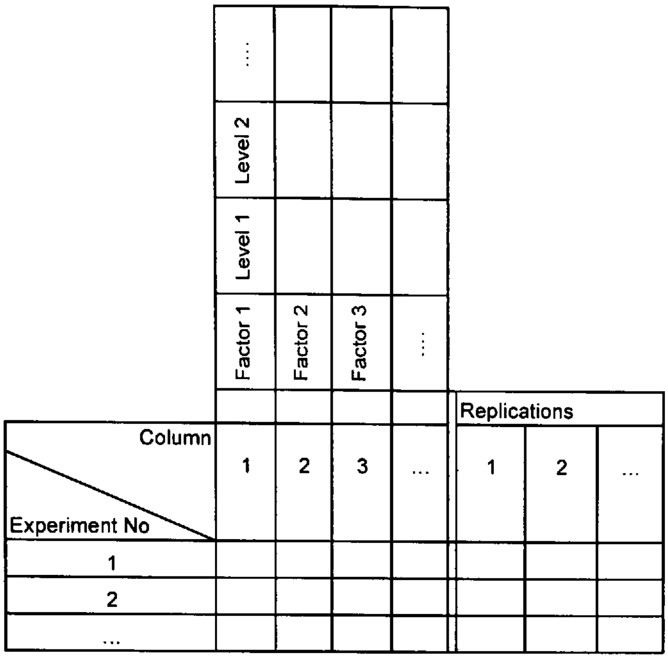


Figure 5.1 The layout of an orthogonal array

Table 5.1 summarises the most widely used orthogonal arrays and the maximum number of factors that can be used in each array compared to the full factorial design. For instance, an L_{27} orthogonal array (27 experiments) has the capacity to accommodate 13 factors at 3 levels, whereas in a full factorial study, over 1.5 million (3^{13}) experiments need to be undertaken.

Orthogonal Array	Maximum Number of Factors	Number of Trials in Full Factorial Experiment
L_4	3 x 2 Level	8
L_8	7 x 2 Level	128
L_9	4 x 3 Level	81
L_{12}	11 x 2 Level	2,048
L_{16}	15 x 2 Level	32,768
L_{18}	1 x 2 Level & 7 x 3 Level	4,374
L_{27}	13 x 3 Level	1,594,323

Table 5.1 Some examples of orthogonal arrays compared to the full factorial experimental design

5.2.7 Conducting the experiment - replications as described by the trials in the orthogonal arrays

Once the orthogonal array has been selected, the experiments are conducted as identified by the level combinations of the array. If noise is included then the optimum conditions, insensitive to the influence of the noise can be found. This results in an expansion of the design since during each trial, more than one observation may be estimated; these are then called *replications* or *replicates*. Replication is a primary tool in analytical methods for studying stability and robustness of factor effects and for increasing the degree of belief in the results. It also plays a primary role in providing a measure of the magnitude of variation in the experiments due to noise variables. Moreover, replication helps to minimise the impact of noise variables on factor effects by enabling noise effects to be averaged out. Replication of trials is constrained by time and cost. It is necessary that all the experiments of the chosen orthogonal array be conducted.

5.2.8 Analysis of the resulting data to determine the optimal factor levels

Analysing the data is a systematic straightforward procedure that allows determining the optimum values of the control factor levels (including the interacting ones), and predicting the performance under these levels. The optimal values are based on the performance characteristics, which is the object of interest of the system response. A performance characteristic can be classified according to its target value: nominal-the-best, smaller-the-better, larger-the-better.

5.2.8.1 Analysis of means

An *average response* (Analysis of means) for each factor level is used when the variance of the results of the replication is small. This comes from the following formula:

$$\overline{F}_l = \frac{\sum_{i=1}^{q} R_i}{q \cdot l_{NoE}} \tag{5.3}$$

where $\overline{F}_l$ is the average effect of factor F at level l, R is the result of each trial i while factor F is at level l, q is the number of trials and l_{NoE} is the number of experiments for each factor level l.

However when variations exist, the level(s) of the control factors that may contribute to reduce this variation can be identified, by Taguchi's use of Signal to Noise (S/N) ratio which reflects the amount of variation present. The S/N ratio is an objective function that takes both the effects of the mean and the variation into account. The S/N ratio is treated as a response of the experiment, which measures the level of each factor against the level of noise factors. The S/N ratio is therefore, a very useful way to evaluate the robustness of the system under study. Better performance, as measured by a high S/N ratio implies a smaller loss. There are several *S/N ratios* (η) available depending on the type of performance characteristic chosen. The *smaller-the-better* S/N ratio is defined by the following formula (Roy, 1990):

$$\eta = -10\log_{10}\left[\frac{1}{q}\sum_{i=1}^{q} R_i^2\right] \tag{5.4}$$

The average response and/or S/N ratio of the level of each factor can be plotted for a visual interpretation and is explained in Section § 5.2.8.3. Analysing graphically the average response and taking into account the type of performance characteristic for a particular system, the optimal values of the factor level can be investigated accordingly (Phadke, 1989). However, lower deviation is always indicated by a higher S/N ratio value and is used to investigate the optimal factor levels despite the performance characteristic that has been chosen as the objective function for the particular system (Roy, 1990). This is because the goal of the experiment for a 'smaller the better' situation is to minimise $\left[\frac{1}{q}\sum_{i=1}^{q} R_i^2\right]$, which is accomplished by maximising η in Equation 5.4.

5.2.8.2 Analysis of Variance (ANOVA)

Taguchi replaces the full factorial experiment with a lean, less expensive, partial factorial experiment. Taguchi's design for the partial factorial is based on specially developed orthogonal arrays. Since the partial experiment is only a sample of the full experiment, the analysis of the partial experiment must include an analysis of the confidence that can be placed in the results. Fortunately, there is a technique called Analysis of Variance (ANOVA)(Fowlkes, 1995), which is routinely used to provide a measure of significance of each factor and any interaction effect together with their percentage contribution. The method does not directly analyse the data, but rather determines the variability (variance) of data or variance ratio called the *F-ratio*. This ratio is used to measure the significance of the factor under investigation with respect to the variance of all the factors included in the error term. The *F*-value obtained in the analysis is compared with a value from standard Fisher's tables (*F*-tables) for a given statistical level of significance. The tables for various significance levels and different degrees of freedom are available in most statistics textbooks (Chatfield, 1983), (Roy, 1990). A more detailed explanation of the mathematical formulas that are used to define the *F*-ratio and the percentage contribution may be found in (Chatfield, 1983). These analyses provide the variance of controllable and noise factors. By understanding the sources and magnitude of variance, robust operating conditions can be predicted. This is one of the benefits of the Taguchi methodology.

5.2.8.3 Graphical representation to present the optimal levels and analyse the factor's interactions

The effects and the interactions of factors used in the experiments can be presented graphically, estimated from the average values of the response variable at the different levels of the factor. This is defined by using a two-dimensional table from the observed data. To plot, for instance, the interaction graph between two factors $A \times B$ the table shown in Table 5.2 is prepared.

		Level of Factor B	
		B_1	B_2
Level of Factor A	A_1	$\frac{\sum_{i=1}^{2} y_{A_1B_1}}{2}$	$\frac{\sum_{i=1}^{2} y_{A_1B_2}}{2}$
	A_2	$\frac{\sum_{i=1}^{2} y_{A_2B_1}}{2}$	$\frac{\sum_{i=1}^{2} y_{A_2B_2}}{2}$

Table 5.2 Two-dimensional table used to calculate the average value of the response variable at the different levels of each factor A, B

In Table 5.2 the rows correspond to the levels of factor *A*, the columns correspond to the levels of factor *B*, and entries correspond to the *average* or *S/N ratio* response for the particular responses of the combinations of levels *A1, A2, B1, B2* of the factors *A* and *B*. Finally, *2* is the number of levels. Using the results of Table 5.2, the co-ordinates used for the plots are defined (Phadke, 1989).

Using these plots, the detection of the interactions can also be defined. Detecting interaction can easily be accomplished by observing the parallelism of the plots as described in Figure 5.2 (Roy, 1990). Note that parallel plots imply no interaction. The measurement of interaction may defined geometrically as described in (Lochner and Matar, 1990).

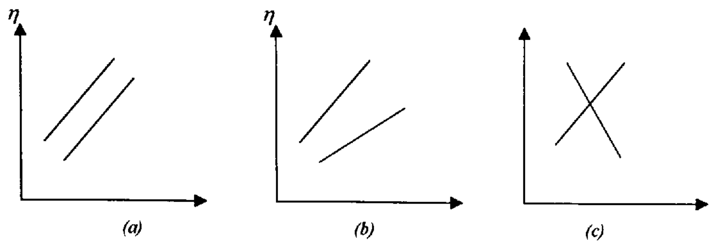


Figure 5.2 Examples of interactions: (a) No interaction, (b) some interaction, (c) high interaction

Note, that during the graphical analysis of these plots, if two or more levels of the same factor have equivalent maximum response their optimal choice is derived according to the following rules:

if the levels belong only to one interaction plot then any one of them can be used

if more than one interaction plot suggest these levels, then the one that belongs to the interaction plot with the highest order of significance is chosen.

if these levels belong to the interaction plot with the highest significance order (ANOVA table) as well as to some other one(s) then the one that is in common should be chosen.

In the cases studies presented in Section § 5.3 it can be seen the applicability of these rules.

5.2.9 Verification (confirmation) of the optimal factor levels

In the analysis of the results described in Section § 5.2.8, for each performance criterion, all the PTFLs are analysed in terms of their performance, significance and robustness and ranked according to their importance. Therefore, the Optimal Tuning Factor Levels (OTFLs) are defined around different performance criteria status and the rules can be easily constructed based on these levels. However, a new experiment with the OTFLs should be conducted to verify their optimum performance. If the performance is drastically different from the predicted one then there is high interaction among the parameters that have not been considered and therefore interaction analysis should be reconsidered.

5.2.10 Constructing fuzzy rules to tune the factor levels

When the OTFLs of the systems parameters are defined their robustness in the performance is obtained (Phadke, 1989). However, if the system's parameter changes and/or the system's environment become non-linear, construction of fuzzy rules is required to tune the factor levels. This construction of the fuzzy rules is designed considering the performance criteria related to the *tuning* and *current* factor levels. However, during the design of the rules some important aspects that have to be considered are proposed:

- *Consistency of the control rules*

  When performance criteria contradict each other an inconsistency may occur. In this case two rules may suggest diverse action and it is necessary to compromise between the rules by taking into account the importance of the criteria. However, if all of the tuning rules are equally important the resulting output will respect all of these rules.

- *Homogeneity of the control rule*

  In some cases two or more different performance criteria implies almost the same state. This means that in the analysis of the results the same factor levels may be considered more than once. This may affect the tuning action as the product of the rules may be more biased in that particular state. Therefore in this situation only one performance criterion should be considered.

- *Priority of the tuning factor levels*

  If there are large number of factors, the tuning of factor levels should be arranged according to their significance ranking, which is determined in the analysis of data considering only the first k important factors and their corresponding levels at a given instant in each tuning cycle (De Silva, 1995). It is important to note however that in some cases, factor(s) may have low significance but high interaction with other highly important factor(s). In that case the factor with low significance should also be considered for further analysis.

The vector of the OTFLs that is used to tune the factors to optimise each of the performance criterion is defined by $l_{f_{lm}} = \{l_{f_{lm}1}, l_{f_{lm}2}, \dots, l_{f_{lm}j}\}$, where m, j are the number of the performance criteria and the number of the tuning factors respectively. However, different levels from the same factor may be used to determine the global optimal factor level for all performance characteristics. Using the combined scheduling method these levels are combined, respecting their weight characteristics as explained in Section § 5.2.11.

Considering the performance criteria the definition of rules to determine which of the factors that should be tuned are constructed and have the form:

R_i: **IF** $p_{S_m} \neq p_{O_m}$ **THEN** tune *factor* f_1 **AND** tune *factor* f_2 **AND**... **AND** tune *factor* f_k

The antecedent part compares the performance state characteristics with the optimal one, whereas the consequence part defines the updated tuning control factors resulting from the fuzzy combined scheduling system approach.

5.2.11 Fuzzy Combined Scheduling System (FCSS)

The *Fuzzy Combined Scheduling System* uses a discriminant function $w = f(Low_{MF}, High_{MF})$. This function defines the weight vector $w = \{w_{t_1}, w_{c_1}, w_{t_2}, w_{c_2}, \dots, w_{t_j}, w_{c_j}\}$ of both *current* and *tuning* factor levels constructed from the weight values of the fuzzy sets of the Performance States. Therefore the construction of the fuzzy rules that defines the relationship between the *Performance State* in the antecedent and the *current* or *tuning* factor levels in the consequence are as:

If *Performance State* is Low **Then** *factor level* is $\underline{l_{f_c}}$

If *Performance State* is High **Then** *factor level* is $\underline{l_{f_{t_m}}}$

Thus, for each performance state output, the degree of the *current* and the *tuning* factor levels are related to the degree of the *Low* and *High* Membership Functions (MFs) therefore defining their weights, accordingly. The shapes of the MFs, as well as their linguistic labels may vary and depend mostly on the set criteria. The overlapping between them is 50% to ensure that both rules are excited for each performance state values. Figure 5.3 shows the setting of these two MFs based on the criterion to minimise the value of a performance state. Therefore, when the performance state increases, the weight of the *current* level should reduce, whereas the weight of the *tuning* factor level should increase accordingly and vice-versa.

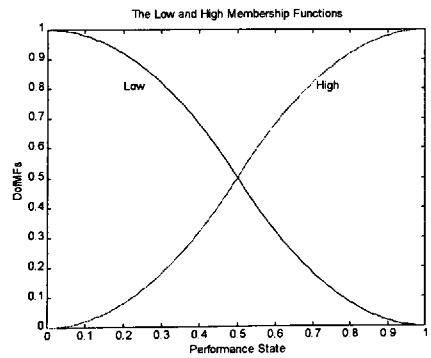


Figure 5.3 The Low and High performance state membership functions where the shapes are Z and S respectively with 50% overlap.

If the discriminant function is such that $w \in [0,1]$ and more than one w is different from 0, then the transition from one factor level to another will be smooth. Therefore, each of the overall factor levels will be optimal even for a non-linear control system, since the weights themselves are functions of the performance states.

The overall optimal factor level vector resulting from the composite *current* and *tuning* factor levels is as in Equation 5.5.

$$L_{oF} = \{L_{oF_1}, L_{oF_2}, ..., L_{oF_j}\} \tag{5.5}$$

where L_{oF_j} is the optimal level for each factor *j*, calculated as the weighted mean (Equation 5.6).

$$L_{oF_j} = \frac{\sum_{k=1}^{m} (w_{t_k} \cdot l_{f_{tm}j} + w_{c_k} \cdot l_{f_c j})}{\sum_{k=1}^{j} (w_{t_k} + w_{c_k})} \tag{5.6}$$

and *m* is the number of the performance states.

In Equation 5.6 multi-performance-criteria is used where more than one objective are maximised and/or minimised. The resulting solution in this case is a single criterion despite the fact that in most multiple criteria approaches a group of solutions is obtained (Ertas and Jones, 1996).

All the performance criteria are assumed to be of equal importance. However, if the significance of each criterion is not weighted equally the fuzzy performance criteria approach can be used for defining the fuzzy multi-criteria decision making (Sousa *et al*, 1999).

The tuning method is based on the availability of both *current* and *tuning* gain values. Section § 5.2.8, explained how to obtain the optimal *tuning* factor levels. However, the *current* gains could imply either the actual SFs values or the representative ones when the dynamics of the system changes.

5.3 Simulation Studies

5.3.1 Structure of Fuzzy like PD Controller

In this section a brief description of the Fuzzy-like PD control is presented. The optimisation and tuning of the parameters of these types of FLC forms the case studies for the proposed approach. The overall structure of the fuzzy-like PD controller is illustrated in Figure 5.4. The inputs of the controller are the error e and the change-of-error Δe. The control output is the variable u. The SFs of the inputs are KE_p, KE_d, and for the output K_u. The output of the FLC is a non-linear function of the inputs e and Δe, and its own SFs.

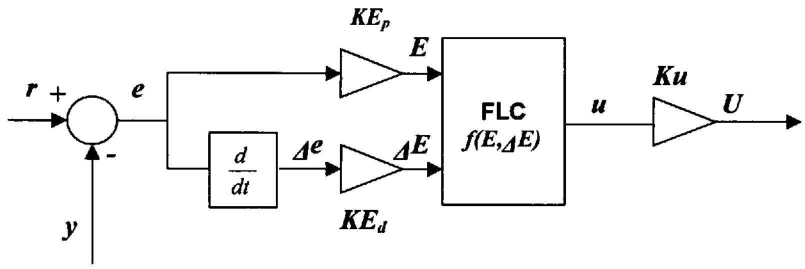


Figure 5.4 Fuzzy-like PD Controller

The MF's for the FLC inputs and output are defined by the common interval [-1 +1]. Thus in the input, triangular MFs are employed (except the ones that are at the extreme ends) with 50% overlap, whereas in the output seven singletons i.e. $[-1,-0.66,-0.33,0,0.33,0.66,1]$ for NB, NM, NS, ZO, PS, PM, PB are applied respectively. When no prior knowledge of the system is available this is the most natural and unbiased choice for MF's since the degree of co-operation between the corresponding rules is equal (Berkan and Trubatch, 1997)

The rule base is presented in table format in Table 5.3 (Reznik, 1997); (Driankov *et al*, 1993). The cell defined by the intersection of the rows and the columns represents a rule such as:

if e is **NB** *and* Δe is **NB** *then* u is **NB**

In the inference engine the product ***and*** is implemented as the ***min*** operation. More information about the design aspect of these types of FLCs can be found in Yager and Filev, (1994b).

e\ Δe	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

Table 5.3 The Rule Base in tabular form

5.3.2 Simulation

The fuzzy-like PD control system, as with other fuzzy control systems, assumes no mathematical model of a controlled process. Without knowledge of the system the optimal tuning parameters are investigated. Limiters employed in the input-output of the FLC avoid saturation.

The performance of the system under study is measured before and after tuning. For a clear comparison between them several performance measures such as percentage overshoot p_{PO}, rise time p_{RT}, settling time p_{ST}, Integral Absolute-Error (IAE) and Integral-of-time-multiplied Absolute-Error (IATE) (Ogata, 1990) are used. The two integral criteria IAE and ITAE are considered because simple visual observations of response curves may not always be enough to make good comparison. In IAE large errors contribute heavily whereas in ITAE large initial errors are weighed lightly and errors occurring late in time are penalised heavily. In this way IAE reflects the transient response and IATE reflects the steady-state response.

Matlab's SIMULINK and Fuzzy Logic Toolbox (MATLAB, 1999) are some of the tools that have been used for the simulation studies. Noise is generated using SIMULINK's Band-Limited White Noise driven by MATLAB's random generator. In each of the examples, a unit step is applied to simulate set-point control due to white noise as well as load disturbances.

5.3.3 Example: Second Order System

In this example a second order process given as:

$$G(s) = \frac{e^{-Ls}}{s(s+1)} \tag{5.7}$$

which represents the position control of an ac motor (Dorf and Bishop, 1995). The saturation range is given as: $u_{\min} = -5$ and $u_{\max} = 25$ (Lee, 1993).

Fuzzy-like PD is the control that is applied to the system. Individual SFs, which may be considered equivalent to the control gains, together with the peaks of the MFs, are the two cases of the parameters that are chosen for optimisation and tuning of the FLC.

5.3.3.1 First case: optimising and tuning SFs

Changing the SFs, changes the universe of discourse and the domain of the membership functions of the fuzzy values of the input/output parameters of the FLC. It has been reported in the literature (Yager and Filev, 1994b) that the performance of the controller is affected dramatically by SF modification. Moreover, the scaling changes even the meaning of the rules in the rule-base of the FLC. Figure 5.5 illustrates the tuning schedule for this type of controller.

As mentioned in Section § 5.1, different approaches have been suggested to define rules for tuning the FLC by manipulating the SFs. Mostly, these rules result in different performance criteria such as percentage overshoot p_{PO}, rise time p_{RT}, settling time p_{ST} by changes in the SFs. Therefore, for the Fuzzy-like PD controller the factor's vector includes the KE_p, KE_d and K_u gains as defined in Equation 5.8.

$$f = \{KE_p, KE_d, K_u\} \tag{5.8}$$

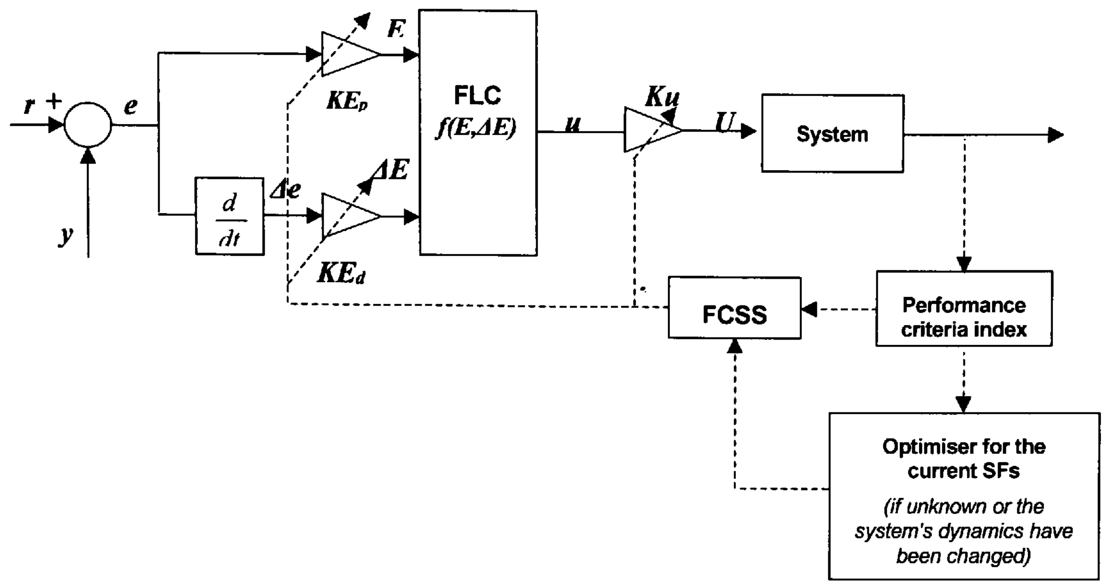


Figure 5.5 Tuning the input/output gain constants of a Fuzzy-like PD Controller

The performance state vector includes the performance state measurements as defined in Equation 5.9.

$$p_s = \{p_{S_{PO}}, p_{S_{RT}}, p_{S_{ST}}\} \tag{5.9}$$

Three different levels are selected for each factor. Therefore, the level vector for the factors KE_p, KE_d and KE_u is defined in Equation 5.10.

$$l_{f_{pi}} = \{l_{f_{pi}KE_p}, l_{f_{pi}KE_d}, l_{f_{pi}KE_u}\} \tag{5.10}$$

where $l_{f_{pi}KE_p}, l_{f_{pi}KE_d}, l_{f_{pi}KE_u}$ are determined in Equation 5.11 to Equation 5.13.

$$l_{f_{pi}KE_p} = \{KE_{p_1}, KE_{p_2}, KE_{p_3}\} = \{0.1, 0.5, 1\} \tag{5.11}$$

$$l_{f_{pi}KE_d} = \{KE_{d_1}, KE_{d_2}, KE_{d_3}\} = \{0.1, 0.5, 1\} \tag{5.12}$$

$$l_{f_{pi}KE_u} = \{K_{u_1}, K_{u_2}, K_{u_3}\} = \{1, 3, 15\} \tag{5.13}$$

These are actually *the* PTFLs. Note that the proportional and derivative factor levels $l_{f_{pi}KE_u}$ and $l_{f_{pi}KE_p}$ chosen assuming that no priory knowledge exist in terms of there between relationship but as the minimum, maximum and between values that could applied in their universe of discourse. However, the output factor levels $l_{f_{pi}K_u}$ are chosen respecting the saturation properties.

From Equation 5.2, the minimum number of experiments that is needed for the parameter optimisation and all two-factor interaction analysis is 19. Thus, the orthogonal array that fits this number of experiments is the L_{27} (Phadke, 1989). Table 5.4 illustrates the L_{27} orthogonal array. The 1st, 2nd and 5th columns are used for the three factors KE_p, KE_d and K_u respectively. The 3rd & 4th, 6th & 7th, 8th & 11th columns are used for the corresponding $KE_p \times KE_d$, $KE_p \times K_u$ and $KE_d \times K_u$ interaction analysis. Note that not all the columns are used in this case. This is because only three factors and three interactions are being studied whereas the L_{27} orthogonal array can analyse up to 13 factors including the interactions. Three replications are performed for each experiment using white noise (with bandwidth magnitude ± 0.025), which is added into the input of the FLC. The results for each performance output are listed in Table 5.5. In analysing the data, the calculation for the average (mean) and the S/N ratio response for each trial/experiment is evaluated by Equation 5.3 and Equation 5.4 respectively and listed in Table 5.4. Only the S/N ratio response for each performance characteristic is considered for further analysis of the results since the robustness of the controller's factors needs to be investigated. Considering only the S/N ratio results, the ANOVA analysis is the next step. Thus, the significance of the three factors and the interaction between them are measured together with their percentage contribution as illustrated in the 7th column of ANOVA Table 5.6. From this table it can be seen that all SFs (main factors) are significant (more than 97.5%) for all performance criteria and therefore all of them are used in the tuning rules. However, only the interaction between $KE_p \times KE_d$ is significant (90% to 99%) for both p_{ST} and p_{PO}, performance criteria. For a full explanation of how to obtain the ANOVA table refer to (Roy, 1990).

Run	KEp	KEd	3	4	Ku	6	7	8	9	10	11	12	13	PO resp ave	PO resp S/N	Rise Time resp ave	Rise Time resp S/N	Set Time resp ave	Set Time resp S/N
1	0.1	0.1	1	1	1	1	1	1	1	1	1	1	1	0.29	10.72	21.4	-26.61	40.4	-32.13
2	0.1	0.1	1	1	3	2	2	2	2	2	2	2	2	0.45	6.85	7.6	-17.62	14.4	-23.17
3	0.1	0.1	1	1	15	3	3	3	3	3	3	3	3	0.91	0.71	3.2	-10.10	5.1	-14.15
4	0.1	0.5	2	2	1	1	1	2	2	2	3	3	3	0.15	16.43	30.35	-29.64	56.9	-35.10
5	0.1	0.5	2	2	3	2	2	3	3	3	1	1	1	0.43	7.18	16.8	-24.51	31.6	-29.99
6	0.1	0.5	2	2	15	3	3	1	1	1	2	2	2	0.46	6.80	11.8	-21.44	22	-26.85
7	0.1	1	3	3	1	1	1	3	3	3	2	2	2	0.22	12.51	41.3	-32.32	77.1	-37.74
8	0.1	1	3	3	3	2	2	1	1	1	3	3	3	0.24	12.20	27.8	-28.88	51.7	-34.27
9	0.1	1	3	3	15	3	3	2	2	2	1	1	1	0.25	11.77	22.5	-27.04	41.8	-32.42
10	0.5	0.1	2	3	1	2	3	1	2	3	1	2	3	2.09	-6.44	3.4	-10.63	5.1	-14.15
11	0.5	0.1	2	3	3	3	1	2	3	1	2	3	1	14.23	-23.07	1.4	-2.92	4.8	-13.62
12	0.5	0.1	2	3	15	1	2	3	1	2	3	1	2	18.48	-25.35	0.52	5.68	3	-9.54
13	0.5	0.5	3	1	1	2	3	2	3	1	3	1	2	0.5	5.74	5	-13.98	9.3	-19.37
14	0.5	0.5	3	1	3	3	1	3	1	2	1	2	3	0.83	1.64	3.2	-10.10	5.1	-14.15
15	0.5	0.5	3	1	15	1	2	1	2	3	2	3	1	1.01	-0.19	2.3	-7.23	4.2	-12.46
16	0.5	1	1	2	1	2	3	3	1	2	2	3	1	0.39	8.10	7.5	-17.50	14.1	-22.98
17	0.5	1	1	2	3	3	1	1	2	3	3	1	2	0.67	3.39	5.2	-14.32	9.8	-19.82
18	0.5	1	1	2	15	1	2	2	3	1	1	2	3	0.82	1.67	4.4	-12.87	8.3	-18.38
19	1	0.1	3	2	1	3	2	1	3	2	1	3	2	12.37	-21.86	1.6	-4.08	5.9	-15.42
20	1	0.1	3	2	3	1	3	2	1	3	2	1	3	28.23	-29.01	0.9	0.92	6.1	-15.71
21	1	0.1	3	2	15	2	1	3	2	1	3	2	1	35.26	-30.95	1.47	-3.35	2.9	-9.25
22	1	0.5	1	3	1	3	2	2	1	3	3	2	1	2.78	-8.89	2.3	-7.23	5.7	-15.12
23	1	0.5	1	3	3	1	3	3	2	1	1	3	2	2.92	-9.33	1.2	-1.58	3.3	-10.37
24	1	0.5	1	3	15	2	1	1	3	2	2	1	3	1.19	-1.56	1.1	-0.83	2	-6.02
25	1	1	2	1	1	3	2	3	2	1	2	1	3	0.51	5.79	3.3	-10.37	6	-15.56
26	1	1	2	1	3	1	3	1	3	2	3	2	1	0.92	0.68	2.4	-7.60	4.5	-13.06
27	1	1	2	1	15	2	1	2	1	3	1	3	2	0.75	2.42	2.23	-6.97	4.1	-12.26
			KEpxKEd	KEpxKEd		KEpxKu	KEpxKu	KEdxKu			KEdxKu								

Table 5.4 The L_{27} orthogonal array that is used to design the experiments together with the response results

Run	KEp	KEd	Ku	PO* (%) Trial 1	PO* (%) Trial 2	PO* (%) Trial 3	Rise time Trial 1	Rise time Trial 2	Rise time Trial 3	Set time Trial 1	Set time Trial 2	Set time Trial 3
1	0.1	0.1	1	0.34	0.22	0.3	21.2	21.5	21.2	40	39.9	40.8
2	0.1	0.1	3	0.47	0.41	0.48	7.8	7.7	7.5	14	14.4	14
3	0.1	0.1	15	1.13	0.86	0.73	3.02	2.92	2.91	5.18	5.1	5.3
4	0.1	0.5	1	0.17	0.13	0.15	30.45	30.22	30.35	56.4	55.7	58.2
5	0.1	0.5	3	0.33	0.43	0.53	16.8	16.8	16.6	30.3	31.4	30.7
6	0.1	0.5	15	0.46	0.43	0.48	12	11.8	11.7	21.8	21.8	21
7	0.1	1	1	0.33	0.2	0.14	41.4	41.3	41.2	78.4	78	77.9
8	0.1	1	3	0.28	0.25	0.2	27.8	27.8	27.7	52	52.4	52.2
9	0.1	1	15	0.29	0.31	0.14	22.8	22.5	22.5	42.06	42.8	43.1
10	0.5	0.1	1	2.36	2.1	1.8	3.3	3.3	3.3	8.5	7.8	5
11	0.5	0.1	3	14.54	13.59	14.55	1.3	1.4	1.3	4.7	4.8	4.7
12	0.5	0.1	15	17.49	18.1	19.84	0.62	0.52	0.51	3.1	3	3
13	0.5	0.5	1	0.36	0.67	0.47	5.4	5	5.3	9.7	9.6	9.5
14	0.5	0.5	3	0.9	0.79	0.79	30.2	2.92	2.92	5.4	5	5.2
15	0.5	0.5	15	0.83	1.11	1.1	2.2	2.3	2.2	4.2	4	4
16	0.5	1	1	0.4	0.4	0.38	7.5	7.4	7.5	14.3	14.5	13.8
17	0.5	1	3	0.7	0.74	0.58	5.2	5.1	5.2	9.6	9.3	9.3
18	0.5	1	15	0.85	0.7	0.91	4.52	4.51	4.31	8.9	8.4	8.4
19	1	0.1	1	12.63	11.43	13.04	1.6	1.65	1.7	7.8	5.7	5.8
20	1	0.1	3	28.1	28.34	28.25	0.9	0.92	0.91	6.4	7.8	4.9
21	1	0.1	15	36.01	36.43	33.35	1.48	1.5	1.5	2.8	2.9	3.7
22	1	0.5	1	2.76	2.78	2.81	2.3	2.3	2.3	6	6	5.6
23	1	0.5	3	2.71	3.27	2.77	1.31	1.31	1.31	3.3	3.5	3.6
24	1	0.5	15	1.28	1.24	1.06	1.06	1.06	1.05	2.1	2.3	1.8
25	1	1	1	0.49	0.59	0.45	3.3	3.3	3.3	6.4	6	6.1
26	1	1	3	0.88	0.98	0.91	2.4	2.4	2.4	4.4	4.8	4.5
27	1	1	15	0.85	0.72	0.69	2.25	2.15	2.16	4.16	3.8	3.9

Table 5.5 Results for each performance after three replicates

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	Significance	% Contribution	Rank
KEd	1907.98	2	953.99	57.67	99%	40.37	1
KEp	1827.38	2	913.69	55.23	99%	38.67	2
KEp x KEd	518.61	4	129.65	7.84	99%	10.97	3
Ku	180.93	2	90.47	5.47	95%	3.83	4
KEd x Ku	94.26	4	23.57	1.42		1.99	6
KEp x Ku	64.55	4	16.14	0.98		1.37	7
Error	132.34	8.00	16.54	1.00		2.80	5
ST	4726.05	26.00				100.00	

(a)

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	Significance	% Contribution	Rank
KEp	1785.38	2	892.69	60.86	99%	64.06	1
KEd	469.00	2	234.50	15.99	99%	16.83	2
Ku	189.15	2	94.58	6.45	97.5%	6.79	3
KEp x KEd	97.86	4	24.47	1.67		3.51	5
KEp x Ku	70.20	4	17.55	1.20		2.52	6
KEd x Ku	58.14	4	14.54	0.99		2.09	7
Error	117.34	8.00	14.67	1.00		4.21	4
ST	2787.08	26.00				100.00	

(b)

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	Significance	% Contribution	Rank
Kep	1384.25	2	692.13	91.96	99%	65.60	1
Ku	273.06	2	136.53	18.14	99%	12.94	2
Ked	186.62	2	93.31	12.40	99%	8.84	3
KEp x KEd	112.63	4	28.16	3.74	90%	5.34	4
KEd x Ku	49.98	4	12.50	1.66		2.37	6
KEp x Ku	43.49	4	10.87	1.44		2.06	7
Error	60.21	8.00	7.53	1.00		2.85	5
ST	2110.24	26.00					

(c)

Table 5.6 ANOVA tables corresponding to (a) PO, (b) rise-time, (c) settling-time responses.

The interaction plots are illustrated in Figure 5.6. After analysing these plots, the OTFLs for each performance criterion are defined and listed in Table 5.7. This is obtained by considering in the interaction plots the factor level's co-ordinates where the maximum response occurs. As can be observed from Figure 5.6 some of the factor levels have almost equal responses (shown by dashed circle in Figure 5.6) and therefore all of them are considered as the OTFLs. As mentioned in Section § 5.2.9 when more than two factor levels have equivalent maximum response, the one that is a member in the most significant interaction plots is chosen as the

OTFLs. Moreover, if their significance ranking is the same then the one that is common between the interaction plots is chosen. For instance, in Table 5.7 (PO interactions) for factor KE_d by looking at the most significant interaction $KE_p \times KE_d$ it can be deducted that both KE_{d_2} and KE_{d_3} can be chosen as possible OTFLs. However, factor level KE_{d_3} is chosen as the OTFL since it is common between $KE_p \times KE_d$ and $KE_p \times K_u$ interactions.

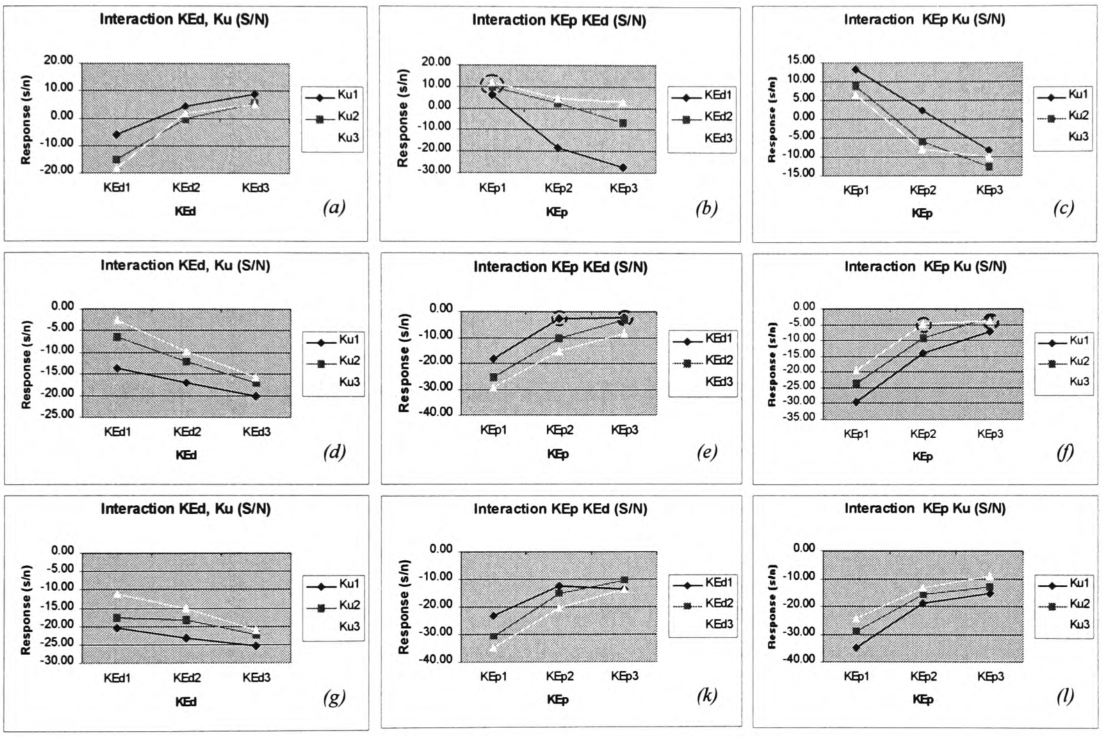


Figure 5.6 Interaction plots between factors $KE_d \times K_u$, $KE_p \times KE_d$ and $KE_p \times K_u$. (a), (b) & (c) resulting from PO S/N ratio analysis, (d), (e), & (f), resulting from rise time S/N ratio analysis, (g), (k) & (l) resulting from settling time S/N ratio analysis.

	Co-ordinates where the maximum response appears and the significance order of											
	PO interaction plot			OTFL	Rise time interaction plot			OTFL	Settling time interaction plot			OTFL
	2	1	3		3	1	2		2	1	3	
	$KE_d \times K_u$	$KE_p \times KE_d$	$KE_p \times K_u$		$KE_d \times K_u$	$KE_p \times KE_d$	$KE_p \times K_u$		$KE_d \times K_u$	$KE_p \times KE_d$	$KE_p \times K_u$	
KE_{p1}		✱	✱	✓								
KE_{p2}						✱					✱	
KE_{p3}						✱	✱	✓		✱	✱	✓
KE_{d1}					✱	✱		✓	✱	✱		✓
KE_{d2}		✱				✱						
KE_{d3}	✱	✱		✓								
Ku_1	✱		✱	✓								
Ku_2							✱					
Ku_3					✱		✱	✓	✱		✱	✓

Table 5.7 The factor level co-ordinates defined from interaction plot. Symbol ✱ represent the intersection between the studied interactions (columns) and their factor levels with the highest response (rows). Symbol ✓ represents the OTFLs for each of the performance criteria.

Thus, for the above analysis the vectors of the tuning factor levels for the first (PO), second (rise time) and third (settling time) performance criteria, PO, are as in Equation 5.14 to Equation 5.16

$$l_{f_{t_1}} = \{KE_{tp_1}, KE_{td_3}, K_{tu_1}\} = \{0.1,1,1\} \tag{5.14}$$

$$l_{f_{t_2}} = \{KE_{tp_3}, KE_{td_1}, K_{tu_3}\} = \{1,0.1,15\} \tag{5.15}$$

$$l_{f_{t_3}} = \{KE_{tp_3}, KE_{td_1}, K_{tu_3}\} = \{1,0.1,15\} \tag{5.16}$$

and the resulting tuning rules are:

R_1: **IF** $p_{S_{PO}} > 0$ **THEN** tune *factor* KE_d **AND** *factor* KE_p **AND** *factor* K_u

R_2: **IF** $p_{S_{RT}} > 0$ **THEN** tune *factor* KE_d **AND** *factor* KE_p **AND** *factor* K_u

R_3: **IF** $p_{S_{ST}} > 0$ **THEN** tune *factor* KE_d **AND** *factor* KE_p **AND** *factor* K_u

The performance states for each criterion p_{PO}, p_{RT} and p_{ST} is measured and the FCSS method is used to optimise the overall tuning factor level vector, that is $L_{oF} = \{L_{oF_{KE_p}}, L_{oF_{KE_d}}, L_{oF_{K_u}}\}$.

From Equation 5.6 the $L_{oF_{KE_p}}, L_{oF_{KE_d}}, L_{oF_{K_u}}$ are defined by:

$$L_{oF_{KE_p}} = \frac{(w_{t_{KE_p}}1 \cdot KE_{tp_1} + w_{c_{KE_p}}1 \cdot KE_{p_{current}}) + (w_{t_{KE_p}}2 \cdot KE_{tp_3} + w_{c_{KE_p}}2 \cdot KE_{p_{current}}) + (w_{t_{KE_p}}3 \cdot KE_{tp_3} + w_{c_{KE_p}}3 \cdot KE_{p_{current}})}{(w_{t_{KE_p}}1 + w_{c_{KE_p}}1) + (w_{t_{KE_p}}2 + w_{c_{KE_p}}2) + (w_{t_{KE_p}}3 + w_{c_{KE_p}}3)} \tag{5.17}$$

$$L_{oF_{KE_d}} = \frac{(w_{t_{KE_d}}1 \cdot KE_{td_3} + w_{c_{KE_d}}1 \cdot KE_{d_{current}}) + (w_{t_{KE_d}}2 \cdot KE_{td_1} + w_{c_{KE_d}}2 \cdot KE_{d_{current}}) + (w_{t_{KE_d}}3 \cdot KE_{td_1} + w_{c_{KE_d}}3 \cdot KE_{d_{current}})}{(w_{t_{KE_d}}1 + w_{c_{KE_d}}1) + (w_{t_{KE_d}}2 + w_{c_{KE_d}}2) + (w_{t_{KE_d}}3 + w_{c_{KE_d}}3)} \tag{5.18}$$

$$L_{oF_{KE_u}} = \frac{(w_{t_{KE_u}}1 \cdot K_{tu_3} + w_{c_{KE_u}}1 \cdot K_{u_{current}}) + (w_{t_{KE_u}}2 \cdot K_{tu_3} + w_{c_{KE_u}}2 \cdot K_{u_{current}}) + (w_{t_{KE_u}}3 \cdot K_{tu_3} + w_{c_{KE_u}}3 \cdot K_{u_{current}})}{(w_{t_{KE_u}}1 + w_{c_{KE_u}}1) + (w_{t_{KE_u}}2 + w_{c_{KE_u}}2) + (w_{t_{KE_u}}3 + w_{c_{KE_u}}3)} \tag{5.19}$$

The weight values come from the DOF of the S and Z MFs as described in Section § 5.2.11, Figure 5.3.

The results of the tuning performance are illustrated in Figure 5.7 and Table 5.8.

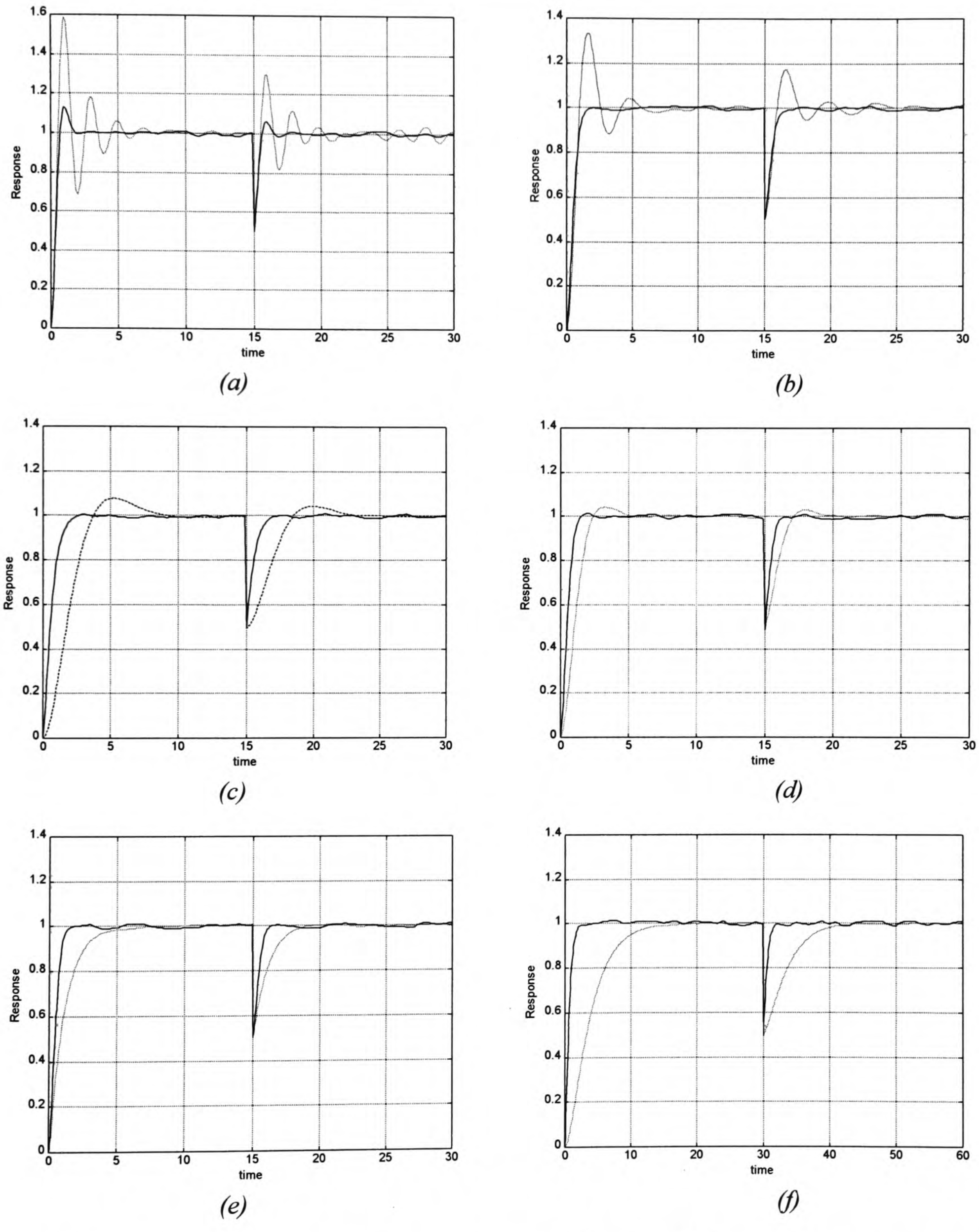


Figure 5.7 Response (of Equation 5.7) resulting from the fuzzy PD controller. -- Before Tuning, – After tuning

No Ex	Performance	Gain KE_p	Gain KE_d	Gain KE_u	p_{PO}	p_{RT}	p_{ST}	IAE	ITAE
a	Before Tuning	0.9	0.01	12	58.6	0.4	7.1	1.92	15.45
	After Tuning	0.87	0.18	14.34	12.9	0.5	1.7	0.72	4.68
b	Before Tuning	0.6	0.05	7	33.6	0.71	5.4	1.76	12.57
	After Tuning	0.73	0.32	14.38	1.16	0.84	1.4	0.91	6
c	Before Tuning	0.9	0.02	0.7	7.8	2.6	7.9	3.1	22.01
	After Tuning	0.7	0.4	14.85	0.8	1.2	2.1	1.08	6.93
d	Before Tuning	0.4	0.2	5	4.7	1.5	4.2	1.77	11.33
	After Tuning	0.74	0.28	13.86	1.39	0.78	1.2	0.89	5.79
e	Before Tuning	0.85	0.9	13	0.81	2.12	4.5	1.83	12.36
	After Tuning	0.93	0.37	14.37	1.12	0.73	1.3	0.82	5.41
f	Before Tuning	0.06	0.04	5	0.18	6.8	12.5	6.06	81.08
	After Tuning	0.67	0.36	14.49	1.31	1.1	2	1.17	16.73

Table 5.8 Tuning results of a fuzzy PD controller

5.3.3.2 Second case: optimising and tuning the input SFs together with the peaks of the MFs

The use of SFs for the tuning of the Fuzzy-like PD controllers is bounded, as their effectiveness is sometimes contradictory by the different performance criteria. For instance, the increase of the SF KE_p decreases the steady-state error and the rise time but usually leads to a large overshoot. Tuning the peaks of the MFs can be a solution to the above problem. Zheng, (1992) suggests a simple and effective way of improving the stability of the system without affecting the responsiveness through shifting the peaks of the MFs. Moreover, by changing a peak value, the definition of only one fuzzy label is changed and hence it can only affect the rules that use this changed fuzzy label. However, the tuning of the peaks of the MFs does not always affect the response satisfactorily for instance in the case of tuning only the peak values of the change-of-error Δe labels. Using therefore both SFs and MFs as the tuning factors could be one of the solutions. The large number of combinations between the tuning factors as well as the analysis of the correlation between them are the key issues that need further investigation.

In this study for the two inputs, *error* and *change-of-error*, the corresponding SFs and MFs' positions of the FLC are chosen as the four tuning factors to be investigated. Moreover, both IAE and ITAE are the performance criteria used to define the resulting effect of the tuning factors. Therefore the tuning factor vector is:

$$f = \{KE_p, KE_d, Le_1, Le_2, Lce_1, Lce_2, \mathrm{Re}_1, \mathrm{Re}_2, Rce_1, Rce_2\} \tag{5.20}$$

where KE_p, KE_d are the input gains of the *error* and *change-of-error* respectively and the Le_1, $Le_2, Lce_1, Lce_2, \mathrm{Re}_1$, $\mathrm{Re}_2, Rce_1, Rce_2$ are the peak values of the four MFs (labelled as NM, NS, PS, PM) on the *Left* and *Right* side of the *error* and *change-of-error* universe of discourses. Note that the left and the right peak values are symmetrical. Thus, $\mathrm{Re}_{1,k} = -Le_{1,k}$, $\mathrm{Re}_{2,k} = -Le_{2,k}$, $Rce_{1,k} = -Lce_{1,k}$, $Rce_{2,k} = -Lce_{2,k}$. Additionally, the performance state vector $p = \{p_{S_{IAE}}, p_{S_{ITAE}}\}$ includes the performance state measurements of IAE and ITAE. Figure 5.8 illustrates the SFs gain as well as the peaks of MFs tuning schedule type of controller.

