

**A NEW INFERENCE METHOD FOR MULTIVARIABLE
FUZZY SYSTEMS WITH APPLICATION IN
INDUSTRIAL CONTROL**

By

Roya Rahbari

B.A.Sc. K.N.Toosi University of Technology, Tehran, Iran

M.A.Sc. Middle East Technical University, Ankara, Turkey

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES
MECHANICAL ENGINEERING

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

April 2001

© Roya Rahbari, 2001

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purpose may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mechanical Engineering

The University of British Columbia

2324 Main Mall

Vancouver, Canada

V6T 1Z4

DATE:

May 14, 2001
✓

Abstract

Fuzzy logic has been proven to be a useful technique for information fusion in intelligent (or expert) systems as well as both direct control and supervisory control of industrial processes. In this technique, knowledge or expertise of humans are formulated in the form of *if-then* rules. Usually a human can correlate two to three input variables (in the *if* part) to an output variable (in the *then* part). However, complex practical systems are typically multi-input-multi-output (MIMO) systems where there are many input variables that relate to one or more output variables. In such a system, it would be rather unrealistic for a human to keep track of a large number of interacting variables and to express the operational expertise as a set of coupled rules consisting of all input variables and output variables. This thesis develops a new inference method to address the decision making of multivariable fuzzy systems. The developed inference method is based on the concepts of fuzzy measure and the fuzzy integral.

The method first separates a coupled rulebase into several single-input-single-output (SISO) modules or, dual-input-single-output (DISO) modules. Then, it assigns a degree of importance to these SISO modules. Finally the inferences of these modules are combined (fused) to obtain an overall inference, through the use of a fuzzy integral. The technique was originally motivated by the requirements of an industrial fish cutting machine that has been developed in the Industrial Automation Laboratory, University of British Columbia. This inference method can be embedded in any commercial expert system, any direct fuzzy logic controller or a supervisory fuzzy control system that deals with multiple fuzzy variables in a coupled rulebase. The validity of this technique is illustrated by implementing it on hydraulic manipulator of the prototype fish cutting machine and through thorough experimentation. Two inference methods--individual -- rulebased method and composition-based method, are implemented on a direct proportional (P)-type fuzzy controller in real time on the prototype hydraulic manipulator.

A proportional-plus-derivative (PD)-type fuzzy controller is developed and implemented in real time on the prototype hydraulic manipulator. These various implementations are used to carry out an extensive experimental investigation and study the significance of selecting and tuning of such components as the membership functions, rulebase, inference method, and the defuzzification technique in practical fuzzy logic systems.

A supervisory fuzzy control system is developed and implemented integral with an intelligent fuzzy tuner for the hydraulic manipulators of the prototype fish cutting machine. This control system monitors the performance of the machine and automatically tunes the parameters of the PD controllers of the hydraulic actuators. The control system has a three-layer hierarchical structure. The middle layer monitors the performance parameters of the PD-controllers, based on a step-input response. The top layer infers the tuning actions for the PD servo units. The knowledge base for tuning the low level controller is developed using expert knowledge and represented as fuzzy rulebase modules. The investigation presented in this thesis makes important contributions for design, development and implementation of fuzzy knowledge-based systems, particularly control systems, in practical applications. In particular, the present work concentrates on inference techniques, fuzzy rulebase, membership functions, and defuzzification in these systems.

Table of Contents

Abstract	ii
List of Tables.....	vii
List of Figures	viii
Acknowledgement.....	xi
1. Introduction	1
1.1 Research Objectives.....	2
1.2 Motivation.....	2
1.3 Thesis Outline	7
2. Experimental System	8
2.1 Novel Prototype Machine for Fish Cutting.....	8
2.2 Controller Considerations	9
2.3 The Hydraulic Manipulator.....	11
2.4 Modeling of the Electrohydraulic System	13
3. PD-Type Fuzzy Control on a Hydraulic Manipulator.....	16
3.1 Introduction.....	16
3.2 PD-like Fuzzy Logic Controller.....	16
3.3 Choice of Membership Functions	20
3.4 Effect of Tuning the Membership Functions	23
3.4.1 Tuning the membership functions of output.....	24
3.4.2 Tuning the membership functions of input 1.....	27
3.4.3 Tuning the membership functions of input 2.....	27
3.5. Effect of Tuning the Rulebase	29
3.6. Comparison of Defuzzification Methods.....	32
4. P-Type Fuzzy Control on a Hydraulic Manipulator	36
4.1 P-Type Fuzzy Logic Controller	36
4.2 Effect of Tuning the Input Membership Functions.....	38
4.3 Effect of Tuning the Output Membership Functions	44
4.4 Effect of Rulebase Tuning	47
5. Application of Composition Based Inference to P-Type Fuzzy Control	51

5.1 Inference Based on Rule Composition	51
5.2 Discussion	65
5.2.1 Effect of number of discretized points of input membership functions	66
5.2.2 Effect of number of discretized points of output membership functions	66
5.2.3 Effect of inference method	67
6. Development of an Intelligent Hierarchical Control System for a Hydraulic Manipulator	69
6.1 Introduction	69
6.2 Development of the Hierarchical Intelligent Control System	71
6.3 Experimental Results	78
6.4 Discussion	80
7. A New Inference Method Based on Fuzzy Integral	81
7.1 Introduction	81
7.1.1 Purpose of the development	82
7.1.2 Motivation	82
7.1.3 Rationale	82
7.1.4 Related literature	83
7.2 Theoretical Background	84
7.2.1 Fuzzy measure	85
7.2.2 λ - Fuzzy measure	85
7.2.3 Fuzzy integral	87
7.2.4 Information fusion using “fuzzy integral”	88
7.3 Development of the inference technique	95
7.3.1 Universal approximation theorem	104
7.3.2 Stone-Weirstrass theorem	105
7.4. Experimental Investigation	106
7.5 Discussion	110
8. Conclusion	114
8.1 Major Contributions	114
8.2 Significance of the Research	115
8.3 Directions for Further Research	116
Bibliography	117

Appendices	125
Appendix A	125
Appendix B	137

List of Tables

2.1. Physical parameters of the hydraulic manipulator (one of two axes).	13
3.1. The rule base of a PD-like fuzzy knowledge-based controller in tabular form.	19
3.2. Rulebase in Case 5.	31
3.3. Rulebase in Case 6.	32
6.1. Rulebase for the roughness of the response.	75
7.1 Calculation of the values of g^i	92
7.2. Computation of the values of the fuzzy integral for class t_1	93
7.3. Computation of the values of the fuzzy integral for class t_2	94
7.4. Memberships of the outputs for case 1.....	100
7.5. Comparison of the methods of computing inference in case 1	101
7.6. Memberships of the outputs for case 2.....	103
7.7. Computation of inference in case 2.....	103
7.8. Computation of the conventional inference in case 2	104
7.9. Output membership functions	109

List of Figures

Figure.1.1.a The automated Iron Butcher (research prototype).....	3
Figure.1. 1.b The cutter assembly of the research machine.....	4
Figure.1.1.c A schematic diagram of the cutter assembly.	4
Figure.1.2 Hierarchical monitoring and supervisory control system of the prototype machine....	6
Figure 2.1.The fish cutting machine (industrial prototype) in the Industrial Automation Laboratory.....	10
Figure 2.2.The observer for velocity estimation.	11
Figure 2.3. The electrohydraulic positioning system (for one axis).	12
Figure 3. 1. Block diagram of the fuzzy logic controller.....	18
Figure 3.2. Membership functions for case1.....	21
Figure 3.3. Membership functions for case2.....	22
Figure 3.4. Membership functions for case3.....	22
Figure 3.5. Membership functions for case4.....	23
Figure 3.6. Effect of tuning the membership functions of output (the position response).	25
Figure 3.7. Effect of tuning the membership functions of output (the control action).	26
Figure 3.8. Effect of tuning the membership functions of input1.....	28
Figure 3.9. Effect of tuning the membership functions of input2.....	29
Figure 3.10. Effect of tuning the rulebase.....	30
Figure 3.11. Membership functions of case 5 and 6.	31
Figure 3.12. Response of the system with centre of gravity defuzzification method.	33
Figure 3.13. Response of the system with centre of average defuzzification method.....	35
Figure 4.1 Block diagram of the fuzzy logic controller.....	37
Figure 4.2 Inference based on individual rules for the P-type controller.	38
Figure 4.3 Membership functions of the input.....	39
Figure 4.4 Membership functions of the output.....	40
Figure 4.5 Rulebase of the P-type fuzzy controller.	40

Figure 4.6 Comparison of the responses of the system in the four cases, with the same initial conditions.....	41
Figure 4.7 Response of the system in case 1.....	41
Figure 4.8 Response of the system in case 2.....	42
Figure 4.9 Response of the system in case 3.....	42
Figure 4.10 Response of the system in case 4.....	43
Figure 4.11 Membership functions of input in three cases.....	43
Figure 4.12 Membership functions of output in three cases.....	44
Figure 4.13 The rulebase of the system in the three cases.....	44
Figure 4.14 Response of the system in case1.....	45
Figure 4.15 Response of the system in case 2.....	46
Figure 4.16 Response of the system in case 3.....	46
Figure 4.17 Rulebase in case 1.....	47
Figure 4.18 Rulebase in case 2.....	47
Figure 4.19 Rulebase in case 3.....	48
Figure 4.20 Membership functions of input and output in all three cases.....	48
Figure 4.21 Rulebase in case1.....	49
Figure 4.22 Rulebase in case2.....	49
Figure 4.23 Response of the system in case3.....	50
Figure 5.1 Discretized membership functions of the input.....	58
Figure 5.2 Discretized membership functions of the output.....	58
Figure 5.3 Membership functions of the input.....	59
Figure 5.4 Rulebase of the fuzzy system.....	60
Figure 5.5 The rulebase in the form of fuzzy relation.....	60
Figure 5.6 Transformation of crisp input signal to fuzzy singleton.....	61
Figure 5.7 Inference engine of the fuzzy system.....	61
Figure 5.8 Defuzzification for the fuzzy system.....	62
Figure 5.9 Response of the system in the Case $N=7$ and $M=9$	63
Figure 5.10 Response of the system in the Case $N=13$ and $M=9$	63
Figure 5.11 Response of the system in the Case $N=7$ and $M=17$	64
Figure 5.12 Response of the system in the Case $N=13$ and $M=17$	64

Figure 6.1 Schematic representation of the intelligent hierarchical controller.	71
Figure 6.2 The observer for velocity estimation.	72
Figure 6.3 The flow chart of the algorithm in the second layer.	73
Figure 6.4 The response of system with different overshoots.	74
Figure 6.5 The response of system with different degrees of oscillations.	75
Figure 6.6 Examples of different speeds of response of the system.	76
Figure 6.7 Knowledgebase of the top level of the hierarchy.	77
Figure 6.8 Membership functions in the knowledgebase.	78
Figure 6.9.a Experimental results (overshoot improvement due to tuning).....	79
Figure 6.9.b Experimental results (improved oscillations due to tuning).....	79
Figure 6.9.c Experimental results (improved speed of response due to tuning).....	80
Figure 7.1. The decomposition of the MIMO rulebase into two SISO modules.	96
Figure.7.2.(a). Rulebase modules for case 1	99
Figure.7.2.(b). Schematic representation of inference in case 1.	100
Figure.7.2.(c). Rulebase modules for case 2	101
Figure 7.2.(d) . Schematic representation of the inference in case 2	102
Figure 7.3. Block diagram of the fuzzy logic controller.	106
Figure 7.4. Membership functions of input and output variables	107
Figure 7.5. Modules of the rulebase in three cases of investigation	108
Figure 7.6. Module 1 and module 2a (case1).....	111
Figure 7.7. Module 1 and module 2b (case2).	112
Figure 7.8. Module 1 and module 2c (case3).....	112
Figure 7.9. Comparison of the responses ($g_1=1$, $g_2=0$).....	113

Acknowledgements

I wish to express my sincere gratitude to my supervisor, Prof. Clarence W. de Silva, for his support, guidance and invaluable patience. His wisdom, patience and proactive attitude toward new ideas, as well as his graceful and professional manner have made him a true role model for his graduate students as well as a leader in technology.

The Funding for my research was provided by supervisor through Natural Sciences and Engineering Research Council of Canada, B.C. Packers, Ltd., and Gartfield Weston Foundation for the Chair of Industrial Automation, which is gratefully appreciated. I would like to thank members of my supervising committee, Prof. Sander Calisal and Prof. Farokh Sassani for their kind and careful attention to the progress of my research. I would also appreciate Prof. Elizabeth Croft as well as the members in Industrial Automation Laboratory for their cooperation. I gratefully acknowledge the strong support of the Faculty of Applied Science, as well as Department of Mechanical Engineering, particularly the heads Prof. Bob Evans, Prof. Dale Cherchas and Prof. Nimal Rajapakse, in addition to my supervisor, Prof. Clarence de Silva, for their true and strong support and encouragement for advancement of women in engineering.

I wish to express my special gratitude to Prof. Ian Gartshore for his true and sincere encouragement of all graduate students. His wisdom and vision have made him a true professional and role model for maintaining peace and integrity in the department.

The kind assistance of Mr. Alan Steeves, Mr. Gerry Rohling, and Mr. Albert Sawchuk to all graduate students is greatly appreciated. Mr. Jon Mikkelsen and Mr. Gordon Wright with their wonderful communication skills have always been very helpful to all teaching assistants and graduate students, this is greatly appreciated as well. I greatly appreciate Mr. Sean Bygrave and Mr. Doug Yuen, for their kind assistance everytime and anytime in the shop facilities. Cooperation of the office staff, Ms. Christine Black, Ms. Barb Murray, Ms. Jan Marsden and Ms. Louise Soga is greatly appreciated.

I thank god who gave me power to overcome the barriers to reach my goal. Last, and definitely not least, I wish to thank my loving, understanding, and cultured parents who have always supported and encouraged me to reach my goals.

*To my parents
who gave me wings to fly,
held me when I fell,
&
their love is the greatest gift of all.*

*To my supervisor, Prof. Clarence W. de Silva
The gift from heaven*

And

To Anxious Patience

Chapter 1

Introduction

Fuzzy sets were first introduced by Zadeh [6,10,92-104]. Following that Mamdani used the concept of fuzzy logic for applications in knowledge-based control [42,51,52,61]. Since then, numerous applications of fuzzy logic have been successfully implemented in such areas as control, image processing, object recognition, and automation. Fuzzy logic has been demonstrated to be useful in information fusion in intelligent systems [4,77], expert systems used in process control [51,52,63], telecommunication networks [73], and many other applications. Expert systems have been developed for a wide range of applications [1,2,40,47,49,63,64,69,71]. In the context of the present research, an important application of fuzzy logic is in expert systems. The name *expert system* is used interchangeably with the term *knowledge-based system*. An expert system is a computer program which embodies human knowledge and understanding in a way that imitates a human expert who can solve specific types of problems [54]. Fuzzy logic is used to capture expertise of human experts and represent in the form of a rulebase of fuzzy variables. Another application area of fuzzy logic is in information fusion. Sugeno, first developed the concept of a *fuzzy integral* based on fuzzy logic and fuzzy measures [72]. Following that fuzzy logic has been used in information processing based on the concept of the fuzzy integral [77]. In these cases relevant knowledge or expertise of humans in the form of linguistic IF-THEN rules needs to be developed. Commercial expert system shells are available, which make the process of development of a rulebase easier. Specifically, the knowledge engineer would extract knowledge of humans (or, domain experts) and represent it as appropriate IF-THEN fuzzy rules. The knowledge includes heuristic and rule-of-thumb information. The expert system processes these IF-THEN rules using a built in inference method.

1.1 Research Objectives

The main objective of the research reported in this thesis is to develop an efficient method for handling multiple fuzzy variables in a rulebase for the purpose of information fusion and decision making. The technique will be particularly suitable for systems that use fuzzy logic for information representation (fusion) and combination. Hence, the research will aim at developing a practical inference method for handling fuzzy systems with multiple variables. A related objective is to implement the technique in an industrial prototype machine for fish cutting and to evaluate the performance of the method through experimentation.

1.2 Motivation

This research is motivated by a novel and automated fish-cutting machine which Industrial Automation Laboratory is developing. The machine that is commonly used in the fish processing industry for head removal of fish is known as the “Iron Butcher”. This machine has been designed and built at the turn of the century and has seen little modification or improvement since then. It is a purely mechanical device, which employs friction to detect the gill of a fish and a tactile engagement mechanism to align the collarbone of the fish with the cutter blade. Due to lack of active sensing and feedback control, this simple mechanism is known to cause considerable wastage by frequently over-feeding the fish into the cutter. In the salmon processing industry of British Columbia, Canada, for example, the wastage of useful meat (due to the poor cutting accuracy of the Iron Butcher) is estimated to be 25 million dollars annually, which corresponds to a wastage of 5% of the useful fish meat. Also, the machine causes under-feed errors, caused when the mechanical positioning foot slides over the collarbone of the fish. In this case the processed fish has to be reprocessed, resulting in a reduced throughput rate.

The Industrial Automation Laboratory has undertaken research aimed at addressing deficiencies of the Iron Butcher by employing modern sensing and actuation

technologies to detect and position the fish accurately at the cutter in order to minimize the wastage [17]. In particular, two automated machines have been developed for head removal of fish [20,21,22,84]. The research prototype as shown in Figure 1.1 represents an attempt toward developing an autonomous and intelligent machine that is able to remove the head of a fish with minimal wastage. The machine has been designed to operate with minimal intervention of human operators, by taking advantage of robotic technology and knowledge based intelligent systems. This machine would be able to learn and reason based on information that is received from various sensors and actuators, which is then processed and analyzed to determine the performance, similar to the way that a human does and then take control actions automatically in order to improve the performance of the system automatically, without intervention of human operators.

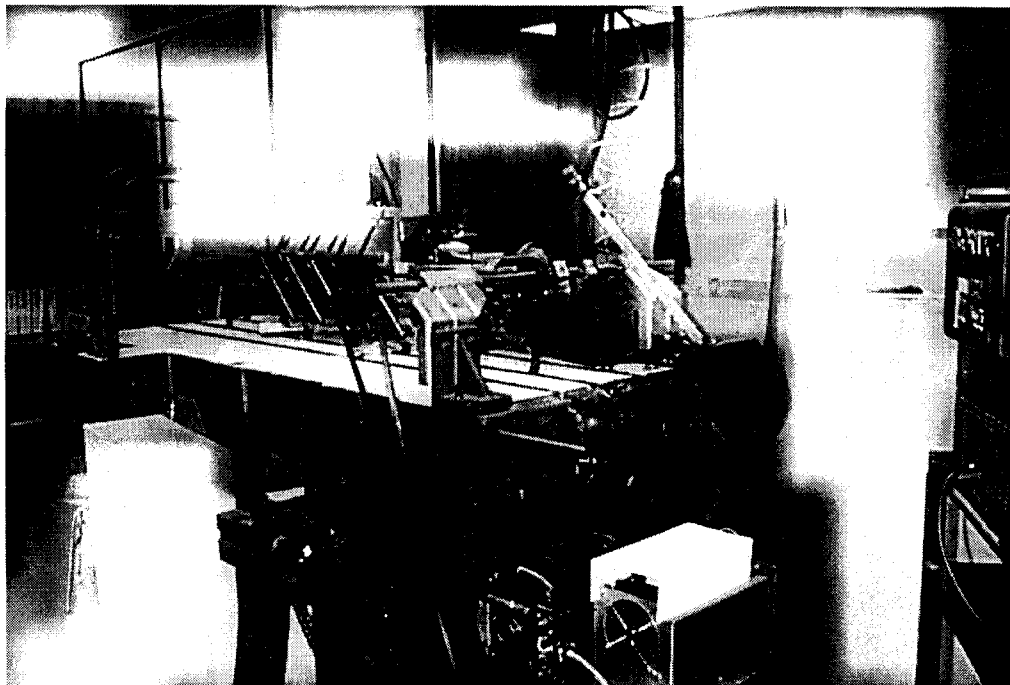


Figure.1.1(a) The automated Iron Butcher (research prototype).