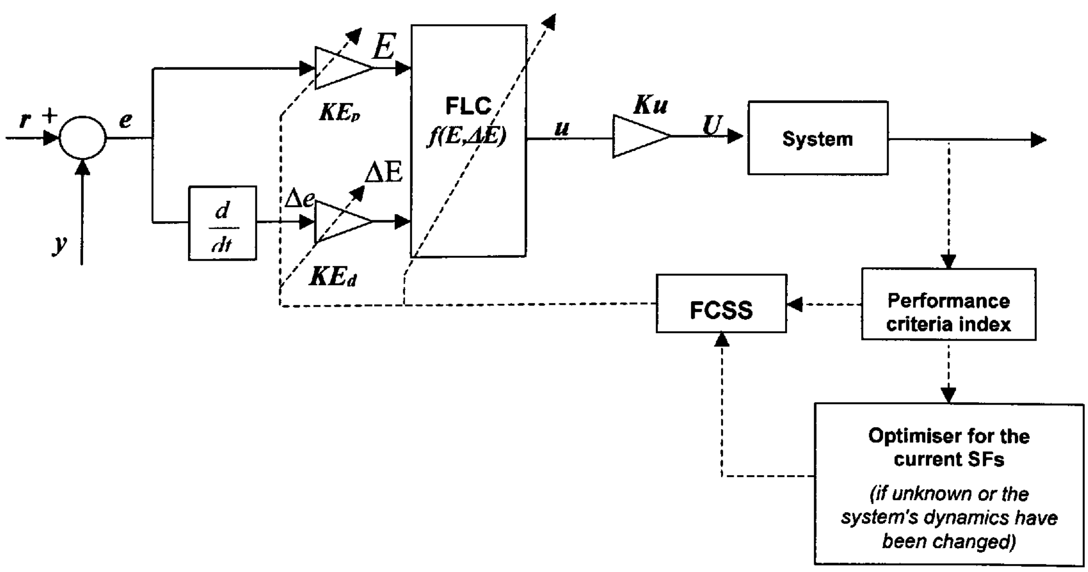


Figure 5.8 Tuning the input gain constants and the peak values of the Fuzzy-like PD Controller

Three levels have been selected for each factor. Thus, the level vector for the tuning factors is defined as in Equation 5.21.

$$l_{f_{pi}} = \{l_{f_{pi}KE_p}, l_{f_{pi}KE_d}, l_{f_{pi}Le_1}, l_{f_{pi}Le_2}, l_{f_{pi}Lce_1}, l_{f_{pi}Lce_2}, l_{f_{pi}\mathrm{Re}_1}, l_{f_{pi}\mathrm{Re}_2}, l_{f_{pi}Rce_1}, l_{f_{pi}Rce_2}\} \quad (5.21)$$

For the gains of the SFs three levels are defined as in Equation 5.22 and Equation 5.23.

$$l_{f_{pi}KE_p} = \{KE_{p_1}, KE_{p_2}, KE_{p_3}\} = \{0.1, 0.5, 1\} \quad (5.22)$$

$$l_{f_{pi}KE_d} = \{KE_{d_1}, KE_{d_2}, KE_{d_3}\} = \{0.1, 0.5, 1\} \quad (5.23)$$

For the peaks of the MFs three main cases are defined in terms of how they shift in the universe of discourse. The first one is when the peaks of the MFs are shifting close to the symmetric axis of the universe of discourse, the second one is when the peaks of the MFs are distributed symmetrically and equally in the universe of discourse and the third one is when the peaks of the MFs are shifting close to the min and/or max point of the universe of discourse. Thus, where the universe of discourse is between $[-1,1]$ the actual set values of the MFs tuning factors are as defined in Equation 5.24 to Equation 5.31:

$$l_{f_{pi}Le_1} = \{Le_{1,1}, Le_{1,2}, Le_{1,3},\} = \{-0.1, -0.33, -0.6\} \quad (5.24)$$

$$l_{f_{pi}Le_1} = \{Le_{2,1}, Le_{2,2}, Le_{2,3},\} = \{-0.26, -0.66, -0.86\} \quad (5.25)$$

$$l_{f_{pi}Lce_1} = \{Lce_{1,1}, Lce_{1,2}, Lce_{1,3},\} = \{-0.1, -0.33, -0.6\} \quad (5.26)$$

$$l_{f_{pi}Lce_1} = \{Lce_{2,1}, Lce_{2,2}, Lce_{2,3},\} = \{-0.26, -0.66, -0.86\} \quad (5.27)$$

$$l_{f_{pi}\mathrm{Re}_1} = \{\mathrm{Re}_{1,1}, \mathrm{Re}_{1,2}, \mathrm{Re}_{1,3},\} = \{0.1, 0.33, 0.6\} \quad (5.28)$$

$$l_{f_{pi}\mathrm{Re}_1} = \{\mathrm{Re}_{2,1}, \mathrm{Re}_{2,2}, \mathrm{Re}_{2,3},\} = \{0.26, 0.66, 0.86\} \quad (5.29)$$

$$l_{f_{pt}Rce_1} = \{Rce_{1,1}, Rce_{1,2}, Rce_{1,3},\} = \{0.1, 0.33, 0.6\} \quad (5.30)$$

$$l_{f_{pt}Rce_1} = \{Rce_{2,1}, Rce_{2,2}, Rce_{2,3},\} = \{0.26, 0.66, 0.86\} \quad (5.31)$$

For convenience in the definitions, the two input MFs *(error and change-of-error)* are defined as MF_e, MF_{ce} respectively. Moreover, their three positions are called as inner (In) MF_{e_1} & MF_{ce_1}, symmetrical (Sim) MF_{e_2} & MF_{ce_2} and outer (Out) MF_{e_3} & MF_{ce_3}. Thus the *PTFLs* for the MFs can be defined as in Equation 5.32 and Equation 5.33 and shown in Figure 5.9.

$$MF_e = \{MF_{e_1}, MF_{e_2}, MF_{e_3}\} \quad (5.32)$$

$$MF_{ce} = \{MF_{ce_1}, MF_{ce_2}, MF_{ce_3}\} \quad (5.33)$$

From Equation 5.24 to Equation 5.31 the factor levels $MF_{e_1}, MF_{e_2}, MF_{e_3}, MF_{ce_1}, MF_{ce_2}, MF_{ce_3}$ are defined as in Equation 5.34 to Equation 5.39

$$MF_{e_1} = \{Le_{1,1}, Le_{2,1}, \mathrm{Re}_{1,1}, \mathrm{Re}_{2,1}\} = \{-0.1, -0.26, 0.1, 0.26\} \quad (5.34)$$

$$MF_{e_2} = \{Le_{1,2}, Le_{2,2}, \mathrm{Re}_{1,2}, \mathrm{Re}_{2,2}\} = \{-0.33, -0.66, 0.33, 0.66\} \quad (5.35)$$

$$MF_{e_3} = \{Le_{1,3}, Le_{2,3}, \mathrm{Re}_{1,3}, \mathrm{Re}_{2,3}\} = \{-0.6, -0.86, 0.6, 0.86\} \quad (5.36)$$

$$MF_{ce_1} = \{Lce_{1,1}, Lce_{2,1}, Rce_{1,1}, Rce_{2,1}\} = \{-0.3, -0.4, 0.3, 0.4\} \quad (5.37)$$

$$MF_{ce_2} = \{Lce_{1,2}, Lce_{2,2}, Rce_{1,2}, Rce_{2,2}\} = \{-0.33, -0.66, 0.33, 0.66\} \quad (5.38)$$

$$MF_{ce_3} = \{Lce_{1,3}, Lce_{2,3}, Rce_{1,3}, Rce_{2,3}\} = \{-0.6, -0.86, 0.6, 0.86\} \quad (5.39)$$

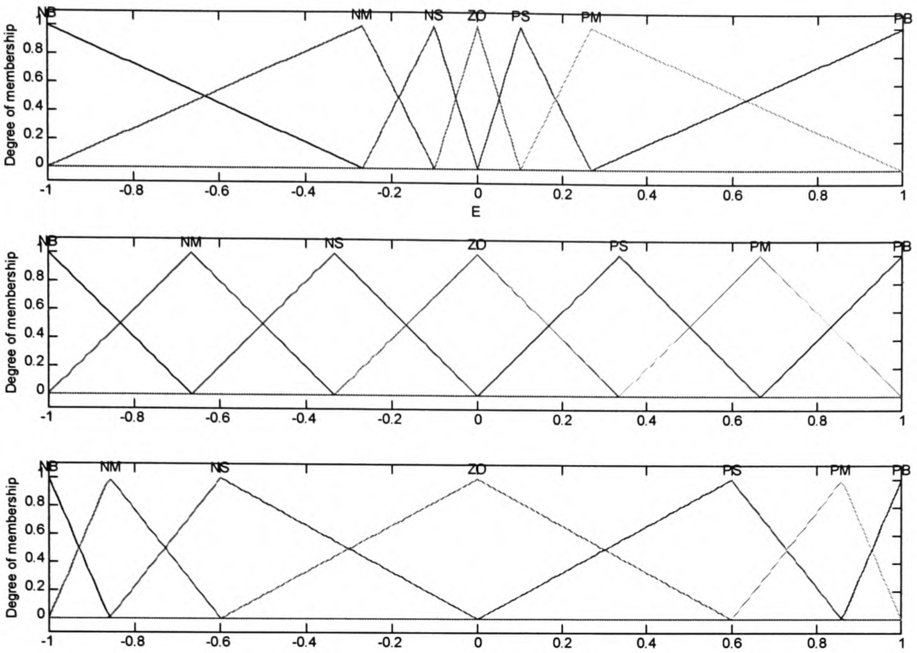


Figure 5.9 Inner, Symmetrical and Outer membership functions

For the four tuning factors (two factors for each input parameter optimisation as well as their interaction analysis) the minimum number of experiments that is needed, including main factors and their interactions is 21 (evaluated by Equation 5.2) and hence L_{27} is thus the orthogonal array that fits this number of experiments. Table 5.9 and Table 5.10 illustrates this orthogonal array. In Table 5.9 the 1st, 2nd, 5th and 9th columns are used for the four factors KE_p, KE_d, MF_e and MF_{ce} respectively. Furthermore, the 4th, 7th, 10th, 11th, 12th and 13th columns are used for the interactions between $KE_p \times KE_d$, $KE_p \times MF_e$, $KE_p \times MF_{ce}$, $KE_d \times MF_e$, $KE_d \times MF_{ce}$, $MF_e \times MF_{ce}$ respectively. However, due to confounding in 3rd, 6th and 8th column only three interactions can be studied. (Confounding occurs when inability to determine which sets of interactions $KE_p \times KE_d$ & $MF_e \times MF_{ce}$, $KE_p \times MF_{ce}$ & $KE_d \times MF_{ce}$ and $KE_p \times MF_{ce}$ & $KE_d \times MF_e$ may be affecting the response variable (Fowlkes, 1995)). The interactions that may be the most significant are the selected ones for further investigation. It was suspected from experimental trials that for both IAE and ITAE, the interaction between $KE_p \times MF_e$, $KE_p \times KE_d$ and $KE_p \times MF_{ce}$ are the most important. Moreover, these are confirmed by the ANOVA Table 5.12 and Table 5.13. Figure 5.10 illustrates the interaction plots between these factors in pairs. In the analysis of these plots the co-ordinates where the maximum response of the factor levels occurs are defined as the OTFLs and listed in Table 5.11.

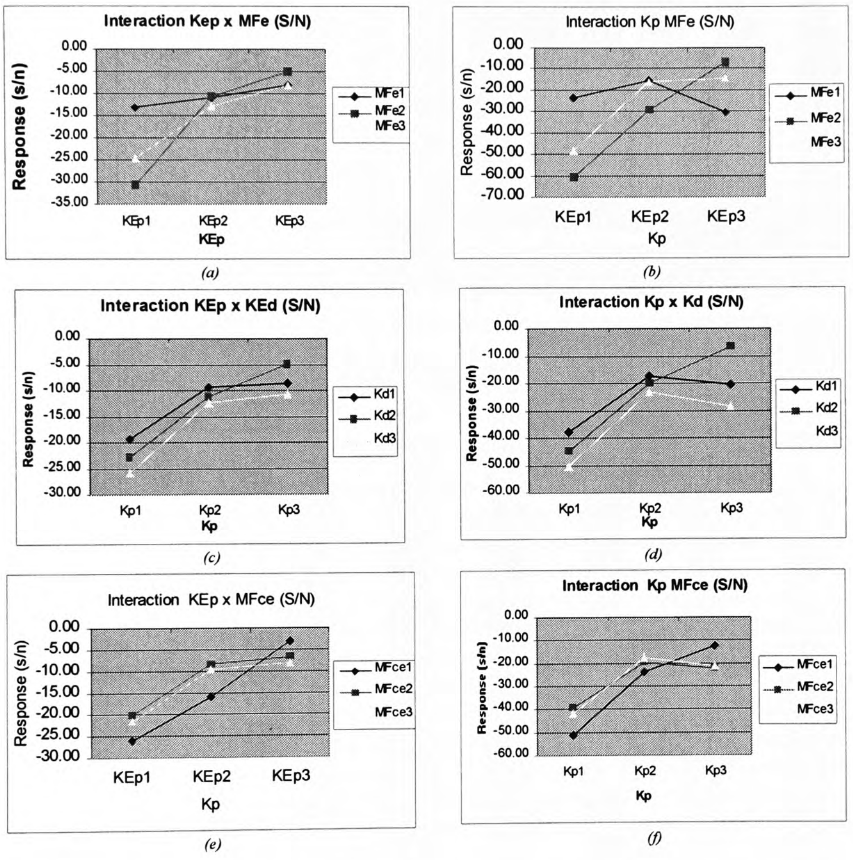


Figure 5.10 Interaction plots between factors $KE_p \times MF_e$, $KE_p \times KE_d$ and $KE_p \times MF_{ce}$ resulting from IAE (a, b, c) & ITAE (d, e, f) S/N ratio analysis

Run	KEp	KEd	3	4	MFe	6	7	8	MFce	10	11	12	13	IAE resp ave	IAE resp S/N	ITAE resp ave	ITAE resp S/N
1	0.1	0.1	1	1	In	1	1	1	In	1	1	1	1	3.9	-11.82	11.88	-21.50
2	0.1	0.1	1	1	Out	2	2	2	Out	2	2	2	2	18.93	-25.54	337.16	-50.56
3	0.1	0.1	1	1	Sim	3	3	3	Sim	3	3	3	3	10.92	-20.76	107.99	-40.67
4	0.1	0.5	2	2	In	1	1	2	Out	2	3	3	3	3.76	-11.50	10.6	-20.51
5	0.1	0.5	2	2	Out	2	2	3	Sim	3	1	1	1	26.94	-28.61	704.49	-56.96
6	0.1	0.5	2	2	Sim	3	3	1	In	1	2	2	2	26.66	-28.52	699.27	-56.89
7	0.1	1	3	3	In	1	1	3	Sim	3	2	2	2	5.93	-15.46	24.33	-27.72
8	0.1	1	3	3	Out	2	2	1	In	1	3	3	3	77.92	-37.83	6029.25	-75.61
9	0.1	1	3	3	Sim	3	3	2	Out	2	1	1	1	15.52	-23.82	228.86	-47.19
10	0.5	0.1	2	3	In	2	3	1	Out	3	1	2	3	2.47	-7.85	8.07	-18.14
11	0.5	0.1	2	3	Out	3	1	2	Sim	1	2	3	1	3.88	-11.78	10.83	-20.69
12	0.5	0.1	2	3	Sim	1	2	3	In	2	3	1	2	2.61	-8.33	4.54	-13.14
13	0.5	0.5	3	1	In	2	3	2	Sim	1	3	1	2	1.92	-5.67	2.8	-8.94
14	0.5	0.5	3	1	Out	3	1	3	In	2	1	2	3	9.51	-19.56	86.91	-38.78
15	0.5	0.5	3	1	Sim	1	2	1	Out	3	2	3	1	2.54	-8.10	4.25	-12.57
16	0.5	1	1	2	In	2	3	3	In	2	2	3	1	3.33	-10.45	8.74	-18.83
17	0.5	1	1	2	Out	3	1	1	Out	3	3	1	2	5.52	-14.84	26.37	-28.42
18	0.5	1	1	2	Sim	1	2	2	Sim	1	1	2	3	3.93	-11.89	13.19	-22.40
19	1	0.1	3	2	In	3	2	1	Sim	2	1	3	2	4.93	-13.86	91.44	-39.22
20	1	0.1	3	2	Out	1	3	2	In	3	2	1	3	1.83	-5.25	2.42	-7.68
21	1	0.1	3	2	Sim	2	1	3	Out	1	3	2	1	2.14	-6.61	5.06	-14.08
22	1	0.5	1	3	In	3	2	2	In	3	3	2	1	1.78	-5.01	2.06	-6.28
23	1	0.5	1	3	Out	1	3	3	Out	1	1	3	2	1.83	-5.25	2.54	-8.10
24	1	0.5	1	3	Sim	2	1	1	Sim	2	2	1	3	1.7	-4.61	2.06	-6.28
25	1	1	2	1	In	3	2	3	Out	1	2	1	3	3.69	-11.34	176.95	-44.96
26	1	1	2	1	Out	1	3	1	Sim	2	3	2	1	2.83	-9.04	7.67	-17.70
27	1	1	2	1	Sim	2	1	2	In	3	1	3	2	3.76	-11.50	14.42	-23.18
			KEpxKd MFexMFce	KEpxKd		KEpxMFe KEdxMFce	KEpxMFe	KEpxMFce KEdxMFe		KEpxMFce	KEdxMFe	KEdxMFce	MFexMFce				

Table 5.9 The L_{27} orthogonal array used to design the experiments together with the dead time response results from the second case study

					IAE				ITAE			
Run	KEp	KEd	MFe	MFce	L=0	L=0.1	L=0.2	L=0.3	L=0	L=0.1	L=0.2	L=0.3
1	0.1	0.1	In	In	4	3.9	3.9	3.8	12.8	12.22	11.58	10.9
2	0.1	0.1	Out	Out	19	18.96	18.91	18.86	342.6	338.96	335.35	331.72
3	0.1	0.1	Sim	Sim	11	10.9	10.92	10.87	111	109	107	104.97
4	0.1	0.5	In	Out	3.83	3.78	3.73	3.68	11.55	10.92	10.29	9.65
5	0.1	0.5	Out	Sim	27.02	26.97	26.92	26.87	710.4	706.54	702.5	698.52
6	0.1	0.5	Sim	In	26.73	26.68	26.63	26.58	703.3	700.7	697.94	695.15
7	0.1	1	In	Sim	6	5.95	5.9	5.85	3.2.8	32.14	31.39	30.6
8	0.1	1	Out	In	77.9	77.98	77.93	77.88	6001	6045	6039	6032
9	0.1	1	Sim	Out	15.59	15.54	15.49	15.44	232.67	229.83	227.58	225.36
10	0.5	0.1	In	Out	1.76	2.05	2.54	3.52	2.67	4.02	7.37	18.22
11	0.5	0.1	Out	Sim	3.96	3.91	3.86	3.81	11.92	11.19	10.45	9.76
12	0.5	0.1	Sim	In	2.67	2.62	2.58	2.56	5.02	4.64	4.34	4.16
13	0.5	0.5	In	Sim	1.73	1.81	1.95	2.19	2.05	2.28	2.81	4.05
14	0.5	0.5	Out	In	9.6	9.5	9.5	9.45	88.35	87.44	86.43	85.43
15	0.5	0.5	Sim	Out	2.56	2.53	2.52	2.53	4.52	4.25	4.1	4.11
16	0.5	1	In	In	3.43	3.37	3.32	3.18	9.25	8.96	8.66	8.08
17	0.5	1	Out	Out	5.6	5.55	5.5	5.45	27.56	26.79	25.97	25.17
18	0.5	1	Sim	Sim	4	3.95	3.9	3.85	13.91	13.44	12.95	12.46
19	1	0.1	In	Sim	1.57	2.01	3.9	12.22	2.17	4.69	38.9	320
20	1	0.1	Out	In	1.9	1.8	1.76	1.84	2.62	2.27	2.22	2.59
21	1	0.1	Sim	Out	1.67	1.89	2.24	2.78	2.61	3.55	5.32	8.77
22	1	0.5	In	In	1.77	1.75	1.75	1.86	1.97	1.86	1.86	2.56
23	1	0.5	Out	Out	1.84	1.77	1.78	1.91	2.45	2.28	2.42	3
24	1	0.5	Sim	Sim	1.62	1.63	1.7	1.84	1.83	1.87	2	2.54
25	1	1	In	Out	1.51	1.65	1.99	9.6	1.59	2.1	4.1	700
26	1	1	Out	Sim	3	2.9	2.78	2.63	8.7	8.12	7.35	6.5
27	1	1	Sim	In	3.99	3.9	3.7	3.46	15.67	15.18	14.19	12.64

Table 5.10 Results for each performance using different dead times $L = 0,\ 0.1,\ 0.2$, and 0.3

	Co-orinates where the maximum S/N response appears						
	IAE Interaction Plots			ITAE Interaction Plots			OTFL
	KE_p x MF_e	KE_p x KE_d	KE_p x MF_{ce}	KE_p x MF_e	KE_p x KE_d	KE_p x MF_{ce}	
KE_{d1}							
KE_{d2}		✱			✱		✓
KE_{d3}							
KE_{p1}							
KE_{p2}							
KE_{p3}	✱	✱	✱	✱	✱	✱	✓
MF_{e1}							
MF_{e2}	✱			✱			✓
MF_{e3}							
MF_{ce1}			✱			✱	✓
MF_{ce2}							
MF_{ce3}							

Table 5.11 The factor level co-ordinates defined from a) IAE, b) ITAE interaction plots. Symbol ✱ represent the intersection between the studied interactions (columns) and their factor level's with the highest response (rows). Symbol ✓ represents the OTFLs for each of the performance criteria

Four different dead times (i.e. $L = 0$, 0.1, 0.2, and 0.3) are used to investigate the controller's robustness in this matter. For these dead times the results for each performance output are listed in Table 5.10. As explained in Section § 5.2.8, analysis of data is evaluated using Equation 5.3 and Equation 5.4 to calculate the average (mean) and S/N ratio response respectively for each trial/experiment and are listed in Table 5.9. However, only the S/N ratio response for both IAE and ITAE response characteristics are considered for further analysis of the results since the robustness of the controller's factors need to be investigated. ANOVA is the next step of the analysis of the S/N ratio results where the significance of the four factors and their interactions are measured and ranked together with their percentage contribution as depicted in Table 5.12 and Table 5.13. From these tables it can be observed that the most significant factors for both IAE and ITAE performance criteria are KE_p (55.96% and 42.11% contribution respectively) as well as KE_d and MF_e. However, from the above analysis results the factor MF_{ce} is not very significant, which means it could be avoided in consideration as a tuning factor. Nevertheless, it is included in the tuning rules as the number of the factors is very small and therefore all of them can contribute in the tuning rules without causing complication in terms of constructing them and computation time. Consequently if a large number of factors were under investigation, only the most significant ones would be included in the controller.

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	Significance	% Contribution	Rank
KEp	1076.1	2	538.05	52.83	99%	55.96	1
KEp x MFe	364.33	4	91.08	8.94		18.95	2
MFe	232.93	2	116.47	11.43	99%	12.11	3
KEd	76.56	2	38.28	3.76		3.98	5
KEp x Kd	47.46	4	11.87	1.16		2.47	6
KEp x MFce	44.09	4	11.02	1.08		2.29	7
Error	81.48	8.00	10.19	1.00		4.24	4
ST	1922.95	26.00				100.00	

Table 5.12 ANOVA table for IAE

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	Significance	% Contribution	Rank
KEp	3692.05	2	1846.03	36.21	99%	42.11	1
KEpxMFe	2642.42	4	660.61	12.96		30.14	2
MFe	564.24	2	282.12	5.53	95%	6.44	3
Ked	547.76	2	273.88	5.37	95%	6.25	4
KEp x KEd	465.63	4	116.41	2.28		5.31	5
KEp x MFce	446.9	4	111.73	2.19		5.10	6
Error	407.82	8.00	50.98	1.00		4.65	7
ST	8766.81	26.00				100.00	

Table 5.13 ANOVA table for ITAE

The evaluation of the above analysis is the vectors of the optimal tuning factor levels for the performance criteria under consideration. Thus, for the IAE and ITAE the OTFLs vectors are as defined in Equation 5.40 and Equation 5.41 respectively:

$$l_{f_{t_1}} = \{KE_{tp_3}, KE_{td_2}, MF_{te_2}, MF_{tce_1}\} \tag{5.40}$$

$$l_{f_{t_2}} = \{KE_{tp_3}, KE_{td_2}, MF_{te_2}, MF_{tce_1}\} \tag{5.41}$$

where $KE_{tp_3} = KE_{p_3}$, $KE_{td_2} = KE_{d_2}$, $MF_{te_2} = MF_{e_2}$, $MF_{tce_1} = MF_{ce_1}$

and the resulting tuning rules are:

R_1: **IF** $p_{S_{IAE}} > 0$ **THEN** tune *factor* KE_d **AND** *factor* KE_p **AND** *factor* MF_e **AND** *factor* MF_{ce}

R_2: **IF** $p_{S_{ITAE}} > 0$ **THEN** tune *factor* KE_d **AND** *factor* KE_p **AND** *factor* MF_e **AND** *factor* MF_{ce}

The performance states for each criterion p_{PO}, p_{RT} and p_{ST} is measured and the *FCSS* method is used to optimise the overall tuning factor level vector, which defines the Equation 5.42.

$$L_{oF} = \{L_{oF_{KE_p}}, L_{oF_{KE_d}}, L_{oF_{MF_e}}, L_{oF_{MF_{ce}}}\} \tag{5.42}$$

From Equation 5.6 the $L_{oF_{KE_p}}, L_{oF_{KE_d}}, L_{oF_{MF_e}}, L_{oF_{MF_{ce}}}$ are defined by Equation 5.43 to Equation 5.46.

$$L_{oF_{KE_p}} = \frac{(w_{t_{KE_p}}1 \cdot KE_{tp_3} + w_{c_{KE_p}}1 \cdot KE_{p_{current}}) + (w_{t_{KE_p}}2 \cdot KE_{tp_3} + w_{c_{KE_p}}2 \cdot KE_{p_{current}})}{(w_{t_{KE_p}}1 + w_{c_{KE_p}}1) + (w_{t_{KE_p}}2 + w_{c_{KE_p}}2)} \tag{5.43}$$

$$L_{oF_{KE_d}} = \frac{(w_{t_{KE_d}}1 \cdot KE_{td_2} + w_{c_{KE_d}}1 \cdot KE_{d_{current}}) + (w_{t_{KE_d}}2 \cdot KE_{td_2} + w_{c_{KE_d}}2 \cdot KE_{d_{current}})}{(w_{t_{KE_d}}1 + w_{c_{KE_d}}1) + (w_{t_{KE_d}}2 + w_{c_{KE_d}}2)} \tag{5.44}$$

$$L_{oF_{MF_e}} = \frac{(w_{t_{MF_e}}1 \cdot MF_{te_2} + w_{c_{MF_e}}1 \cdot MF_{e_{current}}) + (w_{t_{MF_e}}2 \cdot MF_{te_2} + w_{c_{MF_e}}2 \cdot MF_{e_{current}})}{(w_{t_{MF_e}}1 + w_{c_{MF_e}}1) + (w_{t_{MF_e}}2 + w_{c_{MF_e}})} \tag{5.45}$$

$$L_{oF_{MF_{ce}}} = \frac{(w_{t_{MF_{ce}}}1 \cdot MF_{tce_1} + w_{c_{MF_{ce}}}1 \cdot MF_{ce_{current}}) + (w_{t_{MF_{ce}}}2 \cdot MF_{tce_1} + w_{c_{MF_{ce}}}2 \cdot MF_{ce_{current}})}{(w_{t_{MF_{ce}}}1 + w_{c_{MF_{ce}}}1) + (w_{t_{MF_{ce}}}2 + w_{c_{MF_{ce}}}2)} \tag{5.46}$$

The weight values come from the DOF of the S and Z MFs as described in Section § 5.2.11, Figure 5.3.

The results of the tuning performance are illustrated in Figure 5.11 to Figure 5.15 and in the Table 5.14.

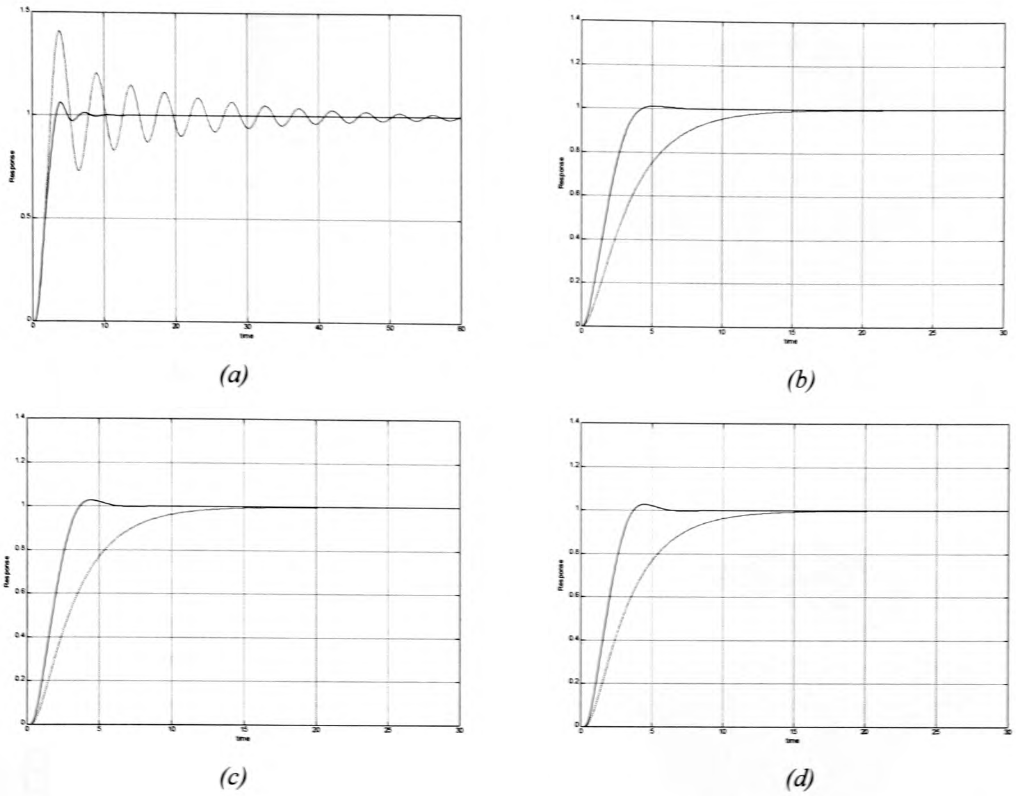


Figure 5.11 Response (of Equation 5.7) resulting before and after tuning the fuzzy PD controller using different values of dead time (L) a) $L = 0.3$, b) $L = 0.05$, c) $L = 0.15$ and d) $L = 0.25$. -- Before Tuning, – After tuning

L	Performance	Gain KEp	Gain KEd	Le_1, Re_1	Le_2, Re_2	Lce_1, Rce_1	Lce_2, Rce_2	PO	Rise Time	Set Time	IAE	ITAE
0.30	Before Tuning	0.43	0.05	0.06	0.2	0.36	0.6	40.9	1.5	49.3	5.05	57.52
	After Tuning	0.71	0.45	0.09	0.27	0.12	0.31	6.17	2.1	6.1	1.91	2.44
0.05	Before Tuning	0.35	0.75	0.265	0.6	0.4	0.73	0	6.55	12.4	3.74	11.33
	After Tuning	0.73	0.60	0.17	0.4	0.22	0.46	1.1	2.55	3.95	1.83	2.12
0.15	Before Tuning	0.35	0.75	0.265	0.6	0.4	0.73	0	6.55	12.4	3.74	11.33
	After Tuning	0.72	0.61	0.175	0.41	0.23	0.45	1.7	2.35	3.7	1.84	2
0.25	Before Tuning	0.35	0.75	0.265	0.6	0.4	0.73	0	6.55	12.4	3.74	11.33
	After Tuning	0.71	0.60	0.165	0.41	0.226	0.47	2.7	2.15	3.2	1.86	2.1

Table 5.14 Factor setting of fuzzy PD controller before and after their tuning

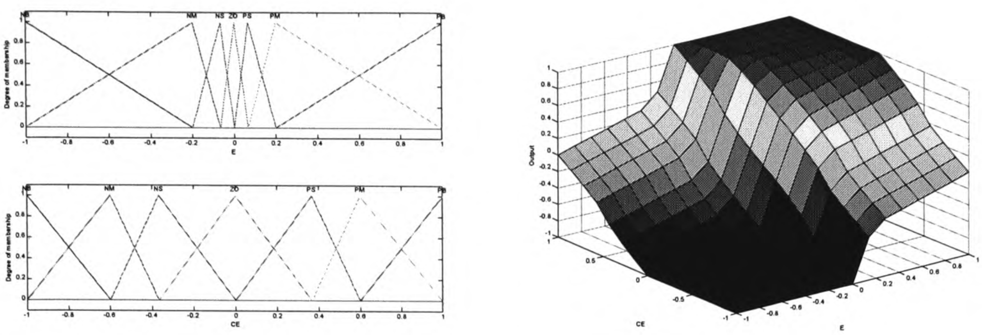


Figure 5.12 a) MFs of the two inputs: error E and change-of-error CE with their peaks as set before tuning (see Table 5.14, 2nd row) and b) the control surface of the fuzzy PD controller

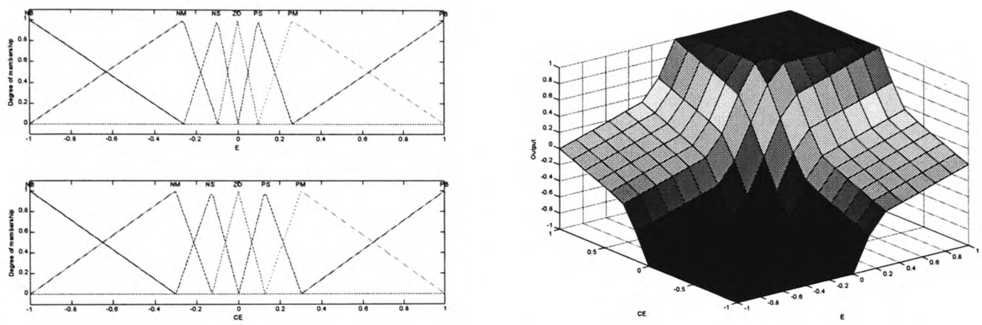
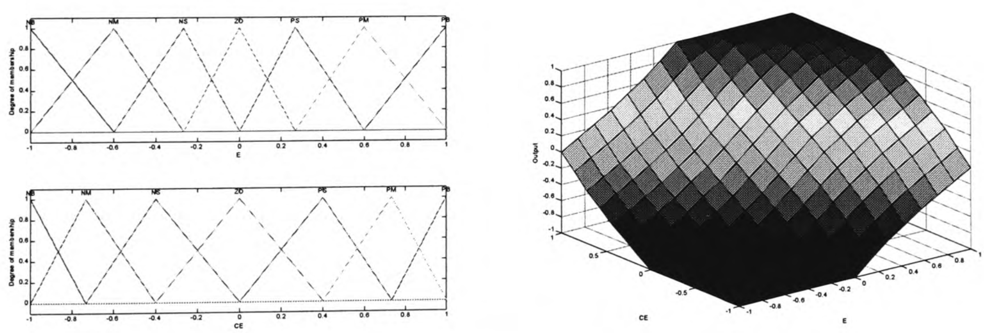


Figure 5.13 a) MFs after tuning with their peaks set as in Table 5.14, 3rd row, and b) the corresponding control surface of the tuned fuzzy PD controller with L = 0.3



(a) (b)

Figure 5.14 a) MFs of the two inputs: error E and change-of-error CE with their peaks as set before tuning (see Table 5.14) b) the control surface of the fuzzy PD controller set by the input from the MFs of (a)

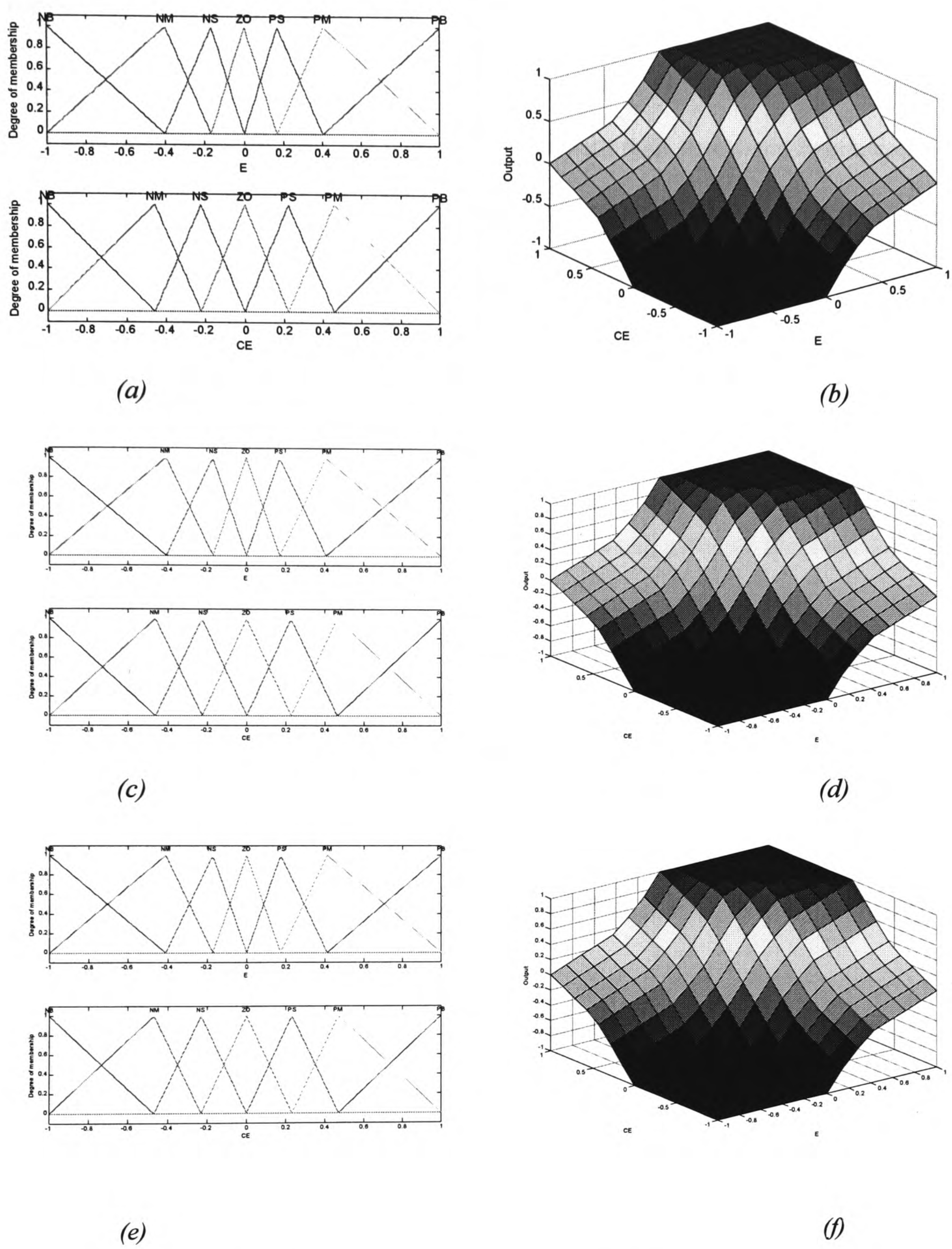


Figure 5.15 a), c), e) MFs after tuning with their peaks set as in Table 5.14 and b) d) f) the corresponding control surface of the tuned fuzzy PD controller with $L = 0.05$, 0.15, 0.25.

5.4 Discussion

The analysis described on this chapter involves eleven major steps, which can be grouped as: defining the performance criteria and factors together with their levels and interactions (step 1 through to 4), planning a matrix experiment to determine the effects of the control factors (step 5 through to 6), conducting the matrix experiment (step 7), analysing and verifying the results to determine the OTFLs (step 8), verification of the optimal factor levels (step 9), constructing the fuzzy rules to tune the factor levels (step 10) and finally developing the tuning engine using the FCSS (step 11).

The steps of the Taguchi methods can be easily processed using one of the available software packages. In this study the WinRobust ® program was used (WinRobust, 1995).

Taguchi Design of Experiments method is used as it:

- *Provides a more efficient way of designing experiments using a minimum number of experiment using its Orthogonal Arrays,*
- *Provides a strategy that analyses and prioritises the significant factors, the factor levels and the interacting factors,*
- *Provides a technique for sensible decision-making to define the tuning rules,*
- *Provides analysis of robustness of the factors,*
- *Gives a process that will provide a better understanding of the controller and the system.*

5.4.1 Performance analysis for the fuzzy-PD controller

5.4.1.1 First Case

Before the analysis of the results some clarification is needed. The fact that all the SFs are significant for all of the performance criteria means that all of them must be included in the structure of the controller as well as in the tuning rules. The interaction between $KE_p \times KE_d$ is significant for both p_{PO}, and p_{ST}. This means that construction of the rules referring to these criteria needs to be applied in a way that both factors KE_p and KE_d should be change.

The performance of the tuning controller has been demonstrated by way of a long series of experiments whose results helped to establish some of its characteristics. Figure 5.7 shows the response of Equation 5.7 in different performance states. The order of Figure 5.7 *(a)* to *(f)* is based on high overshoot and low rise/settling time to low overshoot and high rise/settling time. The dashed line is the response before tuning and the solid line represents the response after tuning. Table 5.8, columns 3 to 5, show the SFs and columns 6 to 8, show the performance characteristics. The odd numbered rows refer to gain settings and response performances before the tuning method is applied, whereas the even numbered rows represent performance after tuning. Examining the 9th & 10th columns of the Table 5.8 it can be seen that the improvements in IAE and ITAE are at least 50% for all of the cases and confirms the suitability of the tuning method. Note that all of the tuning is done by a one-step procedure.

Observe that even in real systems such as the applied ac motor there is a trade off between overshoot and rise time in the results in experiment *(c) & (d)* in Table 5.8 both decreased simultaneously. Moreover, in example *(a) & (b)* the rise time increased a little which may be considered negligible.

5.4.1.2 *Second Case*

In this case different values of the dead times *(L)* have been used to identify the controller performance. These values are different to the initial ones that have been used for the analysis to optimise and tune the input SFs. As a result the applicability of the method for tuning the factor levels having any initial values is demonstrated. Figure 5.11 shows the response of Equation 5.7 before (dash line) and after (solid line) tuning the fuzzy PD controller.

Figure 5.11 *(a)* illustrates the performance results before and after tuning the fuzzy PD controller and the setting of the peaks of the MFs as well as the values of the SFs which are shown in the 2nd row of Table 5.14. The value of the dead time *(L)* is set as $L = 0.3$. Figure 5.12 *(a)* and Figure 5.13 *(a)* illustrates the MFs before and after tuning respectively. Considering the results in Figure 5.11 and Table 5.14, 12th & 13th columns, the performance of the controller has been improved more than 50% for the IAE and 80% for the ITAE.

In Figure 5.11 experiments *(b)* to *(d)* indicate the response when $L = 0.05$ in *(b)*, $L = 0.15$ in *(c)* and $L = 0.25$ in *(d)*. The initial (before tuning) settings for all dead times are identical for both input SFs and the peaks of the MFs as illustrated in the 4th, 6th and 8th row of the Table 5.14. For the setting of the MFs it can be seen from Figure 5.14 that they are almost symmetrically distributed on the universe of discourse and therefore the control surface is almost linear. After tuning both SFs and peaks of the MFs factors their values are defined as illustrated in Table 5.14 rows 5, 7 and 9. Figure 5.15 shows the tuned MFs for all dead times together with their corresponding control surface. Note that in the case of $L = 0.25$ Table 5.14, 9th to 11th columns and 8th to 9th rows, the improvement of both rise and settling time is very high where only small overshoot appears as can also be seen in Figure 5.11 *(d)*. Finally, it should be noted that the control surface in Figure 5.15 *(b), (d) & (f)* are different to the steep[1] surface in Figure 5.13 *(b)* which proves that the parameters of the MFs, thus the control action, change according to Equation 5.6 of the FCSS method.

5.5 Summary

This chapter presents a proposed systematic design and analysis of the FLC's parameters defined by eleven steps. The performance criteria as well as the independent parameters/factors are determined in the first two steps respectively. The selection of the interactions of the factors that may influence the performance characteristics of the systems under study is the third step of the procedure. The definition of possible tuning control factor levels, the number of experiments and the selection of the appropriate orthogonal array are the fourth and fifth steps respectively. The assignation of the factors to the columns of the orthogonal arrays and locating interaction columns are described in the sixth step. The experimental procedure and analysis of the experimental results is defined in the seventh step. The optimal factor levels are verified and a proposed way to construct the fuzzy rules to tune the factor levels is defined in the eight and ninth steps respectively. The construction of the fuzzy rules to tune the factor levels by using a proposed fuzzy combined scheduling system approach are the final two steps of the proposed systematic methodology. Two case studies are used for the application of the proposed method and the results are discussed analytically. In the next chapter the proposed approach is applied to optimise and tune the scaling factors for a fuzzy-like PD controller used in an underwater vehicle.

---

[1] The surface has a steeper slope, or higher gain, near the centre of the table compared to the linear surface, but they have the same values pair-wise in the four corners (Jantzen, 1997)

5.6 Reference

Astrom, J. K., and Hagglund, T. 1995. *PID Controllers.* 2nd end. Instrument Society of America.

Berkan, R. C., and Trubatch, S. L. 1997. *Fuzzy systems design principles building Fuzzy IF-THEN rule bases.* New York: IEEE Press. 0780311515.

Chatfield, C. 1983. *Statistics for Technology.* 3rd end. 0-412-25340-2.

De Silva, C. W. 1995. *Intelligent Control : Fuzzy Logic Applications.*

Derimiggio, J *WinRobust* 1995.

Dorf, R. C., and Bishop, R. H. 1995. *Modern Control System.* 7th end. Addison-Wesley.

Driankov, D.,Hellendoorn, H., and Reinfrank, M. 1993. *An introduction to fuzzy control.* Berlin, New York: Springer-Verlag. 3540563628 (Berlin). 0387563628 (New York).

Ertas, A., and Jones, J. 1996. *The Engineering Design Process.* 2nd end. NY: John Wiley & Sons, Inc.

Fisher, R. A. 1935. *The Design of Experiments.* Oliver and Boyd.

Fowlkes, Y. W. 1995. *Engineering Methods for Robust Product Design Using Taguchi Methods in Technology and Product Development.* 0-201-63367-1.

He, S-Z.,Tan, S., and Xu, F-L. 1993. Fuzzy Self-tuning of PID Controllers. *Fuzzy Sets and Systems,* **56**, pp. 37-46.

Jantzen, J. 1997. *A Robustness Study of Fuzzy Control Rules.* Proceeding Fifth European Congress on Fuzzy and Intelligent Technologies. pp. 1222-1227, Jantzen, J., 15 May 1998. "Design of Fuzzy Controllers." Denmark: Technical University of Denmark.

Lee, J. Nov. 1993. On Methods for Improving Performance of Pl-type Fuzzy Logic Controllers. *IEEE Transactions on Fuzzy Systems,* **1**, pp. 298-301.

Lochner, R., and Matar, J. E. *Design for Quality, An introduction to the best of Taguchi and western methods of statistical experimental design.* 1990. 0412400200.

Mann, G.,Hu, B-G, and Gosine, R. G. 1999. Analysis of Direct Action Fuzzy PID Controller Structures. *IEEE Trans. on Systems, Man and Cybernetics,* **29** (3), pp. 371-388.

MathWorks *MATLAB* 1999.

Mudi, K P., and Pal, R. N. 1999. A Robust Self-Tuning Scheme for PI- and PD - Type Fuzzy Controllers. *IEEE Trans. on Fuzzy Systems,* 7 (1), pp. 2-16.

Ogata, K. 1990. *Modern Control Engineering.* 2nd end. Prentice Hall Int. 0135987318.

Phadke, M. S. 1989. *Quality Engineering using Robust Design.* USA: Prentice-Hall International Inc. 0-13-745167-9.

Procyk, T. J., and Mamdani, E. H. 1979. A Linguistic Self-Organising Process Controller. *Automatica,* **15**, pp. 15-30.

Reznik, L. 1997. *Fuzzy Controllers.* UK: Reed Elsevier plc. 0-7506-3429-4.

Roy, R. 1990. *A primer on the Taguchi Method.* USA: Society of Manufacturing Engineers. 0-442-23729-4.

Sousa, J. M.,Kaymak, U., and Verbruggen, H. B. 1999. Performance Criteria: classical and Fuzzy Design. *In:* Verbruggen, H. B,Zimmermann, H.-J, and Babuska, Robert ed. *Fuzzy* *algorithms for control.* Boston: Kluwer Academic Publishers, pp. 159-183.

Sugeno, M. 1985. *Industrial Applications of Fuzzy Control.* New York: Elsevier Science Pub. Co. 0444878297.

Taguchi, G. 1987. *System of Experimental Design.* New York: Kraus Int. Publications.

Yager, R. R., and Filev, D. P. 1994b. *Essentials of Fuzzy Modelling and Control.* New York: John Wiley. 0471017612 (acid-free).

Yan, J.,Ryan, M., and Power, J. 1994. *Using Fuzzy Logic.* UK: Prentice Hall International (UK) Limited. 0-13-102732-8.

Zheng, L. 1992. A Practical Guide to Tune of Proportional and Integral (PI) Like Fuzzy Controllers. *Proc. FUZZ IEEE,* pp. 633-641.

6

Steering & Depth Control of an Underwater Vehicle (GARBI)

6.1 Introduction

This chapter presents the development of Fuzzy-like Proportional Derivative Controller (Fuzzy-like PD Controller) to control the yaw θ and the depth z of GARBI underwater vehicle (Figure 6.1) in terms of keeping the path of the navigation to the desired one, and/or changing the path according to set point changes. This makes the navigation smoother and safer, the propulsion more economical and more accurate path-keeping.

Fuzzy-like PD Controller is a type of controller that is based on the combination of Fuzzy Logic and conventional Proportional-Derivative (PD) control techniques as discussed in Chapter 5. The main advantage of this type of controller is that they can be applied to systems that are non-linear and where it is difficult to obtain a mathematical or other type of model. Note that the model developed in Chapter 2 describes the dynamics only in terms of rotation about the z-axis, fuzzy controllers, used in the structure of the Fuzzy-like PD Controller, is that they can be designed to apply heuristic rules that reflect the experiences of human experts. Moreover PD controllers can reduce overshoot and permit the use of larger gain by adding damping to the system. The derivative term can help to improve the stability of the system and makes it possible to increase the range of the other tuning parameters of the controller (Santos *et al*, 1996). Fuzzy-like PD Controller is therefore employed, because it performs well in reducing disturbances and keeping the set point to the desired one as will be presented in the experimental results in Section § 6.9.

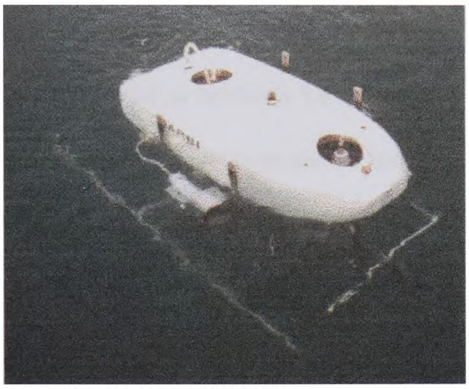


Figure 6.1 Photo of GARBI underwater robot

Structure and parameter designs are important tasks during the building of FLCs. Structure design means to determine the architecture of a controller, the input/output variables of a controller, the format of the fuzzy control rules, and the number of rules. Parameter design means determining the optimal parameters for a fuzzy controller.

For the successful design of FLCs, there is a need to properly select the optimal input and output Scaling Factors (SFs), which scale up or down the entire universe of discourse. Due to their global effect on the control performance and robustness, input and output SF's play a critical role in the Fuzzy-like PD controller and they have the highest priority in terms of tuning and optimisation (Mudi and Pal, 1999).

The rest of the chapter is organised as follows: Section § 6.2 presents the tasks that the controller of GARBI underwater vehicle has to undertake. In Section § 6.3 the procedure of how to design a Fuzzy-like PD controller are described extensively based on the approach proposed in Chapter 5. Analytically, in Section § 6.3.1 the design aspects of the "FLC" part of the Fuzzy-like PD controller are discussed i.e., input/output universe of discourse and linguistic variables – MFs, construction of the rule base, operators and defuzzification method. In Section § 6.3.2 the design aspects of the SFs of the controller are also discussed. The design of experiments to obtain the optimum and tuning values of the parameters of the Fuzzy-like PD Controller for GARBI is presented in Section § 6.4. Moreover, the definition of the performance criteria, of the parameters/factors *(f)*, the possible tuning control factor levels *(l)* and the selection of the appropriate orthogonal array are defined in Sections § 6.4.1 § 6.4.2 § 6.4.3. In Section § 6.5 the results of the experiments conducted in a real environment are presented. Section § 6.6 analyses the results to define the optimal parameters of the Fuzzy-like PD controller using analysis of means (Section § 6.6.1) and analysis of variance (Section § 6.6.2). The way to construct GARBI's Fuzzy-like PD controller's tuning fuzzy rules is described in Section § 6.7. In Section § 6.8 the Fuzzy Combined Scheduling System (FCSS) approach is applied. The verification and tuning of both yaw and depth Fuzzy-like PD controllers' parameters based on new experimental trials is discussed in Section § 6.9. Finally, a Summary of the chapter is presented in Section § 6.10.

6.2 Control tasks of GARBI underwater vehicle

As in any underwater vehicle, the dynamics of GARBI are coupled and highly non-linear. When designing GARBI's controller it is necessary to compensate for its non-linear dynamics and kinematics, non-linearities due to thrusters and pressure hysteresis, barometer dead-zones, and the noise in yaw and depth measurements. Therefore, robust controllers that reliably perform complex tasks in the face of the above uncertainties should be used. Fuzzy-like PD Controller is designed to make the vehicle follow the commands from the pilot in terms of course-changing and course-keeping of both yaw angle and depth of the robot.

Controllers for course-keeping and/or course-changing are normally based on feedback from a gyrocompass measuring the heading for the yaw and air pressure-sensors measuring the difference of the pressure inside and outside of the robot for depth.

The control objective for a course-keeping controller can be expressed as $\theta, z = constant$. For course-changing, the objective is to follow the changes of the pilot commands with the best control performance in terms of small overshoot, settling time and steady state error.

Figure 6.2 shows a simplified scheme of course-keeping/course-changing control configuration. The structure uses independent FLCs for each controlled variable (Yaw and Depth), greatly simplifying the design at the cost of some decrease in performance. The corresponding inputs of these controllers are the error e_θ between the real and the desired yaw angle and the error e_z between the real and the desired heave position as well as the corresponding change of the above errors Δe_θ and Δe_z (see Figure 6.3 and Figure 6.4)

$$e_\theta(nT) = \theta_{sp} - \theta(nT) \tag{6.1}$$

$$e_z(nT) = z_{sp} - z(nT) \tag{6.2}$$

$$\Delta e_\psi(nT) = e_\psi(nT) - e_\psi(nT-1) \tag{6.3}$$

$$\Delta e_z(nT) = e_z(nT) - e_z(nT-T) \tag{6.4}$$

where $e(nT)$, $\Delta e(nT)$ and $\theta, z(nT)$ designate crisp error rate and process output at sampling time nT respectively. The computed rate (Δe) may not be the actual one due to delays and noise of the measurements. To overcome this problem a rate gyro should be used. Unfortunately, in the GARBI underwater vehicle the above device was not available.

The corresponding outputs of the controllers are; for the first controller the moment N around the z-axis and for the second controller the force Z of the two propellers in the z-direction. As can be seen in Figure 2.3, the rotation N is related to the difference of power between the propellers T_1 and T_2 in the x-direction. The force Z in the z-direction relates to the power of the propellers T_3 and T_4, which is always equal and of the same polarity. As a convention, signals are written in lower case before gains/SFs and upper case after gains/SFs, for instance $E = S_e * e$.

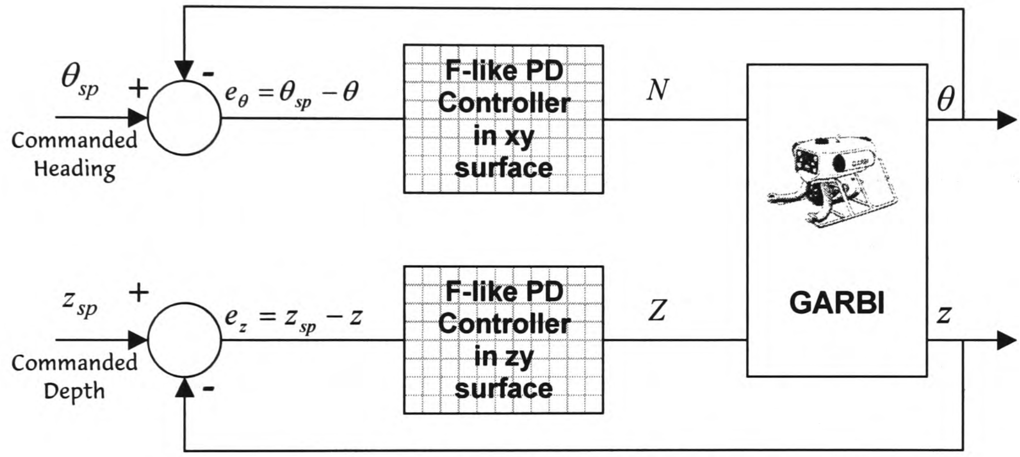


Figure 6.2 Control loop for GARBI

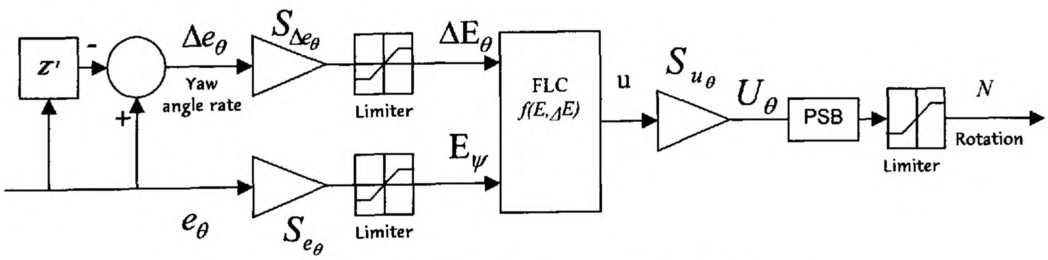


Figure 6.3 Yaw Fuzzy-like PD controller

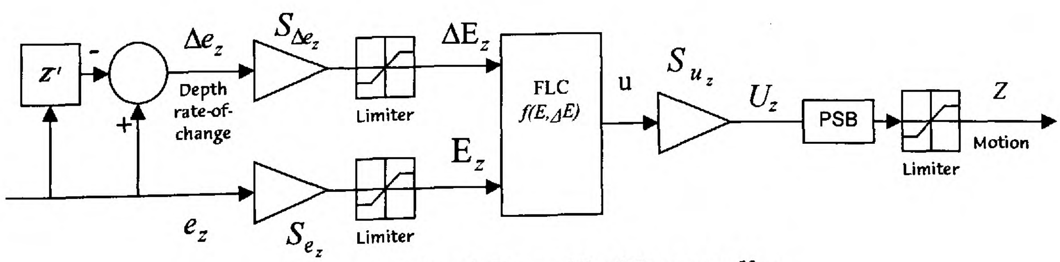


Figure 6.4 Depth Fuzzy-like PD controller

6.3 Designing the Fuzzy-like PD controller for GARBI

In studying the dynamic properties of the fuzzy controller, the model of the process is needed so that the impact of the successive control actions may be monitored. Since a model of GARBI is not available, the dynamic properties of the closed loop structure have to be derived intuitively and experimentally. This is one of the important features of the idea of fuzzy controllers. However, the tuning of Fuzzy-like PD Control systems is a fundamental problem, especially for optimum performance and therefore, more sophisticated procedures than the tuning of conventional controllers is needed. The reason for this is that FLC is an extremely flexible system, whose behaviour is determined by a large number of parameters. There are two different levels of tuning during the design of Fuzzy-like PD controllers. The first level includes the structure, the rule base, the antecedent and consequent MFs together with their distribution, the inference mechanism and the defuzzification strategy. The second level is the tuning of gain parameters. This includes the SFs and other gains used in building the structure. As described in Chapter 5 the dynamic properties of the controller can be adjusted by a series of carefully designed experiments. As the experiments for an underwater vehicle in a real environment are expensive and time consuming, the minimum number of experiments must be executed. However, the risk of losing vital information that can result from large amount of data can be overcome if the experiments are designed using the appropriate orthogonal array of the Taguchi Design of Experiment Method (Fowlkes, 1995) to find the set of optimum input/output parameters. This is explained in detail in Sections § 6.4 to § 6.5.

6.3.1 Design aspects of the "FLC" part of the Fuzzy-like PD controller (first level)

The design aspects of the FLC is a unified approach to determine both its parameters (linguistic labels and reference fuzzy sets) and its structure (rule-base) (Yager and Filev, 1994b).

Specifically the criteria to set the design procedure for the FLC part of the Fuzzy-like PD controller for both yaw and depth control are presented as follows:

6.3.1.1 Input/output universe of discourse.

Each universe is restricted to an interval that is related to the maximal and minimal possible values of the respective variable, that is, to the operating range of the variable. Both inputs of yaw i.e. E_θ, ΔE_θ and depth i.e. E_z, ΔE_z controllers are operated in the whole range of their universe. Therefore, for the universe of the yaw and depth controllers the maximal limits are as in Equation 6.5 to Equation 6.8 and the minimal limits are as in Equation 6.9 to Equation 6.12 respectively

$$E_{\theta_{\max}} = e_{\theta_{\max}} * S_{\theta_e} = YawUniverse_{\max} \quad (6.5)$$

$$\Delta E_{\theta_{\max}} = \Delta e_{\theta_{\max}} * S_{\theta_{\Delta e}} = YawUniverse_{\max} \quad (6.6)$$

$$E_{z_{\max}} = e_{z_{\max}} * S_{z_e} = DepthUniverse_{\max} \quad (6.7)$$

$$\Delta E_{z_{\max}} = \Delta e_{z_{\max}} * S_{z_{\Delta e}} = DepthUniverse_{\max} \quad (6.8)$$

$$E_{\theta_{\min}} = e_{\theta_{\min}} * S_{\theta_e} = YawUniverse_{\min} \quad (6.9)$$

$$\Delta E_{\theta_{\min}} = \Delta e_{\theta_{\min}} * S_{\theta_{\Delta e}} = YawUniverse_{\min} \quad (6.10)$$

$$E_{z_{\min}} = e_{z_{\min}} * S_{z_e} = DepthUniverse_{\min} \quad (6.11)$$

$$\Delta E_{z_{\min}} = \Delta e_{z_{\min}} * S_{z_{\Delta e}} = DepthUniverse_{\min} \quad (6.12)$$

For the yaw controller two different universes are applied. The first is within the range of -90° to $+90^\circ$ degrees and the second is within the range -180° to $+180^\circ$ degrees in the cases where

"Small" or "Big" changes occur in GARBI's navigation path respectively. Moreover for the depth controller the universe is in a range of 10 meters. Thus the depth Fuzzy-like PD controller rules are fired only within this range.

For simplification and unification of the design of the FLC and its computer implementation, however, it is more convenient to operate with *normalised universes of discourse* of the input/output variables of the FLC. The normalised universes are well-defined domains; the fuzzy values of the input/output variables are fuzzy subsets of these domains. In general, the normalised universes can be identical to the real operating ranges of the variables, however, in this application both input error and change-of-error of both yaw and depth fuzzy controllers coincide with the closed interval [-1,1].

In the output of both controllers, the minimum and maximum values ($V_{\min}$, $V_{\max}$) of voltage that can be applied for all of GARBI's propellers are 3V and 10V respectively. However these upper and lower voltage limits may change due to the hardware modifications. Therefore, it could be practically more efficient to normalise between 3 to 10 V. As a result the output of each controller is:

$$N_{xy} = V_{\min} + \frac{U_\theta}{V_{\max}} \tag{6.13}$$

$$Z_{zy} = V_{\min} + \frac{U_z}{V_{\max}} \tag{6.14}$$

U_θ and U_z are the output voltages from both *yaw* and *depth* FLCs respectively where

$$U_\theta = u_\theta \cdot S_{u_\theta} \tag{6.15}$$

$$U_z = u_z \cdot S_{u_z} \tag{6.16}$$

The above normalisation procedure is fitted into a Power Scaling Block (PSB) as can be seen in Figure 6.3 and Figure 6.4. As can also be seen in those figures, limiters are set before the inputs of GARBI's FLC to ensure that the rules will not be fired in case saturation exists due to possible noise disturbances of yaw and depth measurements. Moreover, limiters also exist in the output part of both controllers. That is to ensure that the output commands to the propellers are within their specification properties in case for instance of improper design of the controller's parameters.

6.3.1.2 Input/output linguistic variables - MFs.

The choice of the shape of the antecedent MFs is triangular $(\mu_1(x), \mu_2(x), ..., \mu_n(x))$ with a specific overlap of 0.5. This means that the height of the intersection of the two successive fuzzy sets is:

$$hgt(\mu_i \cap \mu_{i+1}) = \frac{1}{2} \tag{6.17}$$

The overlapping of 50% ensures that each value of the universe is a member of at least two sets, except possibly for elements at the extreme ends. Note that if there is a gap between two sets no rules fire for values in the gap and therefore, the control function is not defined (Jantzen, 1998).

The descriptions of these MFs used in the control algorithm are either in functional form or are defined numerically. Thus, in this case the functional form is used due to their simplicity in calculating the degree of freedom of the rules. Additionally, due to normalisation of the universe of discourse procedure, the functional-type MFs changes their parameters subsequently (Yager and Filev, 1994b). The mathematical construction of these triangular MFs with overlap of 0.5 is as in Table 6.1.

	Triangular MF	
Left	$\mu^L(u)=\begin{cases}1 \\ \max\left\{0,1+\dfrac{c^L-u}{0.5\omega^L}\right\}\end{cases}$	if $u \le c^L$ otherwise
Centre	$\mu^C(u)=\begin{cases}\max\left\{0,1+\dfrac{u-c}{0.5\omega^L}\right\} \\ \max\left\{0,1+\dfrac{c-u}{0.5\omega^L}\right\}\end{cases}$	if $u \le c$ otherwise
Right	$\mu^L(u)=\begin{cases}\max\left\{0,1+\dfrac{u-c^R}{0.5\omega^L}\right\} \\ 1\end{cases}$	if $u \le c^R$ otherwise

Table 6.1 Mathematical Characterisation of Triangular MFs

Notice that c is the centre of the triangular MF and ω is the base-width, c^L specifies the "saturation point" and ω^L specifies the slope of the non-unity and non-zero part of μ^L. The selection of MF has two important characteristics: one is its optimal interface design and the other is its semantic integrity (Pedrycz, 1994). The concept of these characteristics described in the following:

- *Optimal interface design.*

  Error free Reconstruction: In a fuzzy system a numerical value is converted into a linguistic value by means of fuzzification. A defuzzification method should guarantee that this linguistic value can be reconstructed in the same numerical value as in Equation 6.18.

$$\forall x \in [a.b]: \qquad f^{-1}[f(x)] = x \tag{6.18}$$

- *Semantic integrity.*

Justify the labels and number of Elements: The number of sets defined is seven. As also mentioned in Chapter 2, this number comes from the recommendation that the number of the sets should be compatible with the number of "quantifiers" that humans being can handle which actually is within the limit of 7 ± 2 distinct terms (Espinosa and Vandewalle, 1997). The input/output variables of GARBI's FLC are quantified into sets of classes defined by linguistic labels such as "Positive Big" (PB), "Positive Medium" (PM), "Positive Small" (PS), "Zero" (ZO), "Negative Small" (NS), "Negative Medium" (NM) and "Negative Big" (NB). Linguistic labels (terms) and their associated fuzzy sets form a fuzzy partitioning of the (normalised) universe of discourse. Thus, both the *error* and *change of error* term in Figure 6.5 determine a fuzzy partitioning of the universe [-1,1] into seven fuzzy sets. The number of linguistic labels in one set, determines the cardinality of the fuzzy partitioning of that universe of discourse. Cardinalities of the sets associated with the inputs of the FLC define the maximum number of rules contained in its rule-base. Therefore, for GARBI's FLC, the partitioning of the universe of discourse for both the *error* and *change of error* into seven fuzzy sets each, results in the maximal number of 49 rules.

Distinguishability. Each of the linguistic labels should have semantic meaning and the fuzzy sets should clearly define a range in the universe of discourse. So, the MFs should be clearly different. The assumption of the overlap equal to 0.5 assures that the support of each fuzzy set will be different. The distance between the modal values of the MFs is also very important to make sure that the MFs can be distinguished. The modal value of the used MFs is defined as the $a-cut$ with $a=1$

$$\mu_{i(a=1)}(x), \qquad i=1,2,\ldots,N \tag{6.19}$$

Completeness is an important characteristic of the fuzzy sets (Lee, 1990a), (Pedrycz, 1993). It expresses the ability of the fuzzy control algorithm to infer a control action with confidence not less than a minimal level ε, for which the threshold $\varepsilon - cuts$ of all terms cover the interval universe. Decreasing this parameter decreases the fuzziness of the partitioning of the input space of the FLC. Based on heuristic considerations some authors suggest minimal completeness level $\varepsilon = 0.25$ (Kosko, 1992); (Berkan and Trubatch, 1997). As can be seen from Figure 6.5 for both antecedents' MFs the universe of discourse for each variable is uniformly partitioned and the MFs are placed with 50% overlap and therefore the level of completeness is $\varepsilon = 0.5$. This actually is a fixed structure for initial setting of most FLCs.

Normalisation. Due to the fact that each linguistic label has semantic meaning, at least one of the values in the universe of discourse should have a membership degree equal to one. In this particular FLC design all of them do so and therefore all the fuzzy sets are "normal".

The MFs of the consequences are singletons ξ with seven linguistic labels: "Positive Big" (PB), "Positive Medium" (PM), "Positive Small" (PS), "Zero" (ZO), "Negative Small" (NS), "Negative Medium" (NM) and "Negative Big" (NB) equally distributed between –1 to +1 values (see Figure 6.6).

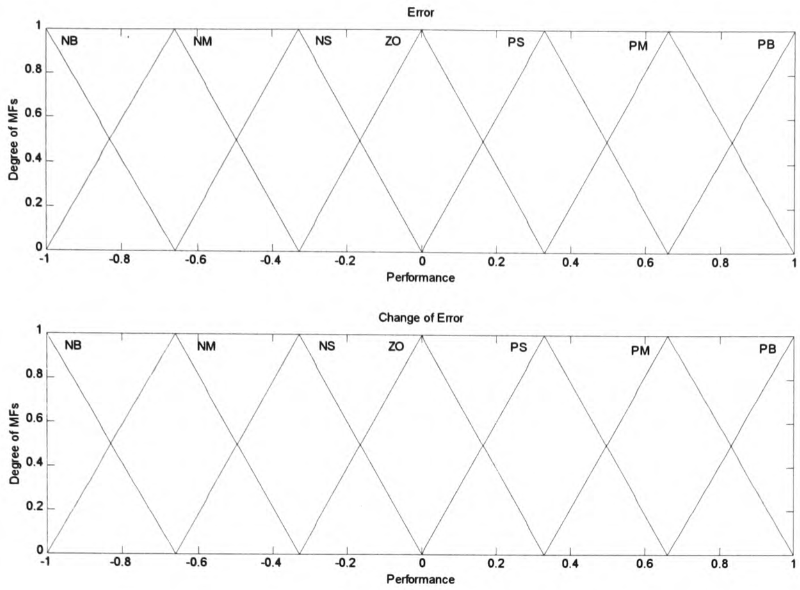


Figure 6.5 Input MFs for GARBI's FLC

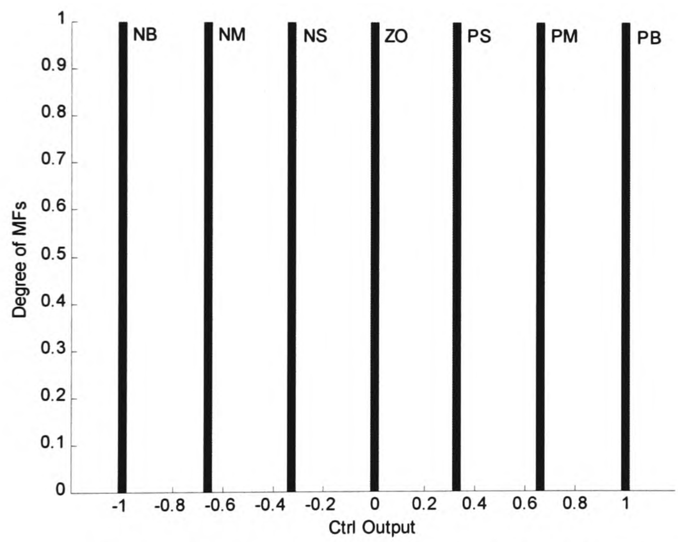


Figure 6.6 Singleton Output sets for the FLCs

By using MFs in the input and singletons in the output of GARBI's control system, the actual Takagi-Sugeno fuzzy system approach (Takagi and Sugeno, 1985) is utilised. Mamdani control approach (Mamdani and Assilian, 1975) is not used due to its computational complexity during the defuzzification procedure which is time consuming. It is well know, however, that fuzzy rules with singletons can be use without losing the performance of the control (Sugeno *et al*, 1993).

Finally it is important to remark that the use of singletons gives more linguistic meaning to the rules, whereas the Takagi-Sugeno model can improve the approximation properties of the controller (Passino *et al*, 1998). It is therefore advisable for real time fuzzy control applications to use singletons ξ in the output. This results in simpler and faster control action (Jantzen, 1998); (Nguyen *et al*, 1999).

6.3.1.3 Construction of the Rule (knowledge) Base

The construction of the rule-base is the crucial and the most difficult aspect of the FLC design (Lee, 1990a). It is also one reason for criticism of fuzzy logic control because, in general, there are no systematic tools for forming the rule-base of the FLC. However there are two main notable methods (Yager and Filev, 1994b) based on:

- *Intuitive knowledge and experience*
- *Use of the concept of a template rule-base*

In the first method, the FLC is designed as a simple expert system and therefore, different sources of knowledge, resulting in formulation of alternative rule bases, can be considered. One reasonable source is the knowledge, based on the experience of an operator, controlling a given system. This allows introduction of "rule of thumb" experience in the control strategy. Usually it is difficult to extract control skills from the operator in a form that can be useful for construction of the rule-base of FLCs. Moreover, there is no reason to believe that these rules are the best control strategies. Most FLC's combine an approach based on the operator's experience with a good understanding of systems and control theory and this has proved satisfactory during the

70's and 80's in industrial applications of fuzzy control (Bernard, 1988); (Sugeno, 1985); (Tong, 1985).

In the second method the *template rule-base* is regarded as a basic tool uniting the common engineering sense and experience in fuzzy logic control. MacVicar-Whelan (MacVicar-Whelan, 1977) developed this type of rule-base template which was introduced in the first FLC, (Mamdani and Assilian, 1975); (King and Mamdani, 1977). The MacVicar-Whelan rule-base summarises the rules used in the rule-bases of these FLCs, and in addition includes situations (combinations of linguistic labels of input and output variables of the FLC) that were not defined. The expansion of the original rule-bases is established on the following three metarules (Tang and Mulholland, 1987):

1. *If both* $e(k)$ $\Delta e(k)$ *are zero, then maintain present control setting*

2. *If conditions are such that* $e(k)$ *will go to zero at a satisfactory rate, then maintain present control setting*

3. *If* $e(k)$ *is not self-correcting, then control action* $\Delta e(k)$ *is not zero and depends on the sign and magnitude (small, medium, large, etc.) of* $e(k)$ *and* $\Delta e(k)$.

As operators' expert knowledge for GARBI's dynamic properties are not available, the MacVicar-Whelan rule-base template is used in the FLC part of GARBI's Fuzzy-like PD Controller. As mentioned in Section § 6.3.1.2, each of the FLC blocks contains 49 rules. The cell defined by the intersection of the first row and the first column represents a rule such as,

$$\textit{if } e(nT) \text{ is NB } \textit{and } \Delta e(nT) \text{ is NB } \textit{then } u(nT) \text{ is NB}$$

The rule base is presented in the table format shown in Table 6.2:

e\ Δe	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

Table 6.2 The Rule Base of a Fuzzy-like PD in tabular form.

Analytical explanation of how the above rule base table is designed can be found in Reznik, (1997).

6.3.1.4 Operators

Using the min operation for the aggregation *AND* (outer product) of the fuzzy rules, the output fuzzy set is given by $\mu_u = \min(\mu_e, \mu_{\Delta e})$. Thus, the Fuzzy-like PD controller is a controller where the output is a non-linear function of the error, *e* and its derivative *de/dt* ($u = F(e, de/dt)$), where F is a non-linear function of two variables.

6.3.1.5 Defuzzification

The control signal results from the defuzzification method that uses the degree of membership functions of the antecedent and the singleton of the consequences of the MFs obtained by:

$$u = \frac{\sum_{i=1}^{R} \xi_i \mu_i}{\sum_{i=1}^{R} \mu_i} \tag{6.20}$$

where μ_i is the degree of MF defined in Table 6.1, ξ_i is the singleton's value and R is the number of rules.

6.3.2 Design aspects of the SFs of the controller (second level)

One way of improving the dynamic properties of the Fuzzy-like PD control systems is to optimise and adjust (tune) the constructing parameters. However, there is no general method for tuning them. Most successful results reported are based on the combination of expert understanding about the controlled object and the use of the analogies between the FL and PID controllers (Zheng, 1992).

As discussed in Chapter 5, SFs are the main parameters used for tuning the FLC. SFs play an important role in the formation of the dynamics of the close-loop structure leading to the desired response of the controlled system (Zheng, 1992). The importance of an optimal choice of input SFs is evidently shown by the fact that inappropriate scaling results is either shifting the operating area to the boundaries or utilising only a small area of the normalised universe of discourse. Additionally, the adjustments of the output SF affects the close-loop gain, which has, direct influence on stability and oscillation tendency.

For the Fuzzy-like PD controller, the corresponding SFs are, for the inputs $S_e, S_{\Delta e}$ and for the output S_u respectively. These factors influence the dynamics of the system as:

- if both S_e and $S_{\Delta e}$ increase, the control becomes more sensitive around the set point until oscillations are observed,
- where S_e and $S_{\Delta e}$ decrease, a tolerance band exists around the set point and a large steady-state error is quite common. So if the SFs are too small, the system gives a poor response,
- S_u affects the proportional gain, so it is desirable to have them as large as possible without creating too much overshoot. If too small, the system will be too slow, and if too large the system might become unstable,

In addition as discussed by (Procyk and Mamdani, 1979):

- High values of S_e results in good responsiveness of the system (low steady-state error, and rise time), but they lead to poor stability (large overshoot). Analogously low values of S_e lead to a poor response.

- Faster convergence is bounded by high values of S_e and $S_{\Delta e}$, and relatively low values of S_u.

Despite the above general directions to optimise and adjust (tune) the SFs, their effectiveness is bounded by the contradictory requirements resulting from different performance measures. Additionally, the interactions and the significance of the SFs vary according to the system. However, a systematic approach (proposed in Chapter 5) to overcome the above drawbacks and determine the parameters together with their tuning and robust performance of GARBI's Fuzzy-like PD controller is applied as described in Section § 6.4.

6.3.3 Sampling time

The sampling time is set to 1 second. This is actually the smallest sampling time that could be used due to GARBI's communication line specifications. However, its sensitivity to the input parameters is investigated during the design of the experiments as explained in Section § 6.4.2. Note that if the sampling time is too small the computation of Δe may become too sensitive to noise. This normally shows up as a "restless" control signal.

6.3.4 Software implementation of the Fuzzy-like PD controller's of GARBI

Where the design properties of the Fuzzy-like PD controller's of GARBI is defined, its software was developed at the Mechatronics Research Centre and tested in the Computer Vision and Robotics Lab at the University of Girona. The software routines were implemented in Borland

C++ ver 5.2 and LabWindows CVI ver 5 under Windows 95/98/NT operating system. The main source code of the control software is illustrated by the flow diagram shown in Figure 6.7. The communication protocol block is to check the communication properties between the surface computer and GARBI's local unit computer; the mission setting block is developed to set the yaw and depth mission properties and the two controllers are working independently to command the propeller power settings. The run time for both controllers' routines was less than 9ms using PENTIUM II 333 MHz processor. This was less that the smallest sampling time (1sec) that can be used. The software was tested in terms of the communication abilities of the robot with the host computer in the Laboratory.

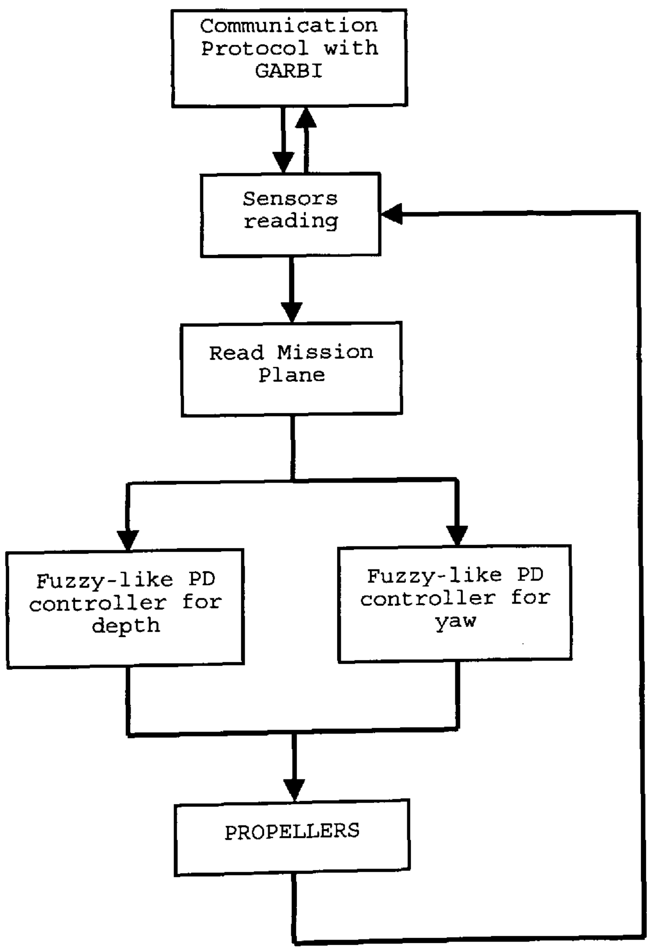


Figure 6.7 Flow diagram of the source code developed for GARBI's controller

6.4 Design of Experiments to obtain the optimum and tuning values of the parameters of the Fuzzy-like PD Controller for GARBI

Optimising a control system's design means determining the best architecture, levels of control factors, and tolerance. In Chapter 5, a systematic methodology of how to identify the optimum values of Fuzzy-like PD controllers in terms of control performance and robustness was developed. This includes defining the relationships between control parameters and performance, investigating which parameters are more important and which are not, and by how much. The proposed method is based on experiments using Taguchi technique of experimentation. The method is applied to GARBI's Fuzzy-like PD Controller to optimise and tune the SFs.

This method includes the following steps:

- *Definition of the performance criteria*
- *Definition of the parameters/factors (f) and the possible tuning control factor levels (l)*
- *Selecting and assigning factors to the columns of the appropriate Orthogonal Array*
- *Conducting the Experiments (in a real environment)*
- *Analysis of the results to define the optimal parameters of the Fuzzy-like PD controller*
- *Constructing GARBI's Fuzzy-like PD controller's tuning fuzzy rules*
- *Applying the Fuzzy Combined Scheduling System (FCSS) approach*
- *Verification and tuning of both yaw and depth Fuzzy-like PD controllers' parameters based on new experimental trials*

Description of the above steps is presented in the following sections.

6.4.1 *Definition of the performance criteria*

The performance criteria of the system has been chosen as the Integral Absolute-Error (IAE) and Integral-of-time-multiplied Absolute-Error (ITAE) (Ogata, 1990). Large errors are penalised heavily using IAE, however, using ITAE, the contribution of large initial errors is reduced whereas errors occurring late in time are penalised heavily. Therefore, IAE reflects the transient response and ITAE reflects the steady-state response. The optimal controller in this case is the one that minimises these integrals. Thus, the performance criteria vector p_o includes the above performance measurements:

$$p_O = \{p_{IAE_{yaw}}, p_{ITAE_{yaw}}, p_{IAE_{depth}}, p_{ITAE_{depth}}\} \tag{6.21}$$

where $p_{IAE_{yaw}}, p_{IAE_{depth}}$ and $p_{ITAE_{yaw}}, p_{ITAE_{depth}}$ are the IAE and ITAE for yaw and depth performance respectively.

6.4.2 *Definition of the parameters/factors (f) and the possible tuning control factor levels (l)*

The SFs are the parameters/factors of both *yaw* and *depth* Fuzzy-like PD controllers of GARBI have been defined. When a SF is changed, it is assumed that the definition of each membership function will be changed by the same ratio. Hence changing of any SF can change the meaning of one part, the IF-part or THEN-part, in any rule. Therefore, it can be said that the change of SFs may affect all of the control rules in rule Table 6.2.

For both *yaw* and *depth* controller, the SFs are applied in their corresponding input (i.e. $S_{e_\theta}, S_{\Delta e_\theta}$, $S_{e_z}, S_{\Delta e_z}$) and outputs (i.e. S_{u_θ}, S_{u_z}) as illustrated in Figure 6.3, Figure 6.4 and therefore the factor's vectors are $f = \{S_{e_\theta}, S_{\Delta e_\theta}, S_{u_\theta}\}$ and $f = \{S_{e_z}, S_{\Delta e_z}, S_{u_z}\}$ respectively.

Three factorial levels are chosen for all SF of both *yaw* and *depth* controllers. The choice of their values is based on min, max and an intermediate value that excites the response of the system. Their values are for:

Input SFs:

$$S_{e_\psi} \quad l_{S_{e_\psi}} = \{S_{e_{\theta 1}}, S_{e_{\theta 2}}, S_{e_{\theta 3}}\} = \{0.5, 0.75, 1\} \tag{6.22}$$

$$S_{e_z} \quad l_{S_{e_z}} = \{S_{e_{z1}}, S_{e_{z2}}, S_{e_{z3}}\} = \{0.5, 0.75, 1\} \tag{6.23}$$

$$S_{\Delta e_\psi} \quad l_{S_{\Delta e_\psi}} = \{S_{\Delta e_{\theta 1}}, S_{\Delta e_{\theta 2}}, S_{\Delta e_{\theta 3}}\} = \{0.5, 1, 2\} \tag{6.24}$$

$$S_{\Delta e_z} \quad l_{S_{\Delta e_z}} = \{S_{\Delta e_{z1}}, S_{\Delta e_{z2}}, S_{\Delta e_{z3}}\} = \{0.5, 1, 2\} \tag{6.25}$$

Output SF:

$$S_{u_\psi} \quad l_{S_{u_\theta}} = \{S_{u_{\theta 1}}, S_{u_{\theta 2}}, S_{u_{\theta 3}}\} = \{3, 7, 10\} \tag{6.26}$$

$$S_{u_z} \quad l_{S_{u_z}} = \{S_{u_{z1}}, S_{u_{z2}}, S_{u_{z3}}\} = \{3, 7, 10\} \tag{6.27}$$

which are the possible optimal tuning factor levels as defined in Chapter 5 (Section § 5.2.4).

The minimum value of all SFs i.e. $S_{e_{\theta 1}}$, $S_{e_{z1}}$, $S_{\Delta e_{\theta 1}}$, $S_{\Delta e_{z1}}$, $S_{u_{\theta 1}}$, $S_{u_{z1}}$ have been derived experimentally as the lowest ones which excites the systems' response when the input/output signals have the smallest values.

For the SFs S_{e_θ}, S_{e_z} the error e input signal is shifting in the universe of discourse from the initial values i.e. where $S_{e_{\theta 3}}, S_{e_{z3}} = 1$ to smaller ones i.e. where $S_{e_{\theta 2}}, S_{e_{z2}} = 0.75$ and $S_{e_{\theta 1}}, S_{e_{z1}} = 0.5$. As a result, investigating the set of rules, in the rule Table 6.2, their excitement in this case is shifted to the labels with small values. Moreover for the SFs $S_{\Delta e_\theta}, S_{\Delta e_z}$ the change-of-error Δe input signal is shifting in the universe of discourse from the initial values i.e. where $S_{\Delta e_{\theta 2}}, S_{\Delta e_{z2}} = 1$ to a smaller one i.e. where $S_{\Delta e_{\theta 1}}, S_{\Delta e_{z1}} = 0.5$ and/or to a bigger one i.e. where $S_{\Delta e_{\theta 3}}, S_{\Delta e_{z3}} = 2$. Therefore, the rules that are fired are shifting to the ones with labels corresponding to small and/or big values. Note that the SFs $S_{\Delta e_\theta}$, $S_{\Delta e_z}$ the level's value is chosen as 2 so that it may assist the response of these factors in case Δe becomes very small and thus meaningless. Hence if the final optimal level's value is 2, this implies that the sampling time should be increased.

The output of the controller is a value that is multiplied by the output SF's, to give the voltage that is be applied to the robot's propellers.

6.4.3 Selecting and assigning factors to the columns of the appropriate Orthogonal Array

The orthogonal array of the Taguchi Method was used to plan the experimentation for the underwater robot. Three factors ($S_e, S_{\Delta e}, S_u$) at three levels each are defined as explained in Section § 6.4.2. With a full factorial, this would result in 3^3 (27) different experiments. However, using Equation 5.1 for a three factor three level experiment, six degrees of freedom exist, so an orthogonal array with nine experimental runs can be employed instead. Table 6.3 shows the orthogonal array (Phadke, 1989) for both *Yaw* and *Depth* experiments that is sufficient for this study.

No Exp	S_e	$S_{\Delta e}$	S_u
1	0.5	0.5	3
2	0.5	1	7
3	0.5	2	10
4	1	0.5	7
5	1	1	10
6	1	2	3
7	0.75	0.5	10
8	0.75	1	3
9	0.75	2	7

Table 6.3 The Orthogonal array with n ine experiments (Phadke, 1989)

6.5 Conducting the Experim ents (in a real environment)

Experimental trials in a real environment (Lake Banyolas, Spain) were held in Oct 1999 to test both depth and yaw Fuzzy-like PD controller. Before the trials into the lake the gyrocompass was calibrated to set its zero degree state with the horizon. Note that as the device is electromagnetic, its calibration is important before the experimental trials.

As mentioned in Section § 6.3.1.1 the voltage used is in the range of 3 to 10 Volts (V). For heading control the opposite voltage between the horizontal propellers (T_1, T_2) was used i.e. $+V_1$, $-V_2$. So, if the heading angle is turned to $\theta°$ clockwise for instance, the voltage in the right propeller T_1 is reduced and the voltage in the left propeller T_2 is increased by the same amount.

Using Table 6.3, the experiments to investigate both yaw and depth control performance were undertaken. The navigation plan for the *yaw* experiments was:

- *initial voltage of the horizontal and vertical propellers is set to 3V for a period of 60 seconds to make sure that the vehicle goes straight ahead. Equal power to the horizontal propellers are employed the vehicle is moving away from the platform in the water*

- *manoeuvring with set point of* 90° *for experiments 1, 5, 6, 7, 9 and* 180° *for the experiments 2, 3, 4, 8. After the manoeuvring, the task is to keep the vehicle moving in the same direction.*

and for the *depth* experiments was:

- *changing the depth course from 0 meters to 10m and then to 5m and then keeping it at this depth.*

Figure 6.8 to Figure 6.16 and Figure 6.17 to Figure 6.25 illustrates the yaw and depth response together with the control output, the error e and the change-of-error Δe respectively.

It should be noted that due to the failure of one of the power cards for the vertical propeller during the experiments it was decided to switch off the vertical propellers and add more weights to keep the vehicle in the equilibrium position a few meters under the water. For the depth experiments however, the power cards were swapped from horizontal to vertical.

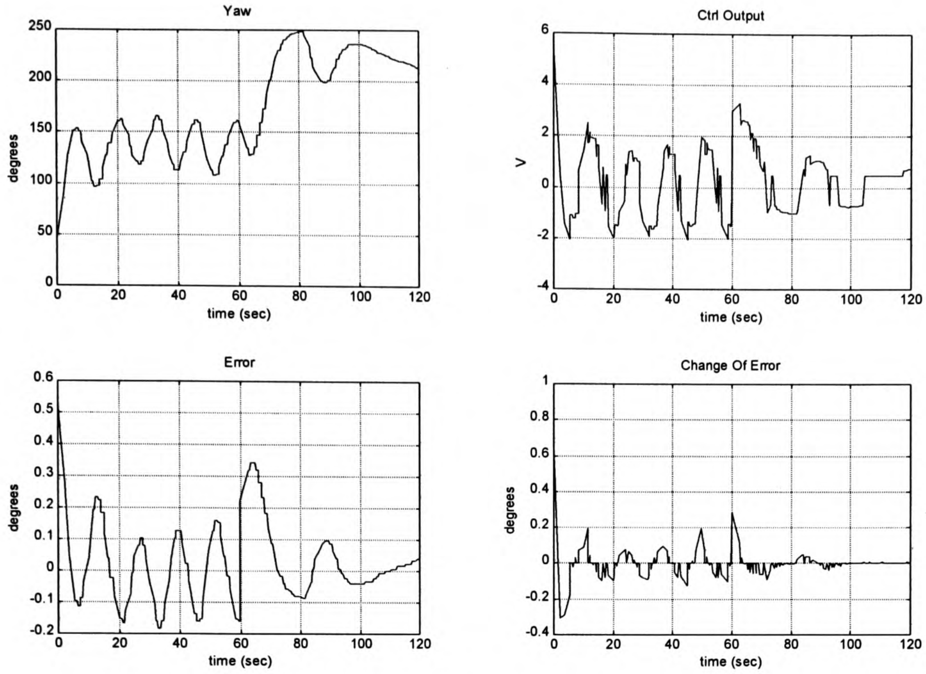


Figure 6.8 Yaw experimental results for the 1st trial ($S_{e_\theta} = 0.5$, $S_{\Delta e_\theta} = 0.5$ *and* $S_{u_\theta} = 3$)

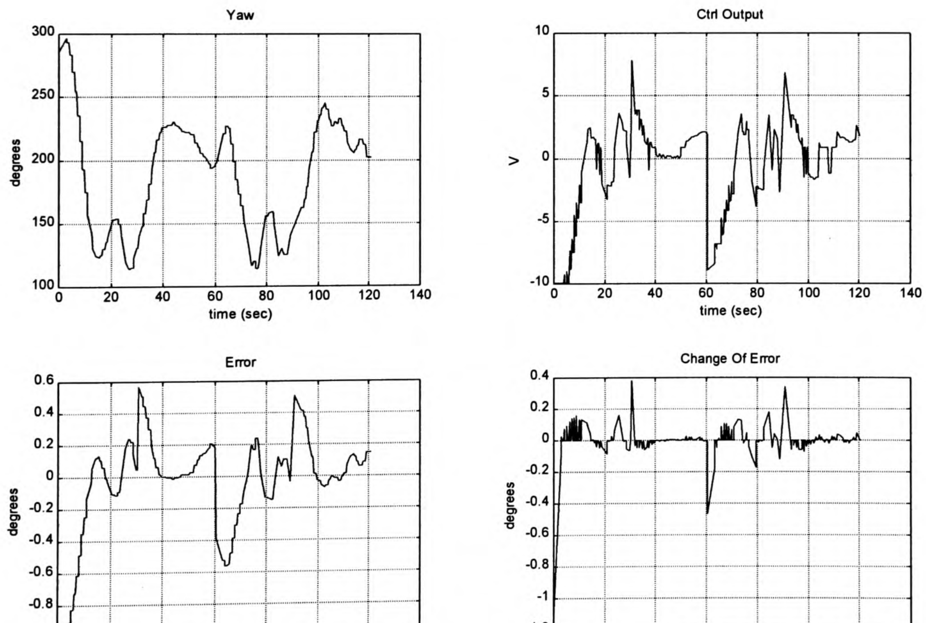


Figure 6.9 Yaw experimental results for the 2nd trial. ($S_{e_\theta} = 0.5$, $S_{\Delta e_\theta} = 1$ *and* $S_{u_\theta} = 7$)

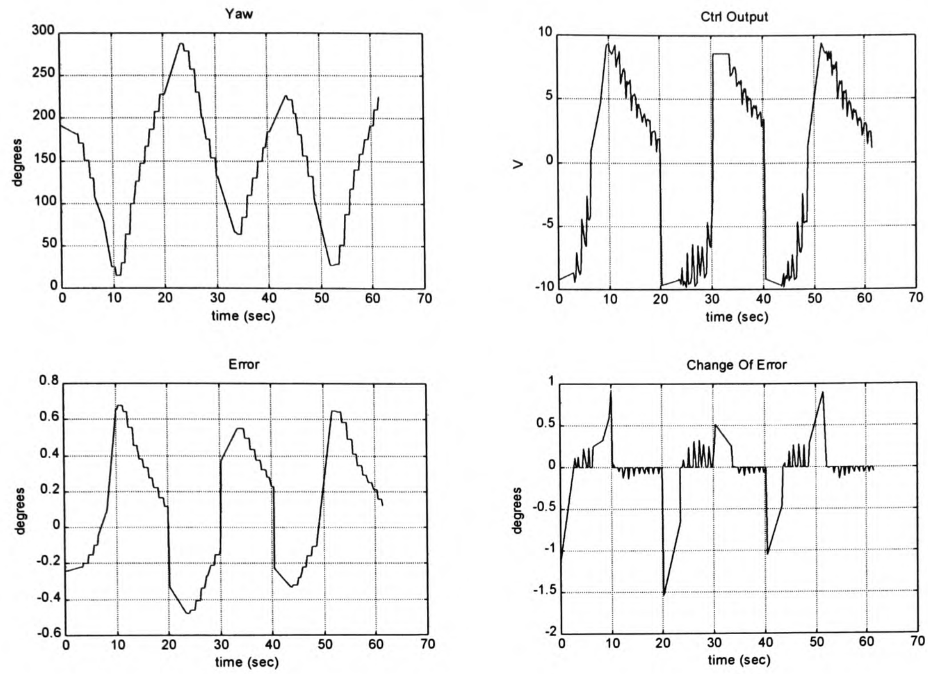


Figure 6.10 Yaw experimental results for 3^{rd} trial ($S_{e_\theta} = 0.5$, $S_{\Delta e_\theta} = 2$ and $S_{u_\theta} = 10$)

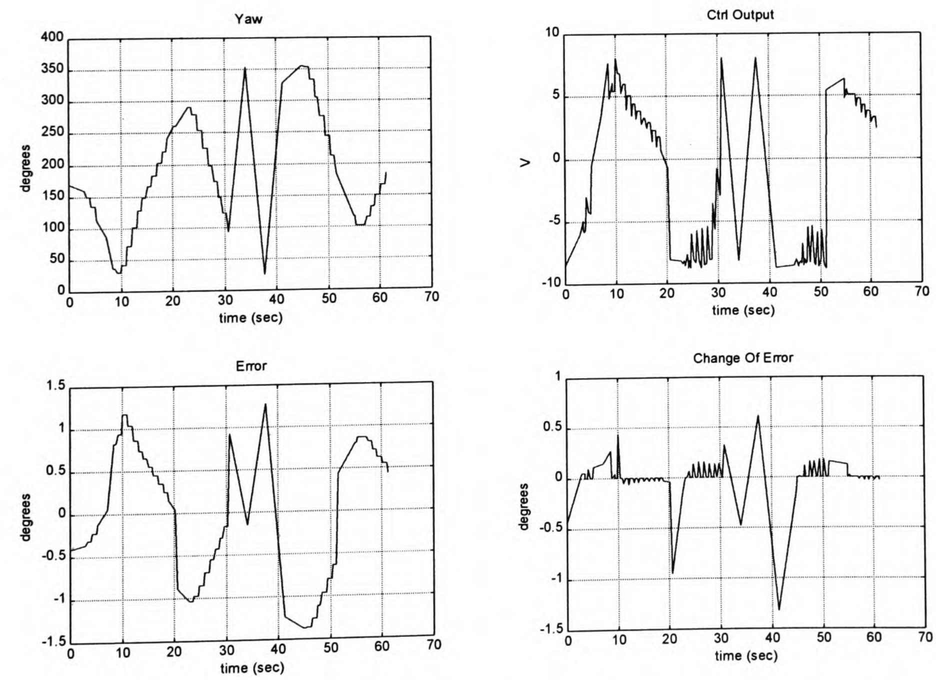


Figure 6.11 Yaw experimental results for the 4^{th} trial ($S_{e_\theta} = 1$, $S_{\Delta e_\theta} = 0.5$ and $S_{u_\theta} = 7$)

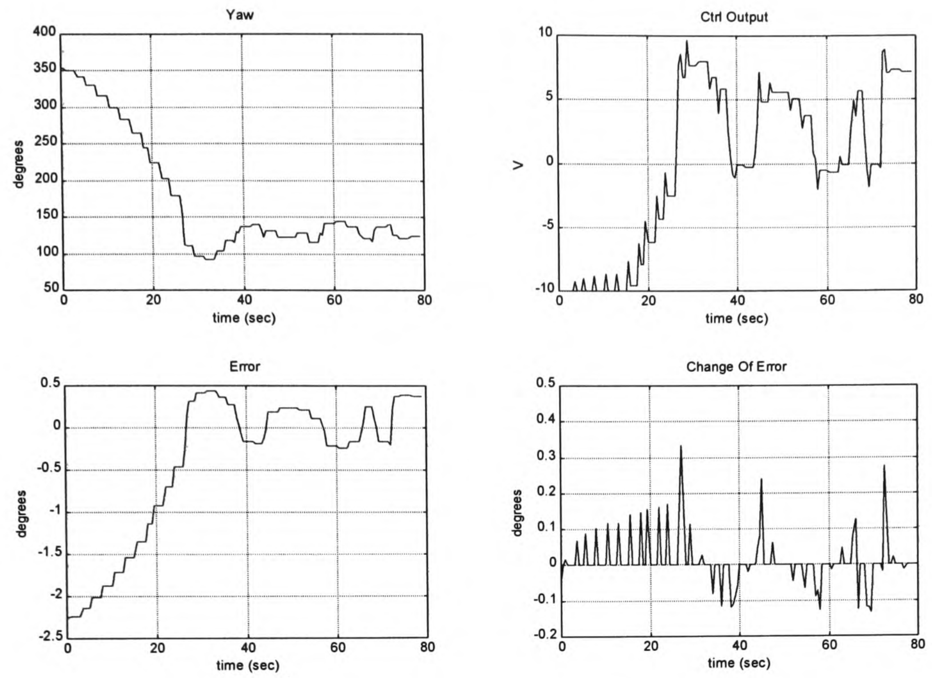


Figure 6.12 Yaw experimental results for the 5th trial ($S_{e_\theta}=1$, $S_{\Delta e_\theta}=1$ *and* $S_{u_\theta}=10$)

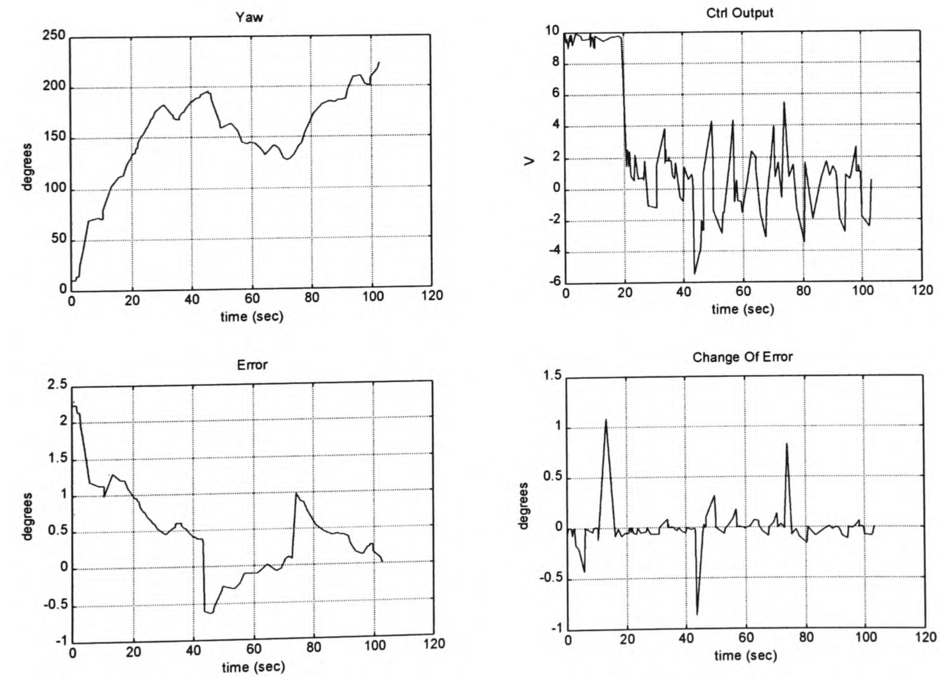


Figure 6.13 Yaw experimental results for the 6th trial ($S_{e_\theta}=1$, $S_{\Delta e_\theta}=2$ *and* $S_{u_\theta}=3$)

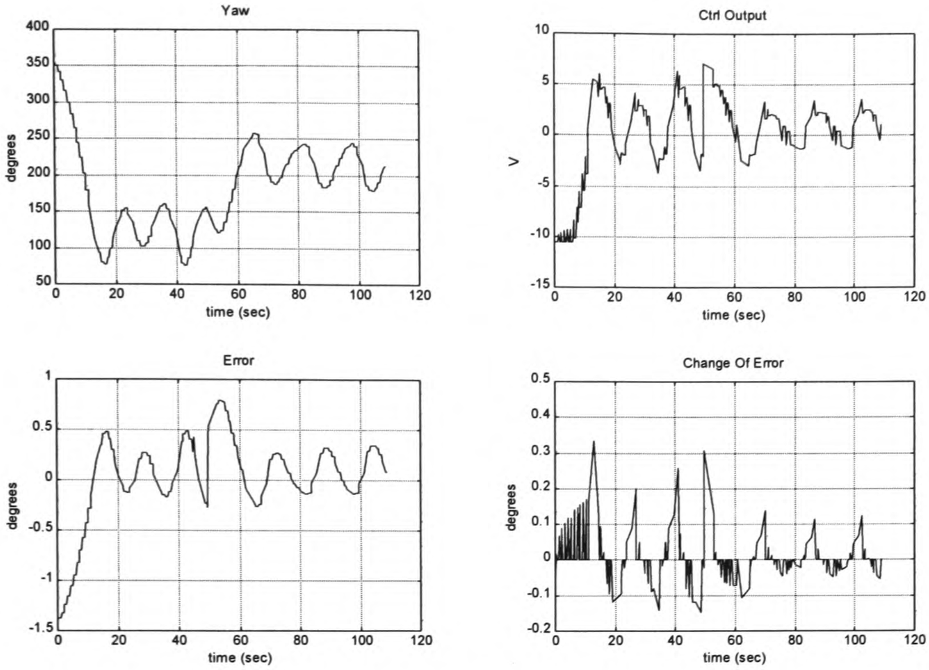


Figure 6.14 Yaw experimental results for the 7^{th} trial ($S_{e_\theta} = 0.75$, $S_{\Delta e_\theta} = 0.5$ and $S_{u_\theta} = 10$)