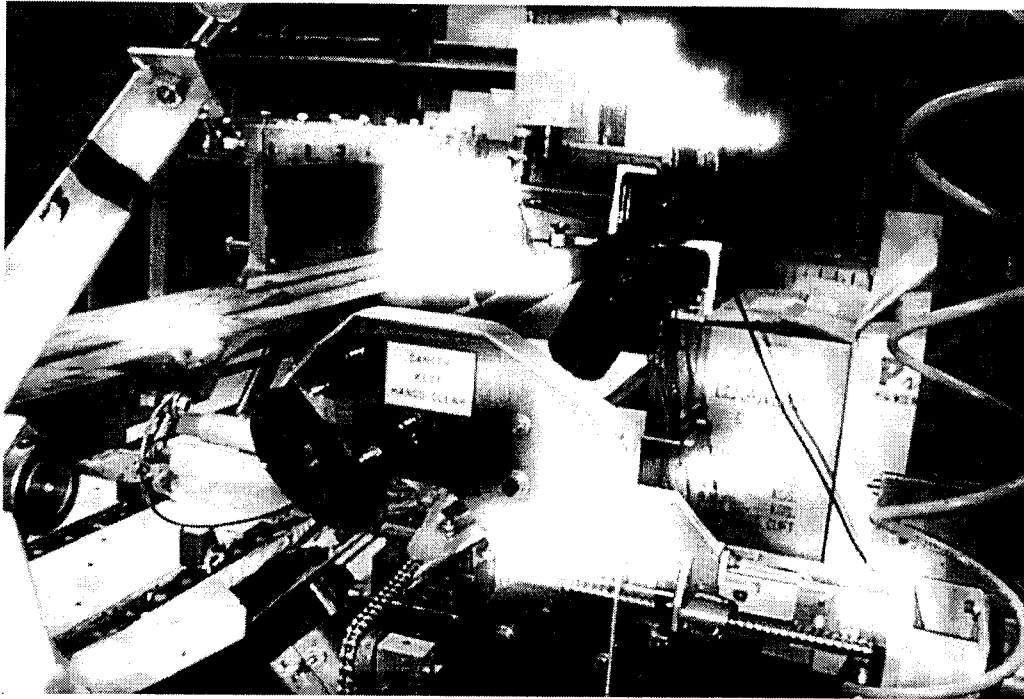


Figure.1. 1(b) The cutter assembly of the research machine.

Figure 1.1(a) and Figure 1.1(b) show two views of the developed research prototype. Figure 1.1(c) shows a schematic diagram of the cutter assembly of the machine.

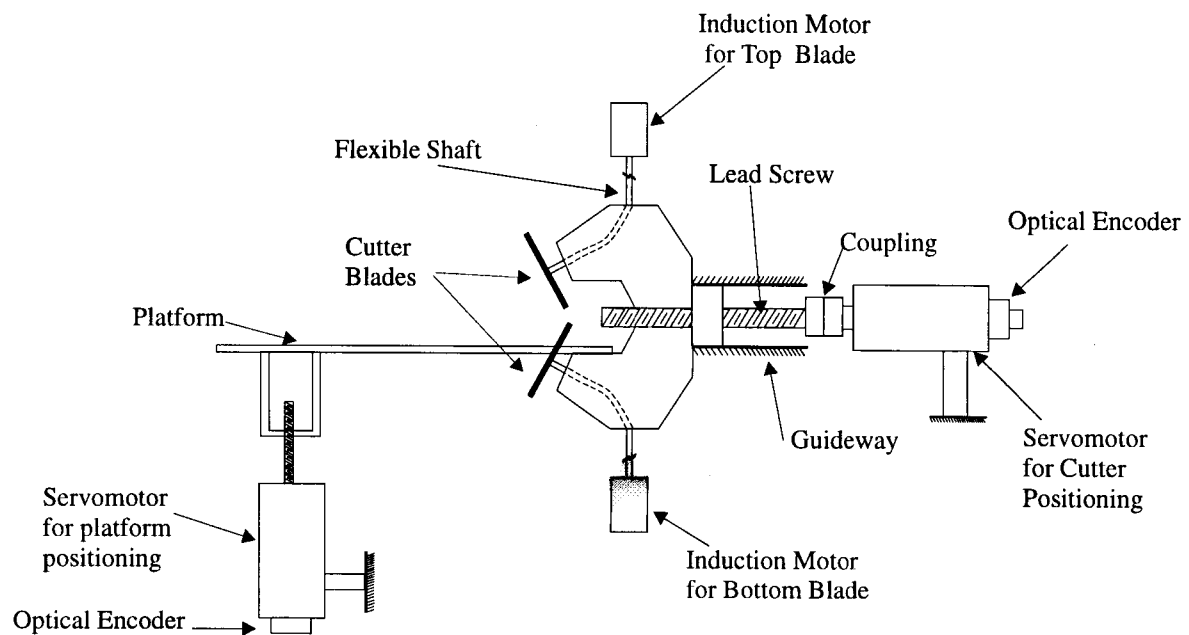


Figure.1.1(c) A schematic diagram of the cutter assembly.

The conveyor moves the fish continuously from the feeding-end to the cutter. The fish are separated by means of the push lugs which are attached to a set of chains around the conveyor. The conveyor has a linear speed of 2 lugs per revolution of the drive shaft, while the distance between two adjacent lugs is 23 cm. The conveyor shaft is driven by a DC motor through a belt drive. For a typical throughput rate of 2 fish/s, the linear speed of the conveyor is 46 cm/s. By the time a fish reaches the cutting zone, the cutter assembly has to be moved to the correct cutting location horizontally. Simultaneously, the feeding platform is adjusted in the vertical direction so as to present the fish to the cutter. The horizontal adjustment is made in order to align the collarbone of the fish with the cutter blades. The vertical adjustment of the platform is necessary so that each fish is delivered symmetrically into the cutter, while compensating for the thickness of the fish in order to minimize the meat wastage. After the fish leaves the cutting zone, it enters the secondary vision system which analyzes the image of the cut fish, in order to determine the quality of the cut. This will provide high level information for the purpose of monitoring the system performance and for machine control. The machine consists of the following components:

1. A conveyor, which carries fish from the feeding point to the cutting location.
2. A hold down mechanism, which is necessary to make sure that the fish are held properly throughout the processes of imaging and cutting.
3. An ultrasonic displacement sensor, which measures the thickness of the fish near the area of the collarbone. This measurement is needed to control the positioning platform in the vertical direction so that each fish is delivered properly into the cutter blades.
4. A primary CCD camera, which grabs the image of each fish on the conveyor, when triggered by an object-detection sensor on the conveyor table. The low-level vision system analyzes the image and determines the cutting locations [31,65].
5. Cutter assembly, which consists of two circular blades arranged in a V-configuration and driven by two independent high-speed induction motors through flexible

- coupling. Motion of the cutter assembly in the horizontal x direction is digitally controlled and actuated by a DC servo motor.
6. An adjustable platform adjacent to the cutter assembly, which places each fish symmetrically with respect to the pair of cutter blades. The position of this platform is adjusted by a DC servomotor, which is digitally controlled.
 7. A secondary CCD camera at the exit end of the cutting zone, which provides images to a high-level vision system, in order to determine visual information related to the quality of the processed fish and in turn the performance of the machine.

This prototype machine is a complex system, which consists of many sensors, actuators, controllers, and mechanical devices. The overall performance of the machine depends on improving the meat recovery and the quality of the cut fish, which are subjective measures. Autonomous operation of the machine relies on a high level control architecture. It needs to be designed so as to process the complex, low-level information from various sensors and actuators, and properly combine them to make a proper decision for product quality assessment and for improvement of performance of the system. Figure 1.2 shows the hierarchical architecture of monitoring and control of the research prototype machine.

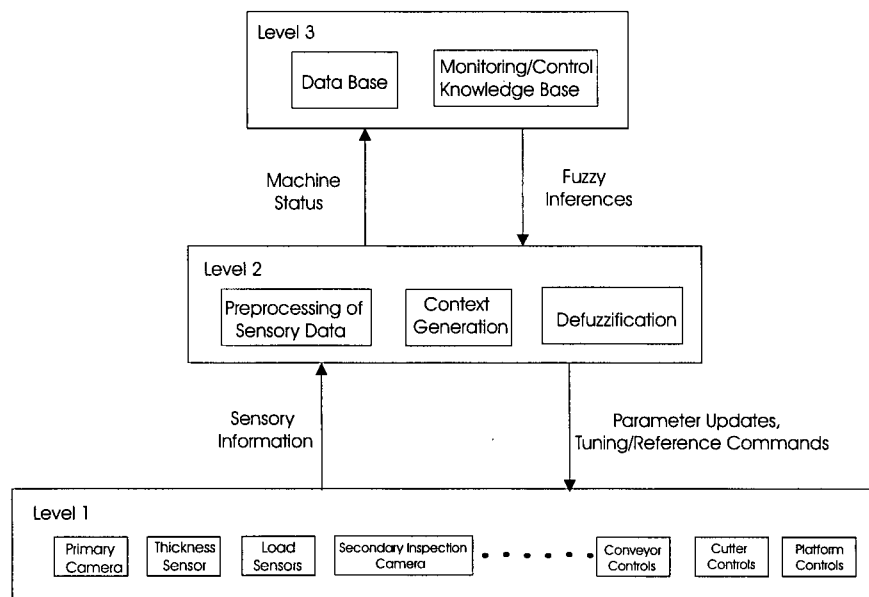


Figure 1.2 Hierarchical monitoring and supervisory control system of the prototype machine.

Since the automated machine consists of many components (sensors, actuators controllers, signal conditioners, etc.) which make it necessary to deal with large amounts of information, a suitable structure for handling the information from various sources is required to be developed.

1.3 Thesis Outline

As a precursor to the development of the new inference scheme for multivariable fuzzy systems, this thesis first studies the real time implementation of different methods of fuzzy logic control; specifically the proportional plus derivative type (PD-type) and proportional type(P-type). A comparison of two practical inference methods is given with experimentation results on an industrial hydraulic manipulator. This foundation of development, implementation, and experimentation with conventional fuzzy decision making and control, using an industrial prototype machine for fish cutting, leads to a new inference method. Chapter 2 explains the industrial hydraulic manipulator (industrial fish cutting machine), which is used for the implementation of the direct low level fuzzy controllers. Chapter 3 describes the development of a PD-type fuzzy controller, which is implemented on the experimental system. Chapter 4 presents the development of a P-type fuzzy controller using “individual rulebase firing”. Chapter 5 describes the development of a P-type fuzzy controller using “composition based inference”. Chapter 6 gives the development and implementation of fuzzy logic control at high level in the form of an intelligent hierarchical control system for tuning the parameters of a conventional PD type controller, which has been implemented in the experimental machine. Chapter 7 presents the novel inference method that is developed, which is an original contribution of the thesis. Chapter 8 concludes the thesis by summarizing the contributions of this thesis, indicating their significance, and providing directions for future research.

Chapter 2

Experimental System

The algorithms and control schemes developed and/or investigated in this research are implemented in the hydraulic manipulator of an industrial prototype of an automated fish cutting machine which has been developed in the Industrial Automation Laboratory. This chapter explains this prototype machine which is used as the test bed for this research, particularly for the direct fuzzy controllers developed and investigated in this research. This is the second generation industrial prototype, which is an outgrowth of the original research prototype outlined in Chapter 1.

2.1 Novel Prototype Machine for Fish Cutting

The Industrial Automation Laboratory of the University of British Columbia has undertaken research aimed at addressing the main deficiencies of the Iron Butcher. The Laboratory has been established primarily for research and development of advanced technology in automation of the fish processing industry [17]. The industrial prototype machine that is used in the present work is an outgrowth of an activity of the laboratory, which first developed the research prototype outlined in Chapter 1. Now the industrial prototype which is the test bed for the present research, is outlined. The machine consists of several parts:

1. An intermittent-motion *conveyor* system to move the fish towards the cutting zone.
2. A *guillotine type blade* that operates by a *pneumatic actuator* in the vertical direction.
3. *Horizontal X-Y table* that carries the cutting blade assembly, which is driven by means of *two hydraulic actuators*.
4. A *digital CCD camera* that takes the image of each fish as it enters the cutting zone.

A view of the prototype machine is shown in Figure 2.1. The conveying mechanism has several parallel arms which are equally spaced along the belt. The fish are pushed

forward intermittently, by means of these arms. Also, there is a matrix of retaining pins which point vertically upwards. These pins can fold in one direction only, thereby restraining any backward motion of the fish. The conveyor pushes a fish forward every 1.24s, which corresponds to the period of the cyclic motion of the mechanism. Accordingly, the rate of processing is about 48 fish/minute. During the first half period the conveyor pushes the fish forward, and in the next half period while the fish are stationary, the imaging of one fish and cutting of the previous fish are carried out. Also during this half period, the conveying arms move backward in order to get ready to push the fish forward in the next cycle. There is an optimum position at which the blade should be located with respect to a fish so that the meat wastage would be minimum. This optimum position is determined by using a computer vision system. While a fish is stationary an image of the fish is taken by the CCD camera and the corresponding digital information is acquired by a frame grabber board of the vision computer. Image processing routines automatically determine the optimum position of the cutter, according to the gill position of the fish [31,65]. This point is used as the reference input (set point) for the hydraulic controller, which moves the cutting blade assembly. After the cutter assembly is moved to the desired position (optimum point), the pneumatic actuator is activated and as a result, the cutter blade is released vertically. The operations of the system are synchronized such that, by the time a fish arrives at the cutting zone, the blade has already been positioned at the desired position.

2.2 Controller Considerations

As mentioned earlier, the period of cyclic motion of the conveyor is 1.24 seconds. This means that image capture of a fish by the computer vision system and the cutting of the previous fish are carried out during the return half cycle of 620ms of the conveyor. Image processing and determination of the desired cutting location is carried out soon after an image is captured and it should be completed during the return half cycle. Positioning of the cutting blade assembly to the desired position needs to be completed during the forward half cycle of 620ms. This means that the hydraulic positioning system has to reach the steady state at the end of the forward half cycle. If the controller fails to reach the steady state by this time, the blade will be released while the cutting blade unit

is still moving, but the fish would be in the stationary position. This may lead to such problems as poor-quality cuts or even complete entanglement of a fish with the cutter. In view of this, it is crucial that the cutting blade unit reaches the set point in a stable manner at the end of the forward half cycle of the conveyor motion. The time required for image processing with the available hardware system is 170ms, which is well within the available time of 620ms for image capture, processing, and the generation of the set point for the cutter position. Note that the cutter cannot be positioned for the next fish before the present fish is cut. The cutting time for one fish using the pneumatic actuator is limited to about 240ms, which is more than the time needed for image capture and set point determination for the next fish. Then a time duration of about 1.0 second would be available for positioning the cutter. Accordingly, the following specifications are used for the low-level controller:

1. Reach the desired position in steady state within 1.0 s.
2. Allowed offset for the cutter position (cutting accuracy) is 5mm.

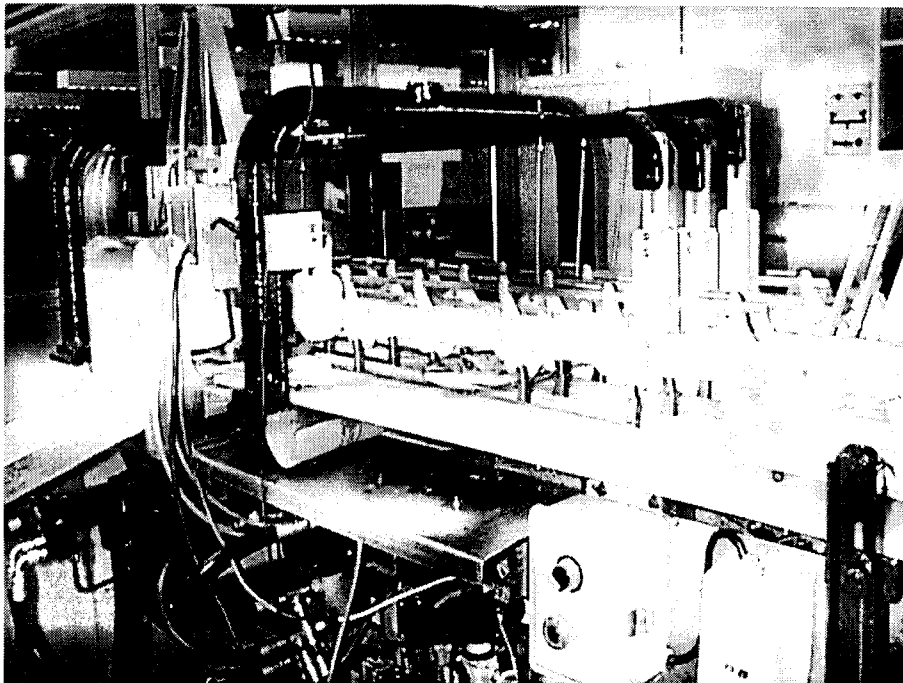


Figure 2.1. The fish cutting machine (industrial prototype) in the Industrial Automation Laboratory.

2.3 The Hydraulic Manipulator

In the prototype system, the manipulator (X-Y table) which positions the cutter blade has two control inputs, which are the currents applied to the electrohydraulic proportional valves corresponding to the X and Y directions of motion. Since the two degrees of freedom in the manipulator are quite similar and also independent, only the X direction is discussed here. Figure 2.2 shows a simplified schematic diagram of the manipulator, for the X axis. The servovalve consists of a torque motor, a flapper pilot stage, and a double sliding spool boost stage. It is balanced by the centering springs which produce a spool position proportional to the differential pressure and therefore to the input current.

For measuring the position of the piston, a Temposonics linear displacement transducer is placed at the head of the hydraulic cylinder. This industrial magnetostrictive sensor precisely senses the position of an external magnet, which is connected to the piston. The sensor in the prototype machine has a resolution of 0.025 mm . In this research, the actuator velocity is estimated using a linear observer which has been developed in [75,76]. The dynamics of the observer is shown in Eq. (2.1) and the block diagram is shown in Figure 2.2, where $k_v = 500$.

$$\begin{cases} \hat{v} = z_v + k_v x \\ \dot{z}_v = -k_v \hat{v} \end{cases} \quad (2.1)$$

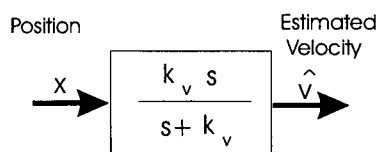


Figure 2. 2. The observer for velocity estimation.

Two gage pressure transducers are also installed on the head and the rod sides of the cylinder, in order to measure the fluid pressures P_1 and P_2 , respectively. Figure 2.3 shows the schematic diagram of setup. A sampling frequency of $f_s = 1$ kHz has been used to implement the digital controllers. A 12 bit PC-based data acquisition system is used for data collection. The computer system that has been used for the experiment setup is an Intel 486, 33 MHz processor with DOS operating system, and the code is written in WATCOM C. The physical parameter values of the prototype manipulator are listed in Table 2.1.

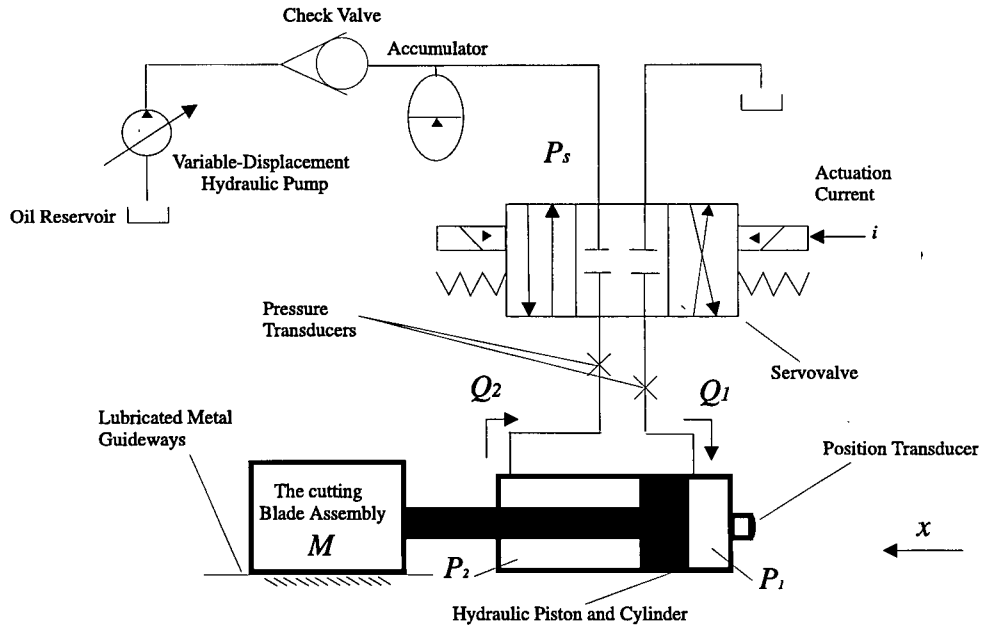


Figure 2.3. The electrohydraulic positioning system (for one axis).