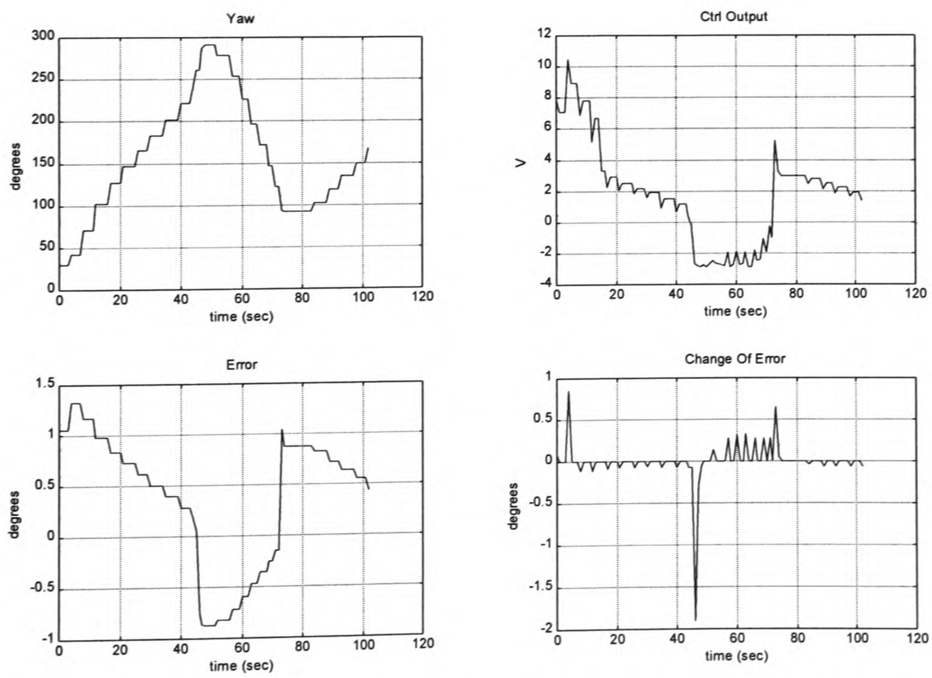


Figure 6.15 Yaw experimental results for the 8^{th} trial ($S_{e_\theta} = 0.75$, $S_{\Delta e_\theta} = 1$ and $S_{u_\theta} = 3$)

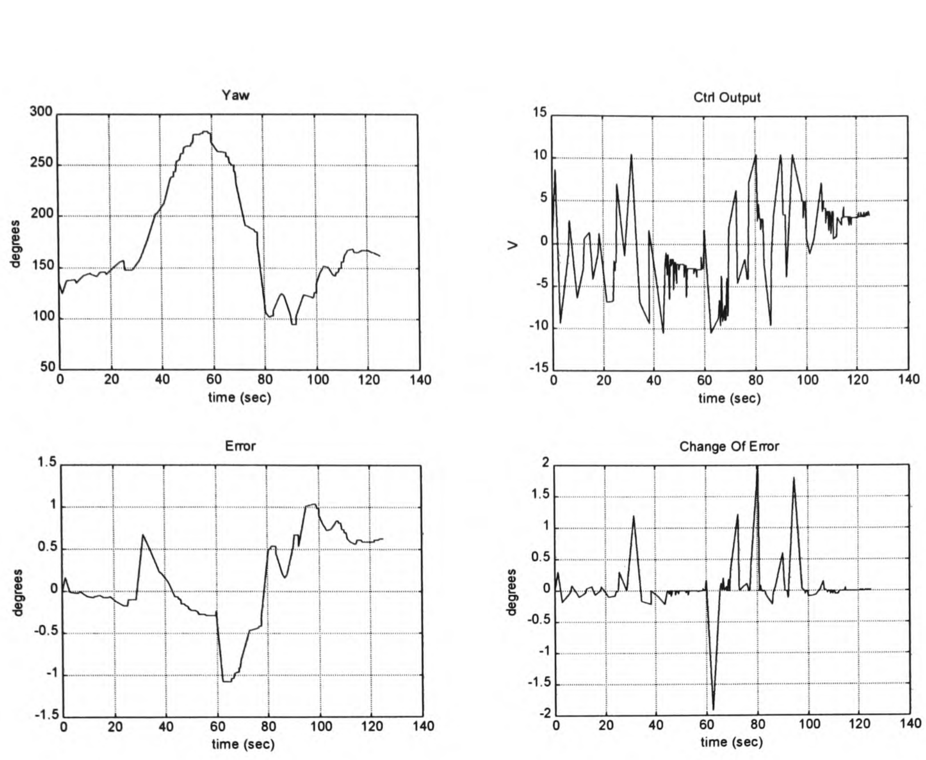


Figure 6.16 Yaw experimental results for the 9th trial ($S_{e_\theta} = 0.75$, $S_{\Delta e_\theta} = 2$ and $S_{u_\theta} = 7$)

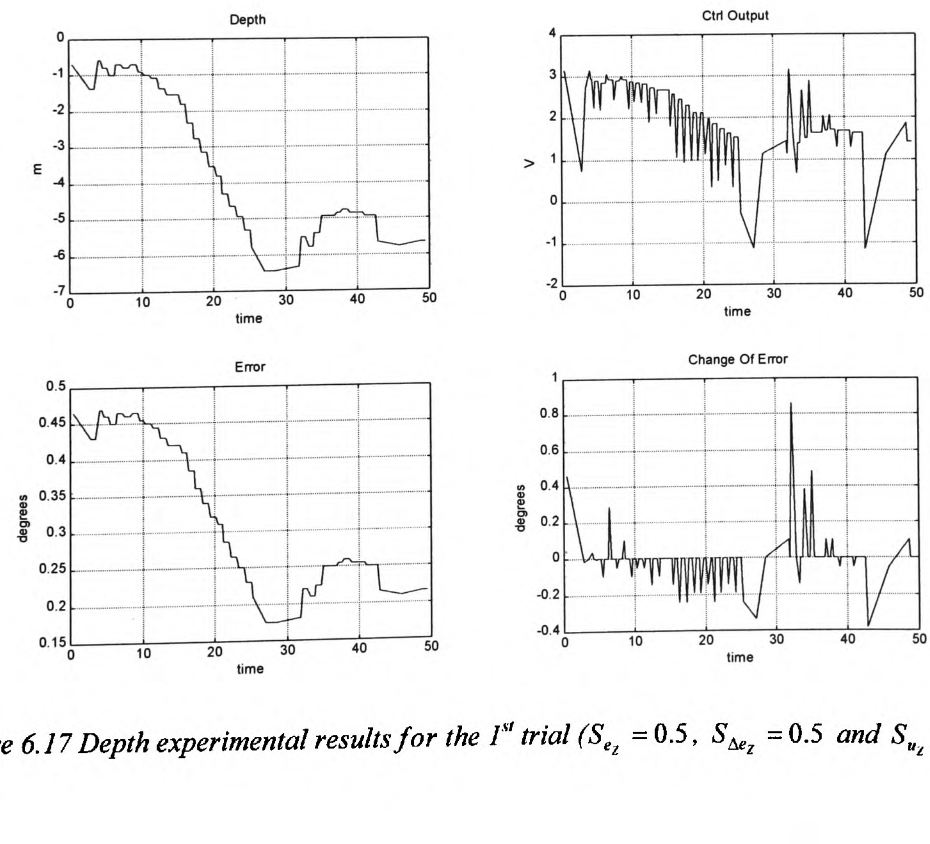


Figure 6.17 Depth experimental results for the 1st trial ($S_{e_z} = 0.5$, $S_{\Delta e_z} = 0.5$ and $S_{u_z} = 3$)

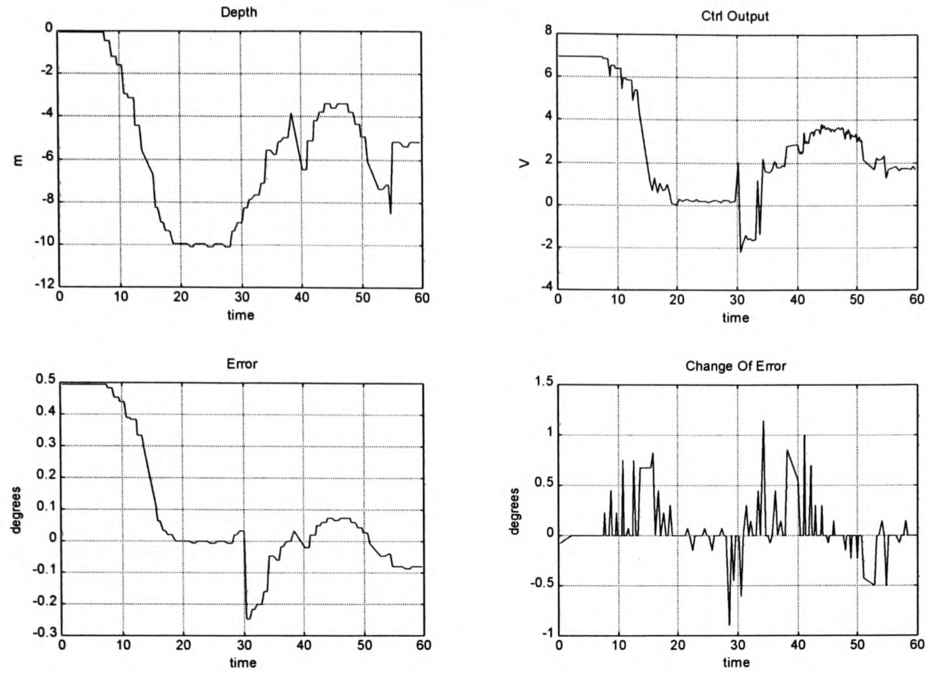


Figure 6.18 Depth experimental results for the 2^{nd} trial ($S_{e_z} = 0.5$, $S_{\Delta e_z} = 1$ and $S_{u_z} = 7$)

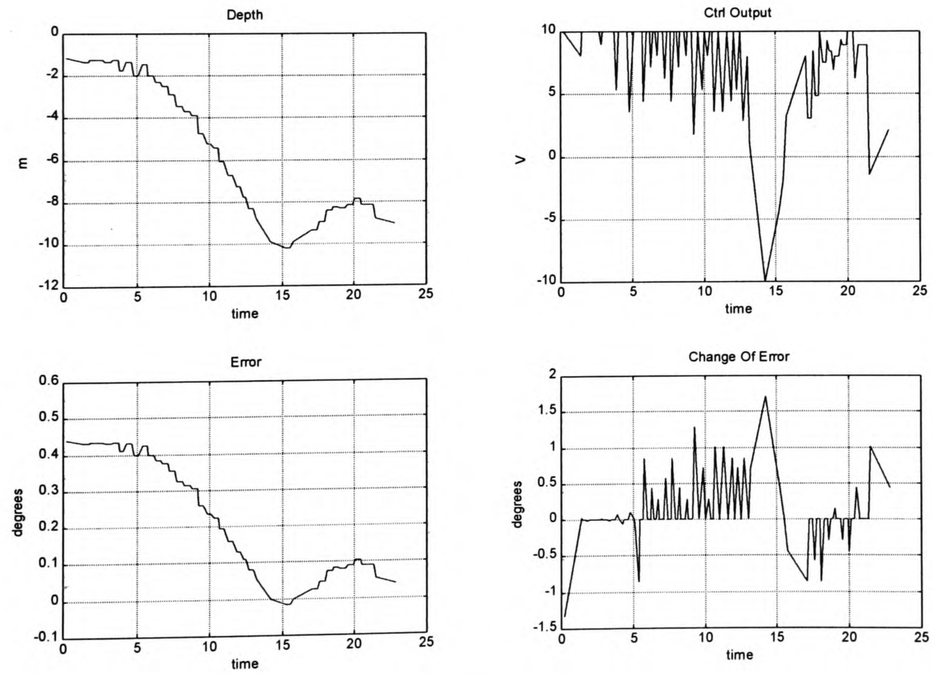


Figure 6.19 Depth experimental results for the 3^{rd} trial ($S_{e_z} = 0.5$, $S_{\Delta e_z} = 2$ and $S_{u_z} = 10$)

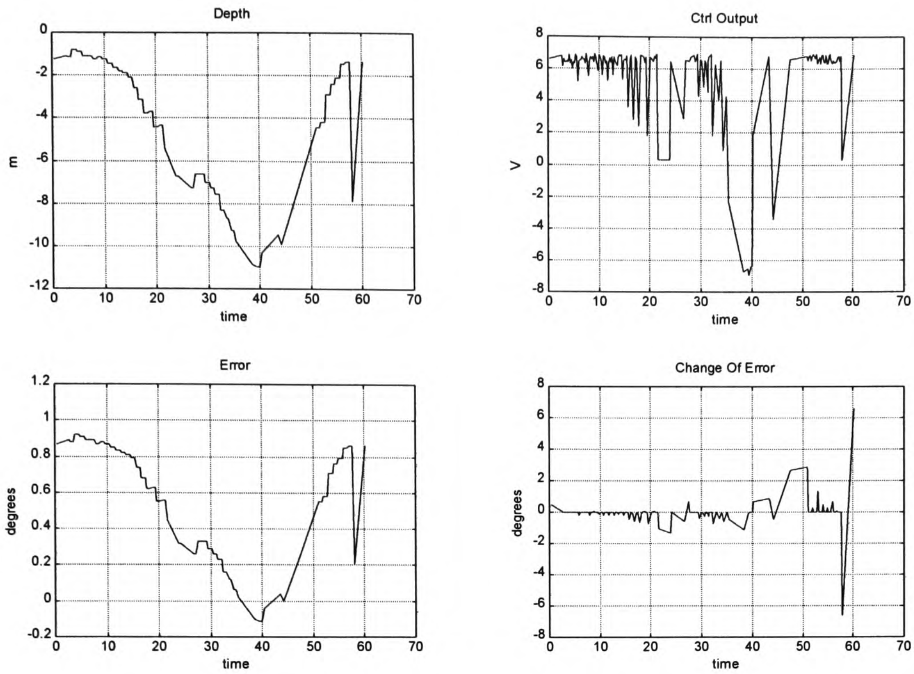


Figure 6.20 Depth experimental results for the 4th trial ($S_{e_z} = 1$, $S_{\Delta e_z} = 0.5$ and $S_{u_z} = 7$)

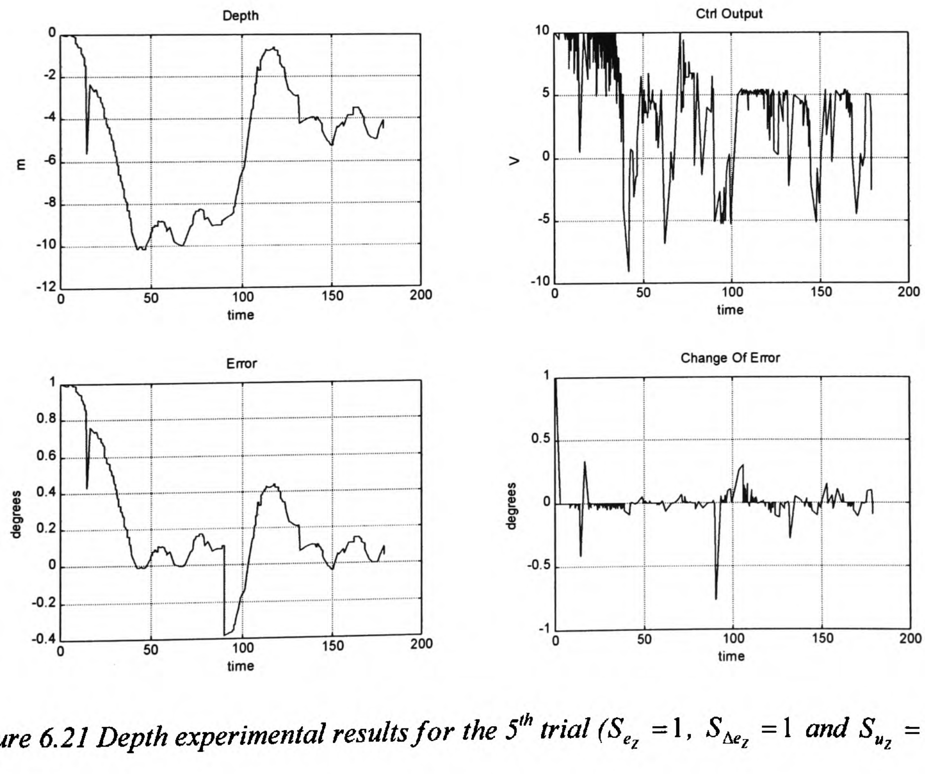


Figure 6.21 Depth experimental results for the 5th trial ($S_{e_z} = 1$, $S_{\Delta e_z} = 1$ and $S_{u_z} = 10$)

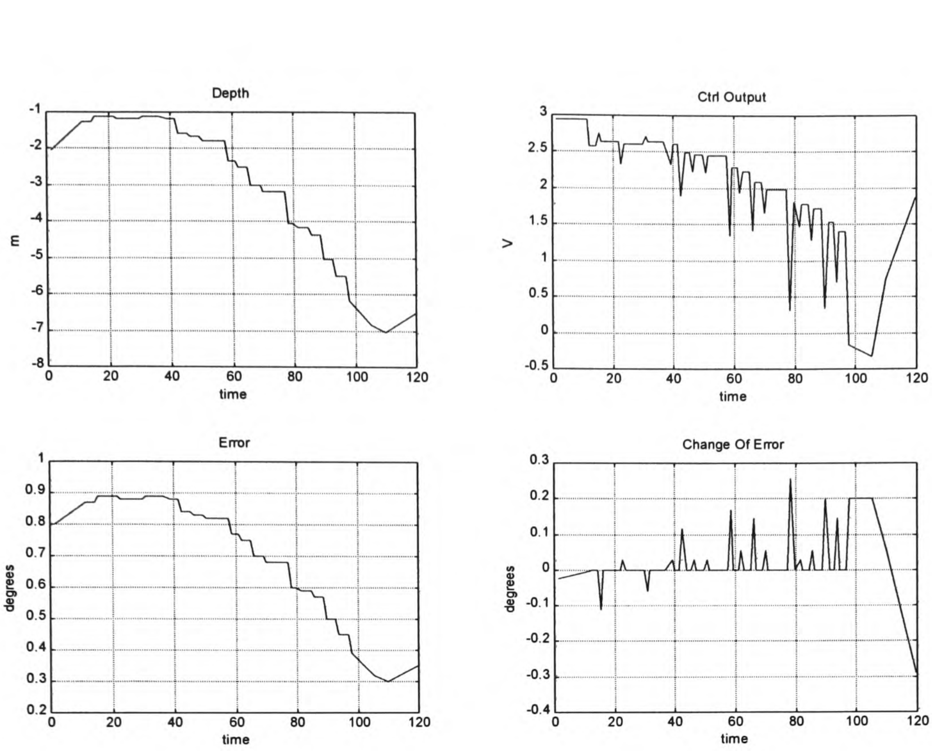


Figure 6.22 Depth experimental results for the 6th trial ($S_{e_z}=1$, $S_{\Delta e_z}=2$ and $S_{u_z}=3$)

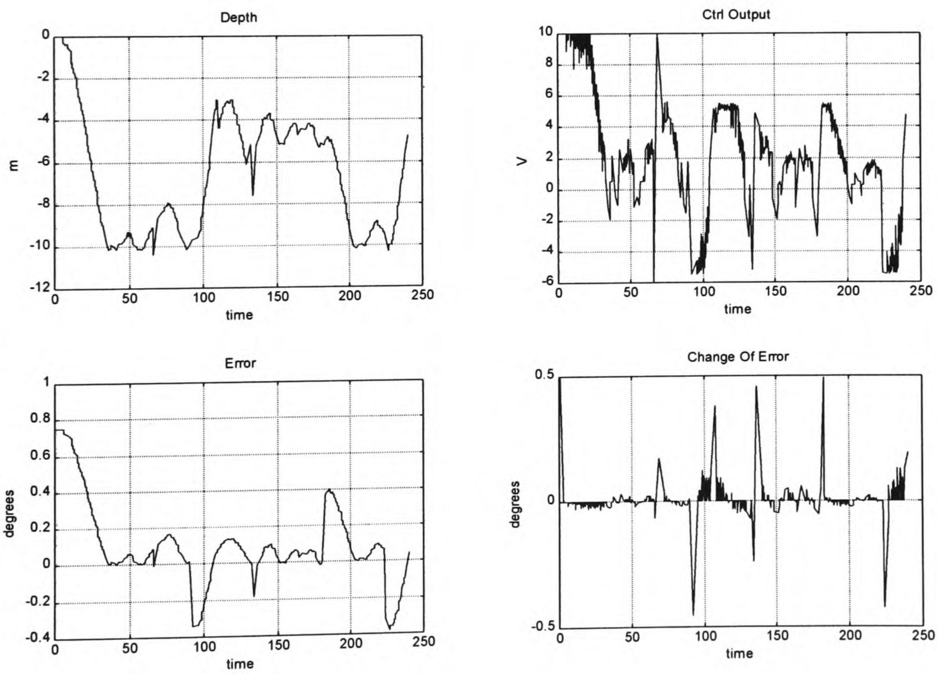


Figure 6.23 Depth experimental results for the 7th trial ($S_{e_z}=0.7$, $S_{\Delta e_z}=0.5$ and $S_{u_z}=10$)

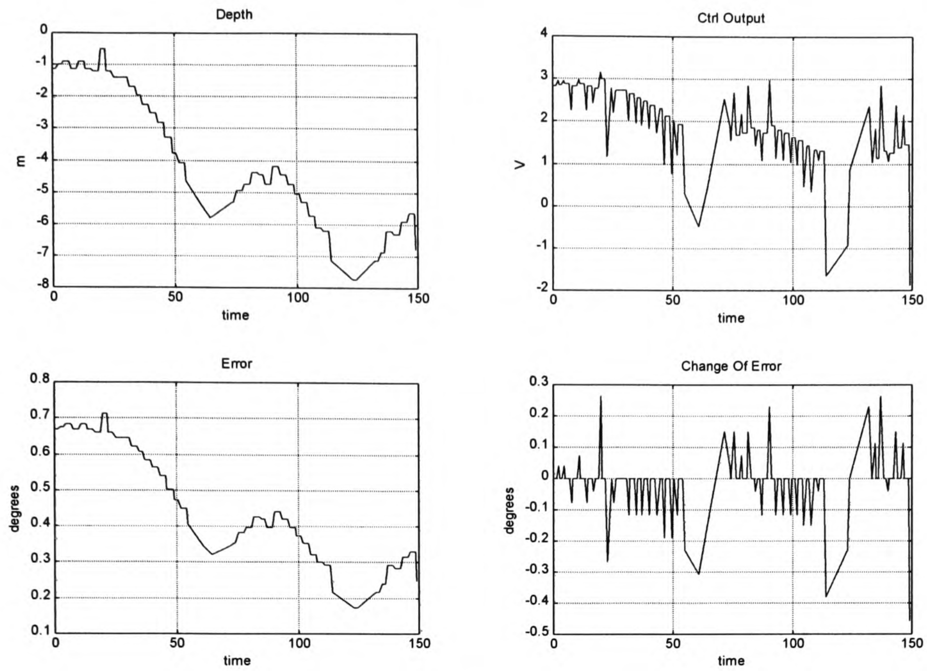


Figure 6.24 Depth experimental results for the 8th trial ($S_{e_Z} = 0.75$, $S_{\Delta e_Z} = 1$ and $S_{u_Z} = 3$)

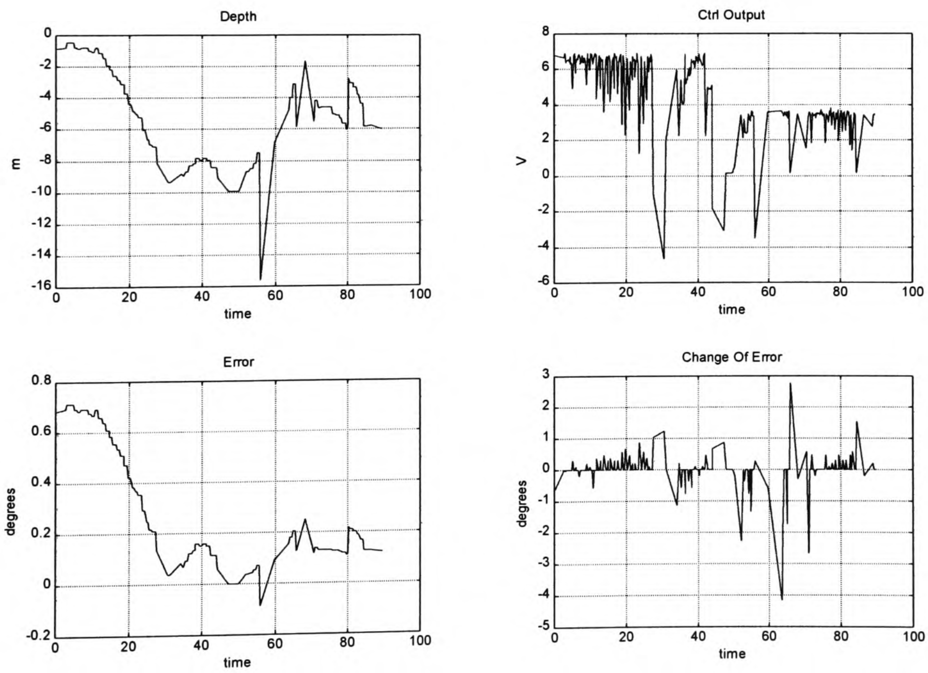


Figure 6.25 Depth experimental results for the 9th trial ($S_{e_Z} = 0.75$, $S_{\Delta e_Z} = 2$ and $S_{u_Z} = 7$)

6.5.1 Experimental results in a tabular form

In this section the results of yaw and depth trials are presented in tabular form. Table 6.4 depicts the results of yaw and Table 6.5 depicts the results in terms of both IAE and ITAE considering the performances vector p_O in Equation 6.21. However from these tables it is very difficult to extract enough information, to identify the optimal parameters of the control system, due to the limited number of experiments. Moreover, the complexity that appears in studying more than one parameter at the same time is another drawback in this type of analysis. Antipodal, a systematic way to investigate and tune these optimal parameters of GARBI's Fuzzy-like PD controller using only the small number of experiments is applied as discussed in Chapter 5 (Section § 5.2.8) and presented in the next Section.

	SCALING FACTORS			YAW PERFORMANCE	
No Experi-ment	S_{e_θ}	$S_{\Delta e_\theta}$	S_{u_θ}	$p_{IAE_{yaw}}$	$p_{ITAE_{yaw}}$
1	0.5	0.5	3	11.38	547.33
2	0.5	1	7	9.06	556.41
3	0.5	2	10	8.914	290.28
4	1	0.5	7	7.9	141.2
5	1	1	10	9.87	277.8
6	1	2	3	14.32	645.37
7	0.75	0.5	10	14.4	670.55
8	0.75	1	3	16.95	729
9	0.75	2	7	10.92	464.5

Table 6.4 Yaw performance in terms of IAE and ITAE from the real experiments. Graphically illustrated in Figure 6.8 to Figure 6.16

No Experi-ment	SCALING FACTORS S_{e_z}	$S_{\Delta e_z}$	S_{u_z}	DEPTH PERFORMANCE $p_{IAE_{depth}}$	$p_{ITAE_{depth}}$
1	0.5	0.5	3	29.92	633.25
2	0.5	1	7	8.515	137.58
3	0.5	2	10	9.36	64.64
4	1	0.5	7	20.86	294.22
5	1	1	10	22.32	1265.25
6	1	2	3	80.62	4190.8
7	0.75	0.5	10	14.89	1118.72
8	0.75	1	3	84.17	5111.58
9	0.75	2	7	21.68	314.2

Table 6.5 Depth performance in terms of IAE and ITAE out from the real experiments. Graphically illustrated in Figure 6.17 to Figure 6.25

6.6 Analysis of the results to define the optimal parameters of the Fuzzy-like PD controller

Analysing the data is a systematic straightforward procedure as introduced in Chapter 5 and is applied in this chapter for the real experimental results as follows:

6.6.1 Analysis of means

The analysis of means for each SF level is applied for the investigation of the OTFLs using the results presented in the previous Section in Table 6.4 and Table 6.5. Thus, following the steps presented in Section § 5.2.8 the OTFLs are defined as follows:

- for each factor level, the average responses are obtained using Equation 5.3
- these responses together with the levels are the co-ordinates that construct the plots used in the analysis of the graphical representation of factor levels.

Thus Figure 6.26, Figure 6.27, Figure 6.28 and Figure 6.29 shows both yaw and depth responses for the plot of IAE and ITAE means respectively. As the aim of the controller is to minimise both IAE and ITAE, the objective characteristic of these target values is *"smaller-the-better"* as discussed in Section § 5.2.8.1. Therefore, for the *yaw* response:

Figure 6.26 shows the "IAE mean response plot" where the smallest *yaw mean (average) responses* are the 9.79, 11.25 and 9.31 (circled values) for S_{e_θ}, $S_{\Delta e_\theta}$ and S_{u_θ} respectively and therefore, the OTFLs are:

$$l_{f_{t1}} = \{S_{e_{\theta 1}}, S_{\Delta e_{\theta 1}}, S_{u_{\theta 2}}\} = \{0.5, 0.5, 7\} \tag{6.28}$$

Figure 6.27 shows the "ITAE mean response plot" where the smallest *yaw mean responses* for S_{e_θ}, $S_{\Delta e_\theta}$ and S_{u_θ} are 354.79, 453.03 and 387.37 (circled values) respectively and therefore the OTFLs are:

$$l_{f_{t2}} = \{S_{e_{\theta 3}}, S_{\Delta e_{\theta 1}}, S_{u_{\theta 2}}\} = \{1, 0.5, 7\} \tag{6.29}$$

For the *depth* response the corresponding IAE means plot is shown in Figure 6.28 where the smallest *"depth average responses"* for S_{e_z}, $S_{\Delta e_z}$ and S_{u_z} are 15.93, 21.89 and 15.52 (circled values) respectively. Thus the OTFLs are:

$$l_{f_{t3}} = \{S_{e_{z1}}, S_{\Delta e_{z1}}, S_{u_{z3}}\} = \{0.5, 0.5, 10\} \tag{6.30}$$

Note here that these OTFLs are not defined as a combination in the experiment (Table 6.3), however, using the Taguchi method it is able to be identified. This is one of the most important features on the above approach as discussed in Chapter 5. That is, it is able to identify experimental combinations that were not originally specified in the orthogonal array.

For the ITAE means plot shown in Figure 6.29 the smallest *"depth average responses"* for S_e, $S_{\Delta e}$ and S_u are 278.49, 682.07 and 248.67 (circled values) respectively. Thus the OTFLs are:

$$l_{f_{t4}} = \{S_{e_{z1}}, S_{\Delta e_{z1}}, S_{u_{z2}}\} = \{0.5, 0.5, 7\} \tag{6.31}$$

Moreover, from the above analysis of figures it can also be observed that more combinations of the SF levels than those in Equation 6.28 to Equation 6.31 could also be used as optimal ones. That is because some of the levels have values that are close the optimal level's values. Table 6.6 and Table 6.7 illustrates the combinations of the SF levels that are Optimal, Close to the Optimal and Not Optimal observed from the IAE and ITAE average yaw and depth responses respectively.

Note that all optimal SFs for the change of error Δe input are 0.5. That means that the sampling time that is chosen for the experiments do not cause sensitivity problems discussed in Section § 6.3.3.

In the above study neither the S/N ratio nor the interaction analysis was under investigation as neither replication nor enough experiments are available respectively.

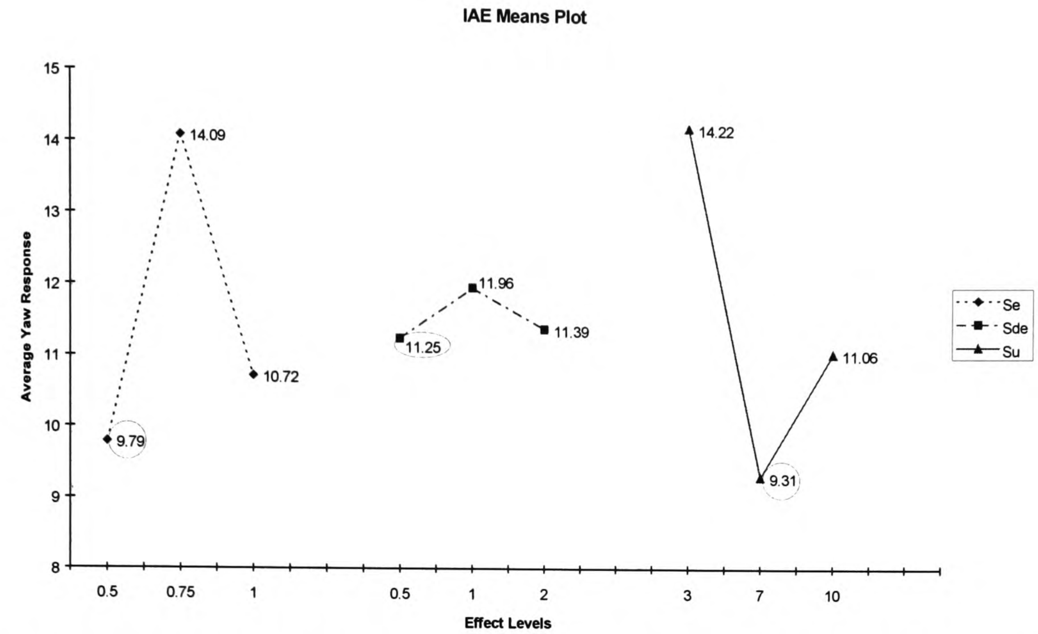


Figure 6.26 Plot used in analysis of means to investigate the optimal levels that minimise the IAE value for the Yaw Response. Investigating for the smallest average response for S_{e_θ}, $S_{\Delta e_\theta}$ and S_{u_θ} the OTFLs are $l_{f_{t1}} = \{0.5, 0.5, 7\}$

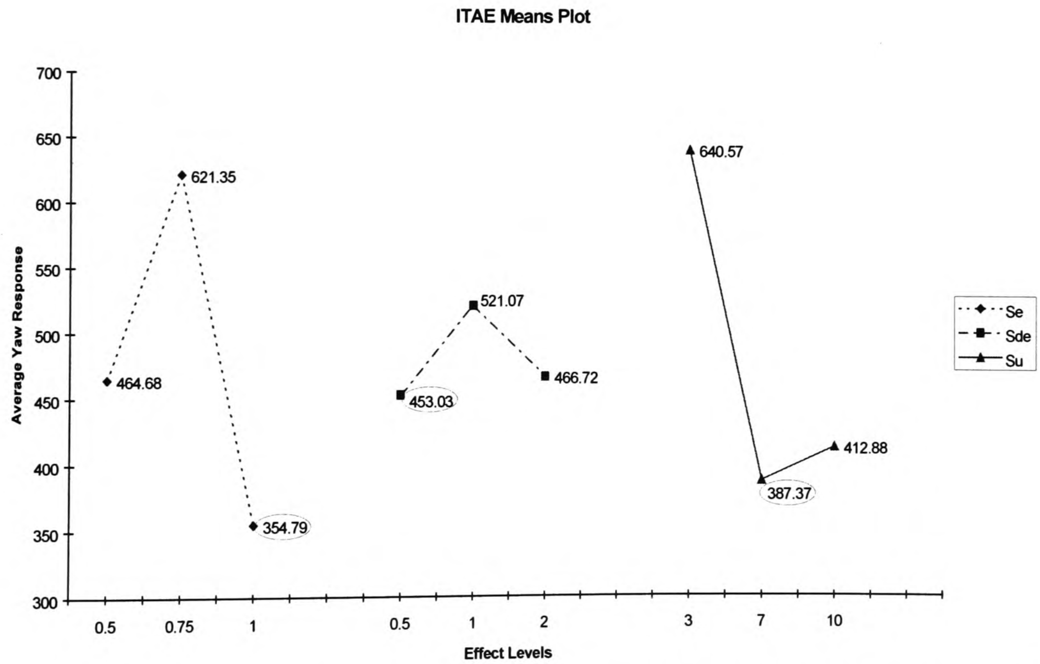


Figure 6.27 Plot used in analysis of means to investigate the optimal levels that minimise the ITAE value for the Yaw Response. Investigating for the smallest average response for S_{e_θ}, $S_{\Delta e_\theta}$ and S_{u_θ} the OTFLs are $l_{f_{t2}} = \{1, 0.5, 7\}$

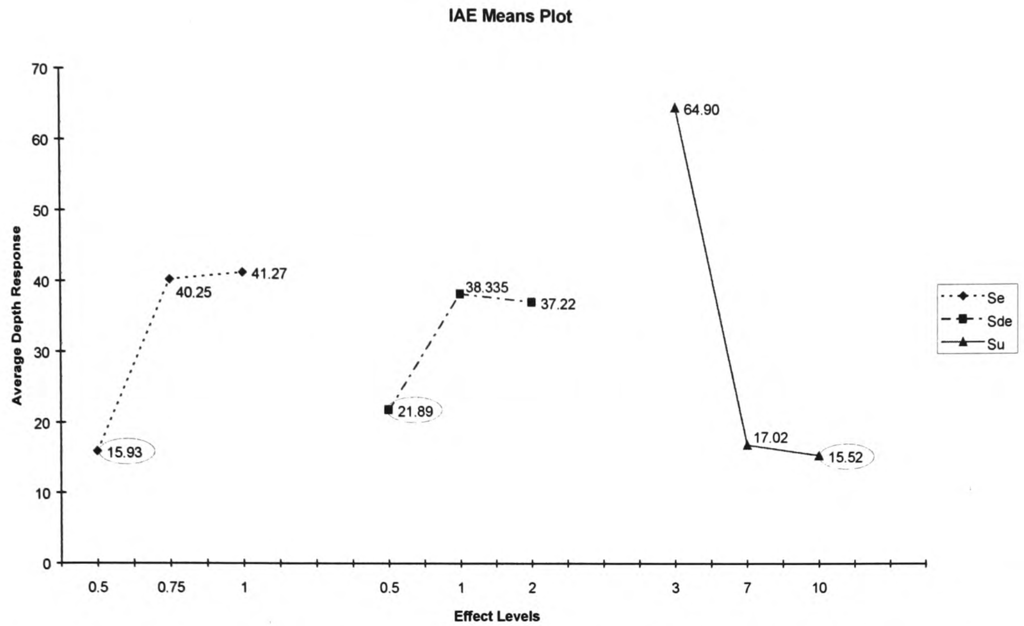


Figure 6.28 Plot used in analysis of means to investigate the optimal levels that minimise the IAE value for the Depth Response. Investigating for the smallest average response for S_{e_z}, $S_{\Delta e_z}$ and S_{u_z} the OTFLs are $l_{f_{l3}} = \{0.5, 0.5, 10\}$

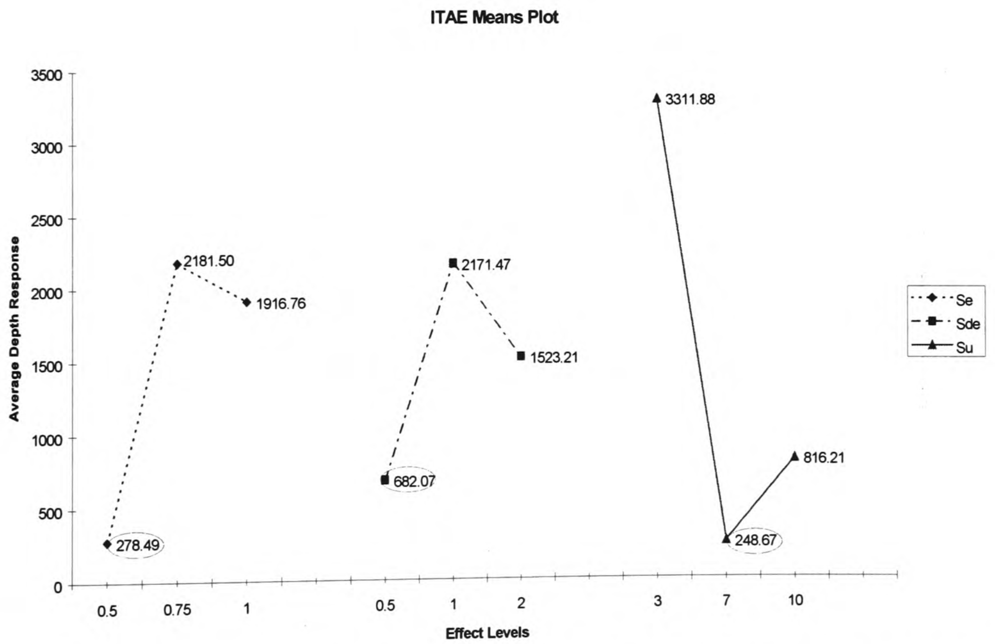


Figure 6.29 Plot used in analysis of means to investigate the optimal levels that minimise the ITAE value for the Depth Response. Investigating for the smallest average response for S_{e_z}, $S_{\Delta e_z}$ and S_{u_z} the OTFLs are $l_{f_{l4}} = \{0.5, 0.5, 7\}$

COMBINATIONS OF THE SCALING FACTOR LEVELS	IAE AVERAGE YAW RESPONSE OF THE SCALING FACTOR LEVELS			ITAE AVERAGE YAW RESPONSE OF THE SCALING FACTOR LEVELS		
	S_{e_θ}	$S_{\Delta e_\theta}$	S_{u_θ}	S_{e_θ}	$S_{\Delta e_\theta}$	S_{u_θ}
Optimal	9.79	11.25	9.31	354.79	453.03	387.37
Close to the Optimal	10.72	11.39 & 11.96	—	—	466.72	412.88
Not Optimal	14.09	—	14.22 & 11.06	464.68 & 621.35	521.07	640.57

Table 6.6 Combinations of the SF levels that are Optimal, Close to the Optimal and Not Optimal, observed from the IAE and ITAE average yaw responses

COMBINATIONS OF THE SCALING FACTOR LEVELS	IAE AVERAGE DEPTH RESPONSE OF THE SCALING FACTOR LEVELS			ITAE AVERAGE DEPTH RESPONSE OF THE SCALING FACTOR LEVELS		
	S_{e_z}	$S_{\Delta e_z}$	S_{u_z}	S_{e_z}	$S_{\Delta e_z}$	S_{u_z}
Optimal	15.93	21.89	15.52	278.49	682.07	248.67
Close to the Optimal	—	—	17.02	—	—	—
Not Optimal	40.25 & 41.27	38.33 & 37.22	64.90	2181.5 & 1916.76	2171.4 & 1523.2	3311.8 & 816.2

Table 6.7 Combinations of the SF levels that are Optimal, Close to the Optimal and Not Optimal, observed from the IAE and ITAE average depth responses

6.6.2 Analysis of Variance

Analysis of Variance is a method, which investigates the significance of each SF as described in Section § 5.2.8.2. From Table 6.8 and Table 6.9 it can be observed that both SFs S_{u_θ} and S_{e_θ} have high significance, with 49.96% and 41.39% contribution corresponding to the IAE yaw response and 35.56% and 32.83% contribution corresponding to the ITAE yaw response respectively. However, SF $S_{\Delta e_\theta}$ has low significance with only 1.16% and 2.37% contribution. Moreover, from Table 6.10 and Table 6.11 the SF S_{u_z} has the highest significance with 69.93% and 56.91% contribution corresponding to the IAE and ITAE depth response respectively whereas the SFs S_{e_z}, $S_{\Delta e_z}$ have lower significance.

From the above observations it can be seen that only some of the SFs for both the yaw and depth controllers have very high significance in the system responses. Nevertheless it was decided that the SF that are not significant should also be included in the structure of yaw and depth controllers. That is because the number of SFs is very small and therefore all of them can contribute to the tuning rules without causing complication in terms of constructing them and computation time.

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	% contribution	Rank
Se	30.74	2	15.37	5.52	41.39	2
Sde	0.86	2	0.43	0.16	1.16	4
Su	37.11	2	18.55	6.67	49.96	1
Error	5.56	2.00	2.78	1.00	7.49	3
ST	74.27	8.00				

Table 6.8 ANOVA table for factors S_{e_θ}, $S_{\Delta e_\theta}$ and S_{u_θ} corresponding to IAE Yaw Response

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	% contribution	Rank
Se	107674.87	2	53837.43	1.12	32.83	2
Sde	7771.79	2	3885.90	0.08	2.37	4
Su	116602.89	2	58301.44	1.22	35.56	1
Error	95885.51	2.00	47942.75	1.00	29.24	3
ST	327935.06	8.00				

Table 6.9 ANOVA table for factors S_{e_θ}, $S_{\Delta e_\theta}$ and S_{u_θ} corresponding to ITAE Yaw Response

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	% contribution	Rank
Se	1233.96	2	616.98	4.19	18.23	2
Sde	506.75	2	253.37	1.72	7.49	3
Su	4733.44	2	2366.72	16.07	69.93	1
Error	294.64	2.00	147.32	1.00	4.35	4
ST	6768.78	8.00				

Table 6.10 ANOVA table for factors S_{e_z}, $S_{\Delta e_z}$ and S_{u_z} corresponding to IAE Depth Response

Factor	Sum Square (SS)	dof	mean sq (MSS)	F-Ratio	% contribution	Rank
Se	6375468.22	2	3187734.11	2.72	22.77	2
Sde	3346086.71	2	1673043.36	1.43	11.95	3
Su	15933725.90	2	7966862.95	6.80	56.91	1
Error	2341646.92	2.00	1170823.46	1.00	8.36	4
ST	27996927.75	8.00				

Table 6.11 ANOVA table for factors S_{e_z}, $S_{\Delta e_z}$ and S_{u_z} corresponding to ITAE Depth Response

6.7 Constructing GARBI's Fuzzy-like PD controller's tuning fuzzy rules

The construction of the fuzzy rules used to tune the factorial levels (when the system may change) considering the performance states and the factor level's characteristics has been explained in Section § 5.2.10. Therefore the resulting tuning rules used to tune both yaw and depth of GARBI's Fuzzy-like PD controllers' parameters are:

for the yaw controller

R₁: **IF** $p_{IAE_{yaw}} > 0$ **THEN** *tune factor* S_{e_θ} **AND** *factor* $S_{\Delta e_\theta}$ **AND** *factor* S_{u_θ}

R₂: **IF** $p_{ITAE_{yaw}} > 0$ **THEN** *tune factor* S_{e_θ} **AND** *factor* $S_{\Delta e_\theta}$ **AND** *factor* S_{u_θ}

for the depth controller

R₃: **IF** $p_{IAE_{depth}} > 0$ **THEN** *tune factor* S_{e_z} **AND** *factor* $S_{\Delta e_z}$ **AND** *factor* S_{u_z}

R₄: **IF** $p_{ITAE_{depth}} > 0$ **THEN** *tune factor* S_{e_z} **AND** *factor* $S_{\Delta e_z}$ **AND** *factor* S_{u_z}

These four rules are actually the ones that are responsible for tuning the SFs resulting in minimising the IAE and ITAE performance characteristics.

6.8 Applying the Fuzzy Combined Scheduling System (FCSS) approach

As explained in Section § 5.2.11, the FCSS approach is applied to optimise the overall tuning factor levels for both yaw and depth Fuzzy-like PD controllers. Therefore, using Equation 5.6 the yaw and depth vectors $L_{S_{yaw}} = \{L_{S_{e\theta}}, L_{S_{\Delta e\theta}}, L_{S_{u\theta}}\}$ and $L_{S_{depth}} = \{L_{S_{e_z}}, L_{S_{\Delta e_z}}, L_{S_{u_z}}\}$ of the updated factorial levels are defined as follows:

for the yaw controller

$$L_{S_{e_\psi}} = \frac{(w_{ts_e 1} \cdot S_{e_\theta 1} + w_{cs_e 1} \cdot S_{e_\theta current}) + (w_{ts_e 2} \cdot S_{e_\theta 3} + w_{cs_e 2} \cdot S_{e_\theta current})}{(w_{ts_e 1} + w_{cs_e 1}) + (w_{ts_e 2} + w_{cs_e 2})} \tag{6.32}$$

$$L_{S_{\Delta e_\psi}} = \frac{(w_{ts_{\Delta e} 1} \cdot S_{\Delta e_\theta 1} + w_{cs_{\Delta e} 1} \cdot S_{\Delta e_\theta current}) + (w_{ts_{\Delta e} 2} \cdot S_{\Delta e_\theta 1} + w_{cs_{\Delta e} 2} \cdot S_{\Delta e_\theta current})}{(w_{ts_{\Delta e} 1} + w_{cs_{\Delta e} 1}) + (w_{ts_{\Delta e} 2} + w_{cs_{\Delta e} 2})} \tag{6.33}$$

$$L_{S_{u_\psi}} = \frac{(w_{ts_u 1} \cdot S_{u_\theta 2} + w_{cs_u 1} \cdot S_{u_\theta current}) + (w_{ts_u 2} \cdot S_{u_\theta 2} + w_{cs_u 2} \cdot S_{u_\theta current})}{(w_{ts_u 1} + w_{cs_u 1}) + (w_{ts_u 2} + w_{cs_u 2})} \tag{6.34}$$

for the depth controller

$$L_{S_{e_z}} = \frac{(w_{ts_e 1} \cdot S_{e_z 1} + w_{cs_e 1} \cdot S_{e_z current}) + (w_{ts_e 2} \cdot S_{e_z 1} + w_{cs_e 2} \cdot S_{e_z current})}{(w_{ts_e 1} + w_{cs_e 1}) + (w_{ts_e 2} + w_{cs_e 2})} \tag{6.35}$$

$$L_{S_{\Delta e_z}} = \frac{(w_{ts_{\Delta e} 1} \cdot S_{\Delta e_z 1} + w_{cs_{\Delta e} 1} \cdot S_{\Delta e_z current}) + (w_{ts_{\Delta e} 2} \cdot S_{\Delta e_z 1} + w_{cs_{\Delta e} 2} \cdot S_{\Delta e_z current})}{(w_{ts_{\Delta e} 1} + w_{cs_{\Delta e} 1}) + (w_{ts_{\Delta e} 2} + w_{cs_{\Delta e} 2})} \tag{6.36}$$

$$L_{S_{u_z}} = \frac{(w_{ts_u 1} \cdot S_{u_z 3} + w_{cs_u 1} \cdot S_{u_z current}) + (w_{ts_u 2} \cdot S_{u_z 2} + w_{cs_u 2} \cdot S_{u_z current})}{(w_{ts_u 1} + w_{cs_u 1}) + (w_{ts_u 2} + w_{cs_u 2})} \tag{6.37}$$

where the weights $w_{ts_e 1}$, $w_{ts_e 2}$, $w_{ts_{\Delta e} 1}$, $w_{ts_{\Delta e} 2}$, $w_{ts_u 1}$, $w_{ts_u 2}$ are defined for the degree of freedom of the "Z" (low) MF and $w_{cs_e 1}$, $w_{cs_e 2}$, $w_{cs_{\Delta e} 1}$, $w_{cs_{\Delta e} 2}$, $w_{cs_u 1}$, $w_{cs_u 2}$ are defined for the degree of freedom of the "S" (high) MF as shown in Figure 6.30. These shapes for the MFs are used, because small errors should not affect the tuning parameters of the FLCs as they are inherently robust to them (Pok and Xu, 1994). Moreover, when the IAE and/or ITAE performance characteristics are getting bigger the parameters should change rapidly close to the OTFLs.

The universe of discourse has been normalised between [0,1] according to

$$\bar{p} = \frac{p - p_{\min}}{p_{\max} - p_{\min}} \tag{6.38}$$

were p is the current value, $p_{\min}$ is the smallest value that the error can have (in this case zero) and $p_{\max}$ the largest value in the series of measurements.

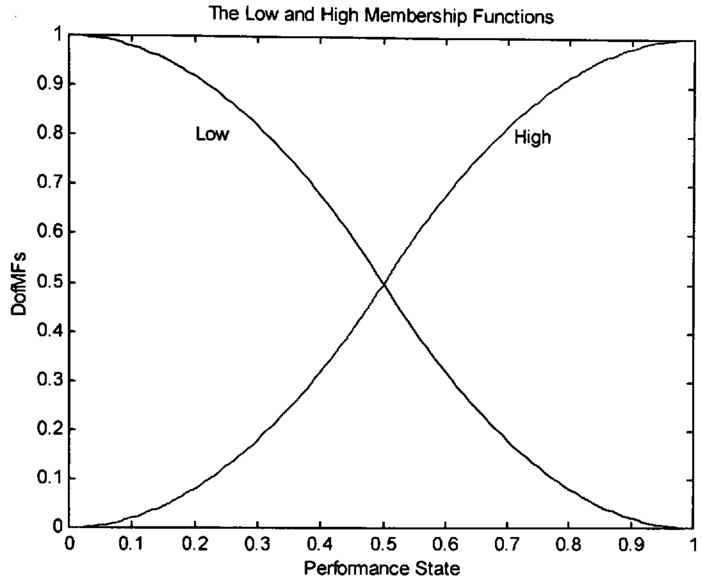


Figure 6.30 The Low and High Performance State Membership Functions

6.9 Verification and tuning of both yaw and depth Fuzzy-like PD controllers' parameters based on new experimental trials

New trials in the lake were undertaken to verify and/or retune the parameters for GARBI's Fuzzy-like PD controller in Oct 1999. Both yaw and depth controllers were working in parallel during these experiments. It was decided to use slightly different values for the optimal SFs to identify firstly, their robustness performance and secondly the FCSS ability to tune themselves according to the system's performance. Thus, the SFs for the *yaw* controller were set at $S_{e_\theta} = 0.5$, $S_{\Delta e_\theta} = 0.6$ and $S_{u_\theta} = 7.5$ whereas for the *depth* controller they were set at $S_{e_z} = 0.65$, $S_{\Delta e_z} = 0.45$ and $S_{u_z} = 6.3$. The mission plan for this experiment was for the heading to change the course from 135° to 0° and keep this direction and for the depth task it was to reach the level of 10 meters.

Figure 6.31 and Figure 6.32 shows the results for the above trial. From Figure 6.31 it can be seen that the yaw controller is responding fast however its performance for the course keeping is rather unsatisfactory as it saturates between 10 to 15 degrees magnitude. Additionally, the calculated performance in terms of IAE and ITAE are $p_{IAE_{yaw}} = 5.79$ and $p_{ITAE_{yaw}} = 155.67$ respectively. In Figure 6.32 the depth controller performs smoothly but not very fast. It took 47 seconds to reach the set point i.e. 10 meters (Note that in the time of 20 seconds a rapid bumping appears in depth. It is believed that this was caused by a sudden overpressure from the air pressure bottle). The calculated depth performance in this case are $p_{IAE_{depth}} = 34.35$ and $p_{ITAE_{depth}} = 698.66$ in terms of IAE and ITAE correspondingly.

Using Equation 6.32, Equation 6.33 and Equation 6.34 the yaw controller SFs are updated. Thus, the resulting SFs' gains are $S_{e_\theta} = 0.5$, $S_{\Delta e_\theta} = 0.5$ and $S_{u_\theta} = 7$. Moreover, from Equation 6.35, Equation 6.36 and Equation 6.37, the SFs for the depth controller are updated to $S_{e_z} = 0.51$, $S_{\Delta e_z} = 0.48$ and $S_{u_z} = 8.05$.

The above values for the SFs are actually the overall evaluation gains considering the new conditions that affect the performance of the system under study. In this particular case these were the hardware modifications and the environmental changes (note that during these experiments the weather conditions were windy and caused some small waves in the lake).

Setting the new SFs the final experiment was performed as follows:

- initial voltage of the horizontal propellers is 3V for a period of 60 seconds.

- change the heading course from 270° degrees to 135° and then to 225° and then keep it in this direction,
- and at the same time change the depth course from 0 to 10 meters and then to 5 meters and then keep it at this depth.

Figure 6.33 illustrates the Yaw performance together with the controller's output power of the horizontal propellers (T_1, T_2) (notice that as mentioned before they have opposite signs), the error e_θ and the change of error Δe_θ between the desired and actual yaw. Additionally, Figure 6.34 shows the depth performance as well as the controller's output power of the vertical propellers (T_3, T_4), the error e_z and the change of error Δe_z between the desired and the actual depth.

The performance of the yaw controller in changing the course (heading/depth) is satisfactory, as both overshoot and rise time are small. Due to buoyancy effects the depth control dynamics vary i.e. the vehicle rises faster than it descends. From Figure 6.34 it can be seen that the controller has accommodated this variation producing very acceptable rise times with very a small overshoot when ascending. In course-keeping control both controllers perform quite well (Figure 6.33 and Figure 6.34). Finally, Figure 6.35 and Figure 6.36 show the pitch and the roll of GARBI measured during the experiments. They mostly result from the heading/depth changes. However, they are very small, no more than 2.5° for the pitch and 5° for the roll, and therefore are considered as fractional and thus negligible and do not need to be considered as a controller variable.

Unfortunately for these experiments the time and therefore the number of trials were very limited due to weather conditions together with some problems that occurred with GARBI's air valve.

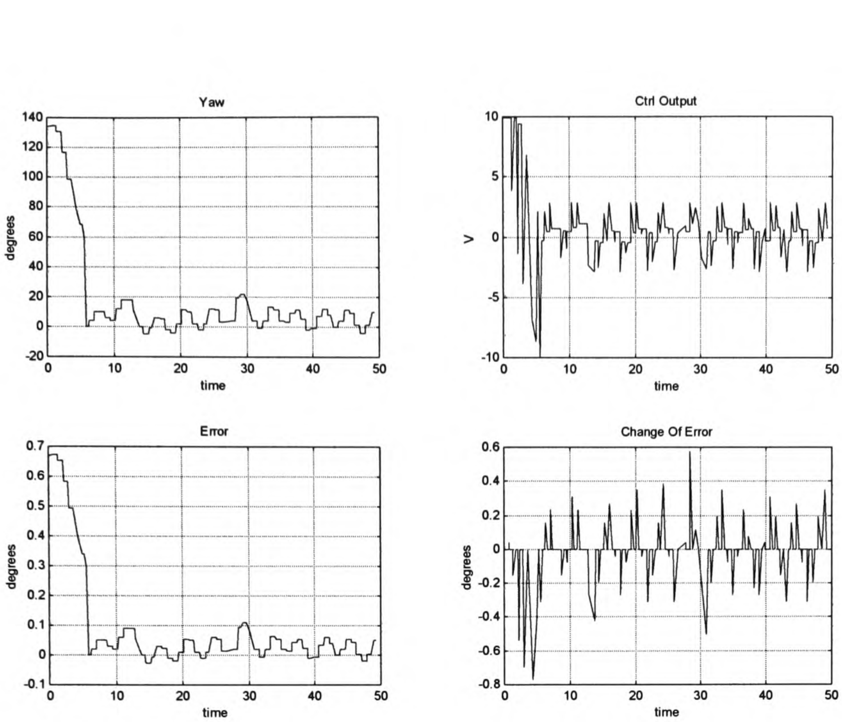


Figure 6.31 Yaw experimental results for trial after optimising the SFs. This figure shows the depth response together with the control output, the error and change of error. The SFs in this experiment are $S_{e_\theta} = 0.5$, $S_{\Delta e_\theta} = 0.6$ *and* $S_{u_\theta} = 7.5$

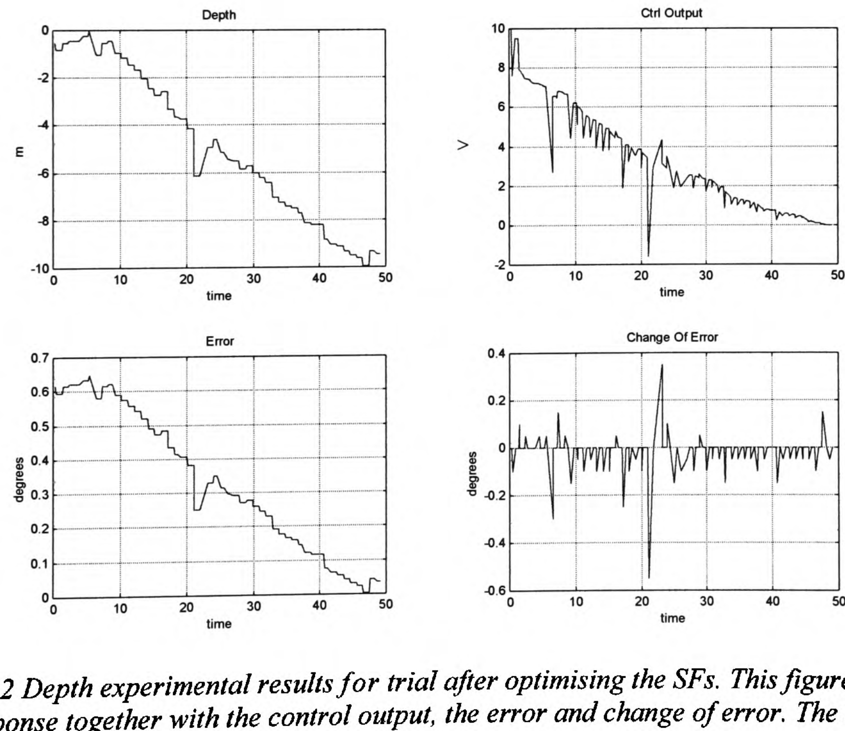


Figure 6.32 Depth experimental results for trial after optimising the SFs. This figure shows the depth response together with the control output, the error and change of error. The SFs in this experiment are $S_{e_z} = 0.65$, $S_{\Delta e_z} = 0.45$ *and* $S_{u_z} = 6.3$

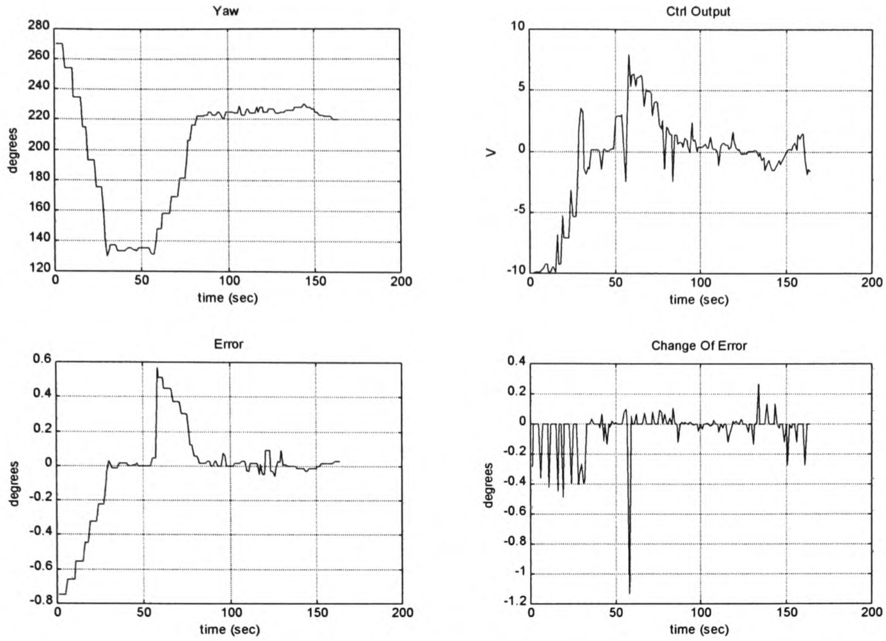


Figure 6.33 Yaw experimental results for trial after optimising the SFs. This figure shows the depth response together with the control output, the error and change of error. The SFs in this experiment are $S_{e_\theta} = 0.5$, $S_{\Delta e_\theta} = 0.5$ *and* $S_{u_\theta} = 7$

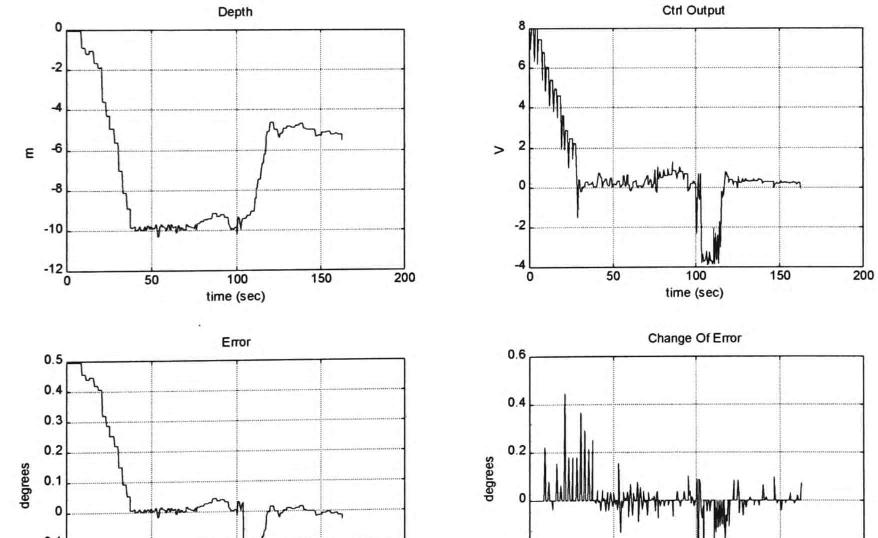


Figure 6.34 Depth experimental results for trial after optimising the SFs. This figure shows the depth response together with the control output, the error and change of error. The SFs in this experiment are $S_{e_z} = 0.51$, $S_{\Delta e_z} = 0.48$ *and* $S_{u_z} = 8.05$

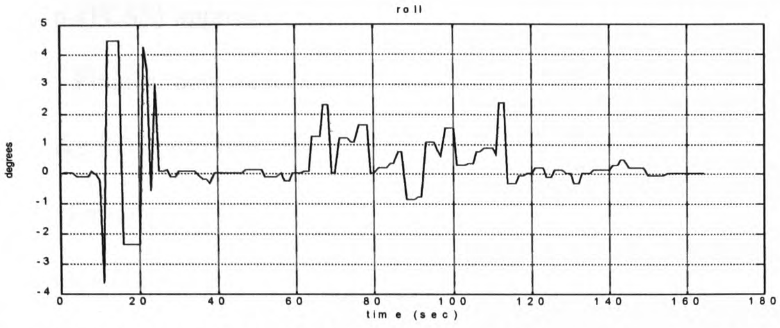


Figure 6.35 The roll due to the final experiment is not more that 5° *degrees*

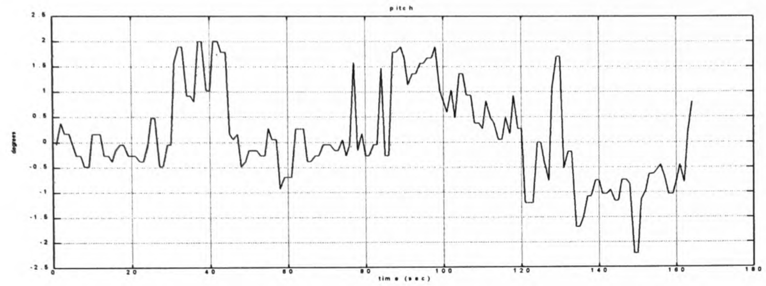


Figure 6.36 The pitch due to the final experiment is not more that 2.5° *degrees*

6.10 Summary

In this chapter the development of a controller for an underwater vehicle is presented based on the approach proposed in Chapter 5. Two independent Fuzzy-like PD Controllers are used to control the *yaw* and the *depth* of the vehicle respectively. The design aspects of these controllers are the input/output scaling factors and the parameters of the fuzzy logic controllers. The design properties of the latter are based on standard techniques coming from fuzzy set theory. General rules of how to design fuzzy logic controllers are proposed. Moreover, the former were chosen due to the global effect on the controller's performance and robustness. The optimisation and tuning of these scaling factors are based on the analysis presented in the previous chapter. Therefore, in the implementation of the designed controller for the underwater vehicle the performance criteria were defined as well as the parameters/factors and the possible tuning control factor levels and the appropriate orthogonal array was selected. Moreover, experiments (in a real environment) were conducted and their analysis after applying the Fuzzy Combined

Scheduling System (FCSS) approach defined the (possible) optimal parameters of the Fuzzy-like PD controller. Finally, new experimental trials were applied to verify and tune both yaw and depth Fuzzy-like PD controllers' parameters. Analysing the results of the final experiments the satisfactory performance of the controller in terms of keeping the path of the navigation to the desired one, and/or changing the path according to set point changes is presented.

6.11 References

Berkan, R. C., and Trubatch, S. L. 1997. *Fuzzy systems design principles building Fuzzy IF-THEN rule bases.* New York: IEEE Press. 0780311515.

Bernard, J. A. 1988. Use of Rule-Based System for Process Control. *IEEE Control Systems Magazine*, **8**, pp. 3-13.

Espinosa, J., and Vandewalle, J 1997. *Fuzzy Modelling and Identification, A guide for the user.* The IEEE International Symposium on Control Theory and Applications. Singapore.

Fowlkes, Y. W. 1995. *Engineering Methods for Robust Product Design Using Taguchi Methods in Technology and Product Development.* 0-201-63367-1.

Jantzen, J., 15 May 1998. "Design of Fuzzy Controllers." Denmark: Technical University of Denmark.

King, P. J., and Mamdani, E. H. 1977. The Applications of Fuzzy Control Systems to Industrial Process. *Automatica,* (13), pp. 235-242.

Kosko, B. 1992. *Neural Networks and Fuzzy Systems, A Dynamic Systems Approach to Machine Intelligence.* Prentice Hall, Inc.

Lee, C. C. 1990a. Fuzzy Logic in Control systems: Fuzzy Logic Controller Part I. *IEEE Transactions on Systems, Man and Cybernetics,* **20**, pp. 404-418.

MacVicar-Whelan, P. J. 1977. Fuzzy Set for Man-Machine Interactions. *Inter. Journal of Man-Machine Studies,* **8**, pp. 687-697.

Mamdani, E. H., and Assilian, S. 1975. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *Int. J. Man-Machine Studies,* 7 (1), pp. 1-13.

Mudi, K P., and Pal, R. N. 1999. A Robust Self-Tuning Scheme for PI- and PD - Type Fuzzy Controllers. *IEEE Trans. on Fuzzy Systems,* 7 (1), pp. 2-16.

Nguyen, H. T., and Prasad, R N. 1999. *Fuzzy Modelling and Control: selected work of M. Sugeno.*

Ogata, K. 1990. *Modern Control Engineering.* 2nd end. Prentice Hall Int. 0135987318.

Passino, K M., and Yurkovich, S. 1998. *Fuzzy control.* Menlo Park, Calif: Addison-Wesley. 020118074X.

Pedrycz, W. 1993. *Fuzzy Control and Fuzzy Systems.* 2nd end. New York: John Willey and Sons.

Pedrycz, W. 1994. Why Triangular membership Functions? *Fuzzy Sets and Systems,* **64**, pp. 21-30.

Phadke, M. S. 1989. *Quality Engineering using Robust Design.* USA: Prentice-Hall International Inc. 0-13-745167-9.

Pok, Y. M., and Xu, J. X. 1994. *Why is Fuzzy Control Robust?* Third Int. Conf. on Fuzzy systems. pp. 1018-1022,

Procyk, T. J., and Mamdani, E. H. 1979. A Linguistic Self-Organising Process Controller. *Automatica,* **15**, pp. 15-30.

Reznik, L. 1997. *Fuzzy Controllers.* UK: Reed Elsevier plc. 0-7506-3429-4.

Santos, M.,Dormido, S., and De La Cruz 1996. Derivative Action in PID-Fuzzy Controllers. *Cybernetics and Systems,* **27**, pp. 413-424.

Sugeno, M. 1985. *Industrial Applications of Fuzzy Control.* New York: Elsevier Science Pub. Co. 0444878297.

Sugeno, M., and Yasukawa, T. 1993. A Fuzzy Logic Based Approach to Qualitative Modelling. *IEEE Trans. on Fuzzy Systems,* **1** (1), pp. 7-31.

Takagi, T., and Sugeno, M. 1985. Fuzzy Identification of Systems and its Applications to Modelling and Control. *IEEE Trans. on Systems, Man and Cybernetics,* **SMC-15**, pp. 116-132.

Tang, K. L., and Mulholland, R. J. 1987. Comparing Fuzzy Logic with Classical Controller Designs. *IEEE Transactions on Systems, Man and Cybernetics,* **SMC-17** (6), pp. 1085-1087.

Tong, R. M 1985. An Annotated Bibliography of Fuzzy Control. *In:* Sugeno, M. ed. *Industrial Applications of Fuzzy Control.* pp. 249-269.

Yager, R. R., and Filev, D. P. 1994b. *Essentials of Fuzzy Modelling and Control.* New York: John Wiley. 0471017612 (acid-free).

Zheng, L. 1992. A Practical Guide to Tune of Proportional and Integral (PI) Like Fuzzy Controllers. *Proc. FUZZ IEEE,* pp. 633-641.

7

Review, Conclusions and Future Work of the thesis

7.1 Introduction

In this chapter the work presented in this thesis is reviewed highlighting the approaches that have been proposed to solve modelling and control problems for non-linear systems. The implementation of these proposed approaches has been achieved by applying them to the modelling the yaw dynamics, the control strategy of object avoidance and developing a controller for an underwater vehicle. Some suggestions for future work are also proposed.

7.2 Approaching the modelling problem of non-linear systems

Chapter 2 addressed the modelling of complex, non-linear, or partially known systems, where very limited experimental data and little knowledge about the systems behaviour are available.

It has been shown that modelling techniques based on fuzzy sets attempt to combine numerical and symbolic processing into one framework. Fuzzy systems are knowledge-based systems consisting of linguistic *If-Then* rules that can be constructed using the knowledge of an expert and/or the design properties of the experiments implemented to collect data from a given field of interest. Moreover, fuzzy systems are universal approximators that can approximate non-linear mappings using a number of available data. This duality allows qualitative knowledge to be combined with quantitative data in a complementary way. The drawback of this type of modelling, however, is its trade-off between accuracy and linguistic meaning. It is well known that with this type of modelling, when the accuracy is increased the linguistic interpretation is reduced.

Using neuro-fuzzy techniques the modelling is based mostly on adaptive neural networks that are functional equivalent to fuzzy inference systems. This type of approach has a high degree of accuracy and they can also be initialised based on some linguistic knowledge. However, often the structure of the fuzzy inference system changes dramatically according to the complexity or non-linearity of the system and thus the resulting modified linguistic meanings may have no practical significance.

By considering the problems of the two modelling approximation techniques described above (fuzzy and neuro-fuzzy) it is concluded that:

- fuzzy modelling provides a more transparent representation of the non-linear systems under study using a linguistic interpretation in the form of rules, but its accuracy depends on good extraction and definition of the knowledge of the system.

- neuro-fuzzy modelling can identify non-linear systems with high accuracy based on function approximation techniques, but its transparency is very low resulting in no significant information.

In chapter 2 the combination of the above two techniques was proposed. The main idea of this hybrid approach is to describe highly non-linear systems globally, using local models with small degrees of non-linearity. The latter can be approximated with high accuracy using neuro-fuzzy modelling techniques, whereas the former defines, using a proposed *fuzzy supervised scheduling system* approach, which of the local models are the representative ones and to which degree. The proposed hybrid approach uses the advantages of the function approximation as well as linguistic acquisition techniques using neuro-fuzzy and fuzzy modelling technique respectively.

It was suggested that the approximation of the local models can be achieved by using the Adaptive Network-based Fuzzy Inference System (ANFIS) architecture. It has been shown that this type of neuro-fuzzy modelling has very high accuracy in approximating non-linear sub-systems utilising only a very small number of training data and algorithmic iterations. The properties of the *fuzzy supervised scheduling system* are defined based on knowledge interpretation techniques, experimental design properties and fuzzy set theory as discussed in chapter 2.

The implementation of the proposed hybrid fuzzy and neuro-fuzzy method was employed for the modelling of the yaw motion of an underwater vehicle. The experimental design properties that were set to extract information in terms of the dynamics of the vehicle's yaw motion, the power of the vertical propellers and the desired yaw angle determined the fuzzy variables that construct the *fuzzy supervised scheduling system* that defines the global model. Using a small number (only nine) of experiments in a real environment, three different groups of local models were defined according to the course-changing action and then approximated using the ANFIS neuro-fuzzy modelling method.

It is appreciated that with only nine experiments it was difficult to have a model with high accuracy. Additional trials/experiments providing new data would be a means of improving the model. However, for this work only the yaw mode was considered. Further work should consider modelling of the depth dynamics.

7.3 The Identification and Modelling of Control Strategies using Fuzzy Clustering Methods

7.3.1 The proposed algorithmic approach

In Chapter 3 it was shown how fuzzy clustering methods can be used as techniques for identifying and modelling of control strategies based on availability of input/output mapping data. The information that these data may contain can be extracted based on similarities between them that actually define the clusters to which they belong. These similarities can be identified using "hard" or fuzzy clustering methods. In non-fuzzy "hard" *clustering*, the boundary between different clusters is crisp, such that one pattern is assigned to exactly one cluster. On the contrary, *fuzzy clustering* provides partitioning results with additional information supplied by the cluster membership values indicating different degrees of belonging. The "hard" clustering method is unrealistic to be used for the investigation of similarities of the data set that represents real systems, as the data cannot always belong to only one cluster, thus a *fuzzy clustering* method is more suitable.

Fuzzy clustering methods are based on algorithms that aim to minimise a cost function regarding the degree to which the data belongs to the clusters and the degree of dissimilarity between them. *Fuzzy c-mean* is one of the first and fundamental algorithms used in clustering analysis. This algorithm provides information on the location of the cluster's centres. However, the shapes of the identifying clusters using this method are usually hyperspheres based on a fixed distance norm. It may therefore force the objective function to prefer clusters of that shape even if they are not present. As a result of this drawback some extensions to the fuzzy c-mean algorithm have been made with the most widely used Gustafson-Kessel algorithm. These types of algorithms have the ability to adapt the distance norm and therefore can define the shape of the cluster, which is approximated as hyperellipsoid, according to the data distribution using fuzzy covariance matrix. The eigenstructure of this matrix provides information about the cluster's shape and the orientation in any n-dimensional input-output space. The size and shape of the ellipsoid show how the inputs and outputs relate to each other in some region of the state space.

The main disadvantage of the above algorithms is that they cannot define the number of possible clusters that the data belongs to. Several approaches have been proposed in the literature and in this thesis a method called mountain clustering is used as a method to determine the number of clusters as well as the approximate position of their centres. These are the initial setting for more sophisticated and accurate clustering algorithms, such as fuzzy c-means and Gustafson-Kessel, to define the actual positions of prototypes and the variances of the clusters.

One of the most satisfying ways to extract the linguistic meaning of the defined clusters in a multi-dimensional space is to project their centres and variances into each dimensional axis. These projections define the set properties of the input/output membership functions assigned by linguistic labels regarding the universe of discourse, which gives a transparency to the system's analysis. Moreover, merging some of them can reduce the number of generated membership function sets in the system under consideration and makes it linguistically more tractable. It was proposed that the merging between neighbouring fuzzy sets should be applied according to: a) their closeness removing therefore highly overlapping fuzzy sets, b) numbers of clusters recommended from the mountain method, and c) observing their linguistic meaning.

It has been shown that the resulting membership functions can construct the rule table with the minimum number of rules. Using these rules the simplified rule base makes it easier to assign qualitatively meaningful linguistic terms to the fuzzy sets, and it reduces the number of terms needed. Thus, it becomes easier for experts to validate the model and users can understand better and more quickly the operation of the system. A model with fewer fuzzy sets and fewer rules is also better suited for the design and implementation of a non-linear controller, or for simulation purposes, and it has lower computation demands.

The main contribution of the work presented in this chapter was an algorithmic methodology based on choices of different fuzzy clustering algorithms, projection of clusters and merging techniques. The idea of the proposed methodology was to combine the best features of well-known clustering methods such as the mountain method, fuzzy c-means and Gustafson-Kessel to produce a strong algorithm. The projection of the prototypes and variables of the clusters is also a recognised approach to infer the information that is included in the data clusters into fuzzy sets. Merging these fuzzy sets based on the proposed guidelines can minimise the number of rules and make the identifying control strategy more transparent. Finally, some improvement of the resulting fuzzy system can be achieved by using optimisation methods such as gradient method. Furthermore, it was shown that these optimisation methods should be applied only in the output singletons and only if their resulting modifications do not lose their linguistic meanings.

Since the number of methods that combine to construct the proposed methodology is high, it gives high flexibility to the designer in terms of what and how the methods (tools) should be used. The proposed methodology is based *on the right choice of the right tools*. The proposed algorithmic method can be described as universal approximation in terms of identifying and modelling non-linear control strategies.

7.3.2 The generation of fuzzy rules for "Avoiding Objects" Control Strategy based on the proposed approach

The proposed algorithm is a systematic straightforward method and it has been applied to optimise the control action of an underwater vehicle in terms of avoiding objects to show its capabilities as discussed in Chapter 4. The applied inputs are the distance and the angle between the vehicle and the objects. The resulting control surface is very close to the original one and thus the algorithm works very satisfactorily i.e. Average Percentage Error (APE) between 0.0817% and 0.0519%.

7.4 Control Systems using Fuzzy Logic & the Taguchi Method

7.4.1 The Design of controllers for non-linear systems and tuning approaches

Chapter 5 described an innovative systematic and synergistic approach to optimise the factor/parameter levels of a Fuzzy-like PD controller aided by a design of experiment method, namely the *Taguchi* method and their tuning using a proposed method named *Fuzzy Combined Scheduling System*. The design structure of a Fuzzy-like PD controller involves a large number of degrees of freedom. Such control systems are inherently non-linear and robust and their performance depend mainly on the appropriate selection of the tuning factor levels. Appropriate selection of levels implies analysis of the most significant factor levels in terms of their performance and interaction characteristics. One way to do this analysis is to apply the Taguchi robust design of experiment method. Robust design is a methodology for finding the optimum setting of control factors to make the system insensitive to noise factors. The important features of this approach is that a minimum number of experiments need to be conducted, which were enough to investigate the significance and robustness of the factors together with their interactions even when these factors are inhomogeneous. For instance, in the second case study shown in chapter 5, although a full factorial would require 81 experiments 27 experiments were enough for the analysis.

The scheme requires the designer to decide initially which factors will be used for tuning and what are their possible levels. For the first decision however, even if an inappropriate choice of factor(s) is made their verification can be done during the Taguchi analysis, where their significance is measured and ranked. This is a very important task especially when large numbers of candidate factors are introduced. For the second decision, the Taguchi method investigates and defines which are the optimal combinations of the set factor levels, and therefore only the significant levels are considered for further use. After extended simulation studies on a Fuzzy-like PD control system that uses more than three levels for each factor, it was observed that no significant differences existed in terms of performance output whereas the analysis become more complicated when more levels are considered. Therefore, in the case studies outlined in chapter 5 only three levels for each factor were used. In this study the factors of Fuzzy-like PD controllers that were chosen to be the tuned ones, were the Scaling Factors (SFs) and the peaks of the fuzzy logic Membership Functions (MF's). Additionally the significance of the interactions between the factors was investigated. When a significant interaction between an important factor and an unimportant factor exist, the unimportant factor has also to be considered due to its interacting effect. It is important to note that, with this approach an interaction between a SF and a peak of MF that is inhomogeneous can be studied. This investigation is usually too complex for other approaches and is usually avoided even when it could be very critical. Thus, the ranking of the significance in terms of robustness characteristics of the factors, levels and interactions can give precise information for the designer to make the right decisions in terms of which factor levels should be used to set the initial optimal factors and design the tuning rules.

In relation to the performance aspects of the factor levels a fuzzy tuning inference mechanism approach was proposed. The antecedent of the rules included the examination of the performance state, whereas the consequence part is constructed by the tuning aggregations that took into account the analysis of the factor levels. However, some consideration should be taken during construction of these rules in terms of (i) consistency of the rules, (ii) equivalence of the factor levels, (iii) homogeneity of the control rules and (iv) priority of the tuning factor levels.

The overall output of the tuning factor levels results from the proposed *fuzzy combined scheduling system* method. The contribution of this method was that for each performance state measurement, both *current* and *tuning* factor levels were considered, defined as two overlapping MFs, where their degrees of membership defined their actual weights.

The performance criteria were used independently and for each of them the setting of two MFs together with the universe of discourse may vary according to the particular criterion. The identification of the *current* SFs (if unknown or the system's dynamics change) is a challenge for this optimisation problem. In this case the SF's value could be optimised using another fuzzy scheduling system that constructs its rule based on the interaction plots resulting from the analysis of mean in the Taguchi method. The rules resulting from this analysis could be used to model the controllers' parameters (e.g. gains) and therefore used as the *current* parameter values.

Another important aspect of the proposed approach was that it does not depend on the performance index characteristics. It was therefore able to deal with any performance criteria and any number of them. However, increasing this number complicates the analysis.

In this approach, the philosophy was that if data, even from a small number of experiments is available, a systematic analysis of the results can give enough information to optimise the factor's/parameter's levels for the performance criteria defined by the designer. This can be achieved using orthogonal arrays to design the experiments that excite all dynamics/parameters of the system. This is a very important issue in the analysis of non-linear systems. Taguchi method analyses the experimental results, which provide information about the significance of each of the chosen control factors as well as the interactions between them. Using this information the detection of the subspace to which the optimal factor levels belong that warranties the robustness of the system was investigated. Then measuring the different performance states and using a fuzzy tuning inference mechanism the overall values of the factor levels were defined regarding the tuning and the current factor levels at a particular time.

The approach therefore, can be very useful in developing control systems where no mathematical model of the system exists and/or it is difficult to undertake large number of experiments. Moreover, the approach can also be used when a good mathematical model and/or a real system is available and simulations or real experiments can be undertaken, extracting information about the design aspects of a required control system.

In chapter 5 the SFs and peaks of MFs of a Fuzzy-like PD controller for a second-order system were used as a case study to test the proposed method and the results are very satisfactory showing a minimum of 50% improvement in IAE and ITAE performance indexes. However, the proposed synergetic approach can easily be adapted as a general approach where other parameters of the FLCs may be studied as tuning factors. The analysis of interactions between inhomogeneous factors is an interesting task using the Taguchi method, where more investigation of different FLC's factors could be useful to identify their inter relationship. Moreover, the performance criteria, which are chosen by the designer, play a critical role in the analysis of the results. Their significance could be defined using the same proposed approach, but in this case the results may by considered as the performance of the tuning factors. Finally, in using the *fuzzy combined scheduling system* approach, the response investigation of the tuning factor levels, that facilitates the extraction of the tuning fuzzy rules, stems from the combined evaluation of all the performance criteria. This is a very important aspect especially when a large number of performance criteria are used and thus their analysis in a global manner is very complicated.

7.4.2 The implementation of the proposed approach to design a controller for an underwater vehicle

The proposed approaches presented in chapter 5 were applied to define Fuzzy-like PD controllers for an underwater robot named GARBI as described in Chapter 6. The design of GARBI's controller involves course-changing and course-keeping for both steering and depth control performances. The architecture of the controller was based on two independent Fuzzy-like PD controllers for each controlled variable, yaw and depth. The design aspects of the FLC as well as the "PD" part for these types of controllers was presented. It was shown that the flexible structure of the FLC leaves the designer to decide about the input/output universe of discourse and linguistic variables in terms of their shape, number and meaning, the construction of the rule base and the meaning of their operators and the defuzzification method. The above decisions were guided from particular design aspects coming from the fuzzy system theory as well as from the expert's knowledge in terms of system's dynamics and behaviour identification. Moreover, the gains of the input/output SFs of the PD part of the controller were defined according to their global effect on the dynamics of the close-loop control system. Although the SFs were defined based on some general instruction mostly from the classical PID design theory in chapter 6, a more systematic procedure to investigate and optimise its controller's parameters was introduced. This optimisation was based on experiments in a real environment and were planned using the Taguchi method for design of experiments.

It was shown that the Taguchi Design of experimental method could help to minimise the number of experiments, using only nine experiments in the orthogonal array, without the risk of losing vital information. This was very important as the experiments were held in a real environment, where time and money is an issue. However, the work that had to be undertaken before and during these real experiments was a very challenging task and required not only the facilities availability (such as hardware/software), but also team working skills.

The preparation before the trials can be summarised as hardware (calibration and possible modifications/improvements) as well as software (developing and testing). During the experiments, moreover, the schedule of the planned experiments was followed. Any change in the conditions such as weather, hardware problems etc. were be recorded and accounted for accordingly.

The results of the experiments were presented and analysed extensively. Using analysis of means and the average response plot resulting from the performance criteria measurements, it was shown that the *optimal tuning factor levels* could be defined. The construction of the tuning fuzzy rules was defined respecting the performance criteria defined for the IAE and ITAE. Finally, the *optimal tuning factor levels* of both yaw and depth controllers' SFs were verified and tuned whilst applying the *fuzzy combined scheduling system* approach in the final trials.

It has, therefore been shown that from the approach presented in chapter 5, successful controllers can be implemented for non-linear systems. Furthermore, using the proposed systematic method, GARBI acquired a Fuzzy-like PD controller, that performs satisfactorily since both course-changing and course-keeping performance is within the desired response with minimal error. It has been shown therefore, that Fuzzy-like PD controllers designed, optimised and tuned by the proposed approach possesses features that are attractive in navigation control problems posed by underwater vehicles. Both the developed yaw and depth controllers are now in operation in GARBI's navigation control system.

In this research it has been shown, also, how a fuzzy controller combined with conventional PD control techniques can help to design a robust controller, dealing with the uncertainties and non-linearities of an underwater vehicle.

7.5 Summary

The purpose of this research was to propose approaches for modelling and control of non-linear systems using fuzzy logic technique. In this thesis it has been shown that bridges have been made between fuzzy logic and other "intelligent" or conventional techniques such as neural networks, Clustering methods, PID controllers and Taguchi method to achieve the main purpose. The main contributions that have been proposed and developed during this research are:

- Modelling non-linear system using a Hybrid Fuzzy and Neuro Fuzzy approach based on the fuzzy supervised scheduling system and neuro-fuzzy modelling techniques with the application to model the yaw dynamics of an underwater vehicle.
- An algorithmic methodology for identification and modelling of a complex system's control actions using fuzzy clustering methods.
- Application of the algorithmic fuzzy clustering approach to construct control strategies for "avoid objects".
- An innovative approach to determine the optimal parameters of fuzzy and fuzzy-like PD controllers in terms of robustness and tuning characteristics based on the Taguchi design of experiments method, the construction of fuzzy rules to tune the factor levels approach and the fuzzy combining scheduling system approach.
- The successful implementation of the Fuzzy-like PD controller for course-changing and course-keeping in an underwater vehicle.

Appendix A

Local models using Neuro-Fuzzy ANFIS Architecture

In this appendix the training data that are used to define the local model that represents the yaw of GARBI underwater vehicle are presented in figures as defined in Table A 2.1 when the power and the angle are as in the 4th, 5th columns.

Training data for			
turning		Angle	Power
30° to 60°	Figure A.1	Low	Low
	Figure A.5	Low	Medium
	Figure A.9	Low	High
60° to 90°	Figure A.13	Medium	Low
	Figure A.17	Medium	Medium
	Figure A.21	Medium	High
90° to 180°	Figure A.25	High	Low
	Figure A.29	High	Medium
	Figure A.33	High	High

Table A 2.1 Figures that shows the training data for turning 30°, 60°, 90° and 180° degrees left and right

The membership functions, the rules and the resulting FIS approximation output of the ANFIS algorithm are presented in figures as defined in Table A 2.2 when the power and the angle are as in the 2nd, 3rd, 5th, 6th, 8th and 9th columns.

Yaw of GARBI underwater vehicle when turns from								
straight ahead to left or right	Angle	Power	left to right	Angle	Power	right to left	Angle	Power
Figure A.2	Low	Low	Figure A.3	Low	Low	Figure A.4	Low	Low
Figure A.6	Low	Medium	Figure A.7	Low	Medium	Figure A.8	Low	Medium
Figure A.10	Low	High	Figure A.11	Low	High	Figure A.12	Low	High
Figure A.14	Medium	Low	Figure A.15	Medium	Low	Figure A.16	Medium	Low
Figure A.18	Medium	Medium	Figure A.19	Medium	Medium	Figure A.20	Medium	Medium
Figure A.22	Medium	High	Figure A.23	Medium	High	Figure A.24	Medium	High
Figure A.26	High	Low	Figure A.27	High	Low	Figure A.28	High	Low
Figure A.30	High	Medium	Figure A.31	High	Medium	Figure A.32	High	Medium
Figure A.34	High	High	Figure A.35	High	High	Figure A.36	High	High

Table A 2.2 Figures that shows the Neuro-fuzzy local model defined by ANFIS algorithm

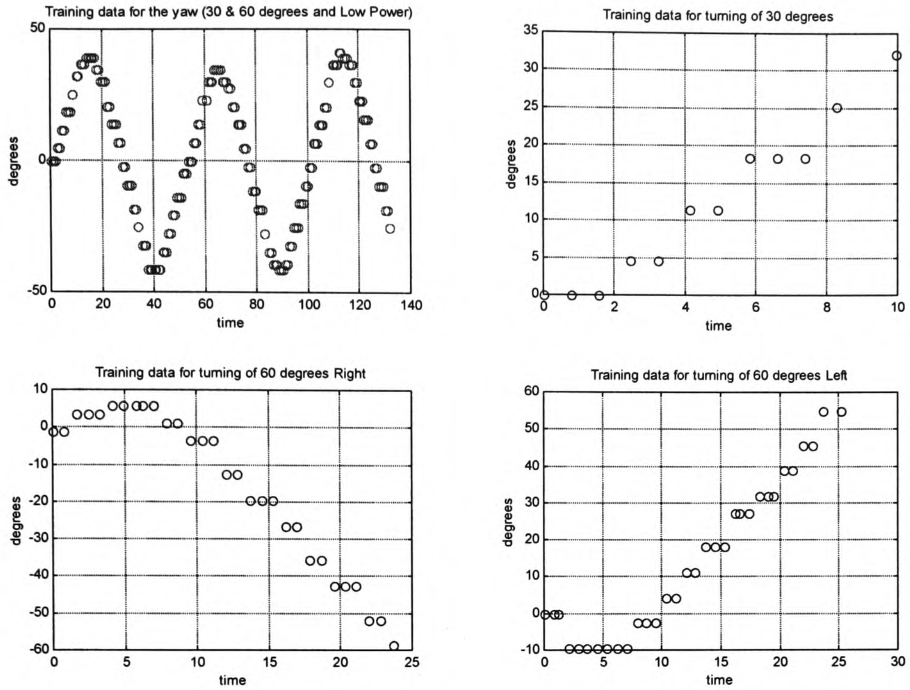
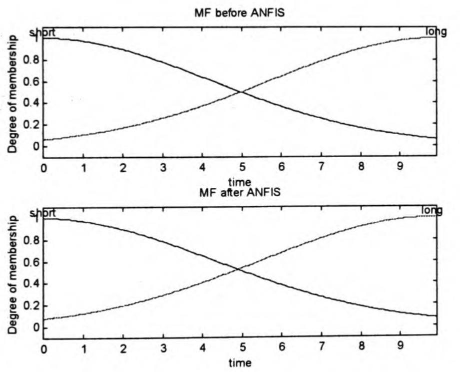


Figure A.1 Training data for turning 30°, 60° *left and right (Low Power-Low Angle)*



Rule 1:

If time is SHORT
Then angle is $\theta = 1.762t - 1.601$

Rule 2:

If time is LONG
Then angle is $\theta = 3.071t + 1.907$

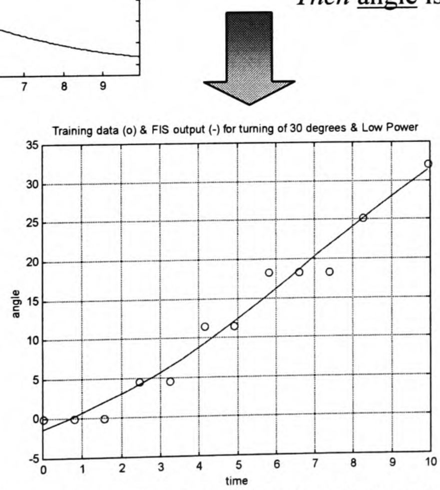
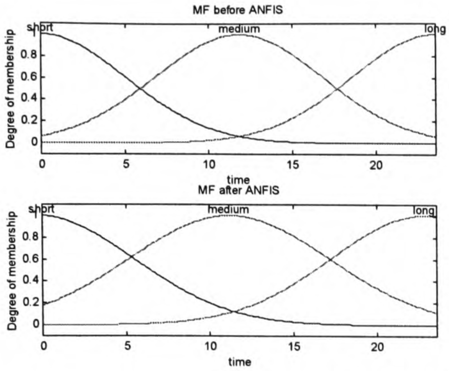


Figure A.2 Neuro-fuzzy local model HLR 1,1 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = 1.403t - 7.103$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = -3.592t - 30.41$

Rule 3:

If time is LONG *Then* angle is
$\theta = -4.345t - 44.93$

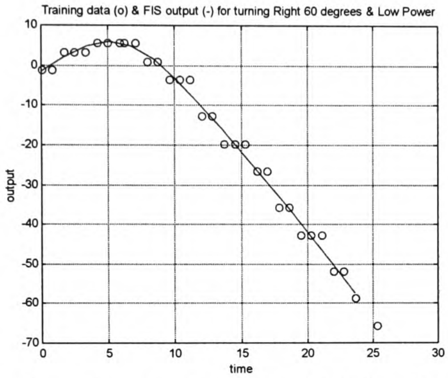
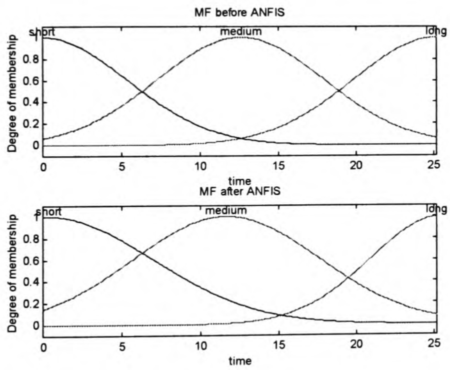


Figure A.3 Neuro-fuzzy local model LR 1,1defined by ANFIS algorithm



Rule 1:

If time is SHORT
Then angle is $\theta = -5.365t + 1.275$

Rule 2:

If time is MEDIUM
Then angle is $\theta = 1.889t + 2.587$

Rule 3:

If time is LONG
Then angle is $\theta = 3.948t - 41.52$

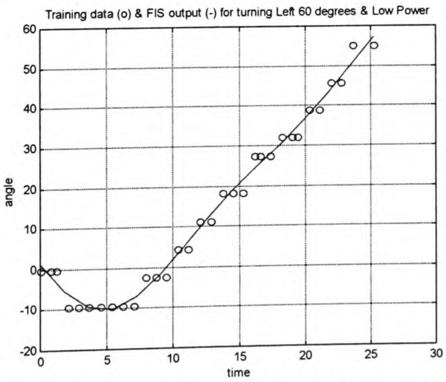


Figure A.4 Neuro-fuzzy local model RL1,1defined by ANFIS algorithm

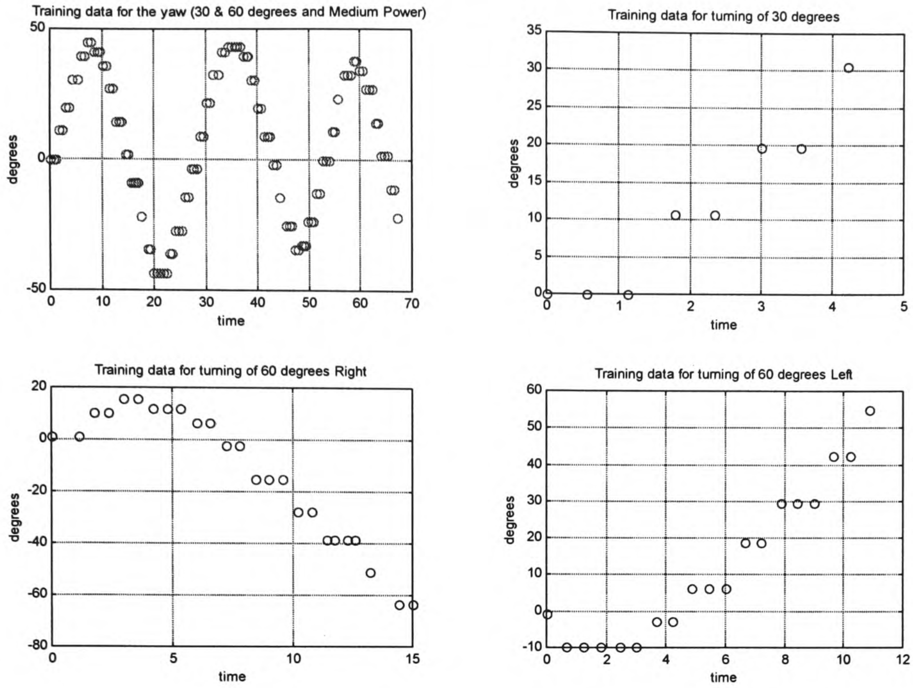
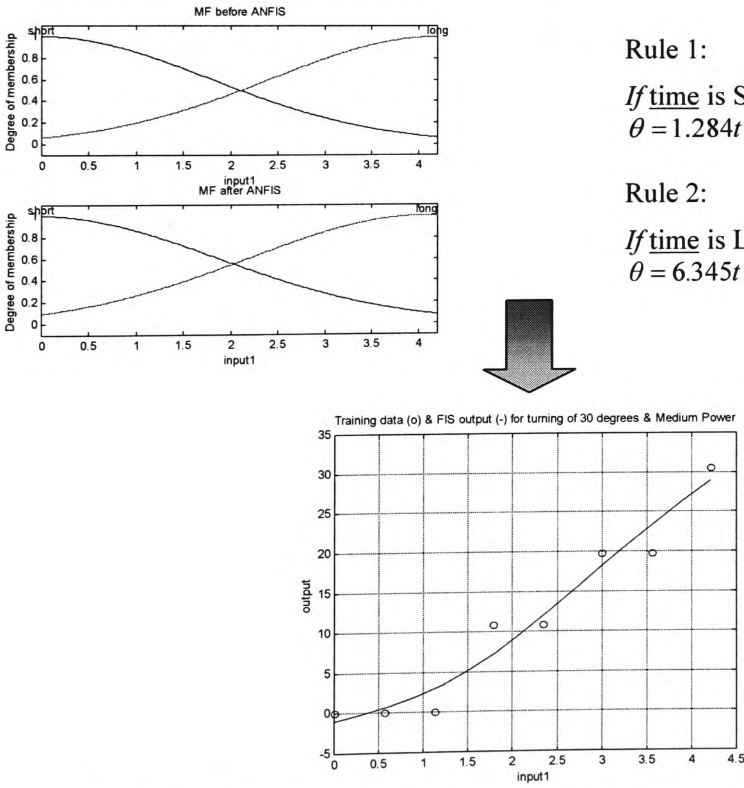


Figure A.5 Training data for turning 30°, 60° *left and right (Medium Power-Low Angle)*



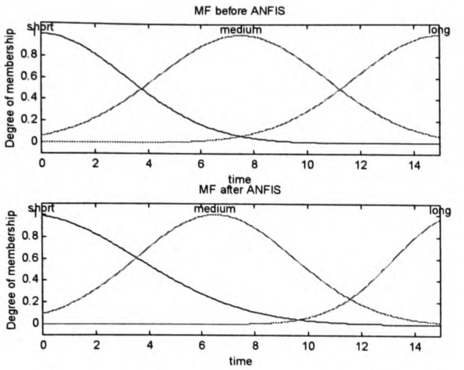
Rule 1:

If time is SHORT *Then* angle is
$\theta = 1.284t - 1.506$

Rule 2:

If time is LONG *Then* angle is
$\theta = 6.345t + 4.282$

Figure A.6 Neuro-fuzzy local model HLR 1,2 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = 0.1795t - 6.041$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = -9.326t + 67.44$

Rule 3:

If time is LONG *Then* angle is
$\theta = -9.942t + 83.32$

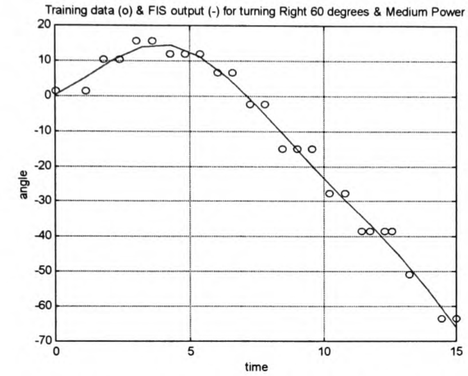
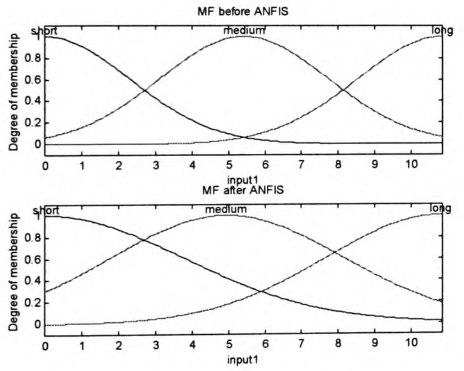


Figure A.7 Neuro-fuzzy local model LR 1,2 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = -16.48t - 8.972$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = 5.773t + 20.95$

Rule 3:

If time is LONG *Then* angle is
$\theta = 13.58t - 96.2$

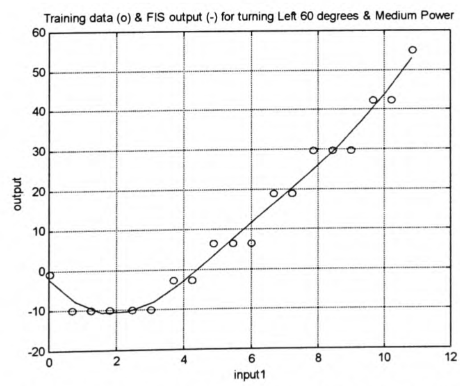


Figure A.8 Neuro-fuzzy local model RL 1,2 defined by ANFIS algorithm

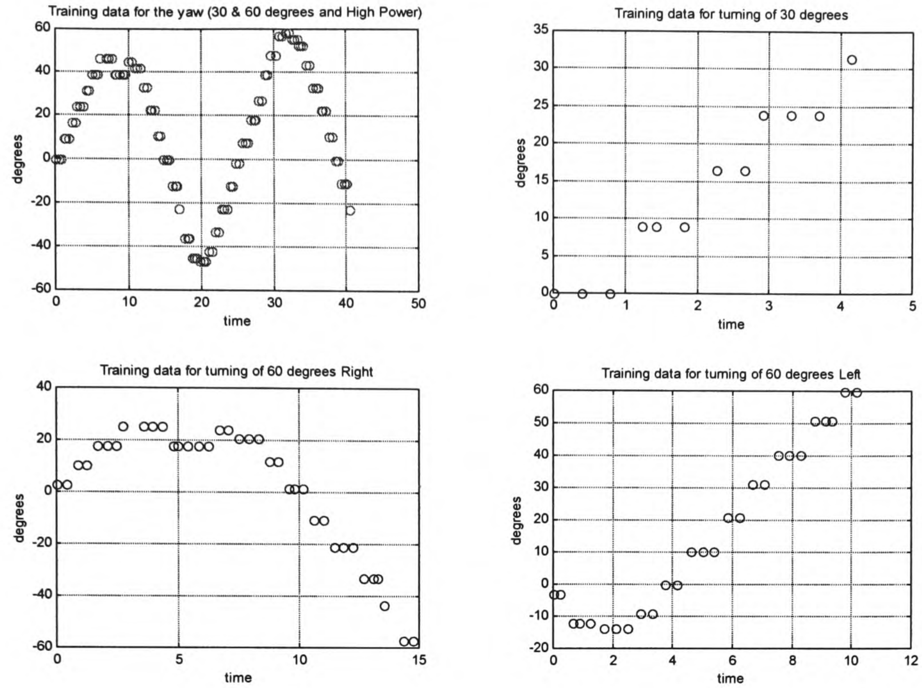
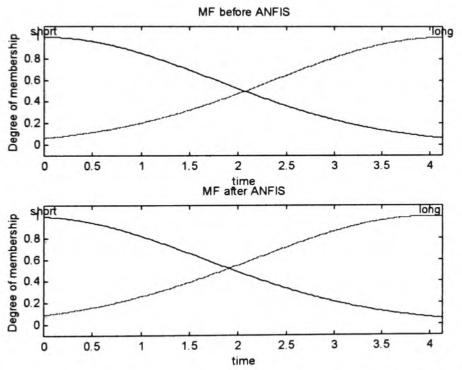


Figure A.9 Training data for turning 30°, 60° *left and right (High Power-Low Angle)*



Rule 1:

If time is SHORT *Then* angle is
$\theta = 3.598t - 2.232$

Rule 2:

If time is LONG *Then* angle is
$\theta = 5.121t + 9.628$

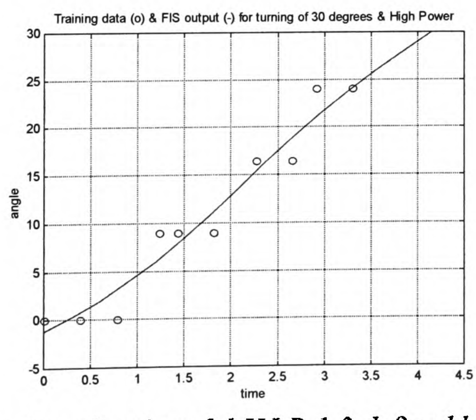
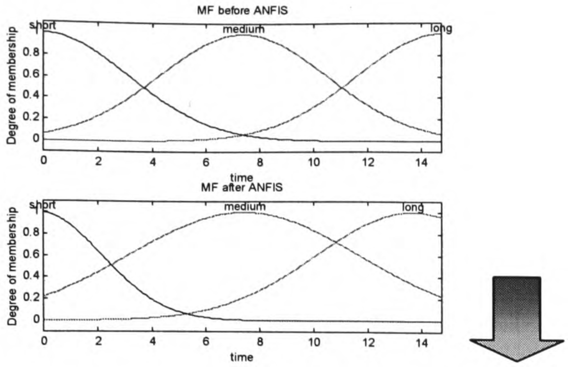


Figure A.10 Neuro-fuzzy local model HLR 1,3 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = 14.63t + 2.919$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = 6.522t - 11.95$

Rule 3:

If time is LONG *Then* angle is
$\theta = -3.792t - 33.91$

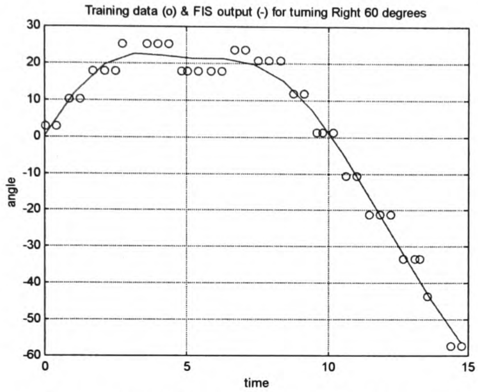
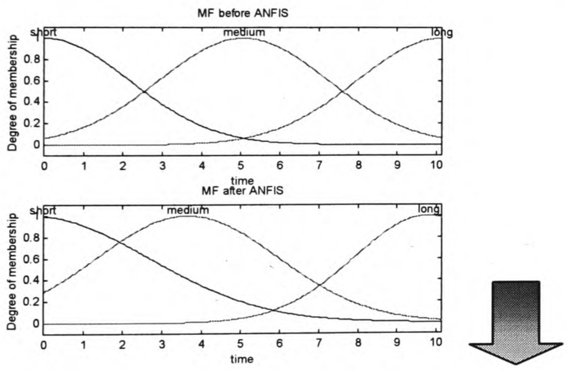


Figure A.11 Neuro-fuzzy local model LR 1,3 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = -19.67t - 1.789$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = 8.231t - 4.45$

Rule 3:

If time is LONG *Then* angle is
$\theta = 8.34t - 23.94$

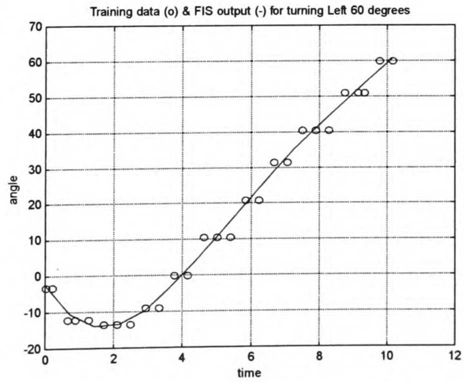


Figure A.12 Neuro-fuzzy local model RL 1,3 defined by ANFIS algorithm

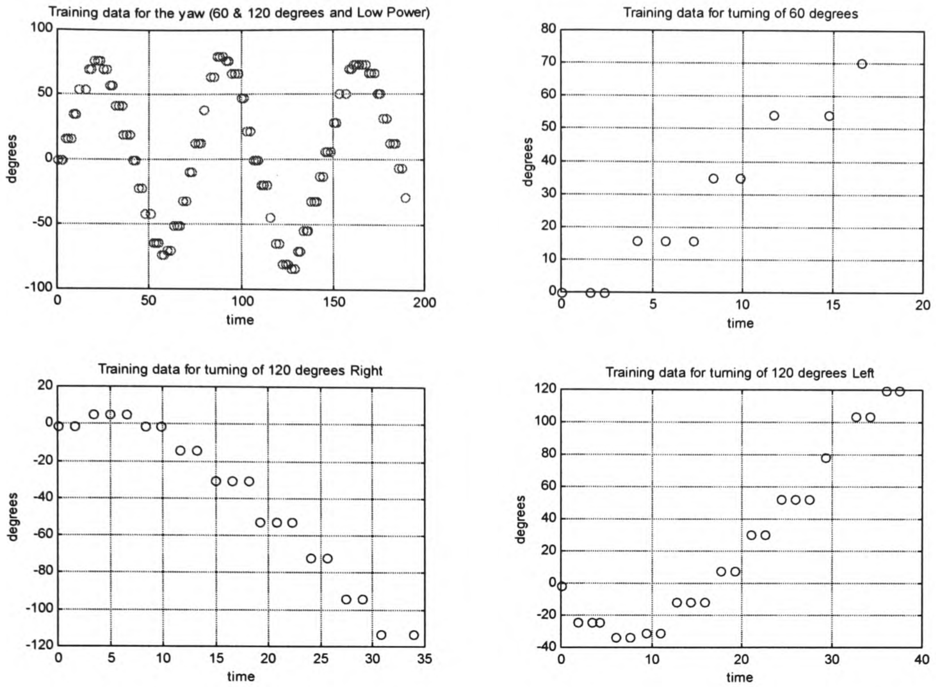
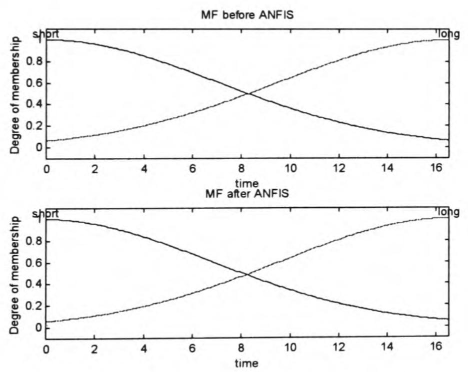


Figure A.13 Training data for turning 60°, 90° *left and right (Low Power-Medium Angle)*



Rule 1:

If <u>time</u> is SHORT *Then* <u>angle</u> is
$\theta = 1.144t - 3.815$

Rule 2:

If <u>time</u> is LONG *Then* <u>angle</u> is
$\theta = 2.048t + 35.91$

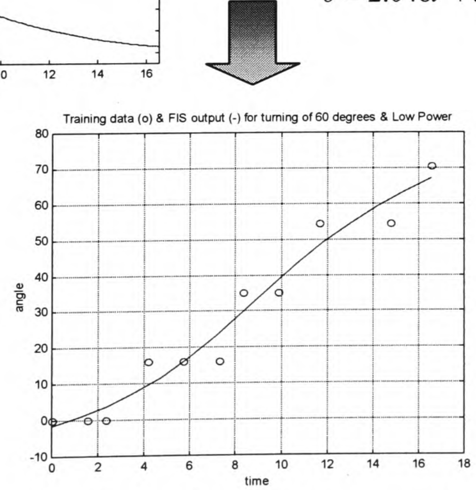
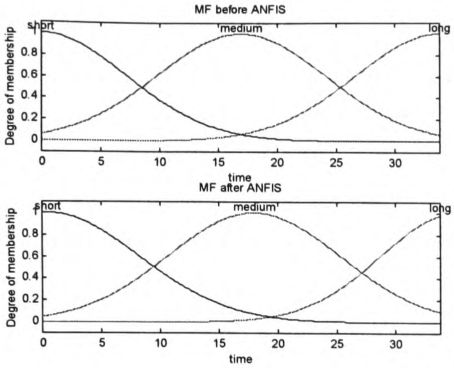


Figure A.14 Neuro-fuzzy local model HLR 2,1 defined by ANFIS algorithm



Rule 1:

If <u>time</u> is SHORT *Then* <u>angle</u> is
$\theta = 2.595t - 1.839$

Rule 2:

If <u>time</u> is MEDIUM *Then* <u>angle</u> is
$\theta = -1.736t - 10.88$

Rule 3:

If <u>time</u> is LONG *Then* <u>angle</u> is
$\theta = -0.01506t - 119$

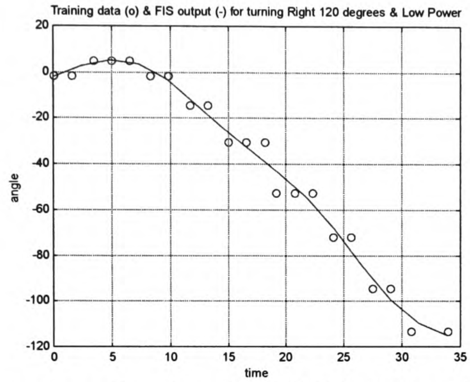
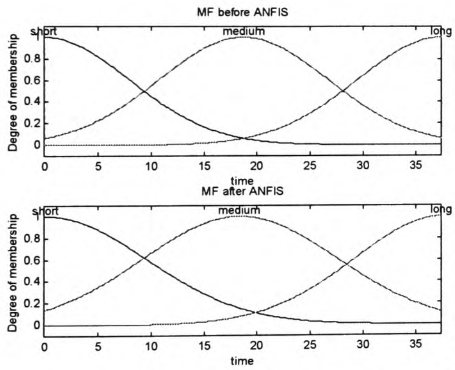


Figure A.15 Neuro-fuzzy local model L R 2,1 defined by ANFIS algorithm



Rule 1:

If <u>time</u> is SHORT *Then* <u>angle</u> is
$\theta = -8.851t - 9.805$

Rule 2:

If <u>time</u> is MEDIUM *Then* <u>angle</u> is
$\theta = 0.1736t + 28.14$

Rule 3:

If <u>time</u> is LONG *Then* <u>angle</u> is
$\theta = 1.889t + 62.4$

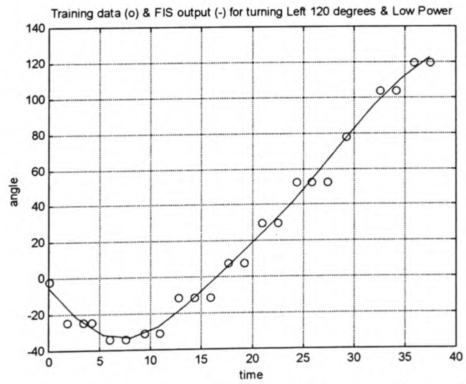


Figure A.16 Neuro-fuzzy local model R L 2,1 defined by ANFIS algorithm

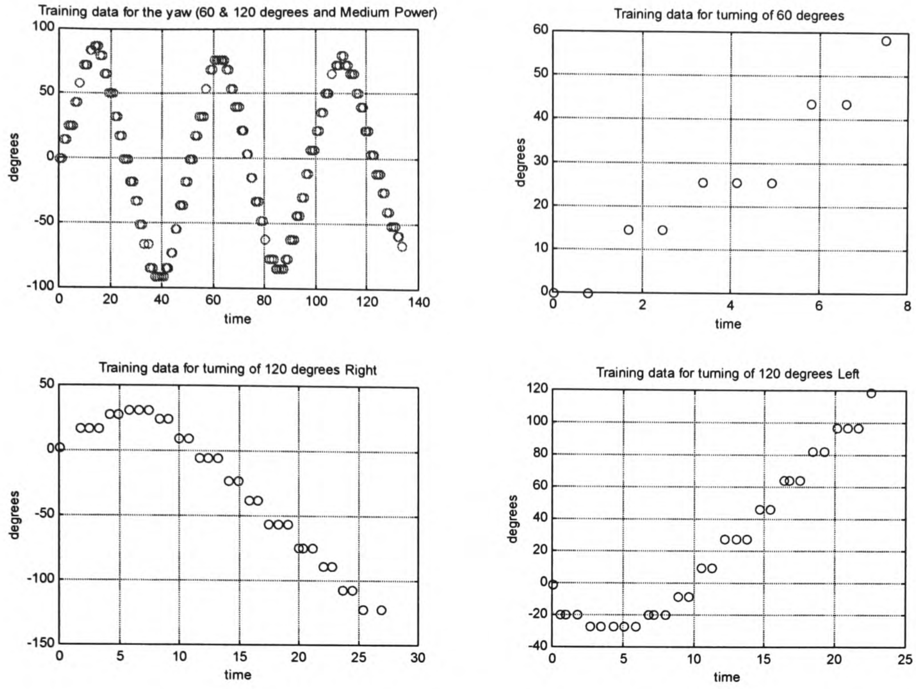
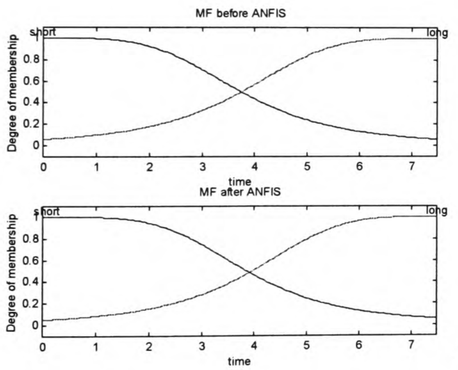


Figure A.17 Training data for turning 60°, 90° *left and right (Medium Power-Medium Angle)*



Rule 1:

If time is SHORT *Then* angle is
$\theta = 9.736t - 0.7143$

Rule 2:

If time is LONG *Then* angle is
$\theta = 12.69t - 38.05$

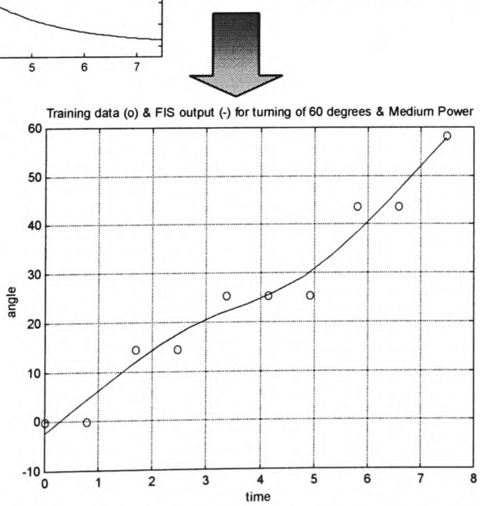
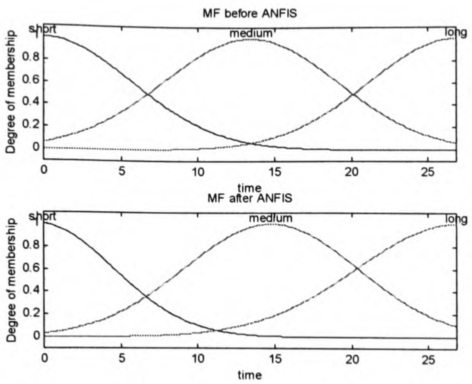


Figure A.18 Neuro-fuzzy local model HLR 2,2 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = 4.374t + 1.817$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = -6.698t - 78.45$

Rule 3:

If time is LONG *Then* angle is
$\theta = -5.803t + 27.93$

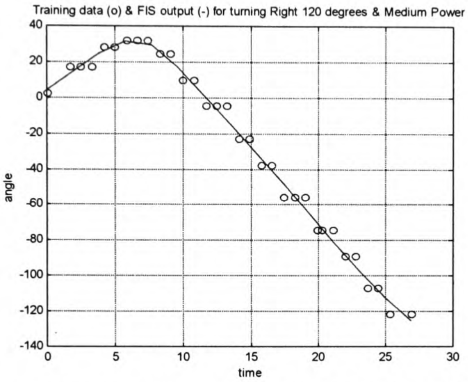
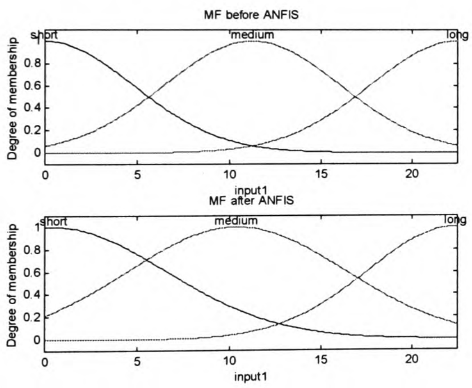


Figure A.19 Neuro-fuzzy local model LR 2,2 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = -10.63t - 7.109$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = 4.973t - 15.48$

Rule 3:

If time is LONG *Then* angle is
$\theta = 7.573t - 54.04$

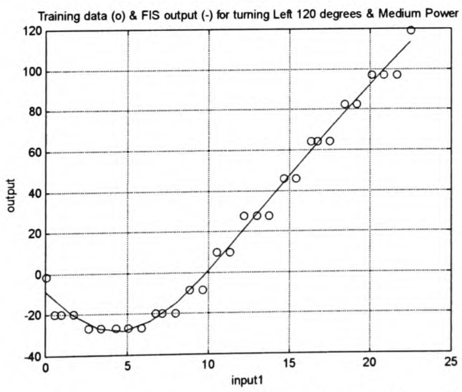


Figure A.20 Neuro-fuzzy local model RL 2,3 defined by ANFIS algorithm

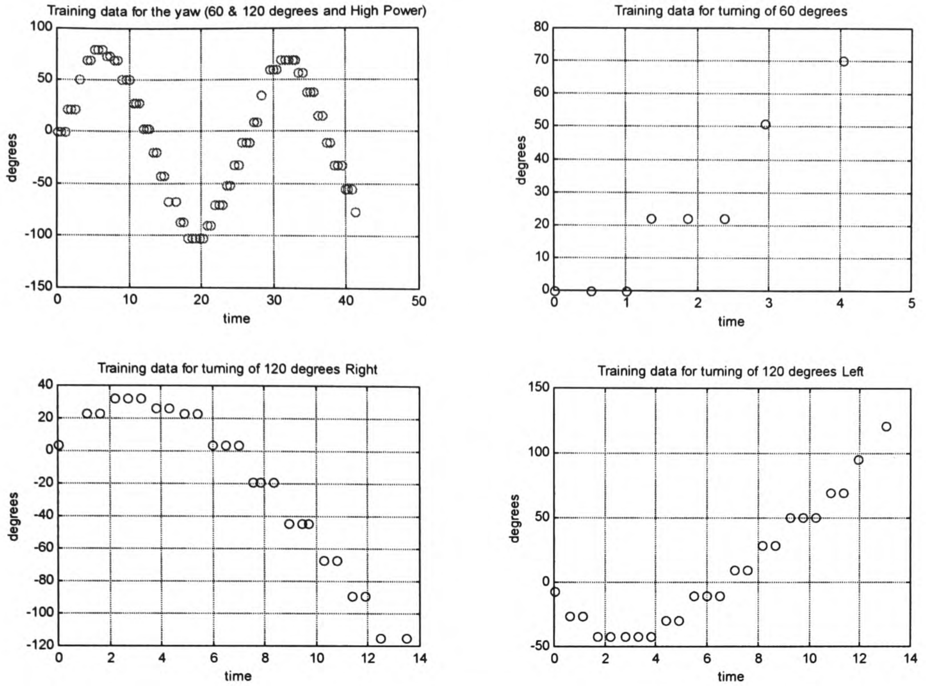
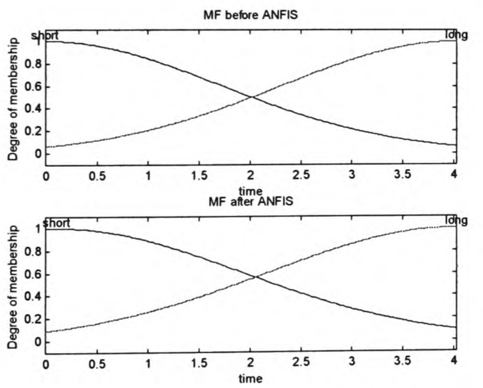


Figure A.21 Training data for turning 60°, 90° *left and right (High Power-Medium Angle)*



Rule 1:

If <u>time</u> is SHORT *Then* <u>angle</u> is $\theta = 1.808t - 0.1103$

Rule 2:

If <u>time</u> is LONG *Then* <u>angle</u> is $\theta = 1.848t + 4.037$

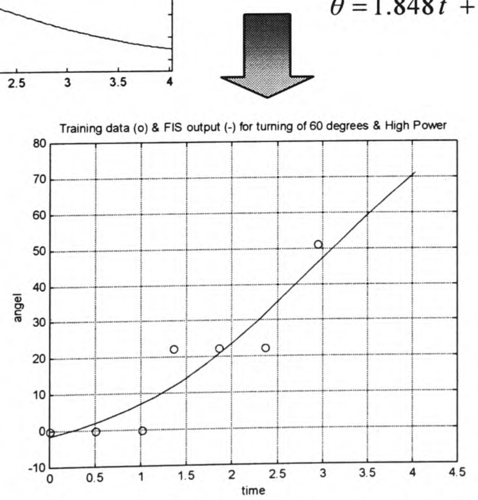
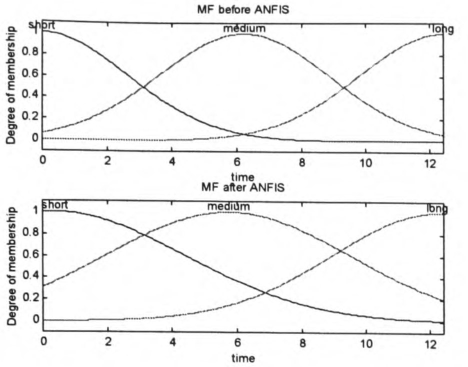


Figure A.22 Neuro-fuzzy local model HLR 2,3 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = 20.6t - 13.17$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = -16.58t + 61.89$

Rule 3:

If time is LONG *Then* angle is
$\theta = -27.45t + 234.1$

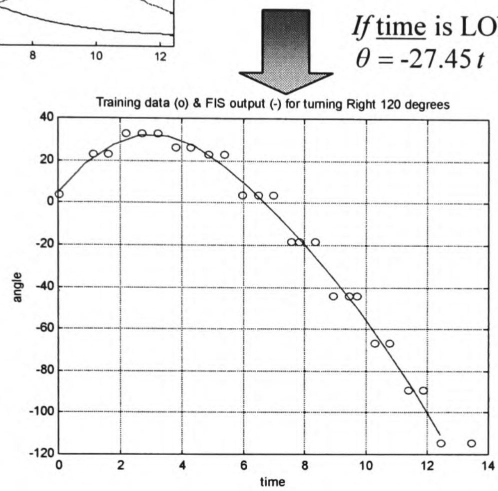
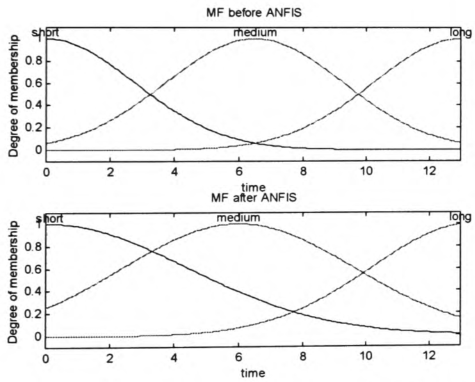


Figure A.23 Neuro-fuzzy local model L R 2,3 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = -32.97\,6t + -6.912$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = 114.75t - 14.63$

Rule 3:

If time is LONG *Then* angle is
$\theta = 30.94t - 283.6$

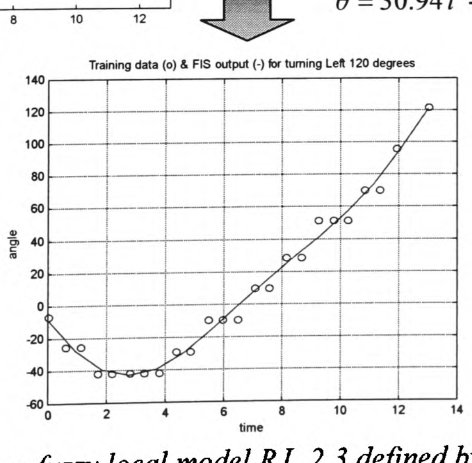


Figure A.24 Neuro-fuzzy local model R L 2,3 defined by ANFIS algorithm

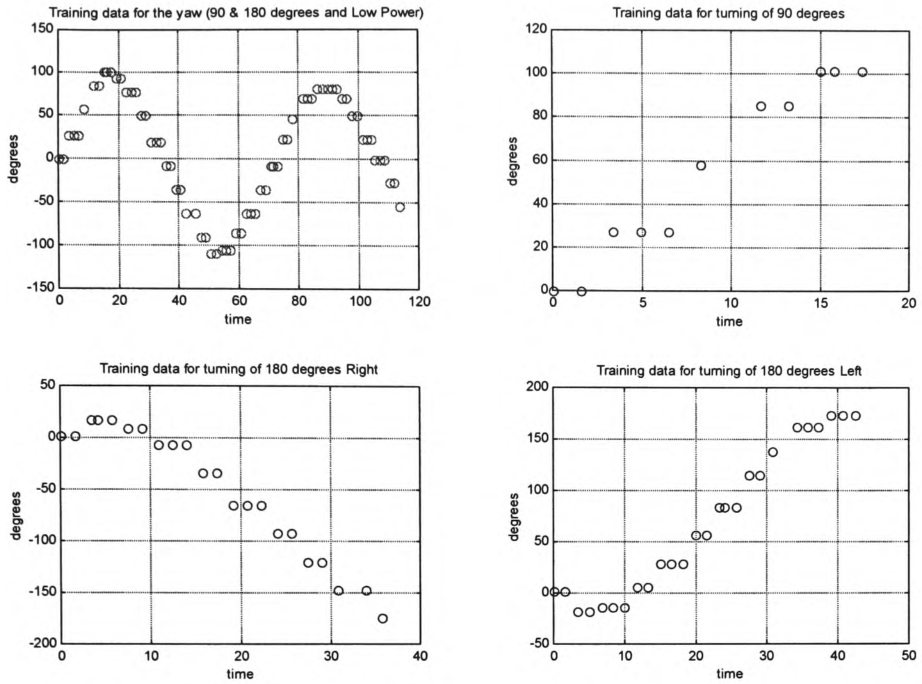
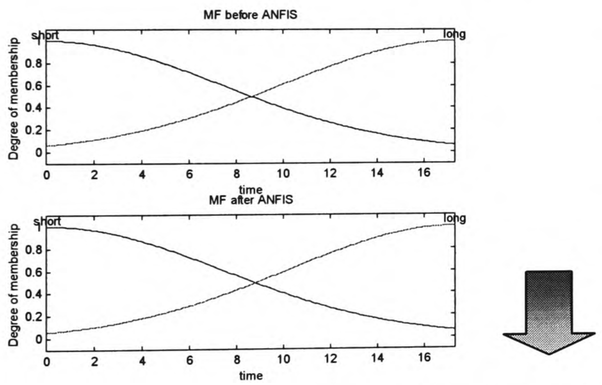


Figure A.25 Training data for turning 90°, 180° left and right (Low Power-High Angle)



Rule 1:

If time is SHORT *Then* angle is
$\theta = 1.793t - 6.404$

Rule 2:

If time is LONG *Then* angle is
$\theta = -0.2111t + 111.8$

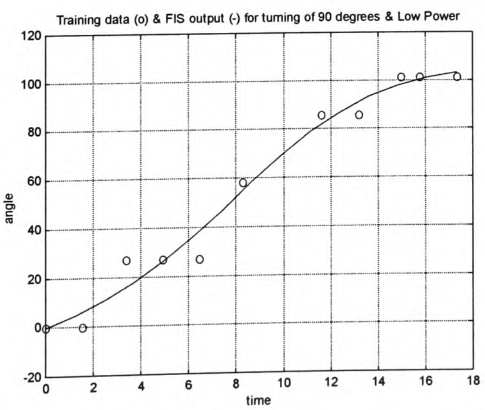
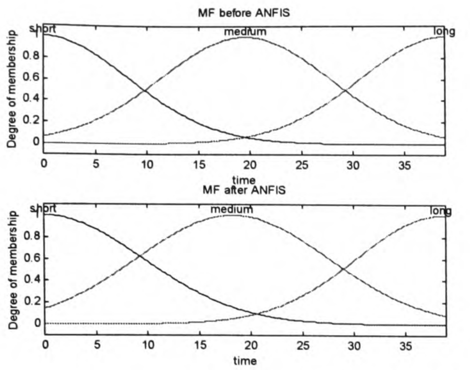


Figure A.26 Neuro-fuzzy local model HLR 3,1 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = 4.908t + 0.1903$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = -4.213t - 10.85$

Rule 3:

If time is LONG *Then* angle is
$\theta = -5.115t + 7.078$

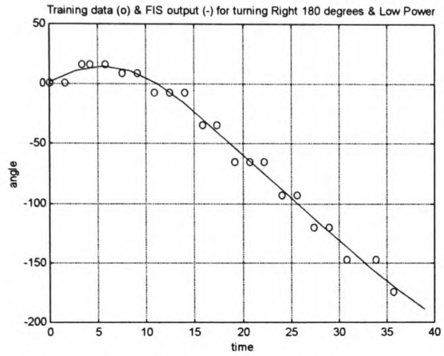
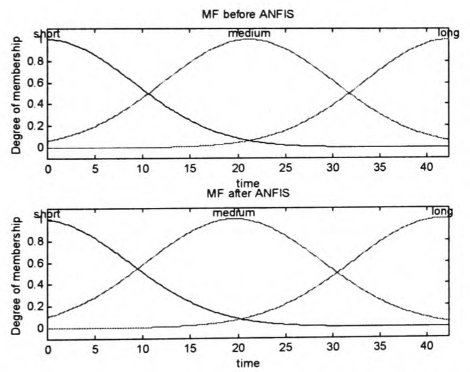


Figure A.27 Neuro-fuzzy local model L R 3,1 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = -6.217t + 1.965$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = 2.115t + 14.99$

Rule 3:

If time is LONG *Then* angle is
$\theta = -1.167t + 225.7$

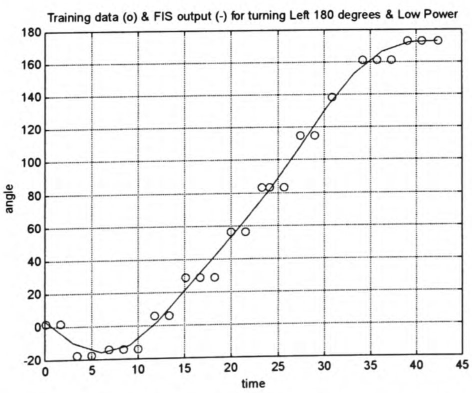


Figure A.28 Neuro-fuzzy local model R L 3,1 defined by ANFIS algorithm

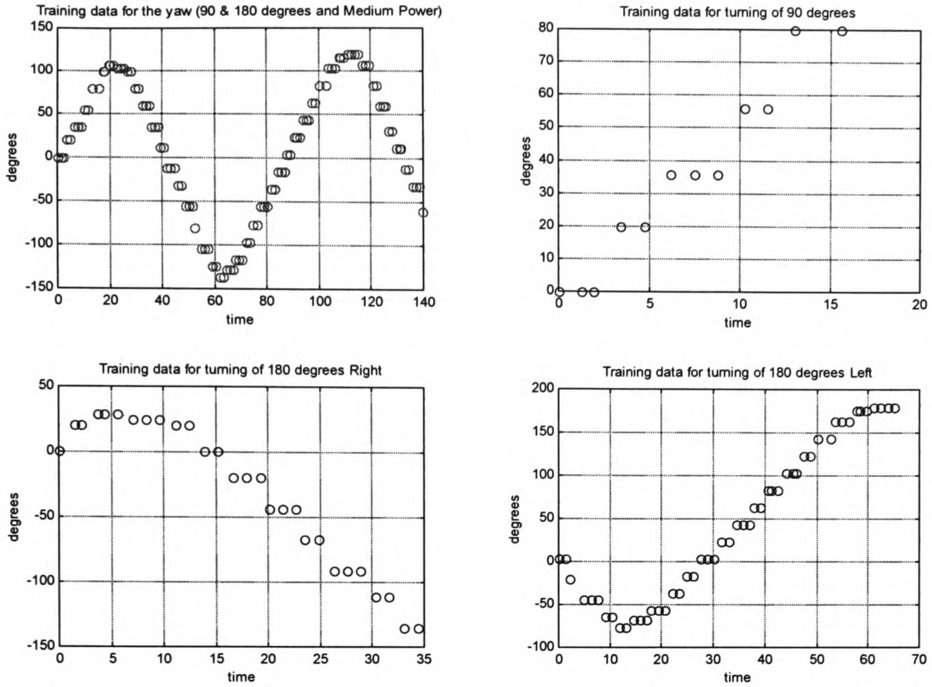
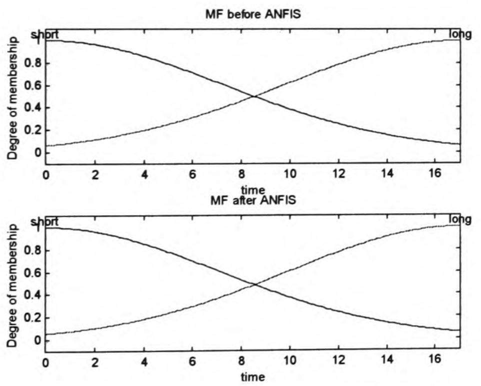


Figure A.29 Training data for turning 90°, 180° *left and right (Medium Power-High Angle)*



Rule 1:

If time is SHORT *Then* angle is
$\theta = 5.636t - 3.252$

Rule 2:

If time is LONG *Then* angle is
$\theta = 6.694t - 16.65$

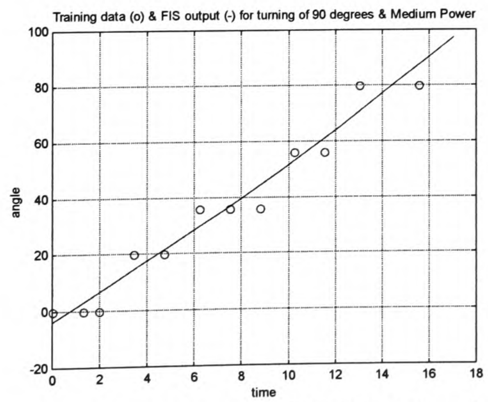
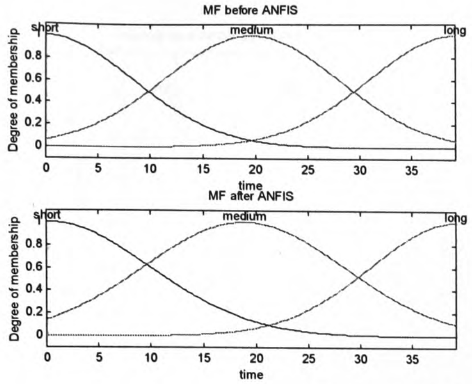


Figure A.30 Neuro-fuzzy local model HLR 3,2 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = 4.339t + 1.584$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = -6.047t + 67.48$

Rule 3:

If time is LONG *Then* angle is
$\theta = -8.644t + 159.5$

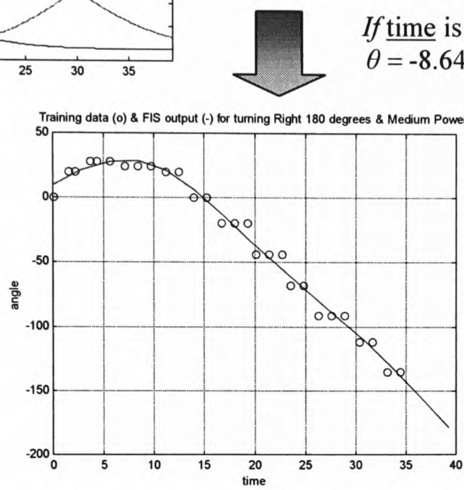
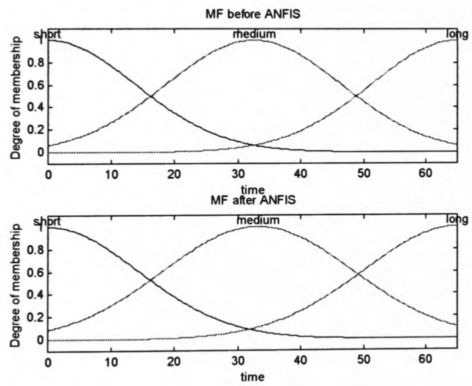


Figure A.31 Neuro-fuzzy local model L R 3,2 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = -10.83t + 4.353$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = 0.02673t + 31.4$

Rule 3:

If time is LONG *Then* angle is
$\theta = -2.483t + 357.6$

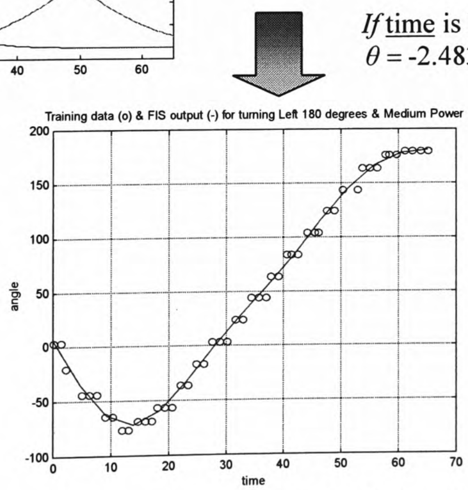


Figure A.32 Neuro-fuzzy local model R L 3,2 defined by ANFIS algorithm

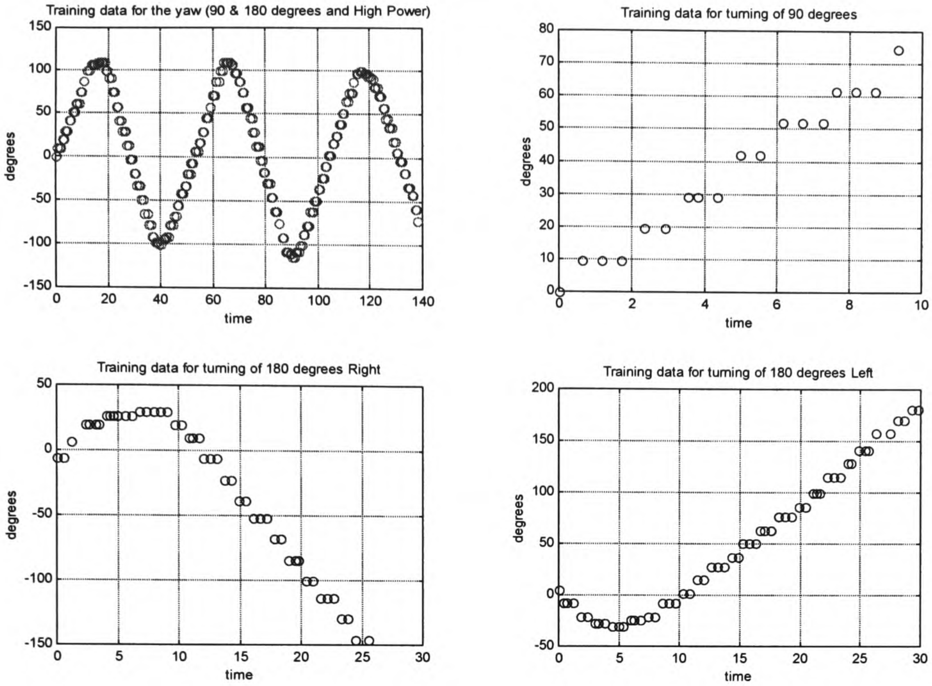
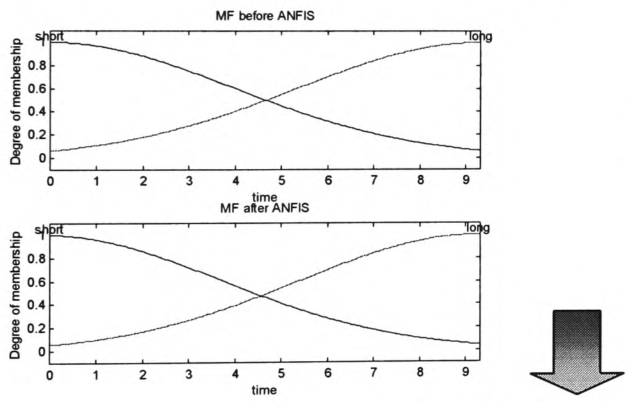


Figure A.33 Training data for turning 90°, 180° *left and right (High Power-High Angle)*



Rule 1:

If <u>time</u> is SHORT *Then* <u>angle</u> is
$\theta = 6.053t + 1.015$

Rule 2:

If <u>time</u> is LONG *Then* <u>angle</u> is
$\theta = 6.165t + 13.82$

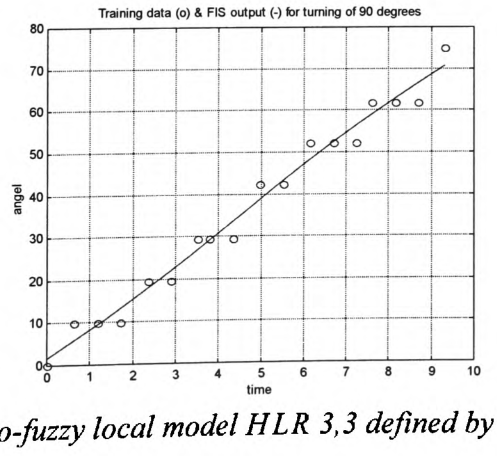
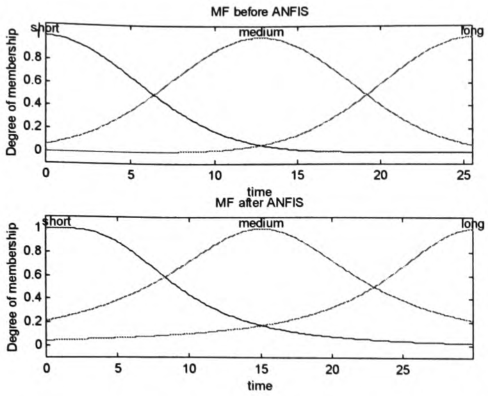


Figure A.34 Neuro-fuzzy local model HLR 3,3 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = 10.04t - 22.07$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = -9.316t + 93.63$

Rule 3:

If time is LONG *Then* angle is
$\theta = -4.464t - 61.65$

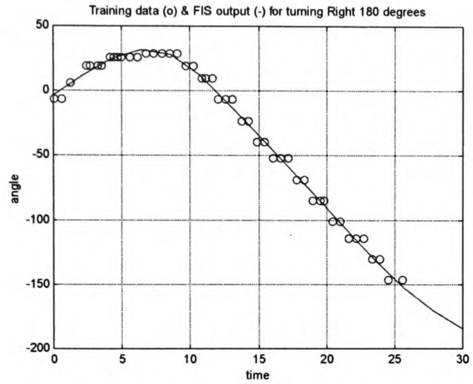
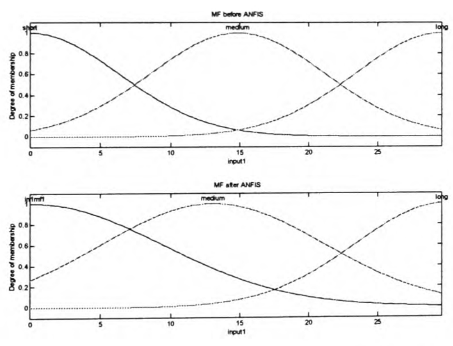


Figure A.35 Neuro-fuzzy local model L R 3,3 defined by ANFIS algorithm



Rule 1:

If time is SHORT *Then* angle is
$\theta = -31.46t - 79.05$

Rule 2:

If time is MEDIUM *Then* angle is
$\theta = -6.712t - 305$

Rule 3:

If time is LONG *Then* angle is
$\theta = 5.351t - 39.35$

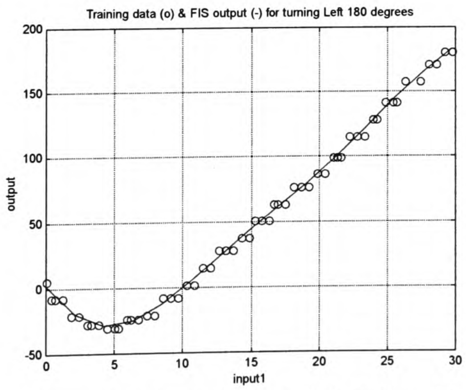


Figure A.36 Neuro-fuzzy local model R L 3,3 defined by ANFIS algorithm

Appendix B

SIMULINK Block Diagrams

Appendix B contains the main block diagrams that were designed in SIMULINK for the modelling simulation defined in Chapter 2.