Table 2.1. Physical parameters of the hydraulic manipulator (one of two axes).

Parameter	Value
Measured moving mass M	32.7 Kg
Head-side piston area A_1	$1.14 \times 10^{-3} \text{ m}^2$
Rod-side piston area A_2	$0.633 \times 10^{-3} \text{ m}^2$
Maximum piston displacement L	$50 \times 10^{-3} \text{ m}$
Volume of each hose between the servovalve and the cylinder V_h	$8.9 \times 10^{-5} \text{ m}^3$

2.4 Modeling of the Electrohydraulic System

A dynamic model is derived in this section for the electrohydraulic manipulator [23,29,55]. This analysis is primarily intended to emphasize the nonlinear and asymmetric nature of the actuator dynamics. There are two types of nonlinearities in the servovalve system; namely, the hysteresis resulting from the electromagnetic characteristics of the torque motor and the flow forces on the valve spools. Generally, servovalve dynamics can be ignored in comparison to the actuator dynamics. Displacement of the valve spools from their null position creates a pressure difference across the single-rod hydraulic actuator, and the resulting fluid flow causes a change in the position of the piston. The force applied by the actuator (given in Eq.(2.2)) can be calculated from the cylinder pressures. Newton's second law applied to the actuator yields:

$$F = P_1 A_1 - P_2 A_2 = M\ddot{x} + F_f \quad (2.2)$$

where F is the actuator force, F_f is the opposing frictional force, x is the piston displacement, and M is the moving mass in the X direction. The amount of fluid flow

to the head-side (Q_1) and from the rod-side (Q_2) of the cylinder is a function of the valve spool position and cylinder pressures. Assuming that the valve is critical center type with matched and symmetrical ports, the relationship can be expressed in the following form :

$$\begin{cases} Q_1 = K.x_v.[s(x_v)\sqrt{P_s - P_1} + s(-x_v).\sqrt{P_1}] \\ Q_2 = K.x_v.[s(x_v)\sqrt{P_2} + s(-x_v).\sqrt{P_s - P_2}] \end{cases} \quad (2.3)$$

where x_v is the valve spool displacement, K is a fixed gain, P_s is the supply pressure, and $s(x_v)$ is a switching function defined as in Eq (2.4):

$$s(x_v) = \begin{cases} 1 & x_v \geq 0 \\ 0 & x_v < 0 \end{cases} \quad (2.4)$$

Using the continuity principle, and taking fluid compressibility into account, we can write:

$$\begin{cases} Q_1 = A_1\dot{x} + \frac{A_1x + V_h}{\beta} \dot{P}_1 \\ Q_2 = A_2\dot{x} - \frac{A_2(L - x) + V_h}{\beta} \dot{P}_2 \end{cases} \quad (2.5)$$

where β is the effective bulk modulus of the fluid and V_h is the volume of the fluid inside each of the hoses that connect the servovalve to the actuator. The state vector is defined as:

$$\mathbf{x} \equiv [x_1 \ x_2 \ x_3 \ x_4]^T = [x \ \dot{x} \ P_1 \ P_2]^T \quad (2.6)$$

By combining equations (2.2), (2.3), and (2.5), the following nonlinear state-space model of the system is obtained:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{1}{M} [A_1 x_3 - A_2 x_4 - F_f] \\ \dot{x}_3 = \frac{\beta}{A_1 x_1 + V_h} [kx_v (s(x_v) \sqrt{P_s - x_3} + s(-x_v) \sqrt{x_3}) - A_1 x_2] \\ \dot{x}_4 = \frac{-\beta}{A_2 (L - x_1) + V_h} [kx_v (s(x_v) \sqrt{x_4} + s(-x_v) \sqrt{P_s - x_4}) - A_2 x_2] \end{cases} \quad (2.7)$$

As can be shown in state space dynamics of the system in Eq (2.7), the actuator dynamics depends on position and velocity of the piston. Considering this information, next chapter develops a fuzzy controller based on information received from position and velocity of the cylinder.

Chapter 3

PD-Type Fuzzy Control on a Hydraulic Manipulator

3.1 Introduction

The application of fuzzy logic control in hydraulic manipulators is rare, particularly in the research community. An application of a fuzzy logic controller has been reported in [107] in which a linear state controller combined with a fuzzy rule evaluation is used to produce what has been termed as fuzzy state controller. Their proposed method indirectly detects the presence of a load, based on the fuzzy rule evaluation of actuator velocity and decides on the gains of the state controller accordingly, in order to make the controller insensitive to load variations. Application of fuzzy logic controller to a class of hydraulically actuated industrial robots is investigated in [13,70]. This chapter studies the development and experimental analysis of a proportional plus derivative (PD)-type fuzzy logic controller, as implemented on the planar hydraulic manipulator which was outlined in Chapter 2. The performance of the fuzzy logic controller is investigated with respect to the control requirements for this machine, explained in Section 2.2, and the effect of tuning of the membership functions is illustrated using experimental results. This foundation will provide an insight into the experimental system and the practical implementation of a fuzzy logic controller in such a system. This investigation will lead to the implementation of another controller, as presented in Chapter 4 and 5. The inference techniques used in these controllers have provided the motivation for the original development given in Chapter 7.

3.2 PD-Type Fuzzy Logic Controller

In conventional control, the PD control law is written in the form:

$$u = K_p \cdot e + K_D \cdot \dot{e} \quad (3.1)$$

where e and \dot{e} are the error and the rate of error, respectively, and K_p and K_d are the associated proportional and differential gain coefficients. In fuzzy logic control, the control law is presented as “if-then” statements; for example,

$$\text{if } X \text{ is } A \text{ and } Y \text{ is } B \text{ then } Z \text{ is } C$$

“ X is A ” and “ Y is B ” are called the *rule antecedents* and “ Z is C ” is called the *rule consequent*. In a fuzzy rulebase, X and Y are the inputs to the fuzzy system and Z is the output of the fuzzy system. Then a PD-type fuzzy logic controller consists of fuzzy rules, with the inputs *error* and *rate of error*, and the output *control action*. The symbolic description of each rule can be given as [28]:

$$\begin{aligned} \text{if } e(k) \text{ is } \langle \text{a fuzzy state} \rangle \text{ and } \dot{e}(k) \text{ is } \langle \text{a fuzzy state} \rangle \\ \text{then } u(k) \text{ is } \langle \text{a fuzzy state} \rangle \end{aligned}$$

where $\langle \text{a fuzzy state} \rangle$ is the symbolic name of a fuzzy linguistic value. The natural language equivalent of the above symbolic rule reads as follows:

For each sampling time k ,

if the value of the *error* is $\langle \text{linguistic value} \rangle$
and the value of the *change-of-error* is $\langle \text{linguistic value} \rangle$
then the value of the *control output* is $\langle \text{linguistic value} \rangle$

For position control of the prototype hydraulic actuator by means of a fuzzy controller, the inputs are *position error* and *velocity*, and the output is *current to the servovalve*. First the membership functions have to be designed for these variables. Velocity is estimated by a linear observer, as outlined in Chapter 2 and shown in Figure 2.2. A block diagram of the fuzzy knowledge-based controller (FKBC) that is used in the present application is shown in Figure 3.1.

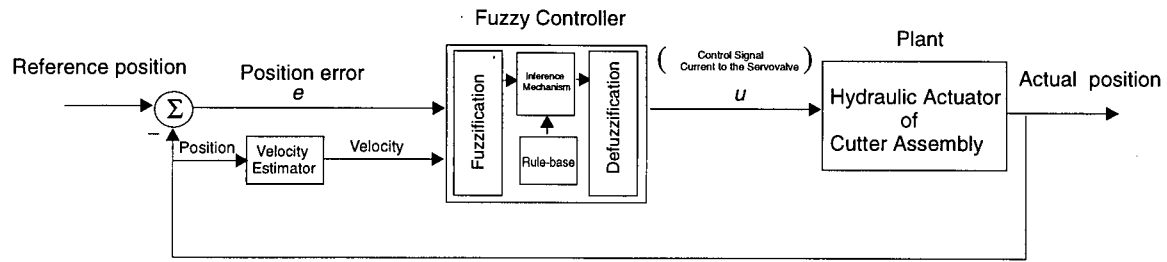


Figure 3. 1. Block diagram of the fuzzy logic controller.

There are several types of inference methods. If we consider the rule:

$$\text{IF } x \text{ is A AND } y \text{ is B THEN } z \text{ is C}$$

There are two operations that are commonly used for “AND” operation; “min” and “algebraic product”. There are many types of implication methods that have been introduced for implication operation “THEN” [78]; min, algebraic product, bounded product, drastic product, Lukasiewicz, Boolean logic, Goedel logic, Goguen. There are two types of operations used for rule aggregation; “max” and “sum”. In the present research “min” is chosen for “AND” operation, “min” is chosen for implication THEN and “max” is chosen for rule aggregation. This type of inference is known as Mamdani’s inference method. However, there are two different inference methods from “mechanism of output generation” point of view; “individual rule-based inference” and “composition based inference” [28]. In “individual rule-based inference” rules are fired individually and then the output of those fired rules are aggregated. This method is explained in Chapter 4, in more details. However, in “composition based inference”, first all the rules are combined into a fuzzy relation, and then the input is composed with the fuzzy relation to compute the output with “composition operation”. This method is explained in Chapter 5 in more details. In the present application, the inference engine of the fuzzy controller uses *individual rule-based inference*, as opposed to *composition based inference*, because of the lower memory requirement of the former in real time processing. The fuzzy sets of the inputs (*position error* and *velocity*) and the output (*current to the servovalve*) all are chosen to be equal in number and contain the same linguistic expressions for the

magnitude part of the linguistic values; i.e., {NL, NM, NS, ZER, PS, PM, PL} where NL denotes negative large, NM denotes negative medium, NS denotes negative small, ZER denotes zero, PS denotes positive small, PM denotes positive medium, and PL denotes positive large. The rule base is shown in Table 3.1 in which $e = x - x_{des}$ is the position error and vel is the estimated velocity.

Table 3.1. The rule base of a PD-like fuzzy knowledge-based controller in tabular form.

	vel						
	NL	NM	NS	ZER	PS	PM	PL
e	NL	PL	PM	PM	PS	PS	ZER
	NM	PM	PM	PS	PS	ZER	ZER
	NS	PM	PS	PS	ZER	ZER	NS
	ZER	PS	PS	PS	ZER	ZER	NS
	PS	PS	PS	ZER	ZER	NS	NM
	PM	PS	ZER	ZER	NS	NS	NM
	PL	ZER	ZER	NS	NS	NM	NL

The rules in the center of the table are related to the steady-state behavior of the process. This means that the current value of the process output variable (position) has deviated from the set-point but is still close to it. When both the position error and the velocity are negative, the position is below the set-point and the cylinder is moving further away. In response the control action should be positive such that it will reduce the position error. When the velocity is positive, while the position error is negative, the cutter is moving towards the set-point and control action should be low enough to slow down the approach to the set-point. Furthermore, note the asymmetry of the rulebase in Table 3.1. This is justified in view of the directional asymmetry of the hydraulic actuator with unequal effective areas on the two sides of the piston, as clear from the formulation in Chapter 2.

3.3 Choice of Membership Functions

For computational effectiveness, efficient use of memory, and the relative ease of performance analysis, a uniform representation of the membership functions would be desirable. Such a uniform representation has been realized in the present application by using membership functions with uniform shape and parametric functional definition. The most popular choice for membership functions is the triangular shape, which can be represented by a parametric functional description. Furthermore, these membership functions can be stored with a minimal use of memory and manipulated efficiently, with respect to real-time requirements, by the inference engine. As can be seen in figures 3.2 through 3.5, there is an overlap between every pair of adjacent membership functions. This means that every crisp value of an input to the controller will belong to at least one membership function, with the degree of membership strictly greater than zero. If this were not the case, then there would be a crisp input value which could not be matched to a rule-antecedent during the fuzzification phase, and none of the rules would fire and hence a value for the control action would not be computed. This situation is what is known as discontinuity in the control output. The present system has been designed in such a way that two adjacent membership functions have a cross-point level of 0.5 and cross-point ratio of 1, (i.e., symmetric 50% overlap). In this way, for each crisp input, two rules will fire and the control action would be continuous. There is an important point that has been considered with regard to the width of membership functions of the controller inputs (rule-antecedents). The membership functions of the nine central rules in the table of rulebase are responsible for the steady state response of the controller. It follows that the peaks of these membership functions are among the parameters that need to be tuned carefully. The membership functions of the controller output (rule-consequent) is symmetrical as well (constant width) so that peak and the centre of gravity of each membership function are identical. In the present controller in this chapter an *individual rule-based* inference method, as outlined before, is employed. The implication is based on Mamdani's method (Min-Max). The "centre of gravity" method is used for defuzzification. With the form of the membership functions that are used for the controller inputs, at each sampling period only two rules will fire for each input value.

Since there are two inputs, the computer processor will deal with only four rules at each sampling period (1 ms). This choice of inference and membership functions significantly reduces the complexity of computation of the fuzzy control law.

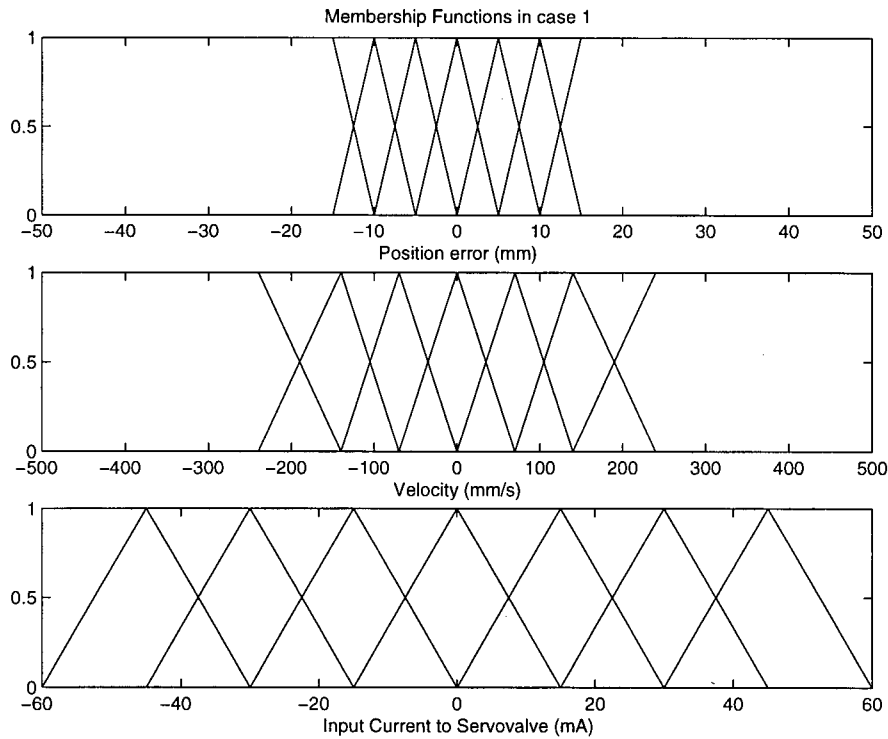


Figure 3.2. Membership functions for case1.

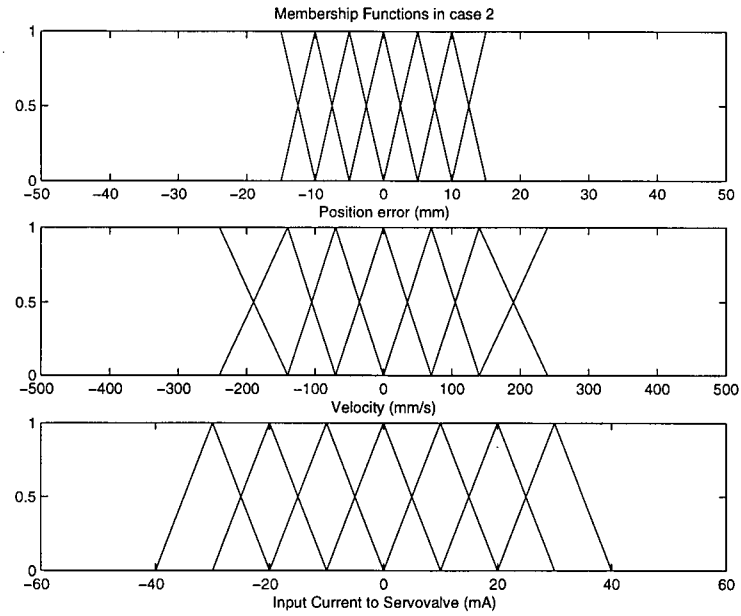


Figure 3.3. Membership functions for case2.

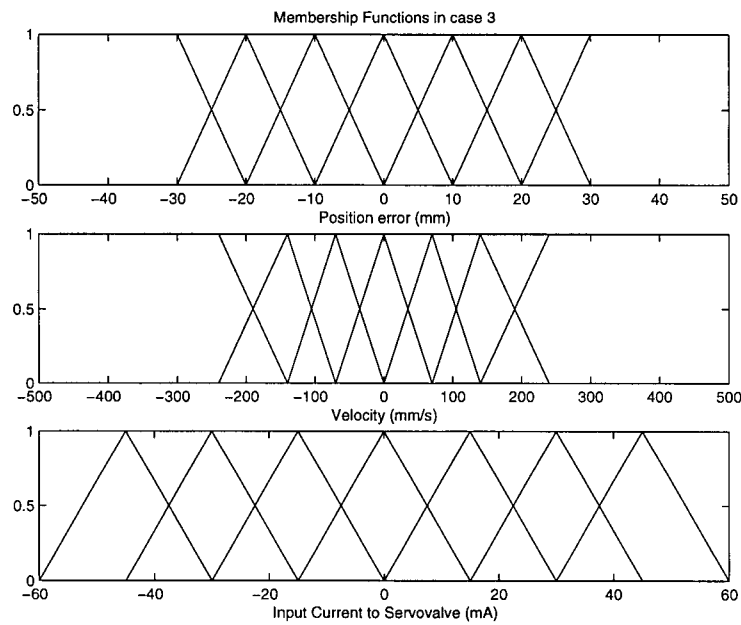


Figure 3.4. Membership functions for case3.

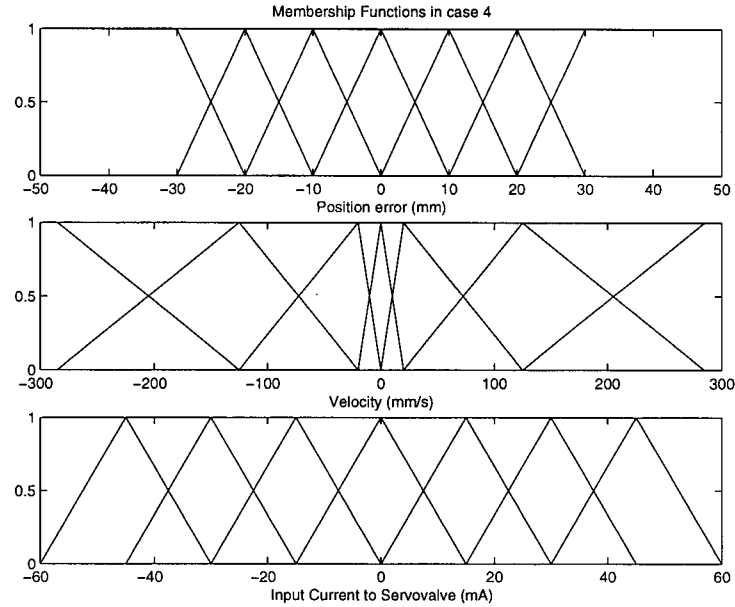


Figure 3.5. Membership functions for case4.