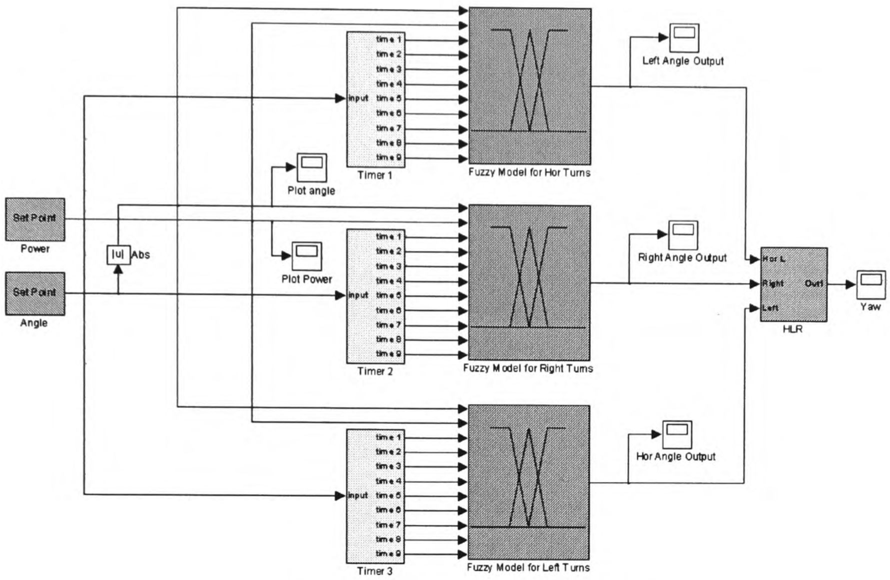


Figure B.1 Main block diagram

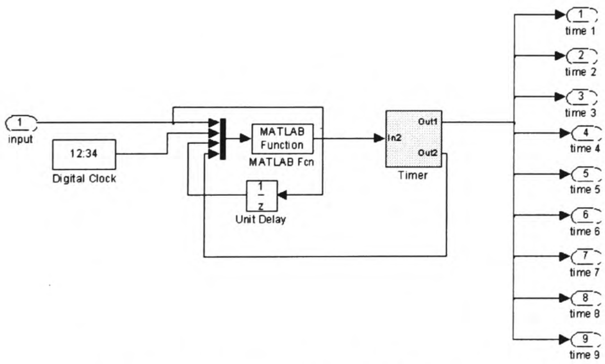


Figure B.2 Timer block diagram1 to 3

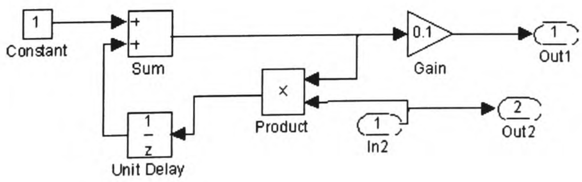


Figure B.3 Timer block diagram

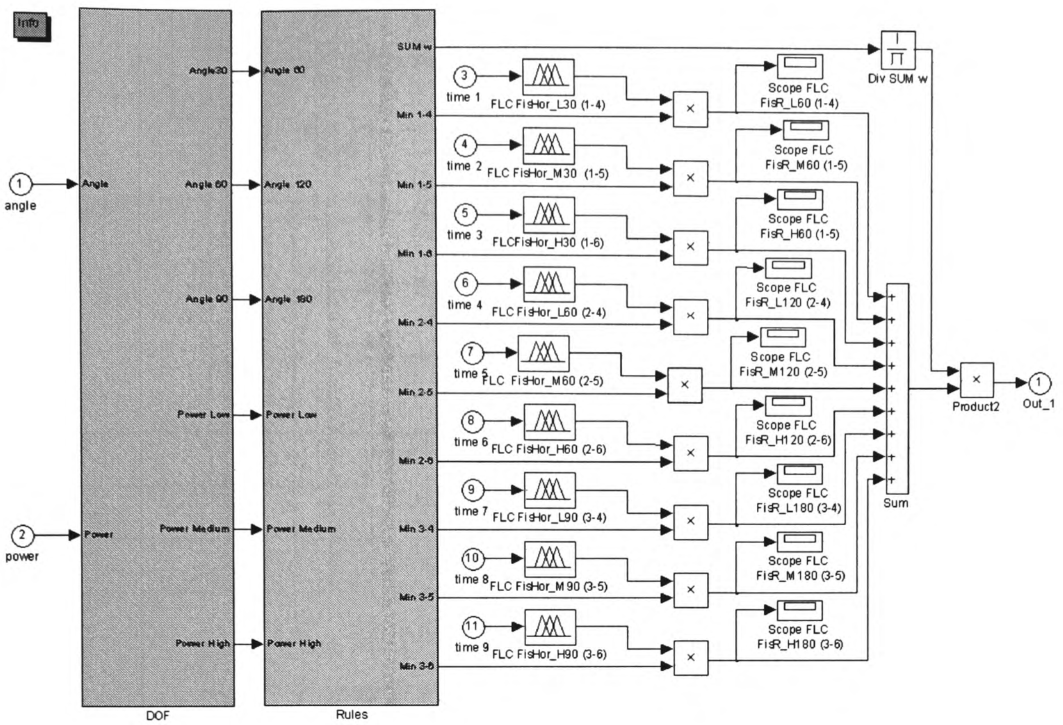


Figure B.4 Fuzzy Model for turns block diagram

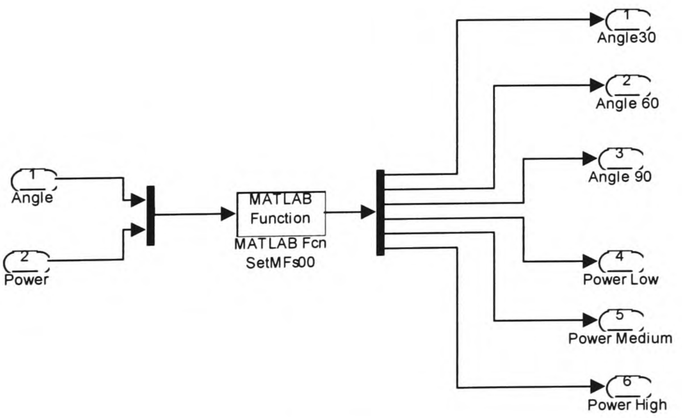


Figure B.5 Degrees of membership functions block diagram

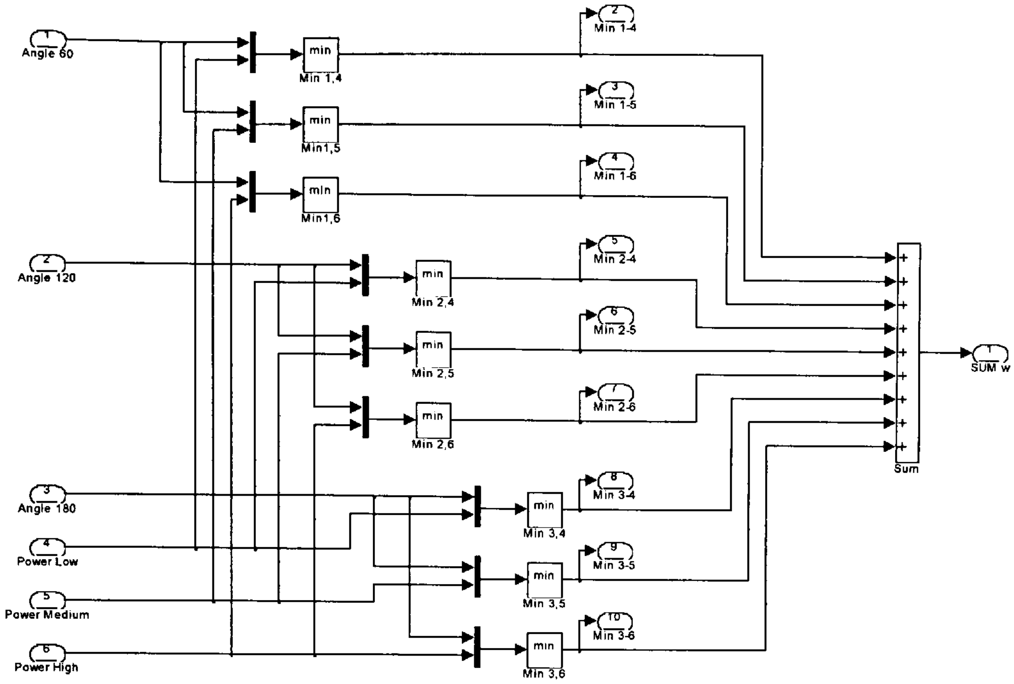


Figure B.6 Fuzzy Rules block diagram

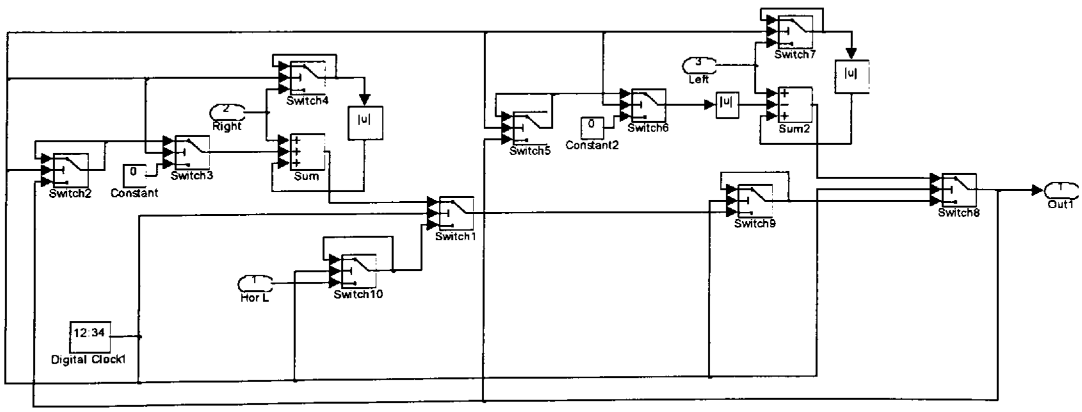


Figure B.7 Monitor block diagram

Appendix C

Merging method

C.1 Merging method

The merging process considering only the closeness of the peak values λ of triangular membership function works as follows:

1. Take the modal values for the membership functions $\mu_1(v), \mu_2(v), \ldots, \mu_{c_n}(v)$ (where c_n is the number of clusters in n-dimension) with more than $\frac{1}{2}$ overlap

$$m_k = \mu_{k(\alpha=1)}(\mathbf{z}) \qquad k = 1,2,\ldots,c_n \tag{C.1}$$

   with

$$m_1 \le m_2 \le \ldots \le m_{c_n} \tag{C.2}$$

2. Define the threshold λ of the distance dm_j, acceptable between the modal values
3. Calculate the difference between successive modal values as:

$$dm_j = m_{j+1} - m_j \text{ where } dm_j > \lambda \quad j = 1,2,\ldots,N-1 \tag{C.3}$$

4. Find all the difference smaller or equal than λ, i.e. $dm_j \le \lambda$.
5. If there is no difference smaller or equal than λ go to 7.
6. Merge all the modal values corresponding to consecutive difference smaller than λ using Equation C.4

$$m_{new} = \frac{\sum_{i=a}^{b} m_i}{D} \tag{C.4}$$

$$D = b - a + 1 \tag{C.5}$$

7. Update N and Go to 3

Appendix D

Update Law for input-output membership function-singletons

D.1 Update Law for input-ou tput membership function-singletons

Considering the error between the output of the fuzzy system $f(u_l \mid \varphi)$ and m trained data y

$$e_m = \frac{1}{2}\left[f(u^m \mid \varphi) - y^m\right]^2 \tag{D.1}$$

where

$$f(u_l \mid \varphi) = \frac{\sum_{i=1}^{R} g_i \mu_i}{\sum_{i=1}^{R} \mu_i} \tag{D.2}$$

and

$$g = \alpha \cdot s_j \tag{D.3}$$

the gradient method seek to minimise e_m by choosing the parameters φ, that can be for the fuzzy system the output singletons and/or the parameters of the input membership functions as follows:

D.2 Input Membership Func tion Update Law

In this case is considering to adjust γ, β, δ in Figure D.1 so minimise e_m.

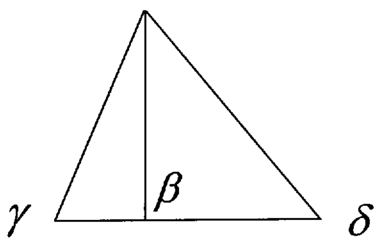


Figure D.1 Triangular membership fu nction

For γ the update law is:

$$\gamma(k+1) = \gamma(k) - \lambda_\gamma \cdot \frac{\partial e}{\partial \gamma} \tag{D.4}$$

$$\frac{\partial e}{\partial \gamma} = (f - y)\frac{\partial f}{\partial \gamma} \quad \text{so that} \quad \frac{\partial e}{\partial \gamma} = \varepsilon \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial \gamma} \tag{D.5}$$

$$\frac{\partial f}{\partial \mu} = \frac{g - f}{\sum_{i=1}^{R} \mu_i} \quad \text{and} \tag{D.6}$$

$$\frac{\partial \mu}{\partial \gamma} = \frac{\partial}{\partial \gamma}\left(\frac{x-\gamma}{\beta-\gamma}\right) = \frac{x-\beta}{(\beta-\gamma)^2} \tag{D.7}$$

$$\gamma(k+1) = \gamma(k) - \lambda_\gamma \cdot \varepsilon \cdot \frac{g-f}{\sum_{i=1}^{R} \mu_i}\left(\frac{x-\beta}{(\beta-\gamma)^2}\right) \tag{D.8}$$

For β the update law is:

$$\beta(k+1) = \beta(k) - \lambda_\beta \cdot \frac{\partial e}{\partial \beta} \tag{D.9}$$

$$\frac{\partial e}{\partial \beta} = (f-y)\frac{\partial f}{\partial \beta} \quad \text{so that} \quad \frac{\partial e}{\partial \beta} = \varepsilon \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial \beta} \tag{D.10}$$

$$\frac{\partial f}{\partial \mu} = \frac{g-f}{\sum_{i=1}^{R} \mu_i} \quad \text{and} \tag{D.11}$$

$$\frac{\partial \mu}{\partial \beta} = \frac{\partial}{\partial \beta}\left(\frac{x-\gamma}{\beta-\gamma}\right) = -\frac{x-\gamma}{(\beta-\gamma)^2} \qquad \gamma \le x \le \beta \tag{D.12}$$

$$\beta(k+1) = \beta(k) + \lambda_\beta \cdot \varepsilon \cdot \frac{g-f}{\sum_{i=1}^{R} \mu_i}\left(\frac{x-\gamma}{(\beta(k)-\gamma)^2}\right) \tag{D.13}$$

For δ the update law is:

$$\delta(k+1) = \delta(k) - \lambda_\delta \cdot \frac{\partial e}{\partial \delta} \tag{D.14}$$

$$\frac{\partial e}{\partial \delta} = (f-y)\frac{\partial f}{\partial \delta} \quad \text{so that} \quad \frac{\partial e}{\partial \delta} = \varepsilon \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial \delta} \tag{D.15}$$

$$\frac{\partial f}{\partial \mu} = \frac{g-f}{\sum_{i=1}^{R} \mu_i} \quad \text{and} \tag{D.16}$$

$$\frac{\partial \mu}{\partial \delta} = \frac{\partial}{\partial \delta}\left(\frac{\delta-x}{\delta-\beta}\right) = \frac{x-\beta}{(\delta-\beta)^2} \qquad \beta \le x \le \delta \tag{D.17}$$

$$\delta(k+1) = \delta(k) - \lambda_\delta \cdot \varepsilon \cdot \frac{g-f}{\sum_{i=1}^{R} \mu_i}\left(\frac{x-\beta}{(\delta-\beta)^2}\right) \tag{D.18}$$

D.3 Output Singletons Update Law

In this case is considering to adjust a so minimise e_m. The update law is

$$a(k+1) = a(k) - \lambda \cdot \frac{\partial e}{\partial a} \tag{D.19}$$

$$\frac{\partial e}{\partial a} = (f - y)\frac{\partial f}{\partial a} \qquad \text{so that} \qquad \frac{\partial e}{\partial a} = \varepsilon \frac{\partial f}{\partial g}\frac{\partial g}{\partial a} \tag{D.20}$$

$$\frac{\partial f}{\partial g} = \frac{\mu_i}{\sum_{i=1}^{R} \mu_i} \qquad \text{and} \qquad \frac{\partial g}{\partial a} = \frac{\partial (a \cdot s_j)}{\partial a} = s_j \tag{D.21}$$

$$a(k+1) = a(k) - \lambda \cdot \varepsilon \cdot \frac{\mu_i}{\sum_{i=1}^{R} \mu_i} s_j \tag{D.22}$$

Appendix E

Papers

Appendix D contains all the papers that have been published as a result of the research described in this thesis.

Akkizidis, I., and Roberts, G. 2000a. *Navigation Control of an Underwater Robot using Fuzzy-like PD Controller.* Control 2000 conference. pp. CD, Cambridge University, UK.

Akkizidis, I., Roberts, G. P. Ridao and J. Batlle, 2000b. *Steering and Depth Control of an Underwater Robot using Fuzzy-like PD Controller.* 5th IFAC Conference on Manoeuvring and Control of Marine Crafts pp. 245-251, Aalborg University, Denmark.

Akkizidis, I., and Roberts, 1998a. *Modelling & Motion Control of an Autonomous Underwater Robot using a Fuzzy & Fuzzy neural approach.* Mechatronics '98. pp. 641-646, Svovde, Sweden.

Akkizidis, I., and Roberts, 1998b. *Fuzzy Modelling & Fuzzy-Neural Motion Control of an Autonomous Underwater Robot.* AMC'98, 5th International Workshop on Advanced Motion Control. pp. 641-646, Coimbra, Portugal.

NAVIGATION CONTROL OF AN UNDERWATER ROBOT USING FUZZY-LIKE PD CONTROLLER

I. S. Akkizidis, G. N. Roberts

University of Wales College, Newport, UK
e-mail: ioannis.akkizidis@newport.ac.uk

ABSTRACT

Underwater vehicles are commonly classified as being highly non-linear uncertain systems and are consequently difficult to control effectively using model-based control methods. Application of a Fuzzy-like Proportional Derivative Controller (Fuzzy-like PD Control) for both steering and depth control of an underwater vehicle is described in this paper. The design of these types of controllers is of interest from the point of view of the course-changing, course-keeping and mission and motion control. The architecture of the controller is based on two independent Fuzzy-like PD controllers for each controlled variable, yaw and depth. Control performance and robustness are the most important issues of the above controllers and their optimum values need to be identified. These parameters may be the scaling factors that are applied in the input/output of the PD part of the controller as well as the fuzzy sets of the fuzzy part of the controller. The parameters of the fuzzy sets are optimised based mostly on the expert's knowledge in terms of systems' dynamics and uncertainty analysis. The values of input/output scaling factors are optimised since they play an important role in the formation of the dynamics of the close-loop structure leading to the desired response of the control system. This optimisation is based on experiments in a real environment and were planned using the Taguchi method for the Design of Experiments. Results are presented and analysed extensively to investigate the capabilities of the controller.

Keywords: Fuzzy like-PD Controller, Yaw and Depth Control, Taguchi Method.

1. INTRODUCTION

Manoeuvring and depth control of an Underwater Vehicle (UV) is discussed in this paper. The difficulty that stems from this type of controller is that they have to be robust. UVs are classified as systems possessing highly non-linear dynamics as well as the environment in which they operate has a lot of disturbances, these give rise to special problems that may be solved using intelligent control techniques.

This paper presents the development of an optimal fuzzy control strategy to control steering and depth of a low-cost Remote Operate Vehicle (ROV) named GARBI developed at the Polytechnic of Barcelona and the University of Girona in Spain. The vehicle, which is illustrated in Figure 1, is used for underwater mission operations such as observations and inspections. An umbilical cable carrying power and providing a communication link, links GARBI to a surface ship or other operating platform. The objective of the Fuzzy Logic Controller (FLC) is to control the yaw ψ and the depth z of the vehicle in terms of keeping the path of the navigation to the desired one, and/or changing the path according to a set point. This makes the navigation smoother and safer, the propulsion more economical and more accurate path-keeping.

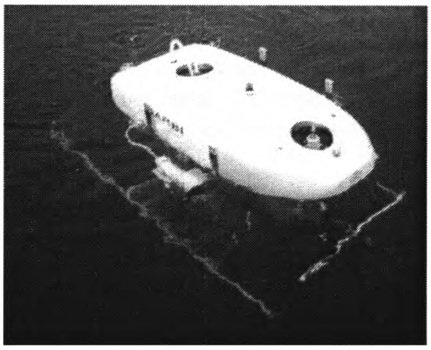


Figure 1. Photo of GARBI underwater robot

Structure and parameter designs are important tasks during the building of FCs. Structure design means to determine the architecture of a controller, the input/output variables of a controller, the format of the fuzzy control rules, and the number of rules. Parameter design means determining the optimal parameters for a fuzzy controller.

The objective of this paper is to describe how to design and apply non-linear Fuzzy-like Proportional Derivative Controller (F-like PD Controller) in an underwater vehicle. This controller's strategy is based on the combination of Fuzzy Logic and conventional proportional-derivative control techniques. The main advantage of the FLC is that it can be applied to systems that are non-linear where their mathematical model are difficult to obtain. Another advantage is that the controller can be designed to apply heuristic rules that reflect experiences of the human experts. F-like PD controller has a fixed set of control rules, usually derived from experts' knowledge. The Membership Functions (MFs) of the associated input and output linguistic variables are generally predefined according to non-linearities of the system. Conventional Proportional Derivative (PD) controllers provide high sensitivity and tend to increase the stability of the overall feedback control systems. Additionally, PD controllers can reduce overshoot and permit the use of larger gain by adding damping to the system. The derivative action is employed because it performs well in reducing disturbances and keeping the set point to the desired one.

For the successful design of FLC's, proper selection of the optimal input and output Scaling Factors (SFs) is required which scale up or down the entire universe of discourse. Due to their global effect on the control performance and robustness, input and output SF's play a critical role in the F-like PD controller and they have the highest priority in terms of tuning and optimisation, Mudi and Pal (2). Analysis of how to investigate their optimal values is presented in this paper. Experimental results of the F-like PD controller are presented and discussed extensively in the following sections.

2. THE HYDRODYNAMIC FORCES AND MOMENTS OF GARBI

The motion study of marine vehicle involves six *Degrees Of Freedom* (DOF), since six independent co-ordinates are necessary to determine the position and orientation of a rigid body. The first three co-ordinates Surge, Sway and Heave and their time derivatives correspond to the position and translational motion along the *x-, y-, and z-axes.* The last three co-ordinates Roll, Pitch and Yaw (or heading angle) and their time derivatives are used to describe orientation and rotational motion. In GARBI the motions in the x and z direction (Surge and Heave) are controlled by the horizontal propellers (T_1, T_2) and vertical propellers (T_3, T_4) respectively (Figure 2). However, no correction in y direction (Sway) is applied.

The structure of GARBI is designed in such a way that Pitch and Roll cross-coupling is virtually non-existent. However, coupling appears between yaw and surge only when the vehicle has initial speed. Nevertheless, this coupling is expected and acceptable due to the navigation properties. Because of feedback control, the longitudinal control that is coupled with pitch and therefore depth are minor and to some extent compensated. No other major couplings are present.

3. CONTROL TASKS OF GARBI UNDERWATER VEHICLE

As in any underwater vehicle the dynamics of GARBI are coupled and highly non-linear. When designing GARBI's controller it is necessary to compensate its non-linear dynamics and kinematics, non-linearities due to thrusters and pressure hysteresis, barometer dead-zones, and the noise in yaw and depth measurements. Therefore, robust controllers that reliably perform complex task in the face of the above uncertainties should be used. Non-linear F-like PD Controller is designed to make the vehicle follow the commands from the pilot in terms of course-changing and the course-keeping of both yaw angle and depth of the robot.

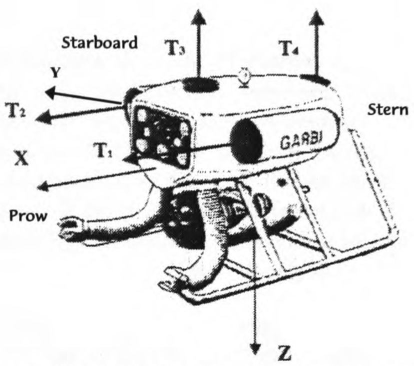


Figure 2. GARBI Body-fixed reference frames

Controllers for course-keeping and/or course-changing are normally based on feedback from a gyrocompass measuring the heading for the yaw and air press-sensors measuring the difference of the pressure inside and outside the robot for depth.

The control objective for a course-keeping controller can be expressed as: $\psi, z = constant$. For the course-changing the objective is to follow the changes of the pilot commands with the best control performance in terms of small overshoot, settling time and steady state error.

Figure 3 shows a simplified scheme of course-keeping/course-changing. The structure uses two independent FLCs for each controlled variable (Yaw and Depth), greatly simplifying the design at the cost of some decrease in performance. The corresponding inputs of these controllers are the error $e_\psi(nT) = \psi_{sp} - \psi(nT)$ between the real and the desired yaw angle and the error $e_z(nT) = z_{sp} - z(nT)$ between the real and the desired heave position as well as the change of the above errors (see Figure 4 & Figure 5) $\Delta e_\psi(nT) = e_\psi(nT) - e_\psi(nT-1)$, $\Delta e_z(nT) = e_z(nT) - e_z(nT-T)$ where $e(nT)$, $\Delta e(nT)$ and $\psi, z(nT)$ designate, crisp error, rate and process output at sampling time nT respectively. Limiters are used to avoid saturation of inputs in the universe of discourse. As a convention, signals are written in lower case before gains/SFs and upper case after gains/SFs, for instance $E = S_e * e$. The corresponding outputs of these controllers are; for the first controller the moment N around the z-axis, and for the second controller the force Z of the two propellers in the z-direction. The rotation N is related to the difference of power between propellers T_1 & T_2 in the x-direction. The equation that describes this relation is given by $N = f(dX_{xy})$. The force Z in the z-direction relates to the power of the propellers T_3 & T_4, which is always equal and of the same polarity.

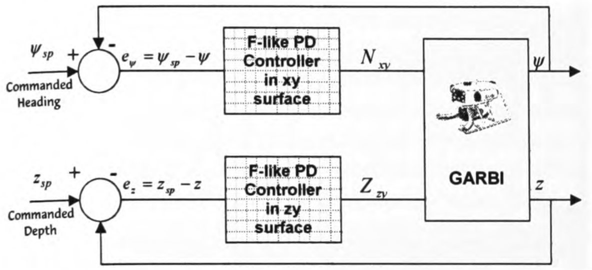


Figure 3. Control loop for GARBI

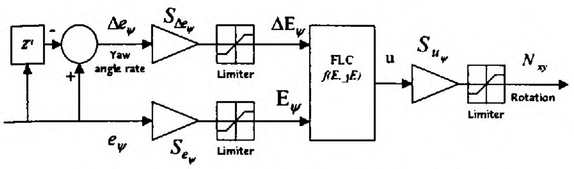


Figure 4. Yaw Fuzzy-like PD controller

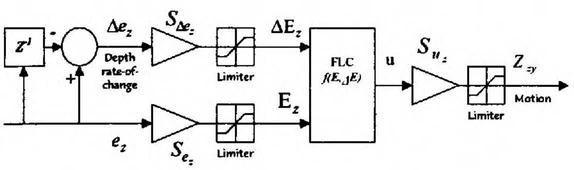


Figure 5. Depth Fuzzy-like PD controller

4. DESIGN THE FUZZY-LIKE PD CONTROLLER FOR GARBI

In studying the dynamic properties of the fuzzy controller, the model of the process is needed so that the impact of the successive control actions may be monitored. Since a model of GARBI is not available, the dynamic properties of the close loop structure have to be derived intuitively and experimentally. This is simply a cornerstone feature of the idea of fuzzy controllers. However, the tuning of F-like PD Control systems is a fundamental problem, specially, for optimum performance. There are two different levels of tuning during the design of F-like PD controllers. The first level includes the structure (as described in Section 3), the rule base, the antecedent and consequent membership functions together with their distribution, the inference mechanism and the defuzzification strategy. The second level is the tuning of gain parameters. This includes the scaling factors and other gains used in building the structure. As mentioned before, the dynamic properties of the controller can be adjusted by a series of carefully designed experiments. As the experiments of GARBI in a real environment are expensive and time consuming the minimum number of experiments must be executed. However, the risk of loosing vital information that can result from large amount of data can be overcome if the experiments are designed using orthogonal array of the Taguchi Method, Fowlkes (1), to find the set of optimum input/output. This is explained in section 5.

The settings in the design procedure for the Fuzzy-like PD controller for both yaw and depth control are as follows:

- Sampling time of 1 sec. If shorter, the computation of Δe may become too sensitive to noise. This normally shows up as a restless control signal.
- Each of the FLC blocks contains 49 rules. The rule base is presented in the table format shown in Table 1. The cell defined by the intersection of the first row and the first column represents a rule such as, *if* $e(nT)$ is NB *and* $\Delta e(nT)$ is NB *then* $u(nT)$ is NB

e\Δe	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

Table 1. The Rule Base of a F-like PD in tabular form.

- The linguistic variables such as PB (positive big), PS (positive small), ZO (zero) in the premise parts are fuzzy variables, while those in the consequence part are singleton value between 0 and 1.

It is well known that generally fuzzy variables in the consequent parts are not needed, Nguyen and Prasad, (3). Additionally singletons make defuzzification simpler and faster. The universe of discourse for each variable is uniformly partitioned and the MFs are placed with 50% overlap. Using the min operation for the aggregation *AND* (outer product) of the fuzzy rules, the output fuzzy set is given by $\mu_u = Min(\mu_e, \mu_{\Delta e})$. Thus, the F-like PD controller is a controller where the output is a non-linear function of the error e and its derivative de/dt ($u = F(e, de/dt)$), where F is a non-linear function of two variables.

Both inputs of yaw i.e. E_ψ, ΔE_ψ and depth i.e. E_z, ΔE_z controllers are operated in the whole range of their universe. Therefore, for the universe of the yaw and depth controllers the maximal limits are $e_{yaw_{\max}} * S_{yaw_e} = YawUe_{\max}$, $e_{depth_{\max}} * S_{depth_e} = DepthU_{\max}$ and the minimal limits are as $e_{yaw_{\min}} * S_{yaw_e} = YawU_{\min}$, $e_{depth_{\min}} * S_{depth_e} = DepthU_{\min}$.

For the yaw controller two different universes are applied. The first is within the range of $-90°$ to $+90°$

degrees and the second is within the range $-180°$ to $+180°$ degrees in the cases of "Small" or "Big" changes occurring in GARBI's navigation path respectively. Moreover for the depth controller the universe is in a range of 10 meters. It should be noted that the depth controller starts using the range of its universe when GARBI is within 10 meters of the set point. In any other case the control output has the maximum value.

One way of improving the dynamic properties is to adjust the SFs used in the construction of the fuzzy controller. SFs play a important role in the formation of the dynamics of the close-loop structure leading to the desired response of the controlled system, Zheng (4).

For the Fuzzy-like PD controller the corresponding SFs are for the inputs $S_e, S_{\Delta e}$ and for the output S_u respectively. The factors that influence the dynamics of the system are:

- if both Se and $S_{\Delta e}$ increase, the control becomes more sensitive around the set point until oscillations are observed.
- where Se and $S_{\Delta e}$ decrease, a tolerance band exist around the set point and a large steady-state error is quite common; so if the scaling factors are too small, the system gives a poor response.
- S_u affect the proportional gain, so it is desirable to have them as large as possible without creating too much overshoot. If too small, the system will be too slow, and if too large the system might become unstable.

In relation to the above factors, design of experiments has been performed to tune the input/output scaling factors.

5. DESIGN OF EXPERIMENTS

A large amount of engineering effort is consumed in conducting experiments (either in hardware, software or simulation) to produce the information required to make decisions about how different factors affect performance under different usage conditions. Experimentation seeks to determine the best material, the best pressure, the best force, the best speed etc., which will operate together within a system to produce a desired response such as settling time, overshoot, steady state error, etc. Designs of experiments are statistical methods that seeks to minimise the number of experiment and still find the optimal combination of the factors under study which are robust in a variety of environmental conditions. A widely used method is the Taguchi Method of Robust Design, which can be applied to a wide variety of problems.

Various types of arrays are used for planning experiments to study one factor at a time, where each individual factor is varied while all the other factors are fixed, full factorial experiments that investigate all possible combinations of all factors and their levels, where the possible combinations can rise to the order of y^x, x being the number of factors and y the different levels. This approach investigates all the possible combinations, maximising the possibility of finding the optimum result, but large numbers of experiments are required. Alternatively, orthogonal array extensively used in the Taguchi Method studies several factors at different levels simultaneously, but only requires a fraction of the full factorial combinations. The orthogonal array imposes an order on the way the experiment is carried out. The combinations are chosen to provide enough information to determine the factor effects using the analysis of mean values. In order to use a standard orthogonal array provided by the Taguchi Method, the degrees of freedom (number of independent measurements available to estimate sources of information) of the factors and levels must be matched with the degrees of freedom for that orthogonal array.

The orthogonal array of the Taguchi Method was used to plan the experimentation for the underwater robot. Three factors ($Se, S_{\Delta e}, S_u$) at three levels each *(Level 1, Level 2, Level 3)* were investigated. With the full factorial, this would result in 3^3 (27) different experiments. For three factor three level experiments, six degrees of freedom exist, so an orthogonal array with nine experimental runs as shown in Table 2 is sufficient for this study.

No Exp	S_e	$S_{\Delta e}$	S_u
1	Level 1	Level 1	Level 1
2	Level 1	Level 2	Level 2
3	Level 1	Level 3	Level 3
4	Level 2	Level 1	Level 2
5	Level 2	Level 2	Level 3
6	Level 2	Level 3	Level 1
7	Level 3	Level 1	Level 3
8	Level 3	Level 2	Level 1
9	Level 3	Level 3	Level 2

Table 2. Orthogonal array with nine experiments

The value of 10 for the 1^{st} level, and 1 for the 3^{rd} level is chosen for both Yaw and Depth Fuzzy-like PD Controllers. For the 1^{st} level the maximum value that can be applied without saturation in either universe of discourse or voltages that are applied to the propellers is chosen. For the 3^{rd} level the value where no proportional gain is added into inputs/output of the FLC is selected. Finally, the value 2 & 0.5 are chosen for the 2^{nd} level of the Yaw & Depth Fuzzy-like PD Controllers respectively. It should be noted that proportional gain values below these resulted, in unacceptable performance.

6. EXPERIMENTS

Experimental trials in a real environment have been held to test both depth and yaw F-like PD controller. The power of the propellers is controlled with power cards for voltage tuning. The voltage used is in the range of 0 to 10 Volts (V). For heading control the opposite voltage between the horizontal propellers (T_1, T_2) is used i.e. $+V_1$, $-V_2$. So, if the heading angle is turned to α^o clockwise for instance, the voltage in the right propeller T_1 is reduced and the voltage in the left propeller T_2 in increased by this amount. If the increase in voltage in propeller T_2 is now more than the predefined voltage range of 1-10 this over-limited voltage is added to the other propellers. The performance of the controller in course-changing and course-keeping navigation is shown in Figure 6 & Figure 7. The navigation plan for the robot is:

- initial power of the horizontal propellers is 3V for a period of 60 seconds.
- change the heading course from 270° degrees to 135° and then to 225° and then keep it in this direction,
- and at the same time change the depth course from 0 meters to 10m and then to 5m and then keep it at this depth.

After the trials as shown in Table 2, and analysis of the results the best combination in terms of performance and robustness is for the Yaw $S_{e_y} = 1$, $S_{\Delta e_y} = 0.5$ & $S_{u_y} = 7$ and for the Depth $S_{e_z} = 0.5$, $S_{\Delta e_z} = 0.5$ and $S_{u_z} = 10$. Note that for the depth, this combination is not outlined in the experiments in Table 2, however, using the Taguchi method it was identified. This is one of the most important features of the Taguchi method. That is, it can identify experimental combinations that were not originally specified in the orthogonal array.

Figure 6(a) illustrates the Yaw, 6(b) the controller output of the power horizontal propellers (T_1, T_2) (notice that as mentioned before they have opposite sign), 6(c&d) the error and the change of error between the desired and actual yaw. Additionally, Figure 7(a) shows the Depth, 7(b) the controller output of the power of the vertical propellers (T_3, T_4), 7(c&d) the error and the change of error between the desired and the actual depth. Finally, Figure 8 and Figure 9 show the corresponding pitch and the roll motions.

The performance of the yaw controller in changing the course (heading/depth) is satisfactory, as both overshoot and rise time are small. Due to buoyancy effects the depth control dynamics vary i.e. the vehicle rises faster than it descends. From Figure 7 it can be seen that the controller has accommodated this variation producing very acceptable rise times with very small overshoot when ascending. In course-keeping control both controllers perform quite well (Figure 6 & Figure 7). Finally in Figure 8 & Figure 9, the pitch and roll appears mostly due to the heading changes. These are very small and can be considered as fractional and thus negligible.

7. CONCLUSIONS

F-like PD control technique possesses features that are attractive in navigation control problems posed by underwater vehicles. The fuzzy logic part helps to model different types of uncertainty and imprecision of the system and together with the scaling factors it allows the building of a robust controller. Scaling factors have been chosen to be tuned to the optimal value, due to their global effect in the dynamics of the close-loop control system. Taguchi experimental method can help to minimise the number of experiments without risk of loosing vital information. This is very important when experiments are held in a real environment and therefore are time and money consuming.

F-like PD controller is much more complicated than conventional controller is. It is in fact a non-linear adaptive controller. This is why F-like PD controller usually presents strong robustness characteristics.

In this work it has been shown how a fuzzy controller combined with conventional PD control techniques can help to design a robust controller, dealing with the uncertainties and non-linearities of an underwater vehicle.

8. ACKNOWLEDGEMENT

The authors wish to acknowledge the support for this work provided by the British Council under the British/Spanish Acciones Integradas programme. The authors would like also to thank Pere Ridao, Marc Carreras and Josep Cortada for their help during the experiments and Vivianne Bouchereau and Hefin Rowlands for their useful comments during the preparation of this manuscript.

9. REFERENCES

1. Fowlkes, Y. W. 1995, "Engineering Methods for Robust Product Design Using Taguchi Methods"
2. Mudi, K P. and Pal, R. N. 1999, "A Robust Self-Tuning Schem for PI- and PD - Type Fuzzy Controllers". IEEE Trans. on Fuzzy Sys. 7(1):2-16.
3. Nguyen, H. T. and Prasad, R N. 1999, "Fuzzy Modelling and Control: selected work of M. Sugeno"
4. Zheng, L. 1992, "A Practical Guide to Tune of Proportional and Integral (PI) Like Fuzzy Controllers". Proc. FUZZ IEEE. 633-641.

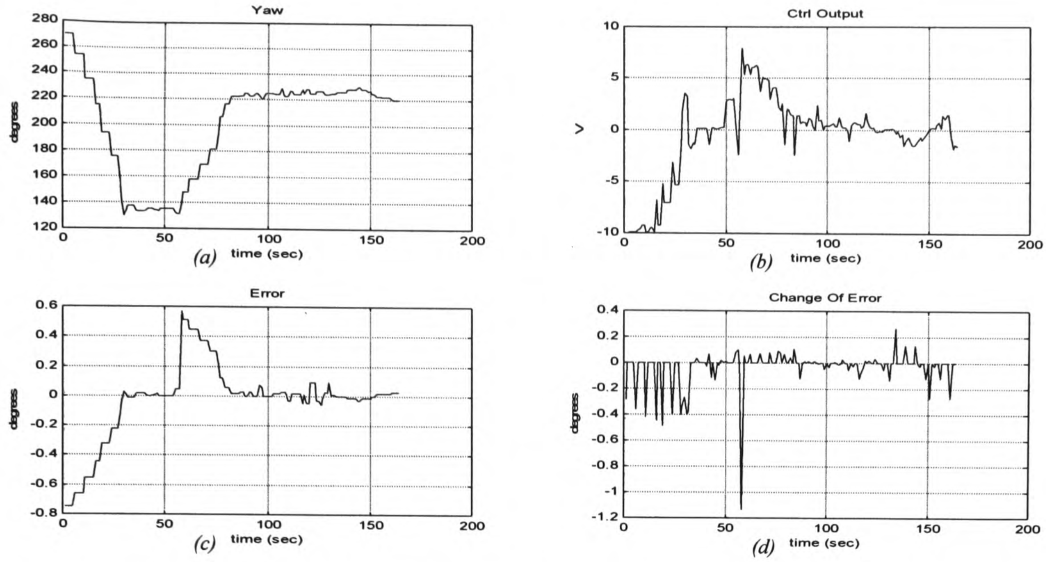


Figure 6. The plan of the "Yaw" experiment (briefly) was as follows: Change the course from 270° degrees to 135° and then to 225° and then keep heading in this direction.

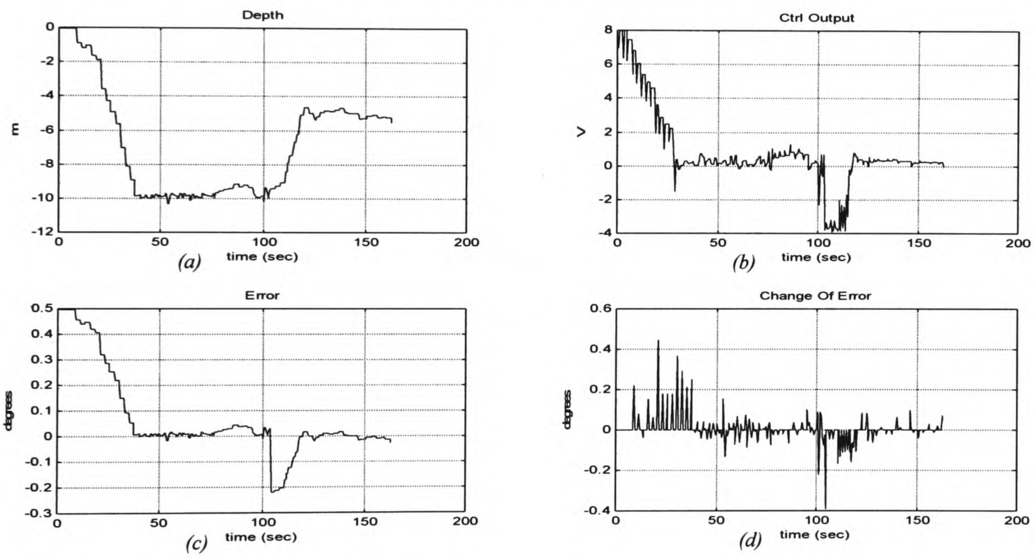


Figure 7. The plan of the "Depth" experiment (briefly) was as follows: Change the course from 0 meters to 10m and then to 5m and then keep depth at this level

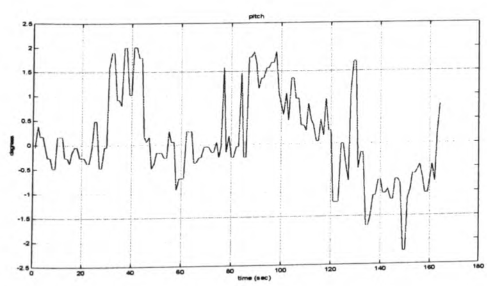


Figure 8. The pitch due to the experiment is not more that 2.5° degrees.

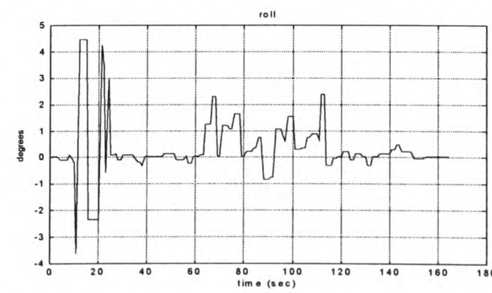


Figure 9. The roll due to the experiments is not more than 5° degrees

STEERING AND DEPTH CONTROL OF AN UNDERWATER ROBOT USING FUZZY-LIKE PD CONTROLLER

I. S. Akkizidis[1], G. N. Roberts[1], P. Ridao[2] and J. Batlle[2]

[1]Mechatronics Research Centre, University of Wales College, Newport,
Allt-yr-yn Campus, PO Box 180, Newport, South Wales, NP9 5XR, UK
Email: ioannis.akkizidis@newport.ac.uk

[2]Computer Vision and Robotics Group
Institute of Informatics and Applications
Edifici Politècnica II, Campus Montilivi, 7071-Girona, SPAIN

Abstract: The design of a steering and depth control of an underwater vehicle is of interest from the point of view of motion stabilisation as well as manoeuvring performance. The paper describes how a Fuzzy-like Proportional Derivative Controller (Fuzzy-like PD Control) is used to control the course-changing and course-keeping tracking mission and motion of an underwater vehicle. The idea for this type of control architecture is to identify the optimum values, in terms of control performance and robustness, of the parameters of the F-like PD Controller. These parameters may be the scaling factors that are applied in the input/output of the PD part of the controller as well as the fuzzy sets of the fuzzy part of the controller. The values of the scaling factors for the error, the change-of-error and the control output are optimised based on experiments. In this work a method has been developed to control the yaw and depth of an underwater robot in terms of course-changing and course-keeping. The controller is synthesised in two parts. Firstly it is applied in horizontal *xy* and secondly in vertical *xz* surface. Experiments in a real environment were planned using the Taguchi Design of Experiments and performed to investigate the capabilities of the controller. The results are presented and analysed extensively.

Keywords: *Fuzzy like-PD Controller, Yaw and Depth Control, Taguchi Design of Experiments.*

1. INTRODUCTION

Manoeuvring and depth control of an Underwater Vehicle (UV) is discussed in this paper. The difficulty that stems from this type of controller is that they have to be robust. UVs are classified as systems possessing highly non-linear dynamics. In addition the environment in which they operate has a lot of disturbances. These give rise to special problems that may be solved using intelligent control techniques.

This paper presents the development of fuzzy controller to control steering and depth of a low-cost Remote Operated Vehicle (ROV) named GARBI developed at the Polytechnic of Barcelona and the University of Gerona in Spain. The vehicle is used for underwater mission operations such as observations and inspections. An umbilical cable carrying power and providing a communication link, links GARBI to a surface ship or other operating platform (Amat *et al*, 1996). The objective of the Fuzzy Logic Controller (FLC) is to control the yaw and the depth of the vehicle in terms of keeping the path of the navigation to a desired one, and/or changing the path according to a set point. This makes the navigation smoother and safer, the propulsion more economical and more accurate path-keeping.

Structure and parameter designs are important tasks during the building of FCs. Structure design means to determine the architecture of a controller, the input/output variables of a controller, the format of the fuzzy control rules, and the number of rules. Parameter design means determining the optimal parameters for a fuzzy controller. The objective of this paper is to describe how to design and apply Fuzzy-like Proportional Derivative Controller (F-like PD Controller) in an underwater vehicle. This controller's strategy is based on the combination of Fuzzy Logic and conventional proportional-derivative control techniques. The main advantage of the Fuzzy Logic Controller (FLC) is that it can be applied to systems that are non-linear where their mathematical model are difficult to obtain. Another advantage is that the controller can be designed to apply heuristic rules that reflect experiences of the human experts. F-like PD controller has a fixed set of control rules, usually derived from experts' knowledge. The Membership Functions (MF's) of the associated input and output linguistic variables are generally predefined according to non-linearities of the system. Conventional Proportional Derivative (PD) controllers provide high sensitivity and tend to increase the stability of the overall feedback control system. Additionally, PD controllers can reduce overshoot and permit the use of larger gain by adding damping to the system. The derivative action is employed because it performs well in reducing disturbances and keeping the set point to the desired one.

For the successful design of FLC's, proper selection of the optimal input and output Scaling Factors (SF's) is required which scales up or down the entire universe of discourse. Due to their global effect on the control performance and robustness, input and output SF's play critical role in the F-like PD controller and they have the highest priority in terms of tuning and optimisation (Mudi & Pal, 1999). Analysis of how to investigate their optimal values is presented in this paper. Experimental results of the F-like PD controller are presented and discussed extensively in the following sections.

2. THE HYDRODYNAMIC FORCES AND MOMENTS OF GARBI

The motion study of marine vehicle involves six *Degrees Of Freedom*, since six independent co-ordinates are necessary to determine the position and orientation of a rigid body. The first three co-ordinates Surge, Sway and Heave and their time derivatives correspond to the position and translational motion along the *x-, y-, and z-axes*. The last three co-ordinates Roll, Pitch and Yaw and their time derivatives are used to describe orientation and rotational motion. In GARBI the motions in the x and z direction (Surge and Heave) are controlled from the horizontal propellers (T_1, T_2) and vertical propellers (T_3, T_4) respectively (Fig. 1). However, no correction in y direction (Sway) is applied.

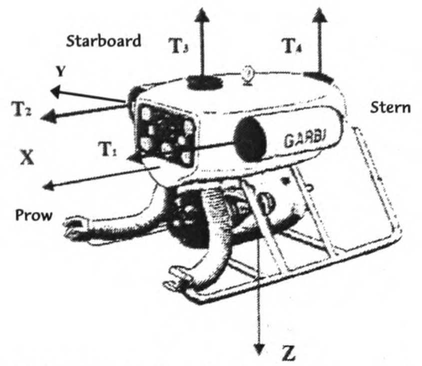


Fig. 1. GARBI Body-fixed reference frames showing the six degrees of freedom

The structure of GARBI is designed in such a way that pitch and roll cross-coupling is virtually non-existent. However, coupling appears between yaw and surge only when the vehicle has initial speed. Nevertheless, this coupling is expected and acceptable. Similarly, coupling between yaw and pitch and yaw and depth is also minor and can be neglected.

3. CONTROL TASKS OF GARBI UNDERWATER VEHICLE

As in any underwater vehicle the dynamics of GARBI are coupled and highly non-linear. When designing GARBI's controller it is necessary to compensate for its non-linear dynamics and kinematics, non-linearities due to thrusters and pressure hysteresis, barometer dead-zones, and the noise in yaw and depth measurements. Therefore, robust controllers that reliably perform complex task in the face of the above uncertainties should be used. F-like PD Controller is designed to make the vehicle follow the commands from the pilot in terms of course-changing and the course-keeping of both yaw angle and depth of the robot.

Controllers for course-keeping and/or course-changing are normally based on feedback from a gyrocompass measuring the heading for the yaw and air press-sensors measuring the difference of the pressure inside and outside of the robot and therefore the depth.

The control objective for a course-keeping controller can be expressed as $\psi, z = constant$. For course-changing, the objective is to follow the changes of the pilot commands with the best control performance in terms of small overshoot, settling time and steady state error.

Fig. 2 shows a simplified scheme of course-keeping/course-changing. The structure uses two independent FLCs for each controlled variable (Yaw and Depth), greatly simplifying the design at the cost of some decrease in performance. The corresponding inputs of these controllers are the error $e_\psi(nT) = \psi_{sp} - \psi(nT)$ between the real and the desired yaw angle and the error $e_z(nT) = z_{sp} - z(nT)$ between the real and the desired heave position as well as the change of the above errors (see Fig. 3 & Fig. 4)

$\Delta e_\psi(nT) = e_\psi(nT) - e_\psi(nT-1)$,

$\Delta e_z(nT) = e_z(nT) - e_z(nT-T)$

where $e(nT)$, $\Delta e(nT)$ and $\psi, z(nT)$ designate, crisp error, rate and process output at sampling time nT respectively. The computed rate (Δe) may not be the actual one due to delays and noise of the measurements. To overcome the above problem a rate giro should be used. Unfortunately, during the experiments the above device was not available. Limiters are used to avoid saturation of inputs in the universe of discourse. As a convention, signals are written in lower case before gains/SFs and upper case after gains/SFs, for instance $E = S_e * e$. The corresponding outputs of these controllers are; for the first controller the moment N around the z-axis, and for the second controller the force Z of the two propellers in the z-direction. The rotation N is related to the difference of power between the propellers T_1 & T_2 in the x-direction. The motion Z in the z-direction relates to the power of the propellers T_3 & T_4, which is always equal and of the same polarity.

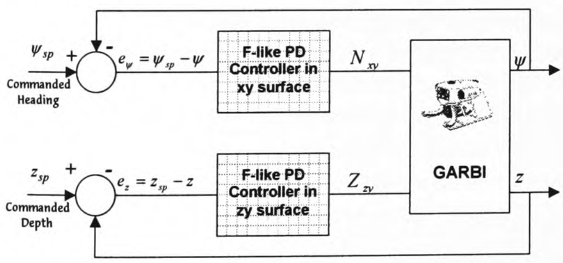


Fig. 2. Control loop for GARBI

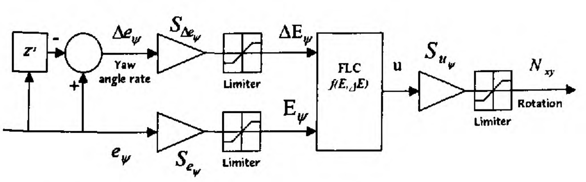


Fig. 3. Yaw F-like PD controller

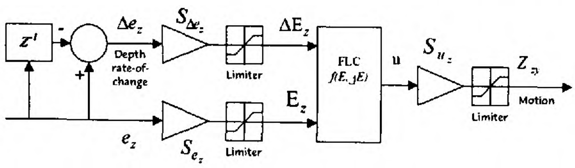


Fig. 4. Depth F-like PD controller

4. DESIGN OF THE F-LIKE PD CONTROLLER FOR GARBI

In studying the dynamic properties of the fuzzy controller, the model of the process is needed so that the impact of the successive control actions may be monitored. Since a model of GARBI is not available (due to its very complicated shape), the dynamic properties of the close loop structure have to be derived intuitively and experimentally. This is simply a cornerstone feature of the idea of fuzzy controllers. However, the tuning of F-like PD controllers systems is a fundamental problem, specially, for optimum performance. There are two different levels of tuning during the design of F-like PD controllers. The first level includes the structure (as described in section 3), the rule base, the antecedent and consequent membership functions together with their distribution, the inference mechanism and the defuzzification strategy. The second level is the tuning of gain parameters. This includes the scaling factors and other gains used in building the structure.

The settings in the design procedure for the F-like PD controller for both yaw and depth control are as follows:

- Sampling time of 1 sec. If shorter, the computation of Δe may become too sensitive to noise. This normally shows up as a restless control signal.
- Each of the FLC blocks contains 49 rules. The rule base is presented in the table format shown in Table 1. The cell defined by the intersection of the first row and the first column represents a rule such as, *if* $e(nT)$ is NB *and* $\Delta e(nT)$ is NB *then* $u(nT)$ is NB
- The linguistic variables such as PB (positive big), PS (positive small), ZO (zero) in the premise parts are fuzzy variables, while those in the consequence part are singleton with values between 0 and 1.

Table 1 The Rule Base of a F-like PD in tabular form.

e \ Δe	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

It is well known that generally fuzzy variables in the consequent parts are not needed (Nguyen & Prasad, 1999). Additionally singletons make defuzzification simpler and faster. The universe of discourse for each variable is uniformly partitioned and the MFs are placed with 50% overlap. Using the min operation for the aggregation *AND* (outer product) of the fuzzy rules, the output fuzzy set is given by $\mu_u = Min(\mu_e, \mu_{\Delta e})$. Thus, the F-like PD controller is a controller where the output is a non-linear function of the error e and its derivative de/dt ($u = F(e, de/dt)$), where F is a non-linear function of two variables.

Both yaw and depth controllers use the whole range of its universe. Therefore, the maximal E and ΔE should equal the limit of the universe. Thus, for the yaw controller

$$e_{yaw_{max}} * S_{yaw_e} = \Delta e_{yaw_{max}} * S_{yaw_{\Delta e}} = YawUniverse_{max}$$

where the universe is in a range of -180° to $+180^{\circ}$ degrees (the positive sign is for port and the negative sign is for starboard turns). The depth controller

$$e_{depth_{max}} * S_{depth_e} = \Delta e_{depth_{max}} * S_{depth_{\Delta e}} = DepthUniverse_{max}$$

where the universe is in a range of 10 meters. In both cases the scale is normalised into the range [-1,1]. It should be noted that the depth controller starts using the range of its universe when the robot is within 10 meters of the set point. In any other case the control output has the maximum value.

One possible way of improving the dynamic properties is to adjust the SFs used in the construction of the fuzzy controller. SFs play important role in the formation of the dynamics of the close-loop structure leading to the desired response of the controlled system (Zheng, 1992). For the F-like PD controller the corresponding SFs are for the inputs $S_e, S_{\Delta e}$ and for the output S_u respectively. In this study design of the experiments has been performed to optimise the input/output scaling factors.

5. DESIGN OF EXPERIMENTS

A large amount of engineering effort is consumed in conducting experiments (either in hardware, software or simulation) to produce the information required to make decisions about how different factors affect performance under different usage conditions. Experimentation seeks to determine the best pressure, the best force, the best speed etc., which will operate together within a system to produce a desired response such as settling time, overshoot, steady state error, etc. Designs of experiments are statistical methods that seeks to minimise the number of experiment and still find the combination of the factors under study which are robust in a variety of environmental conditions. A widely used method is the Taguchi Method of Robust Design, which can be applied to a wide variety of problems (Fowlkes, 1995).