3.4 Effect of Tuning the Membership Functions

For further studying the designed fuzzy logic controller, the system is now analyzed with regard to the effect of tuning the membership functions of the input and output variables. In all the cases, the response of the system is analyzed for four experiments (a,b,c,d). Experiments (a) and (b) are those where the initial position error is less than zero which corresponds to the cases where the piston is moving from left to right. Experiments (c) and (d) are those where the cylinder is moving from right to left. Note that the set points (desired reference positions) are chosen differently in the four experiments.

3.4.1 Tuning the membership functions of output

Figure 3.6 shows the performance of the system in the four experiments of different initial conditions under two cases of membership choices. In these two cases, the membership functions of the two inputs, the rulebase (shown in Table 3.2), the inference method, and the defuzzification method are the same. The only difference between the two cases is the different choice of membership functions of the output. Figures 3.2 and 3.3 show the membership functions in those two cases. Specifically, Figure 3.6 illustrates effect of tuning the membership functions of output. The membership functions of the output in both cases are uniform and symmetric. However, in Case 1 the universe of discourse is $[-60\ 60]$ and in Case 2 is $[-40\ 40]$. Figure 3.6 shows that the response in Case 2 it is slower than in Case 1. The reason is that the larger universe of discourse provides a stronger control action. Accordingly, we can conclude that tuning the universe of discourse of the output membership functions has a direct effect on the speed of response of the system. Figure 3.7 shows the control action (the input signal to the actuator) for these two cases, in the four experiments (a)-(d).

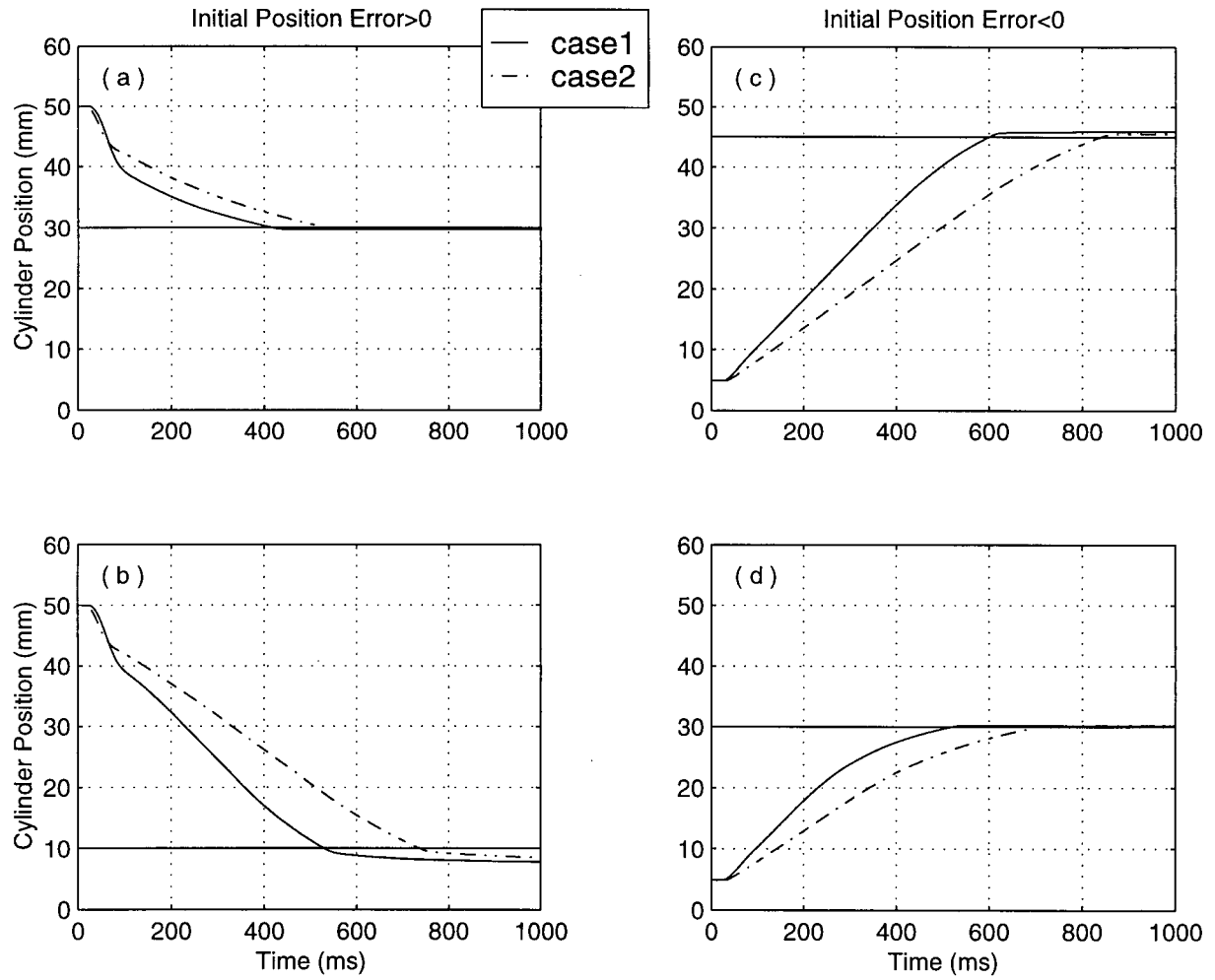


Figure 3.6. Effect of tuning the membership functions of output (the position response).

(Four experiments (a),(b),(c), and(d) with different initial conditions)

Case1: Membership functions in Figure 3.2

Case2: Membership functions in Figure 3.3

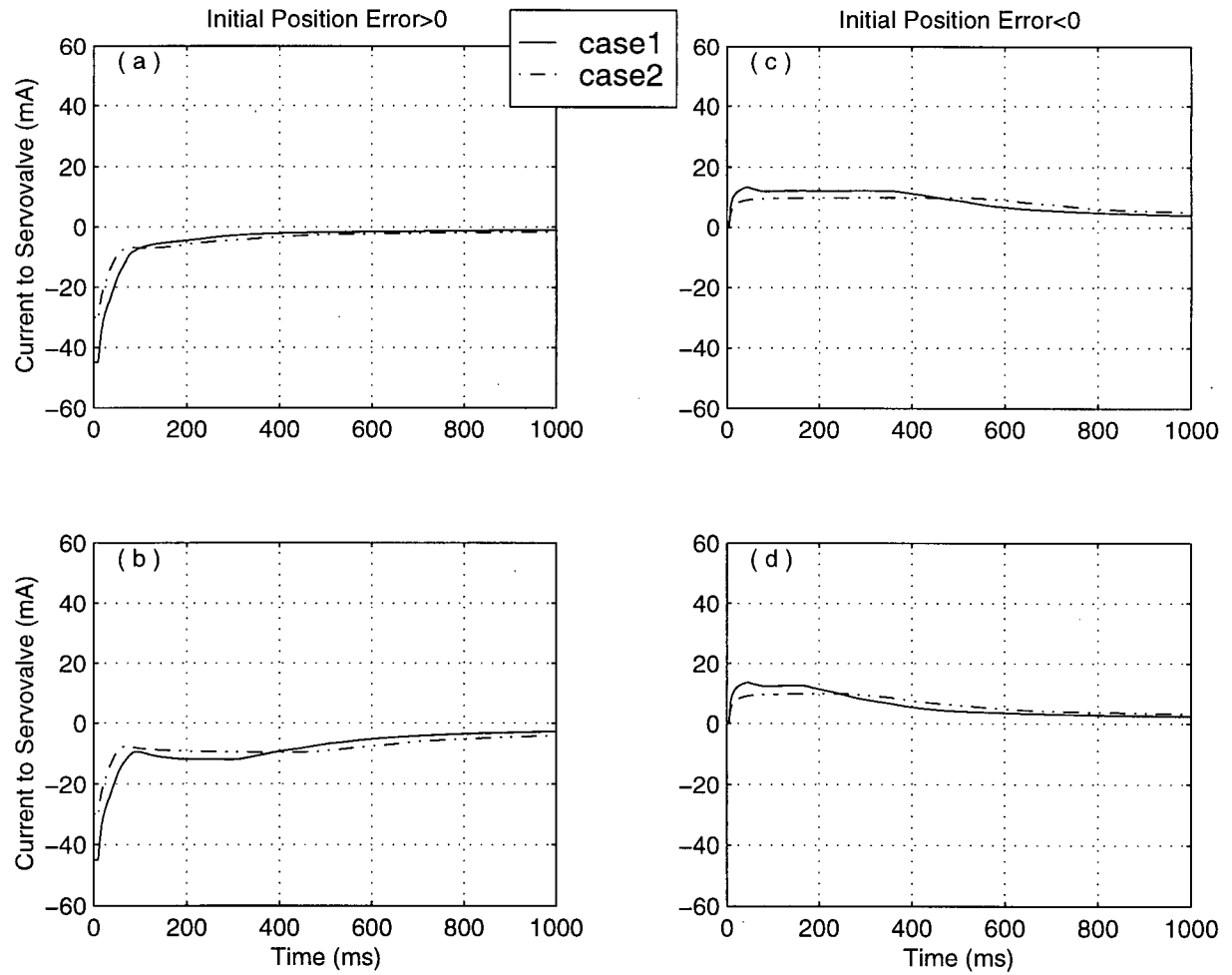


Figure 3.7. Effect of tuning the membership functions of output (the control action).

3.4.2 Tuning the membership functions of input 1

Figure 3.8 shows the effect of tuning the membership functions of input 1 (position error) in four experiments (a)-(d) under two cases: Case1 and Case3. Membership functions for Case 1 and Case3 are given in figures 3.2 and 3.4, respectively. Both sets of membership functions are uniformly distributed and symmetric. As can be seen in Figure 3.8, the response is much slower in Case 3, and the system does not reach the desired position within the allowed time of 1 second. Comparing figures 3.2 and 3.4, we can see that the support set of input 1 is much larger in Case 3 than in Case 1.

3.4.3 Tuning the membership functions of input 2

Figure 3.9 shows the effect of tuning of the second input (velocity of the cylinder) by comparing the response of the system in four experiments (a)-(d) under two cases, Case 1 and Case 4. Figures 3.2 and 3.5 show the membership functions of these two cases. Membership functions of input 1 and the output are common for the two cases, but the membership functions of input 2 (velocity) are different. In Case 1, the membership functions of input 2 are chosen to be symmetric and uniformly distributed, but in case 2 it is not. The effect of this tuning on the response is shown in Figure 3.9. Note that the nonuniform membership functions of the velocity do not provide a desirable response, which is too slow and never reaches the steady state.

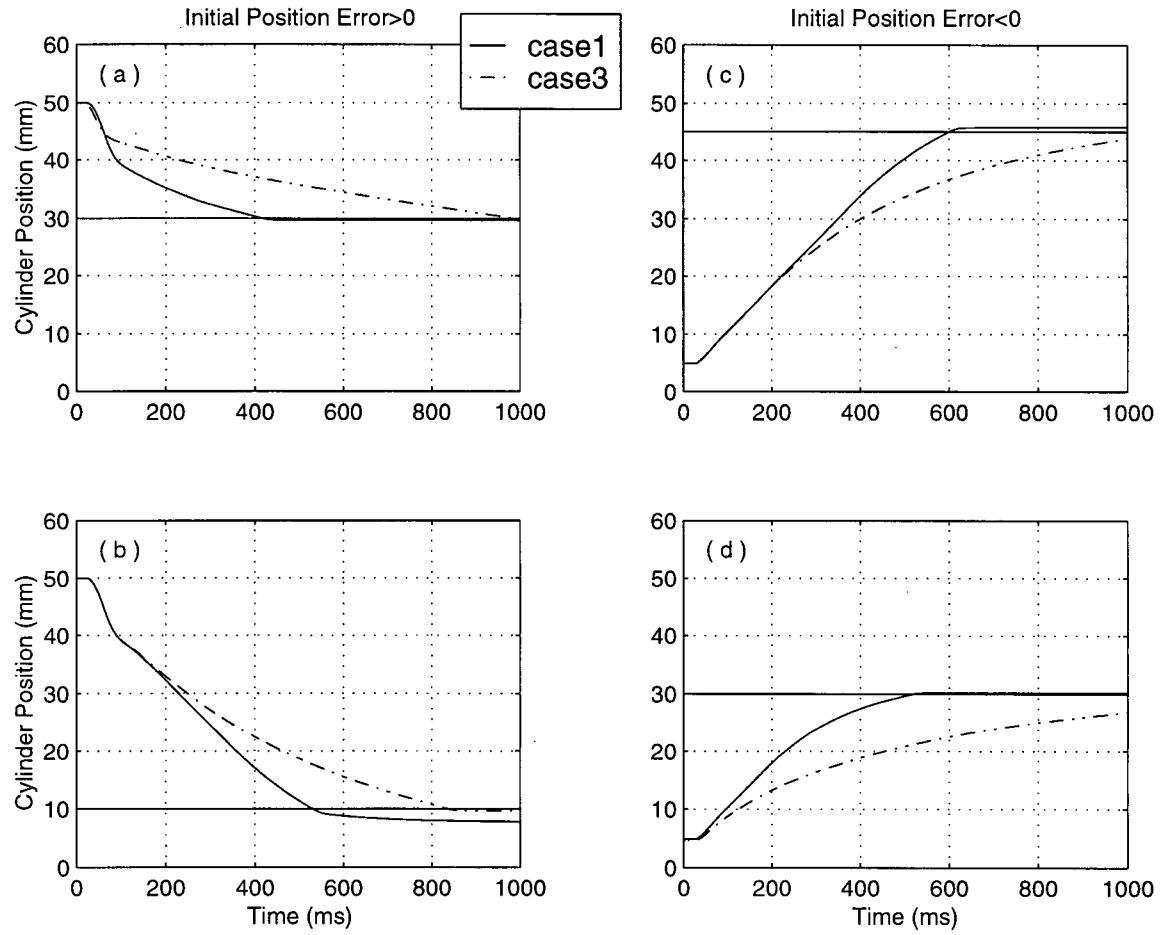


Figure 3.8. Effect of tuning the membership functions of input 1.
 (Four experiments (a)-(d) with different initial conditions)
 Case 1: Membership functions in Figure 3.2.
 Case 3: Membership functions in Figure 3.4.

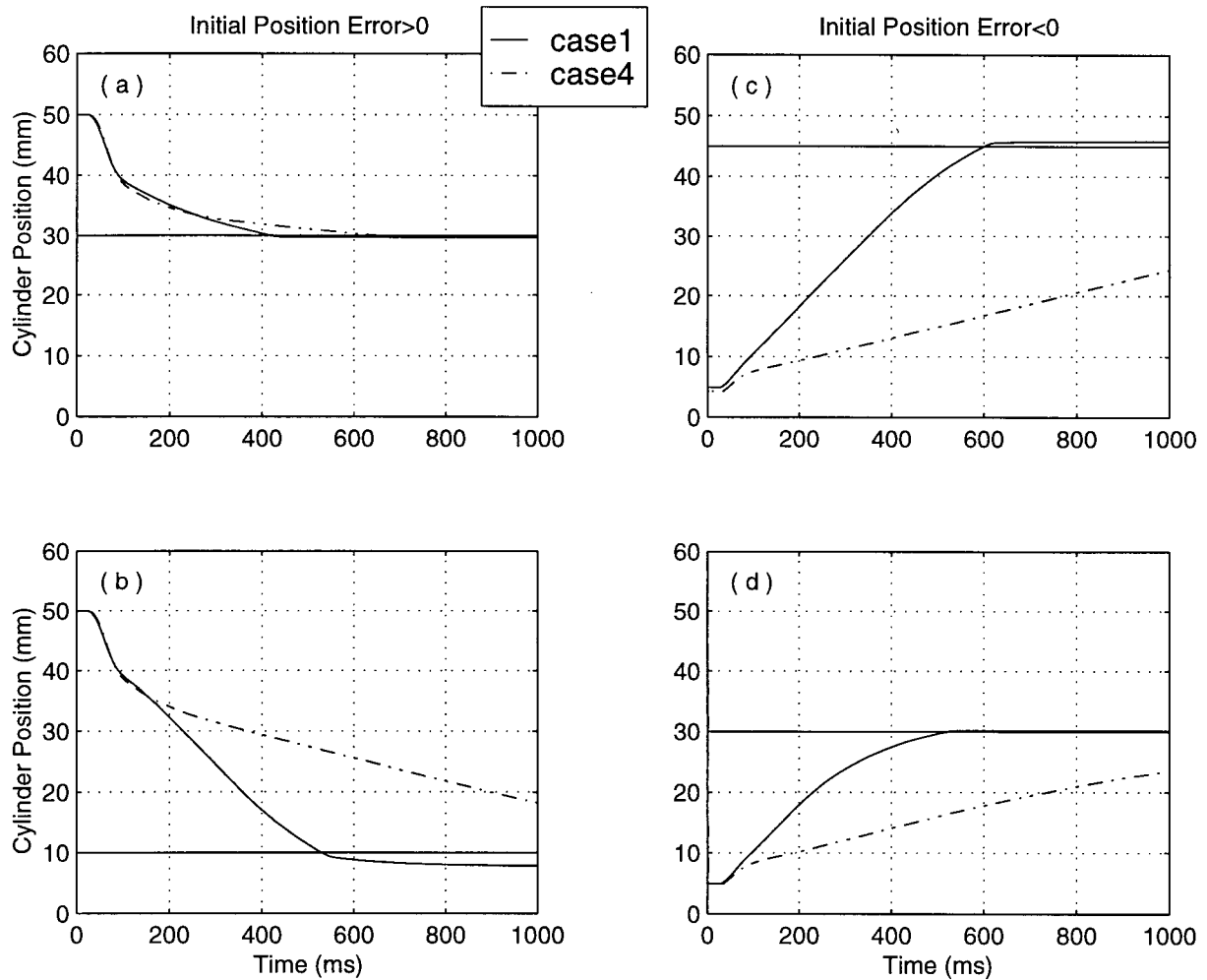


Figure 3.9 . Effect of tuning the membership functions of input 2.
 (Four experiments (a)-(d) with different initial conditions)
 Case1: Membership function in Figure 3.2.
 Case 4: Membership function in Figure 3.5.

3.5. Effect of Tuning the Rulebase

The effect of tuning the rulebase is shown in Figure 3.10. Again four experiments (a)-(d) with different initial conditions are carried out under two choices of rulebase, denoted by Case 5 and Case 6. As can be seen, the speed of response in Case 6 is faster than that in Case 5. The membership functions in cases 5 and 6 are given in tables 3.2 and 3.3, respectively. The membership functions of inputs and output are common for the two

cases, and shown in Figure 3.11. The present comparison shows the difference that the tuning of a rulebase can make in order to achieve an improvement in performance.

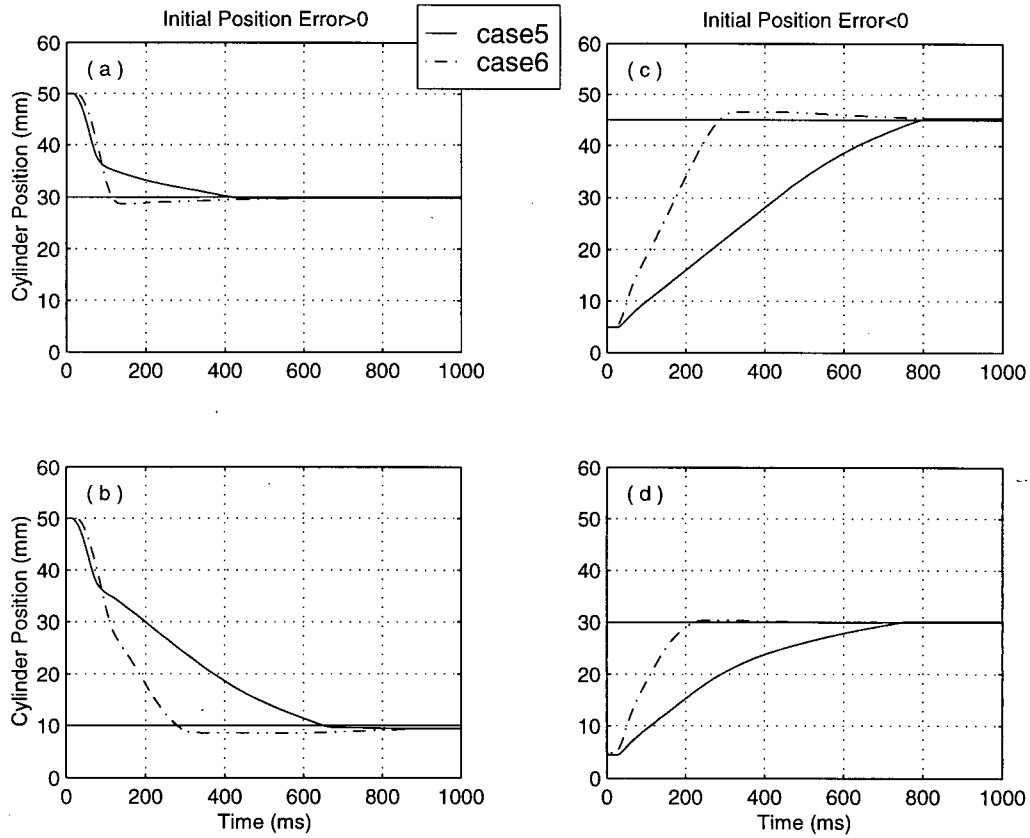


Figure 3.10. Effect of tuning the rulebase.
(Four experiments (a)-(d) with different initial conditions)
Case 5: Rulebase of Table 3.2.
Case 6: Rulebase of Table 3.3.

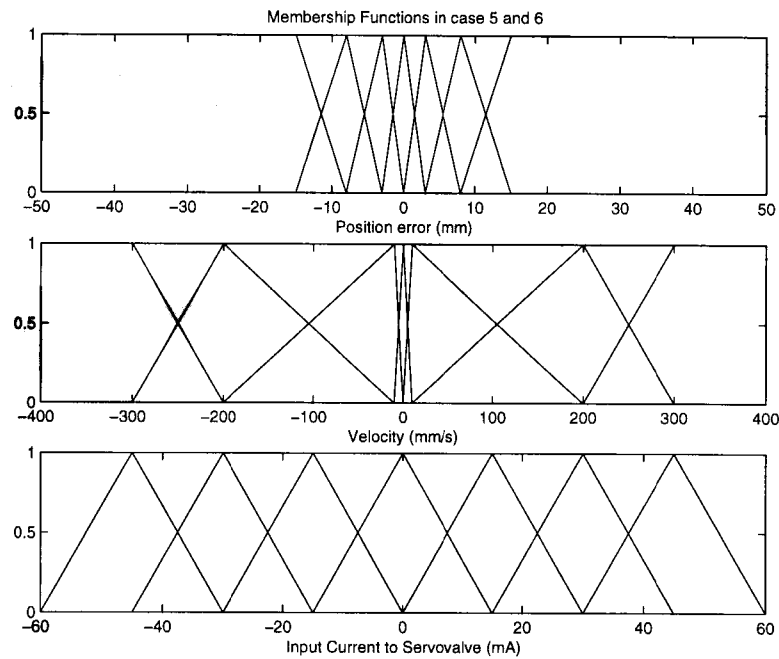


Figure 3.11. Membership functions for case 5 and case 6.

Table 3.2. Rulebase for case 5.

		v						
		NL	NM	NS	ZER	PS	PM	PL
e	NL	PL	PM	PM	PS	PS	ZER	ZER
	NM	PM	PM	PS	PS	ZER	ZER	ZER
	NS	PM	PS	PS	ZER	ZER	ZER	NS
	ZER	PS	PS	ZER	ZER	ZER	NS	NS
	PS	ZER	ZER	ZER	ZER	NS	NS	NM
	PM	ZER	ZER	ZER	NS	NS	NM	NM
	PL	ZER	ZER	NS	NS	NM	NM	NL

Table 3.3. Rulebase for case 6.

	v						
	NL	NM	NS	ZER	PS	PM	PL
e	NL	PL	PL	PM	PM	PS	ZER
	NM	PM	PM	PM	PS	ZER	NS
	NS	PM	PM	PM	PS	ZER	NS
	ZER	PM	PM	PS	ZER	NS	NM
	PS	PM	PS	ZER	NS	NM	NM
	PM	PS	ZER	NS	NM	NM	NL
	PL	ZER	NS	NM	NM	NL	NL

3.6. Comparison of Defuzzification Methods

In this section we compare the performance of the system with common membership functions for inputs and output and a common rulebase, but with two different commonly used defuzzification methods. Case 5 is chosen for this comparison, and two experiments with different initial conditions are carried out. Membership functions of the inputs and the output are shown in Figure 3.11 and the rulebase is given in Table 3.2. The performance of this fuzzy logic controller is compared with two different defuzzification methods. Figure 3.12 shows the performance of the system with the centre of gravity defuzzification method. Figure 3.13 gives the system response when the defuzzification method is changed to the “centre average” method given in Eq. (3.3). These two defuzzification methods are more common because they are easier to compute. In the defuzzification method using the “centre-of-gravity” given in Eq.(3.2), a crisp output y_q^{crisp} is chosen using the centre-of-area and area of each implied fuzzy set, and is given by;

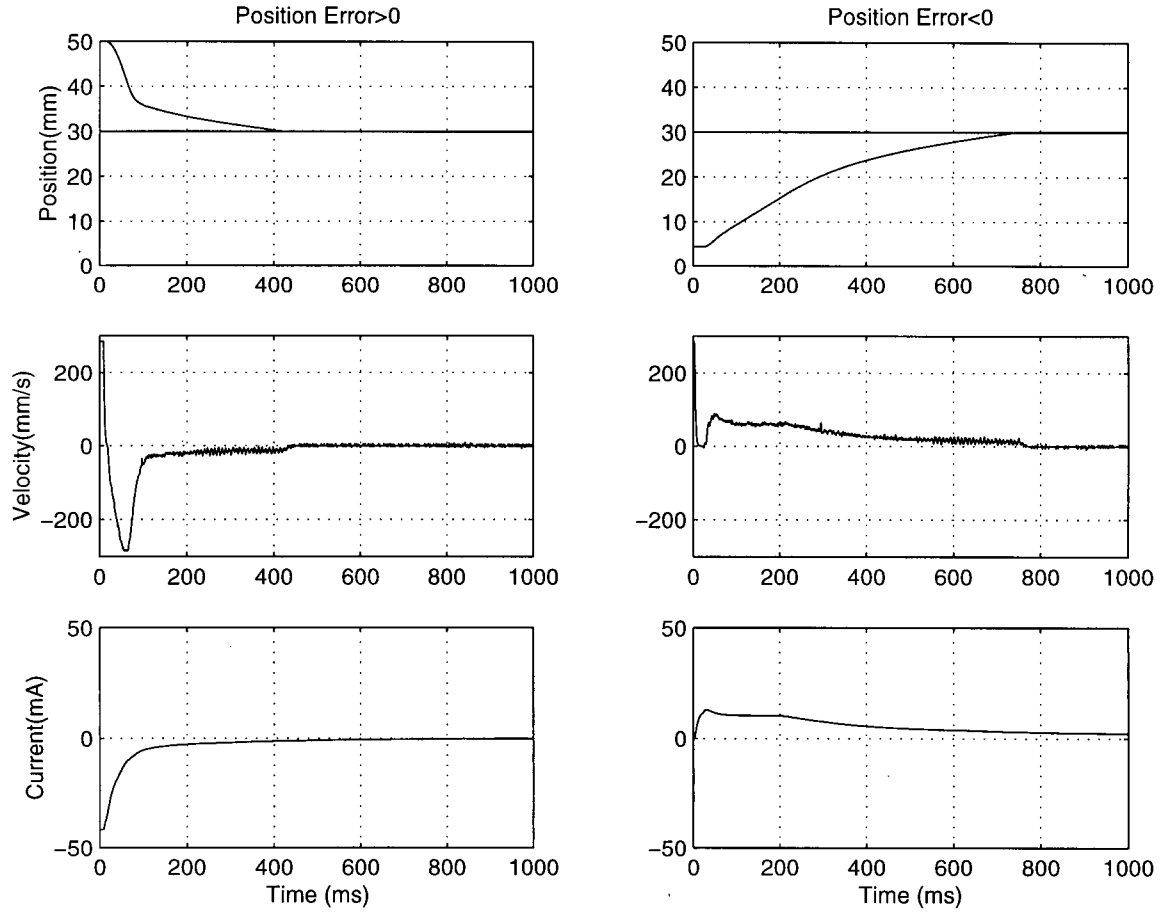


Figure 3.12. Response of the system with the centre of gravity defuzzification method.
(Two experiments with different initial conditions)

$$y_q^{crisp} = \frac{\sum_{i=1}^R b_i^q \int_{y_q} \mu_{\hat{B}_q^i}(y_q) dy_q}{\sum_{i=1}^R \int_{y_q} \mu_{\hat{B}_q^i}(y_q) dy_q} \quad (3.2)$$

where R is the number of rules, b_i^q is the centre of area of the membership function of B_q^p associated with the implied fuzzy set \hat{B}_q^i for the i^{th} rule $(j, k, \dots, l; p, q)_i$, and $\int_{y_q} \mu_{\hat{B}_q^i}(y_q) dy_q$ denotes the area under $\mu_{\hat{B}_q^i}(y_q)$. In the defuzzification method using the centre-average, a crisp output y_q^{crisp} is chosen using the centres of each of the output

membership functions and maximum certainty of each of the conclusions represented with the implied fuzzy sets, and is given by;

$$y_q^{crisp} = \frac{\sum_{i=1}^R b_i^q \sup_{y_q} \{\mu_{\hat{B}_q^i}(y_q)\}}{\sum_{i=1}^R \sup_{y_q} \{\mu_{\hat{B}_q^i}(y_q)\}} \quad (3.3)$$

Observe that the position of the cylinder reaches the desired position within the specified time in Figure 3.12, but in Figure 3.13 the response is slower and cylinder does not reach the desired position in the specified time. However, the membership functions and the rulebase had already been tuned for the case where the defuzzification is by the centre of gravity method. It follows that if we change the defuzzification method, the system needs to be tuned accordingly. However, Figure 3.12 (Defuzzification by the centre of gravity method) shows that the control signal is much smoother than the case with the centre average method.

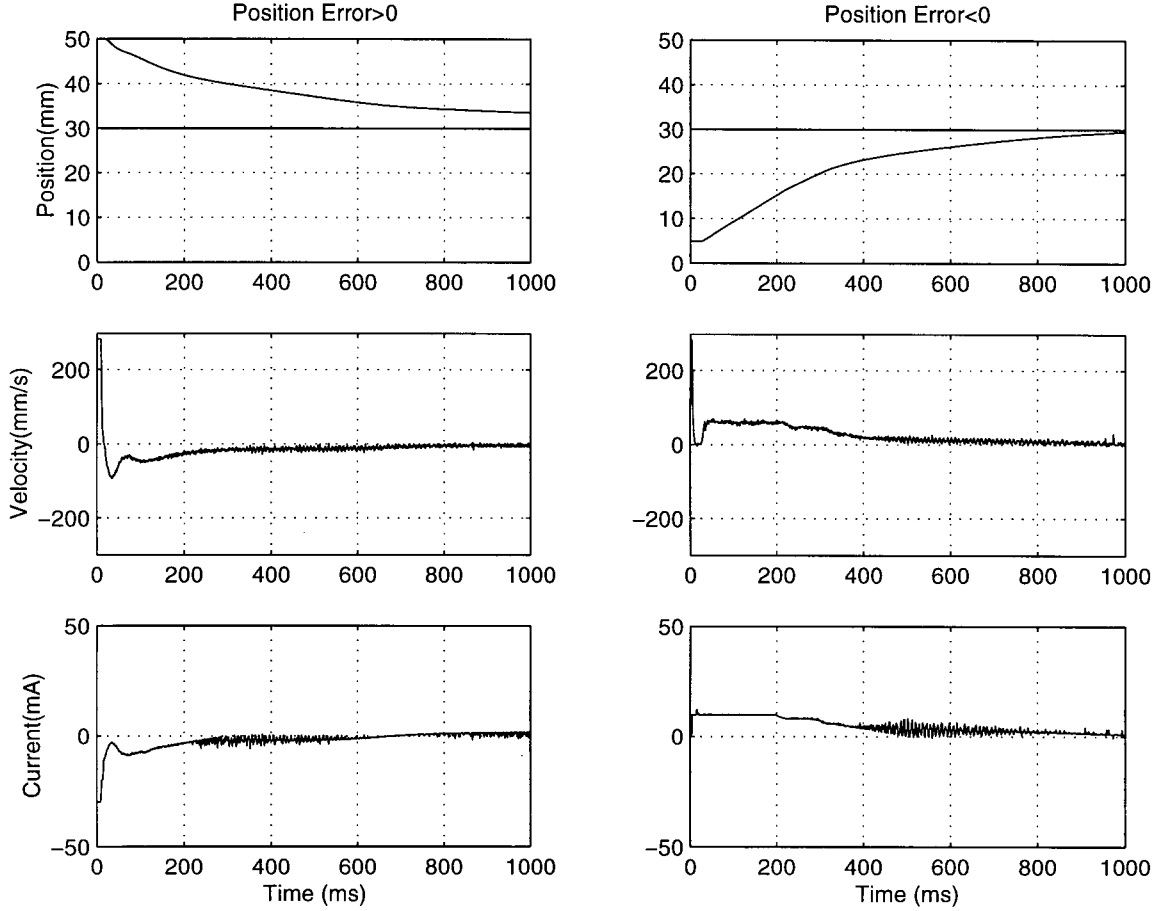


Figure 3.13. Response of the system with the centre of average defuzzification method.
(Two experiments with different initial conditions)

The present comparison shows that tuning of the membership functions of input 1, input 2, and output, and of the rulebase, and also proper selection of the defuzzification method can have a significant effect on the performance of the system response. The actual experimental results using an industrial hydraulic actuator have shown that the designed PD-type fuzzy logic controller is able to meet the specifications of the specific industrial application, assuming that the membership functions, the rulebase, and the defuzzification method are chosen and tuned properly. Since the velocity of the piston is estimated based on the designed observer in Eq 2.1, it takes a few samples (about 50 samples) to converge and make a correct estimation. This is the reason that the velocity of the first 50 ms in the Figures 3.12 and 3.13 are not showing the correct estimation.

Chapter 4

P-Type Fuzzy Control on a Hydraulic Manipulator

Implementation of a proportional-plus-derivative (PD)-type fuzzy logic controller on the prototype hydraulic manipulator was presented in Chapter 3. Issues related to the inference scheme rulebase membership functions, and the defuzzification method were investigated through experimentation. In that controller, there are two inputs. The maximum number of rules in the control rulebase is given by n^m , where n is the number of inputs and m is the number of fuzzy states that each input variable can have. It follows that, depending on the number of fuzzy states, the number of rules can be quite large. In the particular example, given in Table 3.1, the number of developed rules for the PD-type fuzzy controller was $7 \times 7 = 49$ rules. Clearly, it is useful to see whether a comparable performance to the PD controller could be achieved with a simple controller and correspondingly a simpler rulebase. This chapter explores the use of a proportional (P)-type controller for the hydraulic manipulator with the advantage of simplicity.

4.1 P-Type Fuzzy Logic Controller

In this section, the proportional (P)-type fuzzy logic controller, as implemented in the hydraulic manipulator is described. The conventional proportional (P) control law is given in Eq (4.1) by;

$$u = K_p \cdot e \quad (4.1)$$

where e is the response error and K_p is the proportional control gain. Then a P-type fuzzy logic controller, where the input (context) is the response error and the output is the control action, consists of fuzzy rules [28] of the form

$$\text{if } e(k) \text{ is } \langle \text{a fuzzy state} \rangle \text{ then } u(k) \text{ is } \langle \text{a fuzzy state} \rangle$$

The corresponding linguistic form is:

if the value of the response error is $\langle \text{linguistic value} \rangle$

then the value of the control output is $\langle \text{linguistic value} \rangle$

for each sampling period k .

In the present problem of position control of the prototype hydraulic actuator, the input to the fuzzy decision making system is the position error e and the output is the current applied to the servovalve. A block diagram of the fuzzy knowledge-based controller (FKBC) is shown in Figure 4.1.

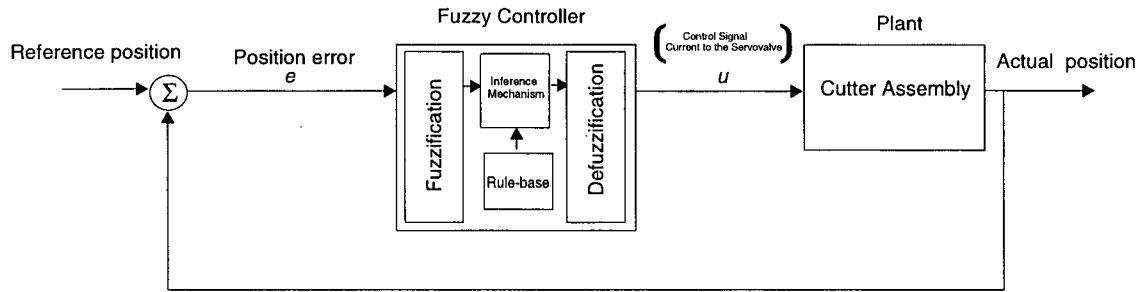


Figure 4.1 Block diagram of the fuzzy logic controller.

The P-type fuzzy logic controller is implemented on the prototype hydraulic manipulator (Chapter 2, and Figure 2.3) in the same manner as with PD-type controller (Chapter 3). The rulebase used for two specific experimental investigations will be given in subsequent sections. The fuzzy states of the input and the output are all chosen to be equal in number and use the same linguistic descriptors: {NL, NM, NS, ZR, PS, PM, PL}; where NL = negative large, NM = negative medium, NS = negative small, ZR = zero, PS = positive small, PM = positive medium, and PL = positive large. The membership functions of these variables, as used in the experimental investigations presented here will be given in subsequent sections. The inference method is based on *individual rules* [28] as outlined in Chapter 3. This inference method, as applied in the present implementation, is shown in Figure 4.2 and can be simply described as follows:

1. Compute the degree of match between the crisp input and the fuzzy states of the rule-antecedent (position error).
2. "Clip" the fuzzy state of the rule-consequent (control action) by the membership degree to which the rule-antecedent is matched by the crisp input.

3. Aggregate the “clipped” values of the control output of each rule, to form the value of the overall control output.
4. Determine the centroid of the aggregated control inference, giving a crisp control action.