Various types of matrices are used for planning experiments to study one factor at a time, where each individual factor is varied while all the other factors are fixed, full factorial experiments that investigate all possible combinations of all factors and their levels, where the possible combinations can rise to the order of y^x, x being the number of factors and y the different levels. This approach investigates all the possible combinations, maximising the possibility of finding the optimum result, but large numbers of experiments are required. Alternatively, orthogonal array, extensively used in the Taguchi Method, studies several factors at different levels simultaneously, but only requires a fraction of the full factorial combinations. The orthogonal array imposes an order on the way the experiment is carried out. The combinations are chosen to provide enough information to determine the factor effects using the analysis of mean values. In order to use a standard orthogonal array provided by the Taguchi Method, the degrees of freedom (number of independent measurements available to estimate sources of information) of the factors and levels must be matched with the degrees of freedom for that orthogonal array.

The orthogonal array of the Taguchi Method was used to plan the experimentation for the underwater robot. Three factors ($Se, S_{\Delta e}, S_u$) at three levels each were investigated. With the full factorial, this would result in 3^3 different experiments. For a three factor, three level experiment, six degrees of freedom exist, so an orthogonal array with nine experimental runs for both Yaw and Depth Fuzzy-like PD Controllers as shown in Table 2 is sufficient for this study. It should be noted that proportional gain values below these resulted in unacceptable performance.

Table 2 Orthogonal array with nine experiments

No Exp	S_e	$S_{\Delta e}$	S_u
1	0.5	0.5	3
2	0.5	1	7
3	0.5	2	10
4	1	0.5	7
5	1	1	10
6	1	2	3
7	0.75	0.5	10
8	0.75	1	3
9	0.75	2	7

6. EXPERIMENTS

Experimental trials were undertaken in Lake Banyoles, Spain in October 1999 to test both depth and yaw F-like PD controller. The power of the propellers is controlled with power cards for voltage tuning. The voltage used is in the range of 0 to 10 Volts. For heading control the opposite voltage between the horizontal propellers (T_1, T_2) is used i.e. $+V_1$, $-V_2$. So, if the heading angle is turned to $\alpha°$ clockwise for instance, the voltage in the right propeller T_1 is reduced and the voltage in the left propeller T_2 in increased by this amount. The performance of the controller in course-changing and course-keeping is shown in Fig. 5 to Fig. 8.

The navigation plan for the robot is:

- initial power of the horizontal propellers is 3V for a period of 60 seconds.
- change the heading course from 270° degrees to 135° and then to 225° and then keep it in this direction,
- and at the same time change the depth course from 0 meters to 10m and then to 5m and then keep it at this depth.

After the trials as shown in Table 2, the best combination of the SFs in terms of performance and robustness for both Yaw and depth controllers is $S_e = 0.5$, $S_{\Delta e} = 0.5$ and $S_u = 7$. Note that this set is not a member of any in Table 2. That is one of the main advantages of Taguchi method analysis.

Fig. 5(a) illustrates the Yaw, 5(b) the controller output of the power horizontal propellers (T_1, T_2) (notice that as mentioned before they have opposite sign), 5(c&d) the error and the change of error between the desired and actual yaw. Additionally, Fig. 6(a) shows the Depth, 6(b) the controller output of the power of the vertical propellers (T_3, T_4), 6(c&d) the error and the change of error between the desired and the actual depth. Finally, Fig. 7 and Fig. 8 show the corresponding pitch and the roll motions.

The performance of the yaw controller in changing the course (heading/depth) is satisfactory, as both overshoot and rise time are small. Due to buoyancy effects the depth control dynamics vary i.e. the vehicle rises faster than it descends. From Fig. 6 it can be seen that the controller has accommodated this variation producing very acceptable rise times with very a small overshoot when ascending. In course-keeping control both controllers perform quite well (Fig. 5 & Fig. 6). Finally in Fig. 7 & Fig. 8, the pitch and roll appears mostly due to the heading changes. These are very small and can be considered as fractional and thus negligible.

7. CONCLUSIONS

F-like PD control technique possesses features that are attractive in navigation control problems posed by underwater vehicles. The fuzzy logic part helps to model different types of uncertainty and imprecision of the system and together with the scaling factors it allows the building of a robust controller. Scaling factors have been chosen to be tuned to their optimal values, due to their global effect in the dynamics of the close-loop control system. Taguchi experimental method can help to minimise the number of experiments without risk of loosing vital information, investigating what is the optimal combination of the SFs. This is very important when experiments are held in a real environment and therefore are time and money consuming.

F-like PD controller is much more complicated than conventional controller. It is in fact a non-linear adaptive controller. This is why F-like PD controller presents strong robustness characteristics.

In this work it has been shown how a fuzzy controller combined with conventional PD control techniques can help to design a robust controller, dealing with the uncertainties and non-linearities of an underwater vehicle.

8. ACKNOWLEDGEMENT

The authors wish to acknowledge the support for this work provided by the British Council under the British/Spanish Acciones Integradas programme. The authors would also like to thank Marc Carreras, Josep Cortada Hortala and Vivianne Bouchereau for the help during the experiments and for their useful comments during the preparation of this manuscript.

9. REFERENCES

Amat, J. ,Batlle, J. ,Casals, A., & Forest, J. (1996). GARBI: The Low Cost ROV, Constraints and Solutions.

Fowlkes, Y. W. (1995). *Engineering Methods for Robust Product Design Using Taguchi Methods in Technology and Product Development.* Addison-Wesley Publishing Company, USA.

Mudi, K P., & Pal, R. N. (1999). A Robust Self-Tuning Schem for PI- and PD - Type Fuzzy Controllers. *IEEE Trans. on Fuzzy Systems,* 7 (1), pp. 2-16.

Nguyen, H. T., & Prasad, R N. (1999). *Fuzzy Modelling and Control: selected work of M. Sugeno.*

Zheng, L. (1992). A Practical Guide to Tune of Proportional and Integral (PI) Like Fuzzy Controllers. *Proc. FUZZ IEEE,* pp. 633-641

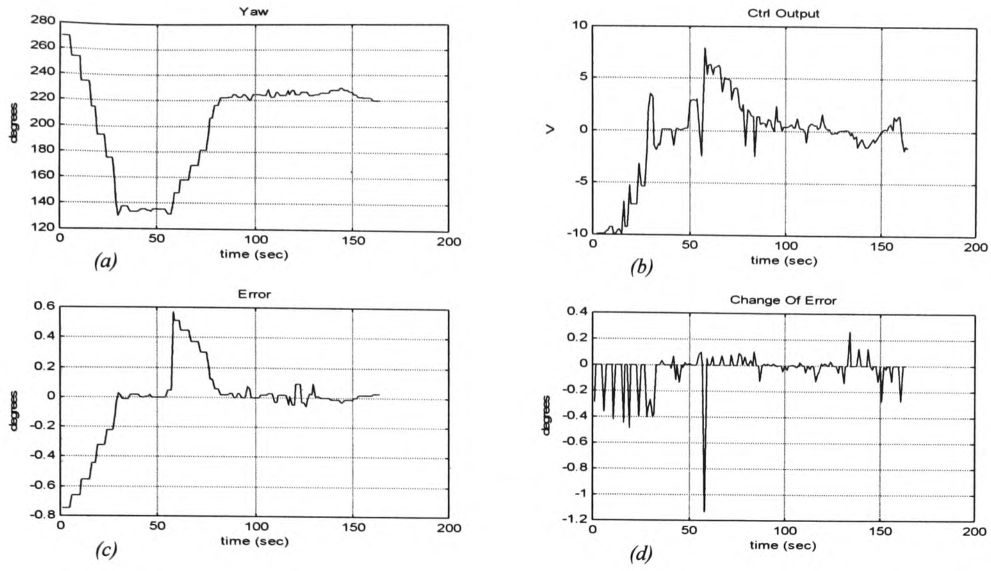


Fig. 5. The plan of the "Yaw" experiment (briefly) was as follows: Change the course from 270° degrees to 135° and then to 225° and then keep heading in this direction

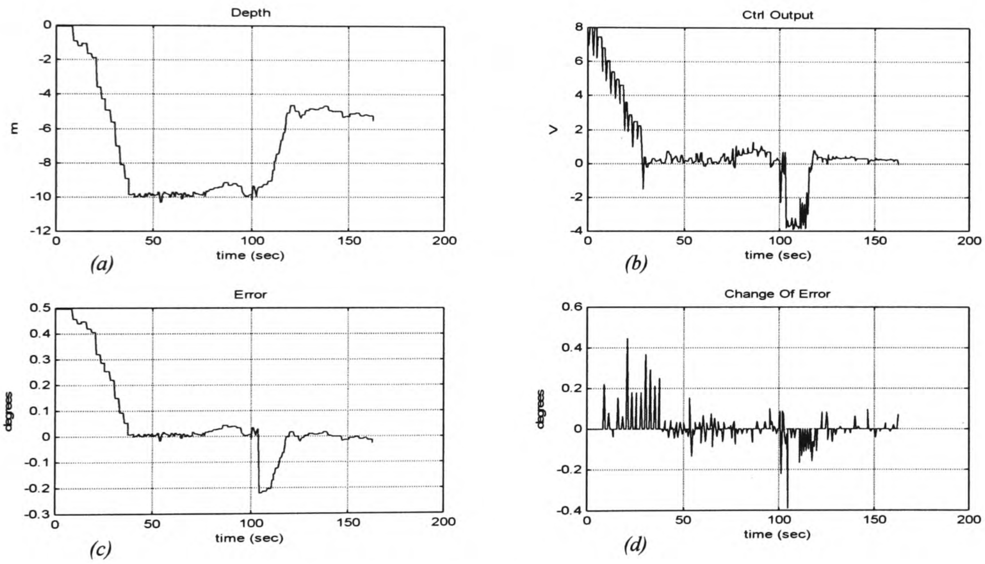


Fig. 6. The plan of the "Depth" experiment (briefly) was as follows: Change the course from 0 meters to 10m and then to 5m and then keep depth at this level

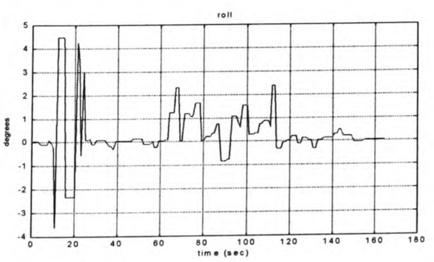


Fig. 7. The roll due to the experiment is not more that 5° degrees

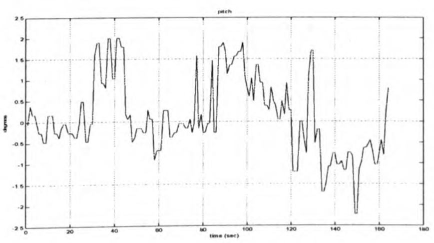


Fig. 8. The pitch due to the experiment is not more that 2.5° degrees

Modelling and Motion Control of an Autonomous Underwater Robot using a Fuzzy and Fuzzy-neural Approach

I. S. Akkizidis and G. N. Roberts

Mechatronics Research Centre, University of Wales College, Newport,
Allt-yr-yn Campus, PO Box 180, Newport, South Wales, NP9 5XR, UK

The design of an autopilot for the control of an Autonomous Underwater Vehicle (AUV) is of interest both from the point of view of motion stabilisation as well as manoeuvring performance. Such vehicles are commonly classified as being highly non-linear uncertain systems and are consequently difficult to control effectively using model-based control methods. The modelling task of these vehicles which consists of three-dimensional equations of motion for its hydrodynamical shape is very complicated. The paper describes a modelling, mission and motion control system for an AUV based on fuzzy and fuzzy neural techniques. The idea for the fuzzy modelling is an on-line supervisory scheduling system, which chooses the linear function that describes the system. Fuzzy mission control and fuzzy neuro motion control strategies, with the ability to adapt membership functions, extract new rules and forget unused rules, are proposed.

1. INTRODUCTION

Manoeuvring control of underwater vehicles is a demanding task. This difficulty stems from the fact that Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) may be classified as uncertain systems possessing highly non-linear dynamics. Therefore developing the model of the AUV dynamics has to be attempted before controller design may be considered. AUVs and ROVs are highly non-linear, multivariable dynamic process which means that controller designs must be able to deal with the non-linearities encountered.

This study described herein is based on a low-cost ROV named GARBI developed at the University of Barcelona and the University of Girona. The vessel, which is illustrated in figure 1, is used for underwater mission operations such as observations and inspections [1,2,3]. GARBI is linked to a surface ship (or with other operating platform) by an umbilical cable carrying power and providing a communication link. This umbilical link imposes limitations to the ROV operations, such as depth limits and danger of cable snagging.

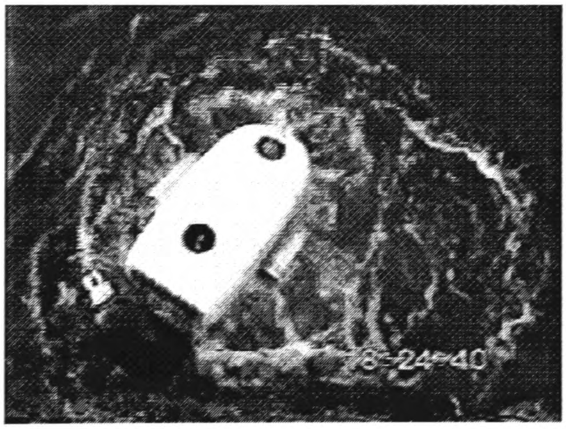


Figure 1. Photo of GARBI underwater robot

By definition AUVs are vessels which have sufficient on-board power and intelligent ability to move purposefully without human intervention in environments that have not been specifically engineered for it. It is clear therefore that AUVs bring a number of advantages: they have no umbilical cable to limit range, or to become entangled in a surrounding structure. They can also undertake missions that would be impractical or impossible with an ROV, such as long range observations and collection of oceanographic data and under-ice surveying.

The motivation for this work is to extend GARBI from operating as an ROV to operating as an AUV. Fuzzy Logic and Neuro Fuzzy techniques are propose for modelling, mission and motion control.

1.1. Comparison between Artificial Neural Network and Fuzzy Logic

A detailed presentation of Fuzzy Systems (FS) and Artificial Neural Networks (ANN) theory is beyond the scope of this paper. A brief comparative study between FS and ANNs related to their operations in the context of knowledge acquisition, uncertainty, reasoning and adaptation is presented in table 1.

1.2. Fuzzy Neural Control systems (FNC)

It can be seen from table 1 that the advantages of the fuzzy approach are mainly the disadvantages of the ANN approach, and vice versa. So the idea is naturally to combine neural networks and fuzzy systems to overcome their disadvantages, but to retain their advantages. The integration of these two fields has given birth to Fuzzy-Neural Systems (FNS). From ANN, the powerful learning capabilities enable these systems either to automatically learn the fuzzy decision rules or to learn from a set of plant measurements, whereas the fuzzy presentation enables designers to extract the learnt information from the expert in a form easily understandable.

2. MODELLING OF AUV

2.1. Fuzzy Modelling

Fuzzy Modelling is the method of describing the characteristics of a system using fuzzy rules. The technique can express complex non-linear dynamic systems by linguistic *if*-then rules [4].

A typical Takagi-Sugeno fuzzy model has the form:

$$R_s^i \;\; if \;\; s = LS^i \;\; then \;\; x_i = f_i(x,u) \qquad (1)$$

where s is the operating point vector $s=\{s_1, s_2, \ldots, s_{n_s}\}$. The vector consists, in general, of state, input and output variables (the n_S is its dimension). LS^i is the i-th fuzzy state vector equal to:

$$LS^i = (LS_1, LS_2, \ldots, LS_{n_s})^T \qquad (2)$$

where LS_{n_s} is the fuzzy values of s with appropriate Membership Function (MF).

The *then*-part of this fuzzy rule defines a linear autonomous open loop model representing the system dynamics within the fuzzy region LS^i specified in the *if*-part of the same fuzzy rule. This model is of the form $x_i=f_i(x,u)$, where f_i is a linear function normally obtained via an identification procedure.

The reason for adopting the fuzzy model described above is that most systems (or plants) are non-linear and therefore cannot be described by a

Table 1. A comparative study between fuzzy systems and artificial neural networks

Skills		Fuzzy Systems	Artificial Neural Networks
Knowledge acquisition	Inputs	Human experts	Sample sets
	Tools	Interaction	Algorithms
Uncertainty	Information	Quantitative and qualitative	Quantitative
	Cognition	Decision making	Perception
Reasoning	Mechanism	Heuristic search	Parallel computations
	Speed	Low	High
Adaptation	Fault-tolerance	Low	Very high
	Learning	Induction	Adjusting synaptic weights
Natural language	Implementation	Explicit	Implicit
	Flexibility	High	Low

single linear model. Instead of constructing complicated non-linear models based on physical laws, an alternative approach can be used, namely the construction of a collection of linear models. In this case, each (local) linear model approximates the original non-linear system around different operating points and a *supervisory scheduling system* determines which local linear model suits the particular operating conditions [5].

The supervisory scheduling system uses a discriminant function:

$$w = f_d(s) \tag{3}$$

This function defines a weight vector $w = \{w_1, w_2, \ldots, w_k\}$, with $w_i \in [0,1]$, for each particular value of the operating point vector. This definition can be applied using *min* or *max* operation. For example, for the particular crisp value $s = \{s_1^*, s_2^*, \ldots, s_{n_s}^*\}$ of the operating point vector the weight vector will be:

$$w = \min(\mu_{LS_1}(s_1^*), \mu_{LS_2}(s_2^*), \ldots, \mu_{LS_{n_s}}(s_{n_s}^*)) \tag{4}$$

where $\mu_{LS_{n_s}}$ is the degree of membership of the crisp value $s_{n_s}^*$ of s_{n_s}.

The overall output X of the composite model is calculated as the weighted mean of the outputs x_i of the local linear open loop models, given as:

$$X = \frac{\sum_{i=1}^{k} w_i \cdot x_i}{\sum_{i=1}^{k} w_i} \tag{5}$$

where k is the number of local open loop linear models.

2.2. Linear modelling study of GARBI

GARBI is a non-linear, multivariable dynamic system. In order to design a controller for GARBI or to undertake simulation studies a dynamic model is needed. Mathematical models, which describe these non-linear dynamics, are very complex as well as limited in terms of only being able to represent the physical laws of the system. An identification method based on experiments is proposed. Using linear control theory (local) models can be investigated. Step or frequency response methods can be applied to identify the transfer functions which describe the relationships between inputs and outputs of the system, in the *s*- or *z*- domain.

The equivalent difference equations of the transfer functions can be written in the form:

$$x(t) = -\sum_{i=1}^{n_a} a_i x(t-i) + \sum_{j=0}^{n_b} b_j u(t-j-1) \tag{6}$$

where the degrees n_a, n_b of the polynomials are given by the order of the backward shift operation, n_a = number of poles of G(s) and $n_b \le n_a$.

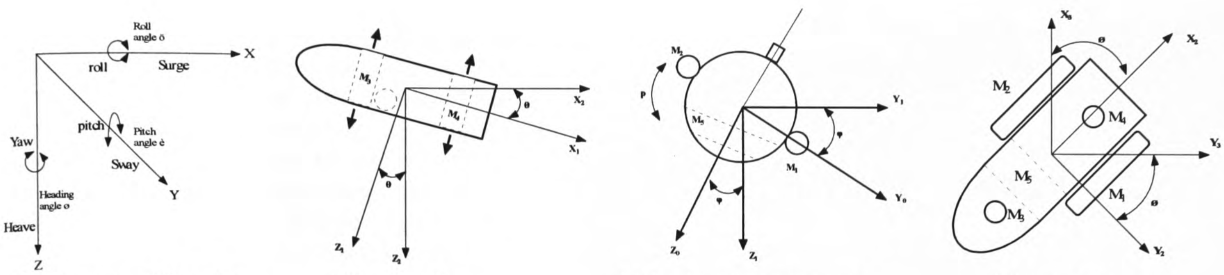
However, using fuzzy model theory the GARBI non-linear system can be described as in section 2.3.

2.3. Fuzzy Modelling of GARBI

The study of the motion of marine vehicle involves six *Degrees Of Freedom* (DOF) (Fig. 2a), since six independent co-ordinates are necessary to determine the position and orientation of a rigid body. The first three co-ordinates (Surge, Sway and Heave) and their time derivatives correspond to the position and translational motion along the *x-, y-, and z-axes.* The last three co-ordinates (roll, pitch and heading angle) and their time derivatives are used to describe orientation and rotational motion.

In order to drive the GARBI on the (x,y), (x,z), (y,z) surface, four different speeds (Zero, Low, Medium and High) of its five propellers M_1 to M_5 (Fig. 2b,c,d) have to be applied. These are the inputs of the system's model. The output however, depends on the six DOF of the vessel. Therefore, the rotation about the *z-axes* may be responsible for roll & yaw, the *y-axes* for pitch, surge & heave, and the *x-axes* for sway, roll & yaw. The functions that characterise the relation between this Multi-Input Multi-Output (MIMO) system represent each (local) linear model of the vessel.

The fuzzy variables with speed in the region *Zero, Low, Medium* and *High* can be used to fuzzify the input of the system's model. It is considered that the shape of the Membership Function within these regions is not so critical. The shapes can be chosen after experimental tests.



(a) degrees of freedom (b) side view (c) horizontal reference (d) top view

Figure 2. Illustration of the GARBI, showing its degrees of freedom and control

The fuzzy model is given as a set of fuzzy rules with each fuzzy rule R_s^i being of the form as shown in (eq.1). Specifically in the GARBI's modelling the rules will have a form such as:

R_s^i : **if** *speed Mi* is High **and** *speed Mj* is Low
then x=f(x,y)

The *then*-part of the above fuzzy rule defines a linear autonomous open loop model representing the system dynamics in terms of the six DOF of GARBI, within the fuzzy regions (Zero, Low, Medium and High) specified in the *if*-part of the same fuzzy rule.

Therefore, during the fuzzification procedure the fuzzy inputs are defined. Using Sugeno type fuzzy rules in the knowledge base, the local open loop linear models are determined (equation. 1). The sum of these models, (equation 5), having different weight, will describe the non-linear GARBI system for any variation in the input and/or output.

3. CONTROLLING THE AUV

3.1. Fuzzy-Neuro Control

Fuzzy control is a successful application of fuzzy theory [6,7] and is used in many applications [8]. Fuzzy Logic Controller (FLC) is a knowledge-based controller, which uses fuzzy logic for knowledge representation and inference [9]. One feature of FLC is that expert knowledge is represented by a compact set of fuzzy *if-then* rules consisting of membership functions in the *if*-part and functions agreeing with the controller in the *then*-part. Therefore, the design of the parameters in FLCs is obtained from the expert. However, FLCs cannot reflect a non-linear system dynamics fully. In addition, the design processes depend on trial-and-error methods or some heuristic algorithm [10]. Moreover, an expert who knows the characteristics of the system may also be needed for setting up the initial rules.

If the actual output of the controller differs from the desired behaviour, either an unsuitable choice of membership functions or missing fuzzy rules is possible. The desired output can be estimated either under simulation or from experimental results.

Multi-layer neural networks with learning algorithms, have been used for *extracting fuzzy rules* or *tuning (adapting) the membership functions* in a linguistic variable based fuzzy control environment.

Here it is proposed to first try and solve the problem using adaptation of membership function technique. If the solution is not satisfactory, a technique for extracting new fuzzy rules is used.

3.1.1. Adaptation of Membership Functions

The adaptation of membership functions is a reverse mechanism that is deduced from the forwarding inference mechanism. The approach comes from ANNs, where the computation of the control values are obtained from the measured input values, which are feeding a feedforward procedure similar to that found in layered neural nets. If the actual output is not able to drive the controlled system to a desired state, an error has to be propagated back through the architecture, changing the parameters, taking into account the feed forward propagation of inputs.

If it is not possible to determine an optimal control action for a given state, the error of the produced output cannot be calculated directly. Therefore, a reinforcement or "non-supervised" learning algorithm is used to evaluate the error of the fuzzy controller corresponding to the given input [11]. However, if the desired control output is known, then a "supervised" learning algorithm is used and the network is trained corresponding to that output. Previous work in this field is reported in [12,13,14]. In each case it is the range of membership functions, which is re-adjusted.

3.1.2. Extraction of new fuzzy rules

In some cases, the expert cannot predict all the rules to construct the base-knowledge of the fuzzy controller. Also it is possible that some of the rules may not be the most suitable. Algorithms to extract new fuzzy rules can be considered as one solution to this problem [15].

An alternative approach, which is used in this study, is to extract new fuzzy rules as above but also to correct existing rules and to forget unused or incorrect rules.

There are several ways to investigate why the fuzzy controller may fail because of improper rules, such as the error of the controller is too high or the controller reaction is too slow. The method is first to try and solve the problem by adding or taking out linguistic variables of the antecedent part of the fuzzy rules. If the improvement is not so good then new rules are created. If a rule is not used, then it is removed.

3.2. Control study for GARBI

The motion control system for GARBI shown in figure 3, involves two control loops: one for mission control and one for navigation. The parameters of the mission target such as depth and field of action are specified using fuzzy logic control techniques. The navigation of GARBI, which is affected by currents, obstacles and turbulence (regarded, as disturbances) is developed using Fuzzy-Neuro control technique as discussed above. Therefore the motion controller using the five propellers (Fig. 2 b,c,d) can reach the mission target and also be robust against disturbances. Typical Mamdany and Sugeno fuzzy model rules are used to construct the knowledge base, including the mission and the control data respectively.

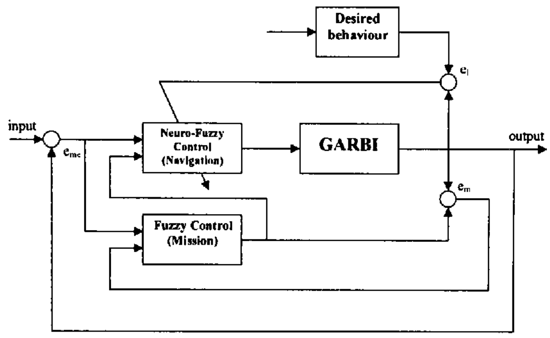


Figure 3. Motion Control System

The fuzzy rules have the form:

For the mission control:

$$R^i_{cm} \quad if \quad cm_a = LA_i, ..., and \quad cm_b = LB_i$$
$$then \quad cm_c = LC \tag{7}$$

For the navigation control:

$$R^j_{cn} \quad if \quad cn_a = LD_j, ..., and \quad cn_b = LE_j$$
$$then \quad y_j = f_j(y, u) \tag{8}$$

where $cm_a, \ldots cm_b$, cm_c, cn_a and cn_b are linguistic variables representing the process state variables and the control variable; $LA_i, \ldots, LB_i$, LC_i, LD_j and LE_j are the linguistic values of the linguistic variables $cm_a, \ldots cm_b$, cm_c, cn_a and cn_b in the universes of discourses $U, \ldots, V$, W, Q and G respectively, $i=1,2,\ldots,n$.

A conjunction of input variables associated with their respective linguistic values determines a linguistic value associated with the output variable. All rules are evaluated in parallel, and their outputs are combined to a fuzzy set, which has to be defuzzified to receive the crisp output value. The conjunction of the inputs is achieved by the *min*-operation, and for aggregating the outputs of the rules, the *max*-operation is utilised. Function y is used for the estimation of the parameters of the fuzzy-neuro controller system.

There exist three kinds of error signals in the proposed system. One of them is the difference between the desired response and the process output. This error e_l, called the learning error, is used to learn correct control action u and will tend to zero in the time interval of interest with increasing iteration number. The other error e_m, called the mission error, is the error between the mission requirements and the process statement. The last error e_{me}, called the measured control error and defined as the difference between the set-point and the process output, is primarily employed to construct the rule-base which will subsequently be used in the fuzzy control system.

4. CONCLUSION

This paper has described how fuzzy and fuzzy neural techniques are used for the development of the fuzzy model and fuzzy neural controller for the motion control of the GARBI underwater robot. The problems and difficulties with applying a model-following approach have been highlighted and an alternative intelligent systems approach has been presented.

The next phase of the work is to undertake an extensive simulation study in order to evaluate the motion control system prior to undertaking trials at sea.

Acknowledgement

The authors wish to acknowledge the support for this work provided by the British Council under the British/Spanish Acciones Integradas programme

REFERENCES

1. Amat, J.,Batlle, J.,Casals, A., & Forest, J. 1996. GARBI: The Low Cost ROV, Constraints and Solutions.
2. --- 1995. *GARBI: The UPC Low Underwater Vehicle.* Undersea Robotics & Intelligent Control, U.S/Portugal Workshop. pp. 19-24.
3. Amat, J.,Codina, J.,Cufi, X., & Puigmal, J. 1995. *Vision Based Control of Underwater Vehicle for Long Duration Observations.* ICAR' 95. pp. 273-277,
4. Takagi, T., & Sugeno, M. 1985. Fuzzy Identification of Systems and its Applications to Modeling and Control. *IEEE Trans. on Systems, Man and Cybernetics,* **SMC-15**, pp. 116-132.
5. Palman, R.,Driankov, D., & Hellendoom, H. 1997. *Model Based Fuzzy Control.* Germany: Springer-Verlag Berlin Heidelberg.
6. Mamdani, E. H. 1974. Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. *Proc. of the IEE,* **121**, pp. 1585-1588.
7. Zadeh, L. A. 1968. Fuzzy Algorithms. *Information and Control,* **12**, pp. 94-102.
8. Terano, T.,Asai, K., & Sugeno, M. 1992. *Fuzzy Systems Theory & its Applications.* Academic Press, Inc.
9. Lee, C. C. 1990a. Fuzzy Logic in Control systems: Fuzzy Logic Controller Part I. *IEEE Trans on Systems, Man and Cybernetics,* **20**, pp. 404-418.
10. Takagi, H. 1990. Fusion Technology of Fuzzy Theory and Neural Networks - Survey and Future directions. *In P. of the Intr. Conf. on Fuzzy Logic and Neural Networks,* pp. 13-26.
11. Nauck, D., & Kruse, R. 1992. A Neural Fuzzy Controller Learning by Fuzzy Error Propagation. *Proc. NAFIPS'92,* pp. 388-397.
12. Sutton, R.,Taylor, H., & Roberts, G. N. 1997. Tuning Fuzzy Ship Autopilots using Artificial Neural Networks. *InstMC Transactions, Measurement and Control,* **19** (2), pp. 94-106.
13. Jang, J. & Sun, T. 1995. Neuro-Fuzzy Modelling & Control. *Proc. IEEE,* **83**, pp. 378-405.
14. Jang, J-S. R. 1993. ANFIS: Adaptive – Network - Based Fuzzy Inference System. *IEEE Trans on Systems, Man and Cybernetics,* **23** (3), pp. 665-685.
15. Nie, J. & Linkens, D. 1995. *Fuzzy-Neural Control Principles, Algorithms and Applications.* Prentice Hall.

Fuzzy Modelling & Fuzzy-Neuro Motion Control of an Autonomous Underwater Robot

I. S. Akkizidis **G. N. Roberts**

Mechatronics Research Centre,
University of Wales College, Newport,
Allt-yr-yn Campus, PO Box 180, Newport
South Wales, NP9 5XR, UK

Tel: +44 (0) 1633 432487 or +44 (0) 1633 432441 Fax: +44 (0) 1633 432442
Email: i.akkizidis@newport.ac.uk and g.roberts@newport.ac.uk

Abstract

This problem addresses the modelling and motion control of an Autonomous Underwater Vehicle (AUV). Such vehicles are commonly classified as being highly non-linear uncertain systems and are consequently difficult to control effectively using model-based control methods. The paper describes a modelling and motion control system for an AUV based on fuzzy and fuzzy neural techniques. The development of a fuzzy model and fuzzy neural motion control strategy is presented.

1 Introduction

Motion control of autonomous underwater vehicles is a demanding task for which no completely satisfactory approaches have yet been developed. This difficulty stems from the fact that Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) may be classified as uncertain systems possessing highly non-linear dynamics.

This study is based on a low-cost ROV, GARBI developed at the University of Barcelona and Girona for underwater mission operations such as observations and inspections [1,2,3]. GARBI is linked to a surface ship (or with other operating platform) by an umbilical cable, which carries power and provides a communication link. There are however, limitations to the ROV, such as limit on depth and danger of cable movement.

Autonomous Underwater Vehicles (AUVs) do not suffer from these problems. AUV means a vessel, which has enough on-board power and intelligent ability to move purposefully and without human intervention in environments that have not been specifically engineered for it. AUV requires a number of heterogeneous capabilities, including the ability to execute elementary goal-achieving actions. For example, reaching a given location; reacting in real time to unexpected events; building, using and maintaining a map of the surrounding environment; determining the robot's position with respect to this map; forming plans that pursue specific goals or avoid undesired situations; and adapting to changes in the environment.

It is obvious therefore that AUV brings a number of advantages: it has no umbilical cable to limit its range, or to become entangled in a surrounding structure. It can also undertake missions that would be impractical or impossible with an ROV, such as long range observations and collection of oceanographic data and under-ice surveying.

The aim of this work is to transfer the GARBI from operating as an ROV to an AUV in order to expand its capabilities. Fuzzy Logic and Neuro Fuzzy techniques are proposed to design the model and the controller of GARBI respectively.

1.1 Fuzzy System Control

Classical control theory is based on mathematical models that describe the behaviour of the plant or system under consideration. The main idea of fuzzy control is to build a model of a human control expert who is capable of controlling the plant without thinking in mathematical model terms [9,16]. The design of fuzzy controllers is normally based on prior knowledge of skilled human operators in the form of linguistic control rules, which are translated into the framework of fuzzy set theory [20]. The process knowledge is expressed in the form of IF-THEN rules as in traditional expert systems. However, a fuzzy logic controller can be regarded as a real-time expert system that employs fuzzy logic to manipulate qualitative variables. The specification of good linguistic rules depends on the knowledge of the control expert. The translation of these rules into fuzzy set theory depends on the choice of certain parameters, such as shape and degrees of Membership Functions, for which particular rules do not exist. Fuzzy controllers are unable to learn by their own experiences and unable to adapt to new conditions in cases of non-linear systems (of high order with uncertainties in parameter and structure). However, their ability to extract new fuzzy rules and change parameters of Membership Functions is necessary for tuning fuzzy controllers. The fuzzy logic approach makes it possible in many cases to build control systems that are more robust, cost-effective and energy-saving, and seems to be the best answer available today for a broad class of challenging control problems.

1.2 Artificial Neural Networks

Artificial Neural Networks (ANN), [13] are designed to model certain aspects of the human brain. They consist of simple processing elements (neurones) that exchange signals along weighted connections. The idea is to take advantage of the knowledge of the ways that the (human) brain learns and functions, to present this information at algorithmic level, and to use the results in computerised problem solving. ANNs need their inputs to be quantitive in nature. The main advantage of neural networks is the ability to learn from examples; moreover good learning algorithms are available for them. However, their Black box behaviour reduces the above advantages. Generally it is neither possible to use prior knowledge to initialise the network, nor can their final state be interpreted in terms of rules.

1.3 Fuzzy Neural Control systems (FNC)

The advantages of the fuzzy approach are mainly the disadvantages of the ANN approach, and vice versa. So the idea is naturally to combine neural networks and fuzzy systems to overcome their disadvantages, but to retain their advantages. The integration of these two fields has given birth to fuzzy-neural systems. From ANN, the powerful learning capabilities enable these systems either to automatically learn the fuzzy decision rules or to learn from a set of plant measurements, whereas the fuzzy presentation enables designers to extract the learnt information from the expert in a form easily understandable. As the former property reduces the time required to create the model, the latter increases the usefulness of the model since there now exists at least some kind of explanation for the model outcome. The learning procedure can also be reinitiated from time to time, or adaptive or learning systems can be considered. Due to the duality in forms of expression, knowledge in IF-THEN form can alternatively be used either in the model initialisation, or during the adaptation to the changing process behaviour.

The goal of neuro-fuzzy combinations in control is to obtain adaptive systems that can use prior knowledge, and can be interpreted by means of linguistic rules. Fuzzy-neural Control systems or Neural-fuzzy Control systems can be divided into *co-operative* and *hybrid* systems. In *Co-operative* approaches the controller itself has a structure resulting from the combination of fuzzy systems and neural networks. In *Hybrid* approaches new control architecture is created using concepts from both paradigms and thus can be interpreted as a neural net and as a fuzzy controller. Besides this, there are *concurrent neural/fuzzy models* in which the resulting control system consists of fuzzy systems and neural networks as independent components performing completely different tasks [7,18,4].

2 Modelling of AUV

2.1 Fuzzy Modelling

Fuzzy Modelling is the method of describing the characteristics of a system using fuzzy rules, and it can express complex non-linear dynamic systems by linguistic if-then rules [16].

A typical Takagi-Sugeno fuzzy model has the form:

$$R_s^i \;\; if \;\; s = LS^i \;\; then \;\; x_i = f_i(x,u) \qquad (1)$$

where s is the operating point vector $s = \{s_1, s_2, \ldots, s_{n_s}\}$. The vector consists, in general, of state, input and output variables (the n_S is its dimension). LS^i is the i-th fuzzy state vector equal to:

$$LS^i = (LS_1, LS_2, \ldots, LS_{n_s})^T \qquad (2)$$

where LS_{n_s} is the fuzzy values of s with appropriate MF.

The *then*-part of this fuzzy rule defines a linear autonomous open loop model representing the system dynamics within the fuzzy region LS^i specified in the *if*-part of the same fuzzy rule. This model is of the form $x_i = f_i(x,u)$, where f_i is a linear function normally obtained via an identification procedure.

The basic idea underlying the above type of fuzzy open loop model is that most systems (or plants) are non-linear and therefore cannot be described by a single linear model. Instead of constructing complicated open loop non-linear models based on physical laws, an alternative approach can be chosen (Fig. 1), namely the construction of a collection of open loop linear models. In this case, each open loop (local) linear model approximates locally the original non-linear system around different operating points and a *supervisory scheduling system* determines which particular local open loop linear model is the relevant one [12].

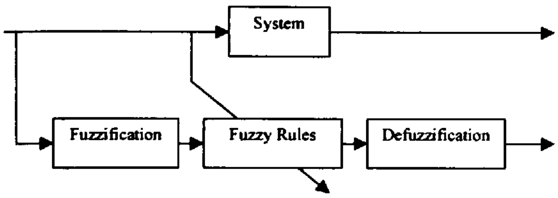


Figure 1. Modified Fuzzy Model structure

The supervisory scheduling system uses a discriminant function:

$$w = f_d(s) \tag{3}$$

This function defines a weight vector $w = \{w_1, w_2, \ldots, w_k\}$, with $w_i \in [0,1]$, for each particular value of the operating point vector. This definition can be applied using *min* or *max* operation. For example, for the particular crisp value $s = \{s_1^*, s_2^*, \ldots, s_{n_s}^*\}$ of the operating point vector the weight vector will be:

$$w = \min(\mu_{LS_1}(s_1^*), \mu_{LS_2}(s_2^*), \ldots, \mu_{LS_{n_s}}(s_{n_s}^*)) \tag{4}$$

where $\mu_{LS_{n_s}}$ is the degree of membership of the crisp value $s_{n_s}^*$ of s_{n_s}.

The overall output X of the composite model is calculated as the weighted mean of the outputs x_i of the local linear open loop models, given as:

$$X = \frac{\sum_{i=1}^{k} w_i \cdot x_i}{\sum_{i=1}^{k} w_i} \tag{5}$$

where k is the number of local open loop linear models.

2.2 Linear modelling study of GARBI

Modelling of marine vehicles involve the study of *statics* and *dynamics*. Statics is concerned with the equilibrium of bodies at rest or moving with constant velocity, whereas dynamics is concerned with bodies having accelerated motion.

GARBI is a non-linear dynamics system. In order to design a controller for GARBI a model is needed. Mathematical models, which describe these non-linear dynamics, are very complex as well as limited in terms of only being able to represent the physical laws of the system. An identification method based on experiments is proposed. Using linear control theory, the open loop (local) models can be investigated. Step or frequency response method can be applied to identify the transfer function of these models (in a *s*- or *z*- domain), and thus the linear functions that describe the relationship between input and output of the system.

The equivalent difference equation of that function can be written as:

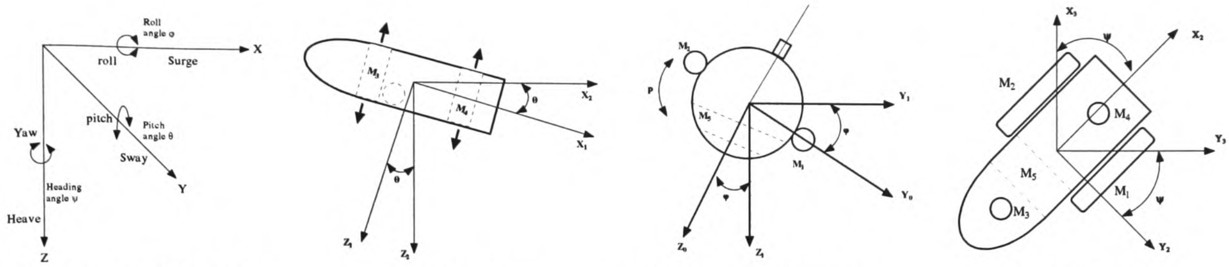
$$x(t) = -\sum_{i=1}^{n_a} a_i x(t-i) + \sum_{j=0}^{n_b} b_j u(t-j-1) \tag{6}$$

where the degrees n_a, n_b of the polynomials are given by the order of the backward shift operation, n_a = number of poles of G(s) and $n_b \le n_a$.

However, using fuzzy model theory the GARBI non-linear system can be described as follows.

2.3 Fuzzy Modelling study of GARBI

The study of the motion of marine vehicle involves six *Degrees Of Freedom* (DOF) (Fig. 2a), since six independent co-ordinates are necessary to determine the position and orientation of a rigid body. The first three co-ordinates (Surge, Sway and Heave) and their time derivatives correspond to the position and translational motion along the *x*-, *y*-, *and z-axes*. The last three co-ordinates (roll, pitch and heading angle) and their time derivatives are used to describe orientation and rotational motion.



(a) degrees of freedom (b) side view (c) horizontal reference (d) top view

Figure 2. Illustration of the GARBI, showing its degrees of freedom and control

In order to drive the GARBI on the (x,y), (x,z), (y,z) surface, four different speeds (Zero, Low, Medium and High) of its five propellers (Fig. 2b,c,d) have to be applied. These are the inputs of the system's model. The output however, depends on the six DOF of the vessel. Therefore, the rotation about the *z-axes* may be responsible for roll & yaw, the *y-axes* for pitch, surge & heave, and the *x-axes* for sway, roll & yaw. The functions that characterise the relation between this Multi-Input Multi-Output (MIMO) system represent each open loop (local) linear model of the vessel.

The fuzzy variables with speed in the region *Zero, Low, Medium and High* can be used to fuzzify the input of the system's model. It is believed that the shape of the Membership Function within these regions is not so critical. The shapes can be chosen after experimental tests.

The open loop fuzzy model [16] is given as a set of fuzzy rules with each fuzzy rule R_s^i being of the form as shown in (eq.1). Specifically in the GARBI's modelling the rules will have a form such as:

R_s^i: **if** *speed Mi* is High **and** *speed Mj* is Low
then x=f(x,y)

The *then*-part of the above fuzzy rule defines a linear autonomous open loop model representing the system dynamics in terms of the six DOF of GARBI, within the fuzzy regions (Zero, Low, Medium and High) specified in the *if*-part of the same fuzzy rule.

Therefore, during the fuzzification procedure the fuzzy inputs are defined. Using Sugeno type fuzzy rules in the knowledge base, the local open loop linear models are determined (eq. 1). The sum of these models, (eq. 5), having different weight, will describe the non-linear GARBI system for any variation in the input and/or output.

3. Controlling the AUV

3.1 Fuzzy-Neuro Control

Fuzzy control is a successful application of fuzzy theory [9,19] and is used in many applications [17]. Fuzzy Logic Controller (FLC) is a knowledge-based controller, which uses fuzzy logic for knowledge representation and inference [8]. One feature of FLC is that expert knowledge is represented by a compact set of fuzzy *if-then* rules consisting of membership functions in the *if*-part and functions agreeing with the controller in the *then*-part. Therefore, the design of the parameters in FLCs is obtained from the expert. However, FLCs cannot reflect a non-linear system dynamics fully. In addition, the design processes depend on trial-and-error methods or some heuristic algorithms [15]. Moreover, an expert who knows the characteristics of the system may also be needed for setting up the initial rules.

If the actual output of the controller differs from the desired behaviour, either an unsuitable choice of membership functions or missing fuzzy rules is possible. The desired output can be estimated under either simulation or experimental results.

Multi-layer neural networks with learning algorithms, have been used for *extracting fuzzy rules* or *tuning (adapting) the membership functions* in a linguistic variable based fuzzy control environment.

The idea is first to try and solve the problem using adaptation of membership function technique. If the solution is not satisfactory, extraction of new fuzzy rules is used.

3.1.1 Extraction of new fuzzy rules

In some cases, the expert cannot predict all the rules to build up the base-knowledge of the fuzzy controller. Therefore, some of the rules may not be the best ones. Algorithms to extract new fuzzy rules can be considered as one solution to this problem [11].

An alternative approach, which is used in this study, is to extract new fuzzy rules as above but also to correct existing rules and to forget unused or incorrect rules.

There are several ways to investigate why the fuzzy controller may fail because of improper rules, such as the error of the controller is too high or the controller reaction is too slow. The idea is first to try and solve the problem by adding or taking out linguistic variables of the antecedent part of the fuzzy rules. If the improvement is not so good then new rules are created. If a rule is not used, then it is removed.

3.1.2 Adaptation of Membership Functions

The adaptation of membership functions is a reverse mechanism that is deduced from the forwarding inference mechanism. The idea comes from ANNs, where the computation of the control values are given as the measured input values, which are feeding a feedforward procedure similar to that found in layered neural nets. If the actual output is not able to drive the controlled system to a desired state, an error has to be propagated back through the architecture, changing the parameters, taking into account the feed forward propagation of inputs.

If it is not possible to determine an optimal control action for a given state, the error of the produced output cannot be calculated directly. Therefore, a reinforcement or "non-supervised" learning algorithm can be used to evaluate the error of the fuzzy controller corresponding to the given input [10]. However, if the desired control output is known, then a "supervised" learning algorithm can be used and the network is trained corresponding to that output. Previous work in this field [14,5,6]. In each case it is the range of membership functions, which is re-adjusted.

3.2 Control study for GARBI

The motion control system for GARBI shown in figure 3, involves two control loops: one for mission control and one for navigation. The parameters of the mission target such as depth and field of action are specified using fuzzy logic control techniques. The navigation of GARBI, which is affected by currents, obstacles and turbulence (regarded, as disturbances) is developed using Fuzzy-Neuro control technique as discussed above. Therefore the motion controller using the five propellers (Fig. 2 b,c,d) can reach the mission target and also be robust against disturbances. Typical Mamdany and Sugeno fuzzy model rules are used to build up the knowledge base, including the mission and the control data respectively. The fuzzy rules have the form:

For the mission control:

$$R^i_{cm} \quad if \quad cm_a = LA_i,\ldots, and \quad cm_b = LB_i \quad then \quad cm_c = LC \quad (7)$$

For the navigation control:

$$R^j_{cn} \quad if \quad cn_a = LD_j,\ldots, and \quad cn_b = LE_j \quad then \quad y_j = f_j(y,u) \quad (8)$$

where $cm_a,\ldots cm_b$, cm_c, cn_a and cn_b are linguistic variables representing the process state variables and the control variable; $La_i,\ldots,Lb_i$, Lc_i, LD_j and LE_j are the linguistic values of the linguistic variables $cm_a,\ldots cm_b$, cm_c, cn_a and cn_b in the universes of discourses $U,\ldots, V$, W, Q and G respectively, $i=1,2,\ldots,n$.

A conjunction of input variables associated with their respective linguistic values determines a linguistic value associated with the output variable. All rules are evaluated in parallel, and their outputs are combined to a fuzzy set, which has to be defuzzified to receive the crisp output value. The conjunction of the inputs is achieved by the *min*-operation, and for aggregating the outputs of the rules, the *max*-operation is utilised. Function y is used for the estimation of the parameters of the fuzzy-neuro controller system.

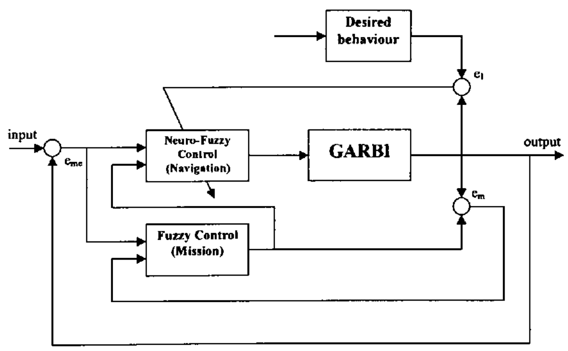


Figure 3. Motion Control System

There exist three kinds of error signals in the proposed system. One of them is the difference between the desired response and the process output.

This error e_l, called the learning error, is used to learn correct control action u and will tend to zero in the time interval of interest with increasing iteration number. The other error e_m, called the mission error, is the error between the mission requirements and the process statement. The last error e_{me}, called the measured control error and defined as the difference between the set-point and the process output, is primarily employed to construct the rule-base which will subsequently be used in the fuzzy control system.

4. Conclusion

In this paper the development of the fuzzy model and fuzzy neural controller to investigate the motion control for the GARBI underwater robot has been described. The next phase of the work is to undertake an extensive simulation study in order to evaluate the motion control system prior to undertaking trials at sea.

Acknowledgement

The authors wish to acknowledge the support for this work provided by the British Council under the British/Spanish Acciones Integradas programme

References

[1] Amat, J.,Batlle, J.,Casals, A., & Forest, J. 1996. GARBI: The Low Cost ROV, Constraints and Solutions.

[2] --- 1995. *GARBI: The UPC Low Underwater Vehicle.* Undersea Robotics & Intelligent Control, U.S/Portugal Workshop. pp. 19-24.

[3] Amat, J.,Codina, J.,Cufi, X., & Puigmal, J. 1995. *Vision Based Control of Underwater Vehicle for Long Duration Observations.* ICAR' 95. pp. 273-277,

[4] Brown, M., & Haris, C. J. 1992. Least mean square learning in associative memory networks. *IEEE Symp. on Intelligent. Control.*

[5] Jang, J. & Sun, T. 1995. Neuro-Fuzzy Modelling & Control. *Proc. IEEE,* **83**, pp. 378-405.

[6] Jang, J-S. R. 1993. ANFIS: Adaptive – Network - Based Fuzzy Inference System. *IEEE Trans. on Sys, Man & Cyb,* **23** (3), pp. 665-685.

[7] Kosko, B. 1992. *Neural Networks and Fuzzy Systems, A Dynamic Systems Approach to Machine Intelligence.* Prentice Hall, Inc.

[8] Lee, C. C. 1990a. Fuzzy Logic in Control systems: Fuzzy Logic Controller Part I. *IEEE Trans on Sys, Man & Cyber,* **20**, pp. 404-418.

[9] Mamdani, E. H. 1974. Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. *Proc. of the IEE,* **121**, pp. 1585-1588.

[10] Nauck, D., & Kruse, R. 1992. A Neural Fuzzy Controller Learning by Fuzzy Error Propagation. *Proc. NAFIPS'92,* pp. 388-397.

[11] Nie, J. & Linkens, D. 1995. *Fuzzy-Neural Control Prin, Algorithms & Applic.* Pren Hall.

[12] Palman, R.,Driankov, D., & Hellendoom, H. 1997. *Model Based Fuzzy Control.* Germany:

[13] Patterson, D. W. 1996. *Artificial Neural Networks Theory and Applications.* Prent. Hall.

[14] Sutton, R.,Taylor, H., & Roberts, G. N. 1997. Tuning Fuzzy Ship Autopilots using Artificial Neural Networks. *InstMC Transactions, Measurement and Control,* **19** (2), pp. 94-106.

[15] Takagi, H. 1990. Fusion Technology of Fuzzy Theory and Neural Networks - Survey and Future directions. *In P. of the Intr. Conf. on Fuzzy Logic and Neural Networks,* pp. 13-26.

[16] Takagi, T., & Sugeno, M. 1985. Fuzzy Ident of Sys & its Applic to Modelling & Control. *IEEE Trans. on Systs, Man and Cyb,* **15**, pp. 116-132.

[17] Terano, T.,Asai, K., & Sugeno, M. 1992. *Fuzzy Systems Theory & its Applic.* Academic Press.

[18] Yamaguchi, T.,Takagi, T., & Mita, T. 1992. Self Organising Control using Fuzzy Neural Networks. *Int. J. Cont,* **56**, pp. 415-439.

[19] Zadeh, L. A. 1968. Fuzzy Algorithms. *Information and Control,* **12**, pp. 94-102.

[20] Zimmermann, H. J. 1996. *Fuzzy Set Theory and its Applications.* Kluwer Acad Publication.