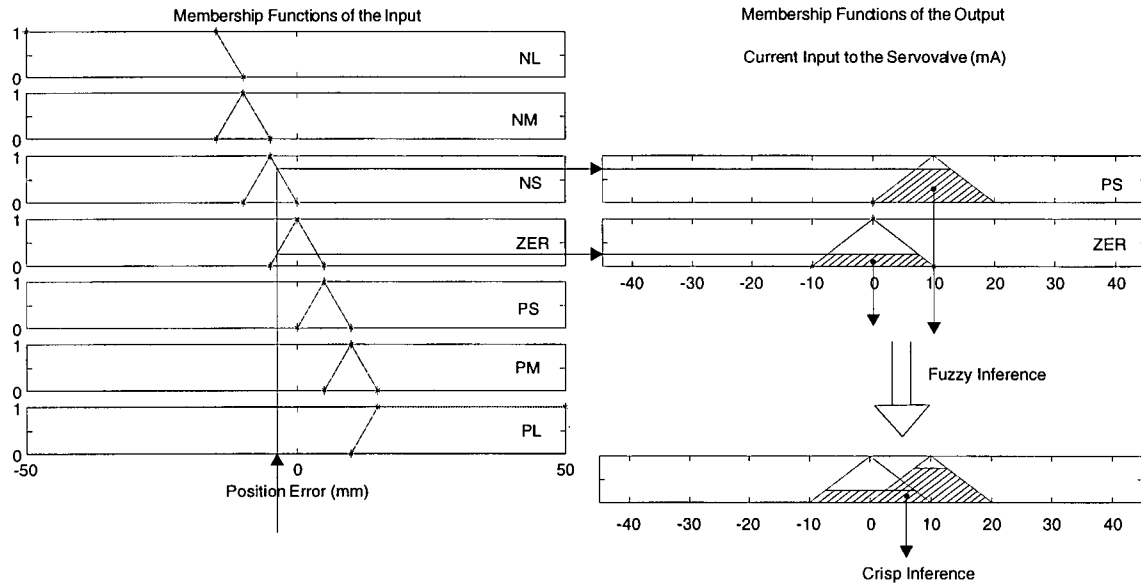


Figure 4.2 Inference based on individual rules for the P-type controller.

4.2 Effect of Tuning the Input Membership Functions

The effect of tuning the input membership function for the P-type controller is investigated by implementing the controller in four different cases. Membership functions of the input to the fuzzy system in the four cases are shown in Figure 4.3. The common membership function of the output for all the four cases is shown in Figure 4.4. The comparison has been conducted under the common rulebase as given in Figure 4.5. A comparison of the response of the hydraulic actuator for the same setpoint and the same initial condition is shown in Figure 4.6. Note that Case 1 which has the highest input resolution, provides the fastest response. Case 4 which has a nonuniform resolution, with a low resolution for large errors and a finer resolution for small errors gives the next best response. Figures 4.7 through 4.10 show position of the cylinder and the input current to the servovalve with two different initial conditions and setpoints

giving two directions of motion, as denoted by (a) and (b). Comparing Figures 4.3 through 4.6, we can see that case 1 shows the fastest speed of response and Figure 4.8 shows that the membership functions of Case 1 is finer in the centre of the universe of discourse. Also, as before, Case 3 gives a very good response in view of the nonuniform resolution of the input membership function. This shows how tuning the membership functions can improve the speed of response of the system.

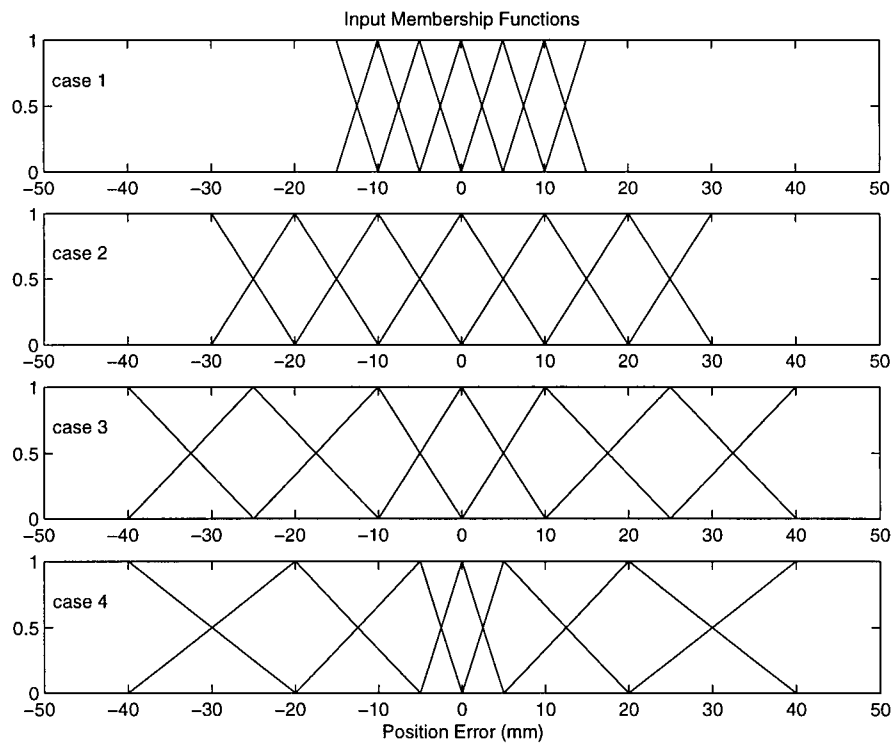


Figure 4.3 Membership functions of the input.

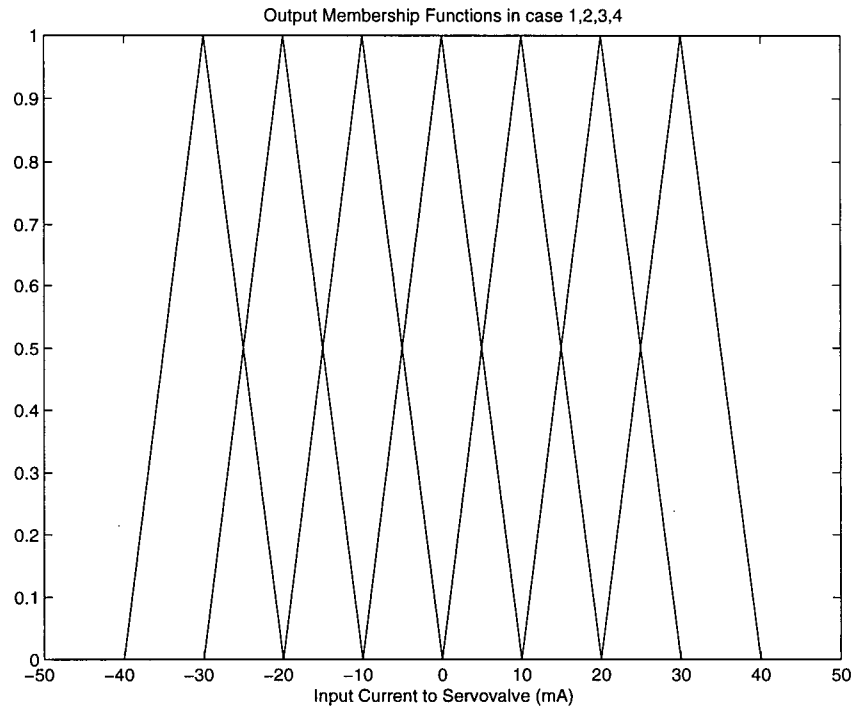


Figure 4.4 Membership functions of the output.

```

IF Pos_err is NL THEN control signal is PL
IF Pos_err is NM THEN control signal is PM
IF Pos_err is NS THEN control signal is PS
IF Pos_err is ZR THEN control signal is ZR
IF Pos_err is PS THEN control signal is NS
IF Pos_err is PM THEN control signal is NM
IF Pos_err is PL THEN control signal is NL
    
```

Figure 4.5 Rulebase of the P-type fuzzy controller.

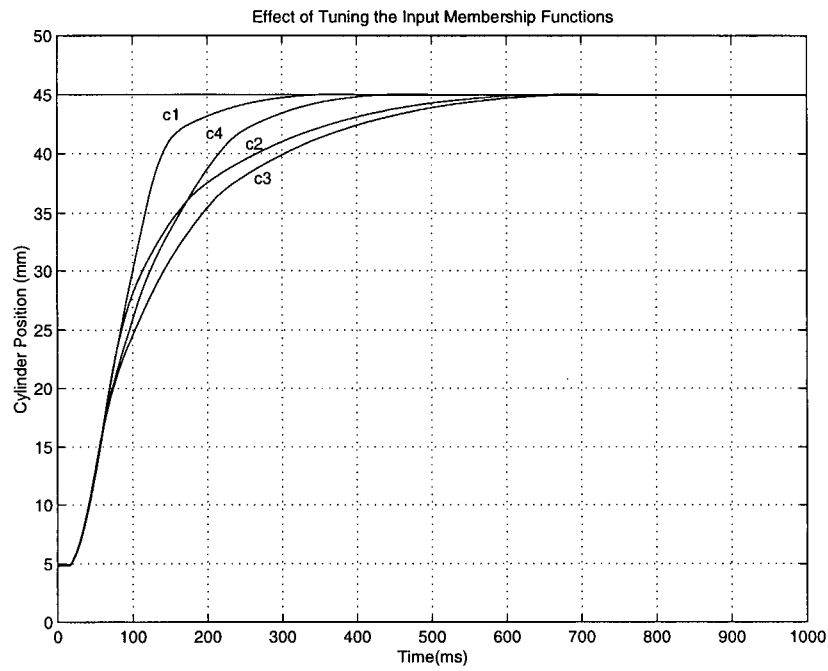


Figure 4.6 Comparison of the responses of the system in the four cases, with the same initial conditions.

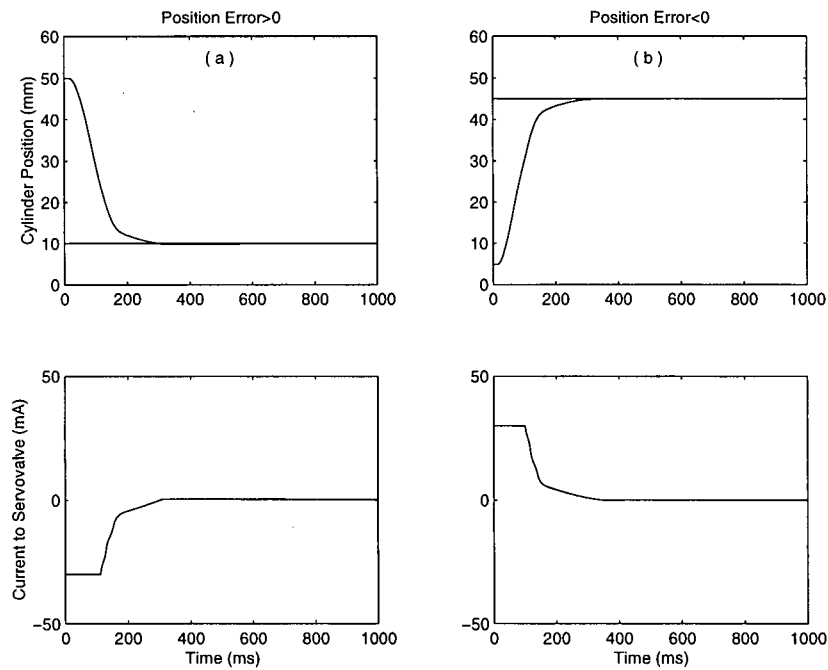


Figure 4.7 Response of the system in Case 1.

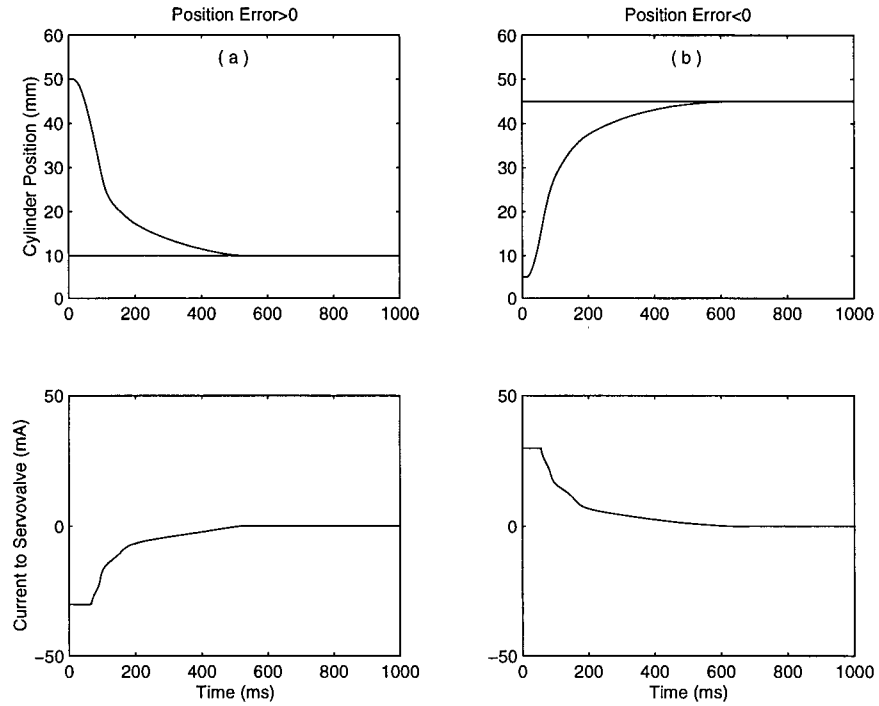


Figure 4.8 Response of the system in Case 2.

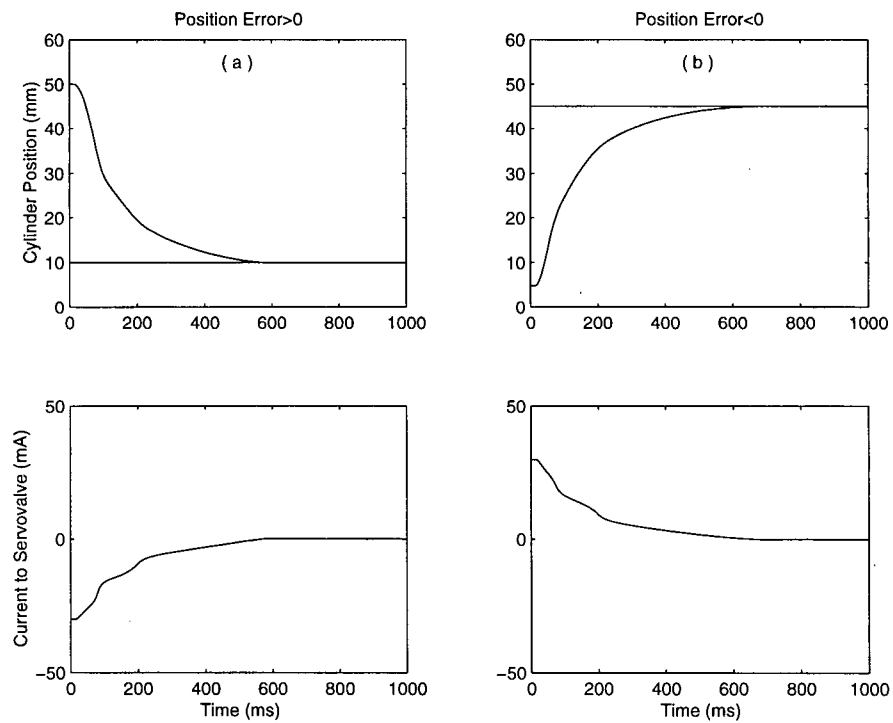


Figure 4.9 Response of the system in Case 3.

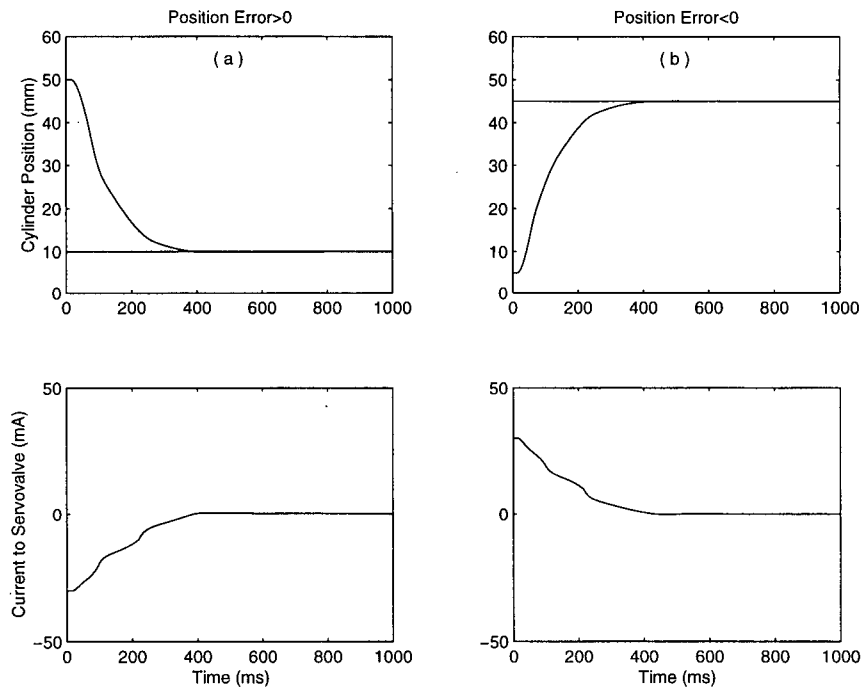


Figure 4.10 Response of the system in Case 4.

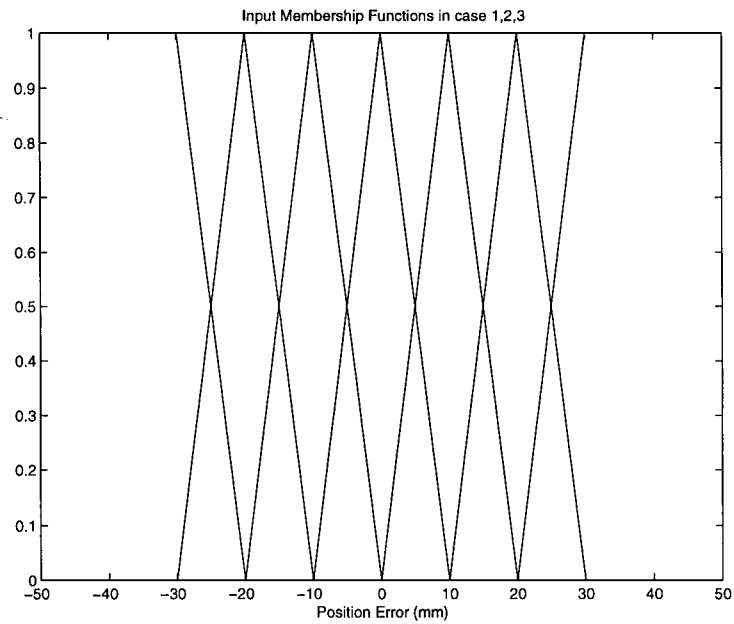


Figure 4.11 Membership functions of input in three cases.

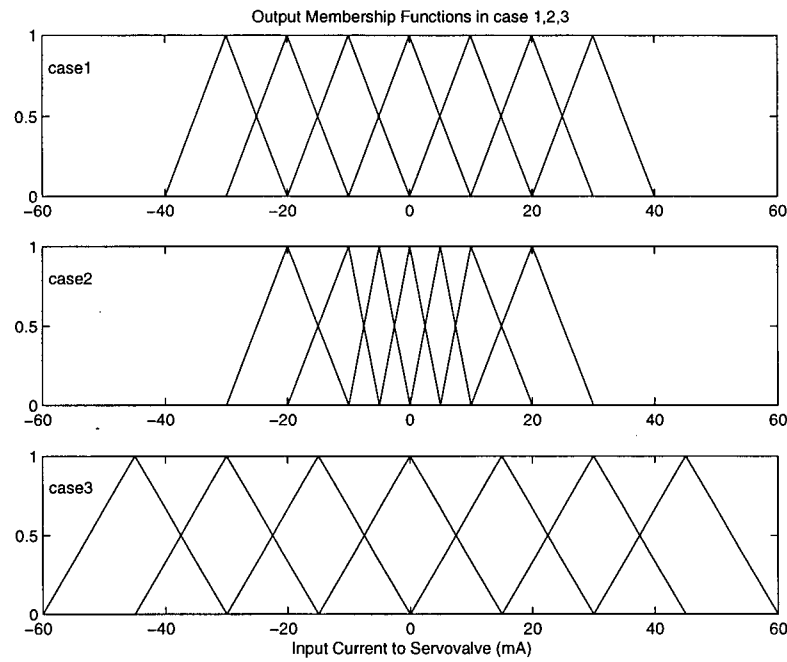


Figure 4.12 Membership functions of output in three cases.

<p>IF Pos_err is NL THEN control signal is PL IF Pos_err is NM THEN control signal is PM IF Pos_err is NS THEN control signal is PS IF Pos_err is ZR THEN control signal is ZR IF Pos_err is PS THEN control signal is NS IF Pos_err is PM THEN control signal is NM IF Pos_err is PL THEN control signal is NL</p>

Figure 4.13 The rulebase of the system in the three cases.

4.3 Effect of Tuning the Output Membership Functions

Next, a set of experiments is conducted to study the effect of tuning the output membership functions. The experiments are carried out in three different cases. The common membership function of the input variable for all three cases is shown in Figure 4.11. The output membership functions for the three cases are shown in Figure 4.12. The common rulebase for all

three cases is shown in Figure 4.13. The performance of the system in three cases is shown in figures 4.14 through 4.16 for two different initial conditions and set points giving two directions. It is clear that the response in Case 1 (Figure 4.14) is the best among the three. By examining the associated membership functions of output, it is seen that Case 1 has a finer and uniform resolution. This case has resulted in a fast speed of response with a setting time of less than 600ms. Figure 4.16 demonstrates the case where only one rule is fired and that is “IF the position error is large THEN control signal should be large”. Referring to output membership functions in case 3, the output membership function of “LARGE” corresponds to signal more than 45 mA. In the code, the signals higher than 45mA is set to 5 mA for showing this particular case. So even though the the control decision is set to be *large*, it actually sends a low current to the servovalve, therefore always the position error is large and always the same rule “IF the position error is large THEN control signal should be large” is fired. This shows that hardware limitations need to be considered while designing the output membership functions.

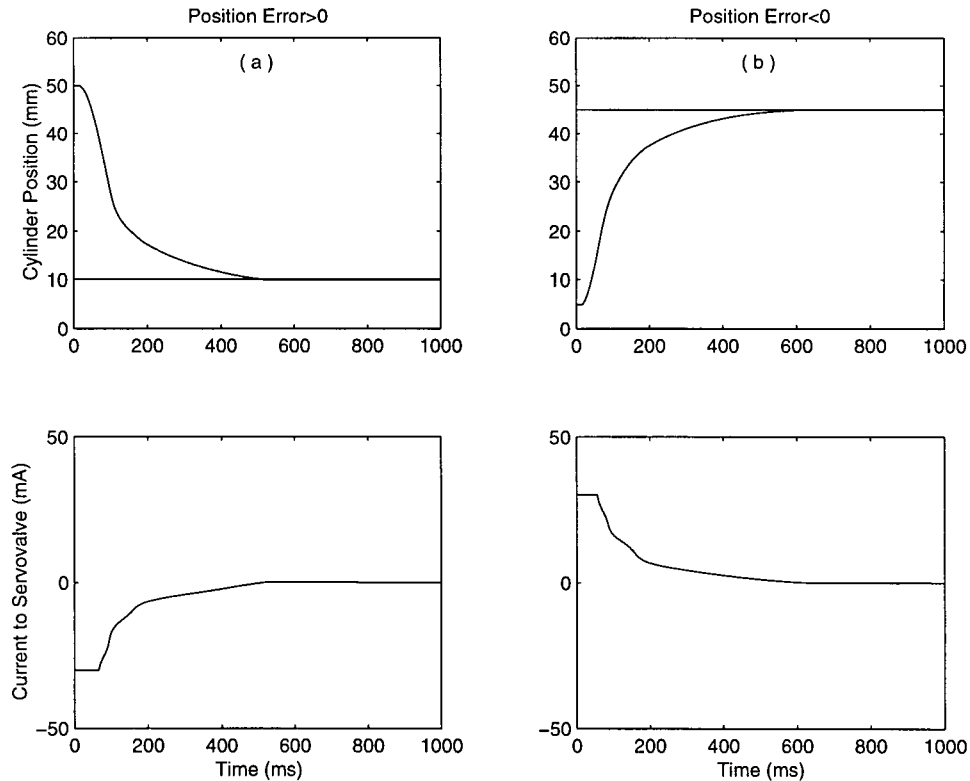


Figure 4.14 Response of the system in Case 1.

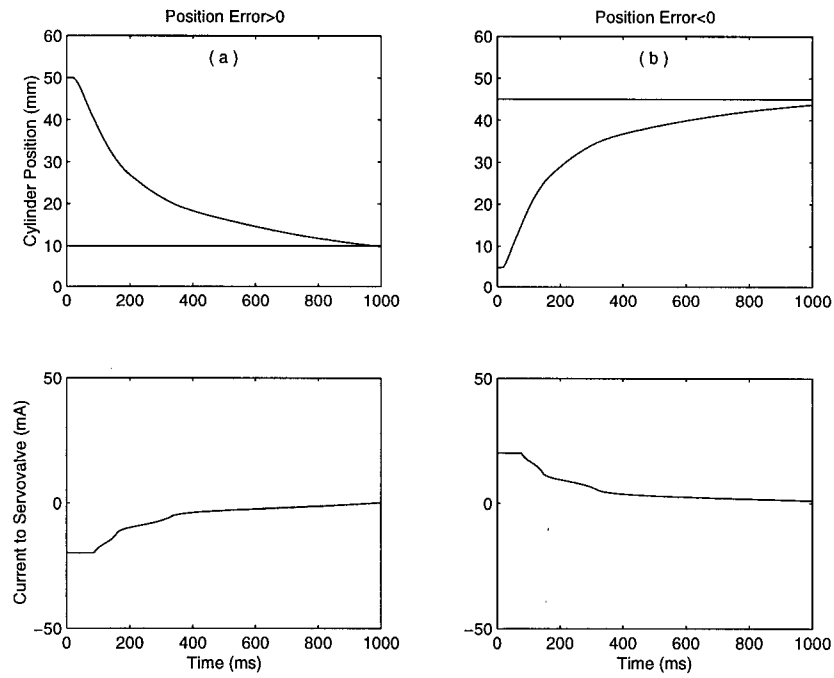


Figure 4.15 Response of the system in Case 2.

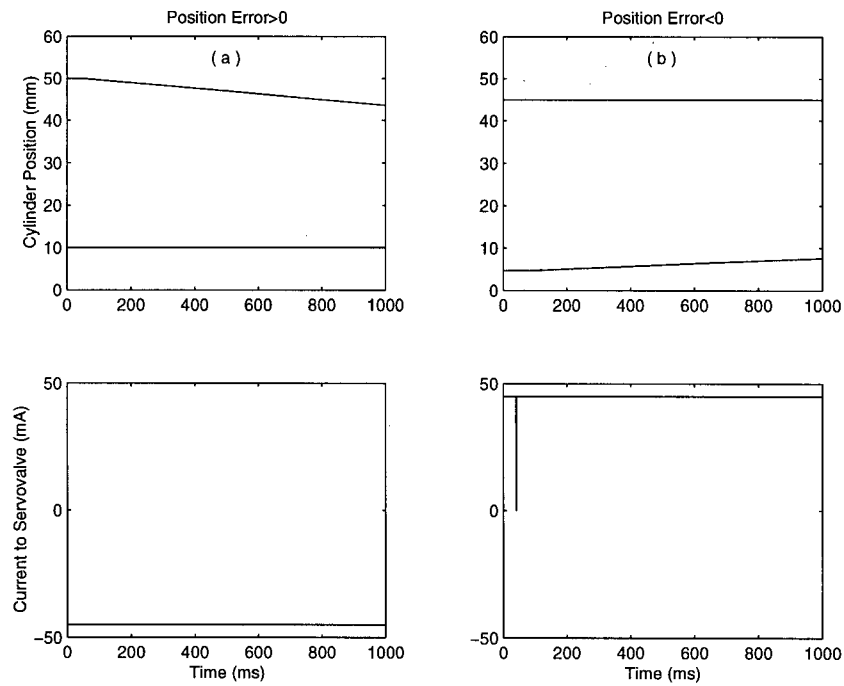


Figure 4.16 Response of the system in Case 3.

4.4 Effect of Rulebase Tuning

A final set of experiments is conducted to investigate the effect of tuning the rulebase. Here three different rulebases, as given in figures 4.17 through 4.19, are used. The membership functions of the input to the fuzzy system (position error) and the control output (input current to the servovalve) are shown in Figure 4.20. The system responses for the three cases of rulebase are shown in figures 4.21 through 4.23 where two sets of initial conditions and set points are used, giving two directions of motion as denoted by (a) and (b). Rulebase in Case 3 (Figure 4.23) provides a stronger control signal, which results in a fast speed of response. The rulebase in Case 2 provides a milder control signal, which results in a slower speed of response as shown in Figure 4.22.

```

IF Pos_err is NL THEN control signal is PL
IF Pos_err is NM THEN control signal is PM
IF Pos_err is NS THEN control signal is PS
IF Pos_err is ZR THEN control signal is ZR
IF Pos_err is PS THEN control signal is NS
IF Pos_err is PM THEN control signal is NM
IF Pos_err is PL THEN control signal is NL

```

Figure 4.17 Rulebase in Case1.

```

IF Pos_err is NL THEN control signal is PM
IF Pos_err is NM THEN control signal is PS
IF Pos_err is NS THEN control signal is PS
IF Pos_err is ZR THEN control signal is ZR
IF Pos_err is PS THEN control signal is NS
IF Pos_err is PM THEN control signal is NS
IF Pos_err is PL THEN control signal is NM

```

Figure 4.18 Rulebase in Case2.

IF Pos_err is NL THEN control signal is PL
 IF Pos_err is NM THEN control signal is PL
 IF Pos_err is NS THEN control signal is PL
 IF Pos_err is ZR THEN control signal is ZR
 IF Pos_err is PS THEN control signal is NL
 IF Pos_err is PM THEN control signal is NL
 IF Pos_err is PL THEN control signal is NL

Figure 4.19 Rulebase in Case3.

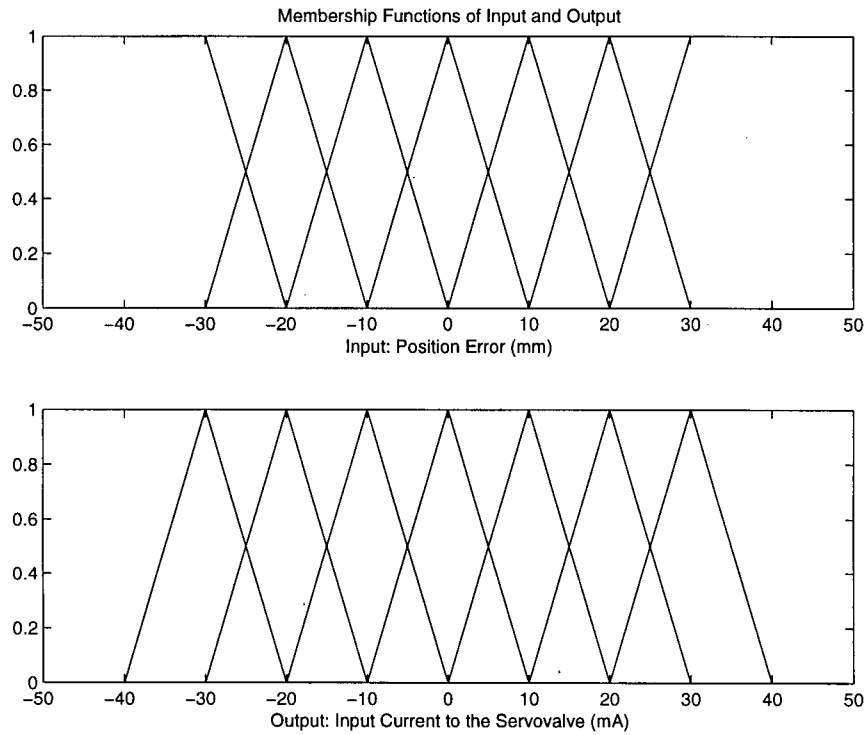


Figure 4.20 Membership functions of input and output in all three cases.

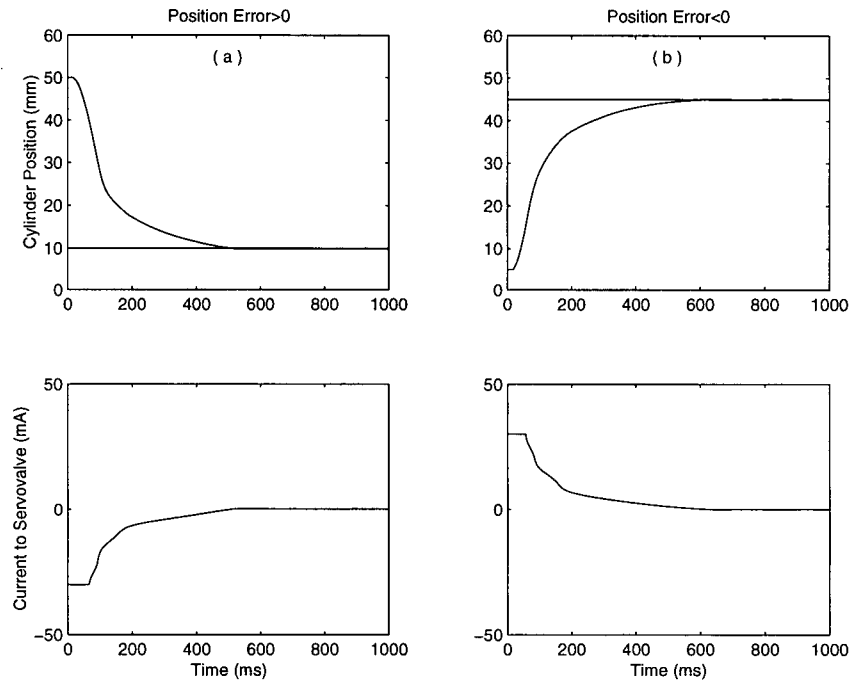


Figure 4.21 Response of the system in Case1.

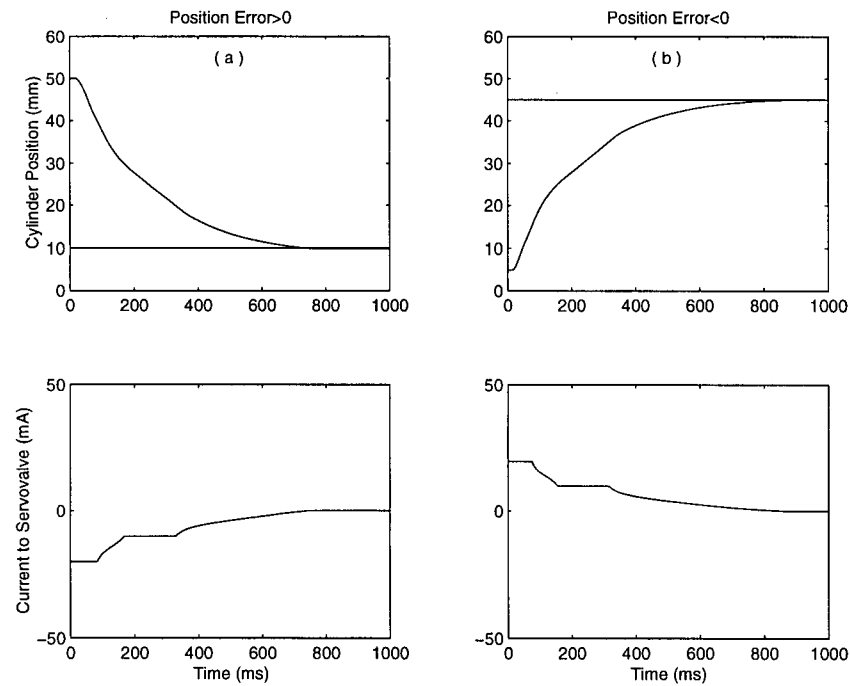


Figure 4.22 Response of the system in Case2.

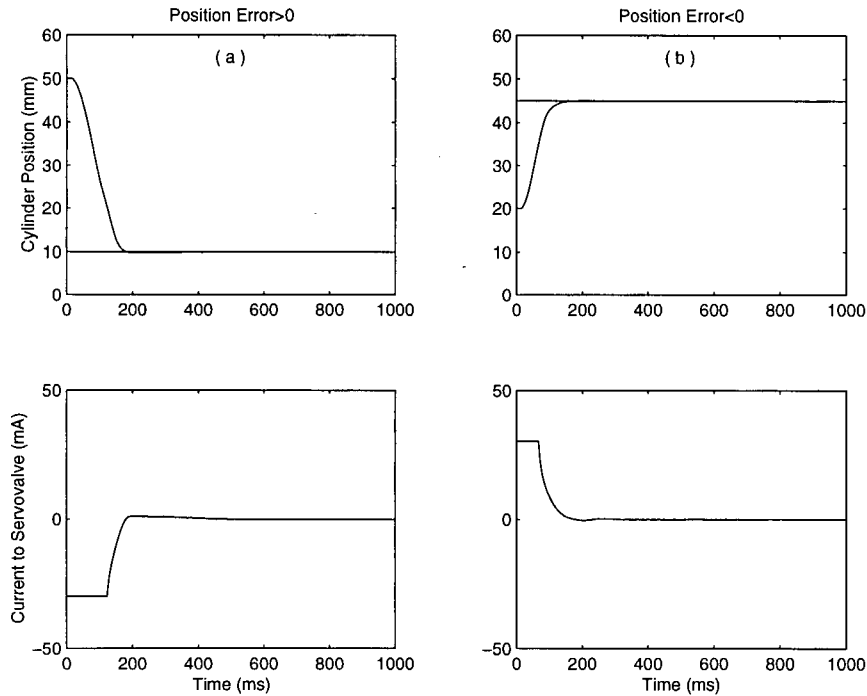


Figure 4.23 Response of the system in Case3.

In summary, the present comparisons show the importance of tuning of the membership functions and the rulebase in order to obtain a satisfactory performance. The experiments also showed that the simplest type of fuzzy control (P-type) can provide a performance that is comparable to what is obtained with a PD-type fuzzy controller in an industrial prototype of a hydraulic manipulator.

Chapter 5

Application of Composition Based Inference to P-Type Fuzzy Control

In the thesis, thus far, two specific types of control (PD-type and P-type) have been implemented on a prototype hydraulic actuator and investigated with regard to such aspects as the membership functions of inputs and outputs and the rulebase. Experience has been gained as well concerning the generation of the control inference. The specific inference technique, which has been used is the individual-rule based approach. Since inference generation in a knowledge-based system is a major focus of the present research, it is useful to investigate another common inference technique. In this chapter, the same P-type fuzzy control (with the same rulebase and the same membership functions) as investigated in Chapter 4 is implemented with the composition based inference method. The performance of the hydraulic actuator with the P-type fuzzy controller will be compared, under the two inference methods. Such a practical comparison of the performance of a fuzzy controller under these two types of inference techniques is not found in the literature.

5.1 Inference Based on Rule Composition

The inference method based on rule composition [28,78], also called *Compositional Rule of Inference*, is defined in [96] as:

If R is a fuzzy relation (see Appendix A) from U to V , and x is a fuzzy subset of U , then the fuzzy subset y of V which is induced by x is given by the composition of R and x , given in Eq (5.1);

$$y = x \circ R \quad (5.1)$$

in which x plays the role of a unary (one dimensional) relation.

Consider the generalized modus ponens:

$$\text{Premise 1: IF } x \text{ is } A \text{ THEN } y \text{ is } B \quad (5.2)$$

$$\text{Premise 2 : } x \text{ is } A'$$

$$\text{Consequence: } y \text{ is } B'$$

where A, A', B, B' are fuzzy sets. First we need to convert the IF-THEN rule in premise 1 into the fuzzy relation $R_{A \rightarrow B}$. Then the conclusion B' is inferred from the fuzzy relations $R_{A \rightarrow B}$ and A' of premise 2, applying the compositional operation given in Eq (5.3):

$$B' = A' \circ R_{A \rightarrow B} \quad (5.3)$$

The reasoning process based on (5.2) is called the *compositional rule of inference*. Note that the rule $A \rightarrow B$ is converted to a fuzzy relation. The conversion methods depend on how the implication $A \rightarrow B$ is interpreted. Implication based on Mamdani's method is commonly used to compose the fuzzy relation R , particularly in control applications given in Eq (5.4), and is defined as follows :

$$R = A \rightarrow B = A \times B = \int_{X \times Y} (\mu_A(x) \wedge \mu_B(y)) / (x, y) \quad (5.4)$$

This conversion is based on the definition of the Cartesian product of two fuzzy sets given in Eq (5.5):

$$a \rightarrow b = a \wedge b \quad (5.5)$$

Implication $A \rightarrow B$ based on Mamdani's method can be summarized as Eq (5.6):

$$\mu_R(x, y) = \mu_A(x) \wedge \mu_B(y) \quad (5.6)$$

This is the case where there is one input and one output or, single-input-single-output (SISO) fuzzy system. Now consider a fuzzy system with two inputs and one output, as:

$$\text{IF } x \text{ is } A \text{ and } y \text{ is } B \text{ THEN } z \text{ is } C \quad (5.7)$$

where A , B and C are the fuzzy subsets of X , Y , and Z , respectively. The rule (5.7) can be simplified as:

$$A \text{ and } B \rightarrow C \quad (5.8)$$

The cylindrical extension [Appendix A.16] is applied as follows:

$$“x \text{ is } A \text{ and } y \text{ is } B” \Leftrightarrow (x, y) \text{ is } (A \times Y) \cap (X \times B) \quad (5.9)$$

$$(x, y) \text{ is the cross product } A \times B \quad (5.10)$$

Therefore, the fuzzy relation “ A and B ” can be expressed as Eq (5.11):

$$A \text{ and } B = A \times B = \int_{X \times Y} (\mu_A(x) \wedge \mu_B(y)) / (x, y) \quad (5.11)$$

Hence, the rule “ A and $B \rightarrow C$ ” can be translated as:

$$R = A \text{ and } B \rightarrow C = A \times B \times C \quad (5.12)$$

$$= \int_{X \times Y \times Z} (\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)) / (x, y, z) \quad (5.13)$$

$$\mu_R(x, y, z) = \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z) \quad (5.14)$$

When there are more than one rule to process, the foregoing approach can be easily extended. For example, consider the following two rules:

Rule 1: IF x is A_1 THEN y is B_1

Rule 2: IF x is A_2 THEN y is B_2

These two rules may be rewritten as:

$$\begin{aligned} A_1 &\rightarrow B_1 \text{ else} \\ A_2 &\rightarrow B_2 \end{aligned} \quad (5.15)$$

These two rules are converted into two fuzzy relations as follows:

$$\begin{aligned} A_1 &\rightarrow B_1 \Rightarrow R_1 \\ A_2 &\rightarrow B_2 \Rightarrow R_2 \end{aligned} \quad (5.16)$$

In Mamdani's method "else" is interpreted as an "OR" operation. Hence the compiled fuzzy relation R is given as:

$$R = R_1 \cup R_2 \quad (5.17)$$

Example:

$X = \{x_1, x_2, x_3\}$ and $A_1, A_2 \subset X$

$Y = \{y_1, y_2, y_3\}$ and $B_1, B_2 \subset Y$

Suppose that the fuzzy sets A_1, A_2, B_1 and B_2 are given by:

$$\begin{aligned} A_1 &= \frac{1.0}{x_1} + \frac{0.6}{x_2} + \frac{0.0}{x_3} \\ A_2 &= \frac{0.0}{x_1} + \frac{0.8}{x_2} + \frac{1.0}{x_3} \\ B_1 &= \frac{1.0}{y_1} + \frac{0.6}{y_2} + \frac{0.1}{y_3} \\ B_2 &= \frac{0.2}{y_1} + \frac{0.8}{y_2} + \frac{0.9}{y_3} \end{aligned} \quad (5.18)$$

First we should convert the rule: Rule 1 $A_1 \rightarrow B_1$ into the fuzzy relation R_1 . With the "min" operation, we get:

(5.19)

$$R_1 = \begin{array}{cc} & \begin{array}{ccc} \mu_{B_1}(y_1) & \mu_{B_1}(y_2) & \mu_{B_1}(y_3) \\ 1.0 & 0.6 & 0.1 \end{array} \\ \begin{array}{c} \mu_{A_1}(x_1) \\ \mu_{A_1}(x_2) \\ \mu_{A_1}(x_3) \end{array} & \begin{array}{c} 1.0 \\ 0.6 \\ 0.0 \end{array} \end{array} \begin{array}{|ccc|} \hline 1.0 & 0.6 & 0.1 \\ 0.6 & 0.6 & 0.1 \\ 0.0 & 0.0 & 0.0 \\ \hline \end{array}$$

or,

$$R_1 = \begin{bmatrix} 1.0 & 0.6 & 0.1 \\ 0.6 & 0.6 & 0.1 \\ 0 & 0 & 0 \end{bmatrix} \quad (5.20)$$

Next we convert the rule: Rule 2 $A_2 \rightarrow B_2$ into the fuzzy relation R_2 .

(5.21)

$$R_2 = \begin{array}{cc} & \begin{array}{ccc} \mu_{B_2}(y_1) & \mu_{B_2}(y_2) & \mu_{B_2}(y_3) \\ 0.2 & 0.8 & 0.9 \end{array} \\ \begin{array}{c} \mu_{A_2}(x_1) \\ \mu_{A_2}(x_2) \\ \mu_{A_2}(x_3) \end{array} & \begin{array}{c} 0.0 \\ 0.8 \\ 1.0 \end{array} \end{array} \begin{array}{|ccc|} \hline 0.0 & 0.0 & 0.0 \\ 0.2 & 0.8 & 0.8 \\ 0.2 & 0.8 & 0.9 \\ \hline \end{array}$$

or,

$$R_2 = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.2 & 0.8 & 0.8 \\ 0.2 & 0.8 & 0.9 \end{bmatrix} \quad (5.22)$$

Finally, we form the overall rulebase relation R as:

$$R = R_1 \cup R_2 = \begin{bmatrix} 1.0 & 0.6 & 0.1 \\ 0.6 & 0.8 & 0.8 \\ 0.2 & 0.8 & 0.9 \end{bmatrix} \quad (5.23)$$

Once R is formed in this manner, for a given input of fuzzy set A' on X , the output fuzzy set B' on Y can be obtained as:

$$B' = A' \circ R \quad (5.24)$$

where “ \circ ” is the *composition* operation. This is the principle of composition-based inference. The composition by “ \circ ” is quite similar to the multiplication of matrices. Specifically, if the multiplication and the addition in a matrix multiplication are replaced by the min-operation and the max-operation, respectively, then the composition operation is realized.

As an illustrative example, suppose that the input fuzzy set A' on X is

$$A' = \frac{0.8}{x_1} + \frac{0.3}{x_2} \quad (5.25)$$

The outcome (inference) of the reasoning, which is B' on Y , is given by

$$B' = A' \circ R \quad (5.26)$$

$$\begin{array}{ccccccc}
 & & & & y_1 & y_2 & y_3 \\
 x_1 & x_2 & x_3 & x_1 & 1.0 & 0.6 & 0.1 \\
 = [0.8 & 0.3 & 0] \circ & x_2 & 0.6 & 0.8 & 0.8 \\
 & & & x_3 & 0.2 & 0.8 & 0.9
 \end{array} \quad (5.27)$$

Then,

$$\begin{aligned}
 B' = & [(0.8 \wedge 1.0) \vee (0.3 \wedge 0.6) \vee (0 \wedge 0.2), \\
 & (0.8 \wedge 0.6) \vee (0.3 \wedge 0.8) \vee (0 \wedge 0.8), \\
 & (0.8 \wedge 0.1) \vee (0.3 \wedge 0.8) \vee (0 \wedge 0.9)]
 \end{aligned} \quad (5.28)$$

or,

$$B' = [0.8 \vee 0.3 \vee 0 \quad 0.6 \vee 0.3 \vee 0 \quad 0.1 \vee 0.3 \vee 0] \quad (5.29)$$

which gives,

$$B' = [0.8 \quad 0.6 \quad 0.3] \quad (5.30)$$

This composition-based inference method is implemented in real time on the prototype hydraulic manipulator. In order to be able to compare the results of this method with the individual rule-based inference method used in Chapter 4, we implement the P-type fuzzy logic controller with

the same membership functions shown in Figure 4.20 and the rulebase given in Figure 4.17. In this manner, responses shown in Figure 4.21 can be compared with the results obtained in the present chapter. The main steps of the present implementation are given below:

Step 1: The membership functions of the input and the output (Figure 4.20) are discretized (Figure 5.1 and 5.2). Pseudo code of the algorithm of this method, formulates the memberships of the input in the form of a two dimensional array, where one element of the pair indicates the fuzzy state of the membership function (NS,ZR,PS,...) and the other element indicates the number of the discretized point. For example, the membership functions of the input can be represented in the form :

$$\text{Mf1_input} [I] [j] \quad (5.31)$$

where i represents the fuzzy state and j represents the point number.

N = number of points of the *Input* membership function

M = number of points of the *Output* membership function

and $i = 0, 1, \dots, 6$

$j = 0, 1, \dots, N$

It follows that, for example, $\text{Mf1_input} [0] [j]$ represents the input membership function of Negative Large (NL), and so on. Accordingly, the membership functions of the input variable are as given in Figure 5.3.

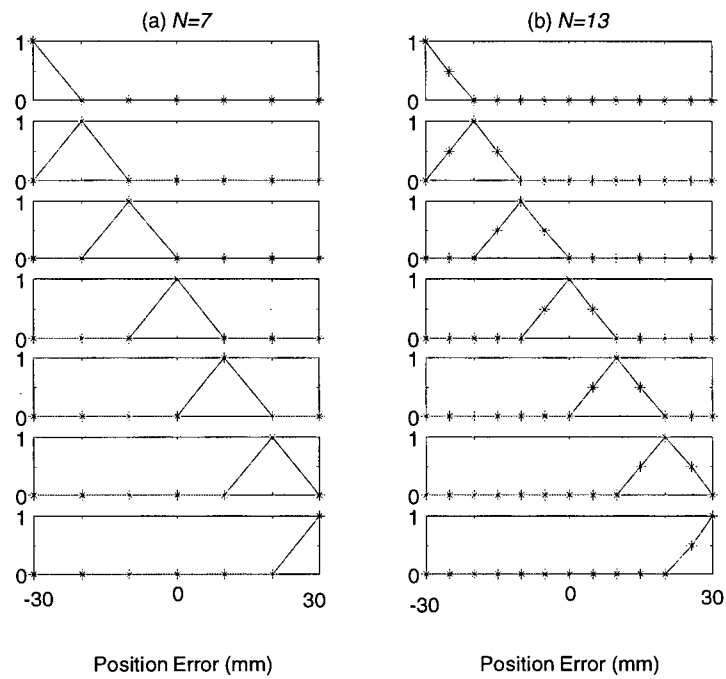


Figure 5.1 Discretized membership functions of the input.

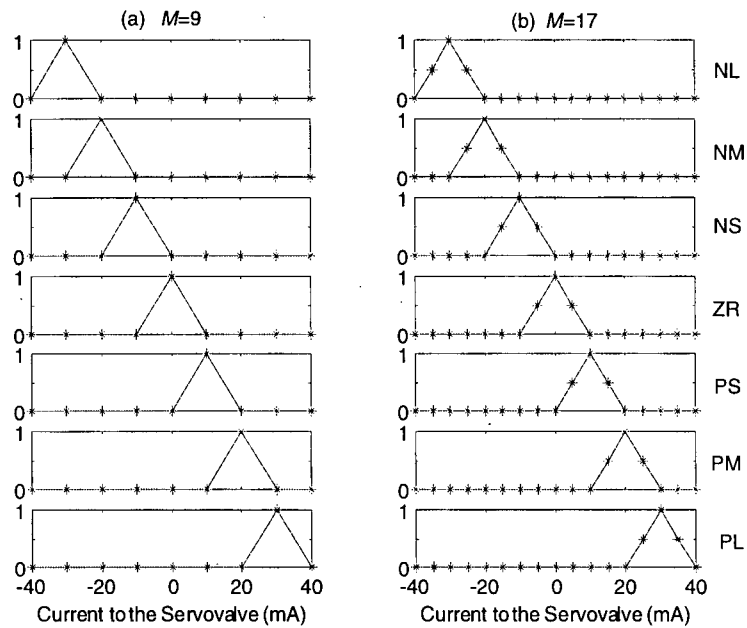


Figure 5.2 Discretized membership functions of the output.

Mf1_input [0] [j] is NL
Mf1_input [1] [j] is NM
Mf1_input [2] [j] is NS
Mf1_input [3] [j] is ZER
Mf1_input [4] [j] is PS
Mf1_input [5] [j] is PM
Mf1_input [6] [j] is PL

Figure 5.3 Membership functions of the input.

Figure 5.1 shows two cases of discretized membership functions of the input variable, where the universe of discourse is divided into N points with $N=7$ or 13. Figure 5.2 shows two cases of discretized membership functions of the output variable, where the universe of discourse is divided to M points with $M=9$ or 17. The effect of the number of discretization points on the performance of the fuzzy system is in this chapter, through experimentation.

Step 2 : The rulebase is generated in the form of a fuzzy relation, based on the discretized input and output membership functions as determined in Step1. This relation is generated based on the Mamdani min-max implication method. Note that,

N = number of discretized points in the universe of discourse of an input membership function

M = number of discretized points in the universe of discourse of an output membership function

mf1= two dimensional array of the input membership functions

mf3= two dimensional array of the output membership functions

The rulebase is given in Figure 5.4. This rulebase is the same as what is given in Figure 4.13 (Case 1). Figure 5.5 shows the pseudo code for generating the rulebase, in the form of a fuzzy relation.

pos_err is NL (mf1[0])	THEN output is	PL (mf3[6])
pos_err is NM (mf1[1])	THEN output is	PM (mf3[5])
pos_err is NS (mf1[2])	THEN output is	PS (mf3[4])
pos_err is ZR (mf1[3])	THEN output is	ZR (mf3[3])
pos_err is PS (mf1[4])	THEN output is	NS (mf3[2])
pos_err is PM (mf1[5])	THEN output is	NM (mf3[1])
pos_err is PL (mf1[6])	THEN output is	NL (mf3[0])

Figure 5.4 Rulebase of the fuzzy system.

```

MAKERULE (mf1[mf1_states][ N], mf3[mf3_states][ M], N, M)
{
    for (i=0; i<=N ;i++)
        for(k=0; k<=M;k++)
        {
            rule1 = min(mf1[0][i],mf3[4][k]); // pos_err is NL (mf1[0]) THEN output is PS(mf3[6])
            rule2 = min(mf1[1][i],mf3[4][k]); // pos_err is NM (mf1[1]) THEN output is PS(mf3[5])
            rule3 = min(mf1[2][i],mf3[4][k]); // pos_err is NS (mf1[2]) THEN output is PS(mf3[4])
            rule4 = min(mf1[3][i],mf3[3][k]); // pos_err is ZR (mf1[3]) THEN output is ZR(mf3[3])
            rule5 = min(mf1[4][i],mf3[2][k]); // pos_err is PS (mf1[4]) THEN output is NS(mf3[2])
            rule6 = min(mf1[5][i],mf3[2][k]); // pos_err is PM (mf1[5]) THEN output is NS(mf3[1])
            rule7 = min(mf1[6][i],mf3[2][k]); // pos_err is PL (mf1[6]) THEN output is NS(mf3[0])

            max1=max(rule1,rule2);
            max2=max(max1,rule3);
            max3=max(max2,rule4);
            max4=max(max3,rule5);
            max5=max(max4,rule6);
            max6=max(max5,rule7);
            rulebase[i][k]=max6;
        }
    return (rulebase[i][k]);
}

```

Figure 5.5 The rulebase in the form of fuzzy relation.

Step 3: The crisp input is changed to a fuzzy singleton shown in Figure 5.6 (where it has membership value of 1.0 at the representative point of the universe of discourse, all the other points having membership values of 0.0). In this manner the crisp input is fuzzified and becomes an array of N elements where all the elements are zero except for one which has the membership grade of 1.0. The fuzzy singleton is rounded to the closest discretized point in the universe of discourse.

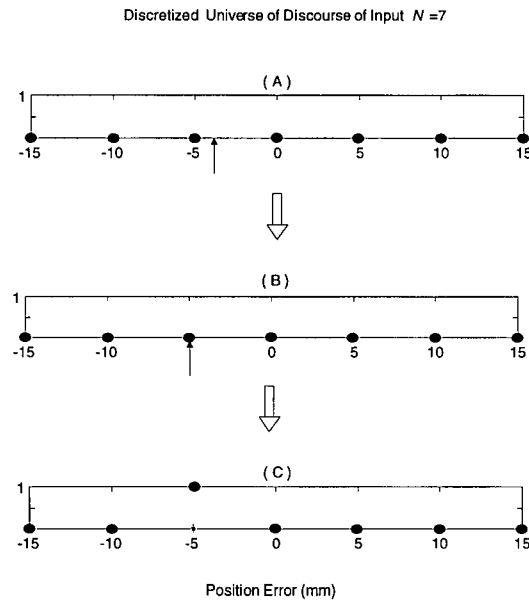


Figure 5.6 Transformation of crisp input signal to a fuzzy singleton.

Step 4: The fuzzified input (fuzzy singleton) is composed with the rulebase, using the compositional rule of inference. The result of this operation is a fuzzy decision (control inference). The fuzzy inference is in the form of a vector with M number of discretized output values. The inference engine of this fuzzy system is given in Figure 5.7.

N = number of points of the input membership function

M = number of points of the output membership function

```

INFERENCE( input [N], rulebase [N][M], N, M)
{
    for (i=0; i<=mf1_points;i++)
        for(k=0; k<=mf3_points ;k++)
        {
            m=min(input[i], rulebase[i][k]);
            infer[k]=max(infer[k],m);
        }
    return(infer[k]);
}
    
```

Figure 5.7 Inference engine of the fuzzy system.

Step 5: The crisp decision as a signal to the actuator is generated from a fuzzy decision (inference). The “centre of gravity” is used as the defuzzification method. The associated pseudocode is shown in Figure 5.8.

```

DEFUZ( infer[mf3_points], mf3_points, X3[mf3_points])
{
    for (k=0; k<=mf3_points; k++)
    {
        weighted_sum+=X3[k]*infer[k];
        sum+=infer[k];
    }
    signal=weighted_sum/sum;
    return(signal);
}

```

Figure 5.8 Defuzzification scheme for the fuzzy system.

In implementation of the “Composition Based Inference” in real time, steps 1 and 2 are carried out off line. However, the steps 3, 4 and 5 are performed at each sampling period. Figures 5.9 through 5.12 show the response of the system for four different cases of discretization levels for the input and output membership functions, where;

N = number of discretized points in the universe of discourse of input membership function.

M = number of discretized points in the universe of discourse of output membership function.

In each case, for experiments (a)-(d) are conducted with different initial conditions and set points, giving two directions of actuator motion.

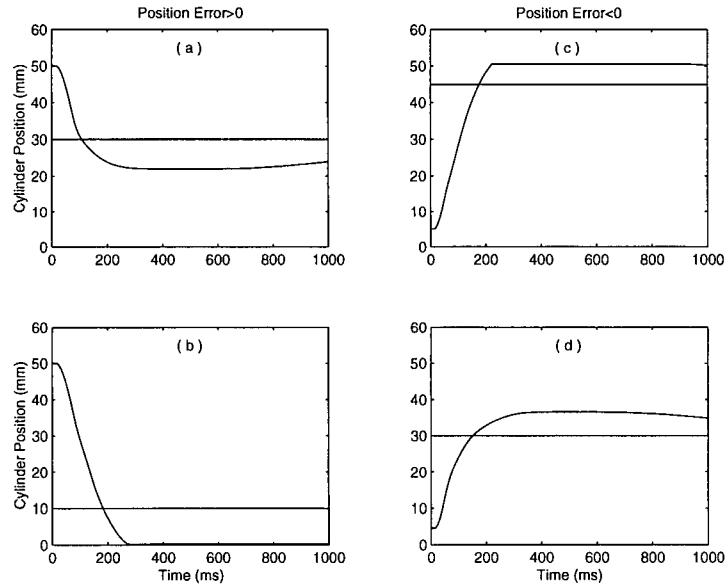


Figure 5.9 Response of the system in the case $N=7$ and $M=9$.
(Four initial error conditions (a)-(d)).

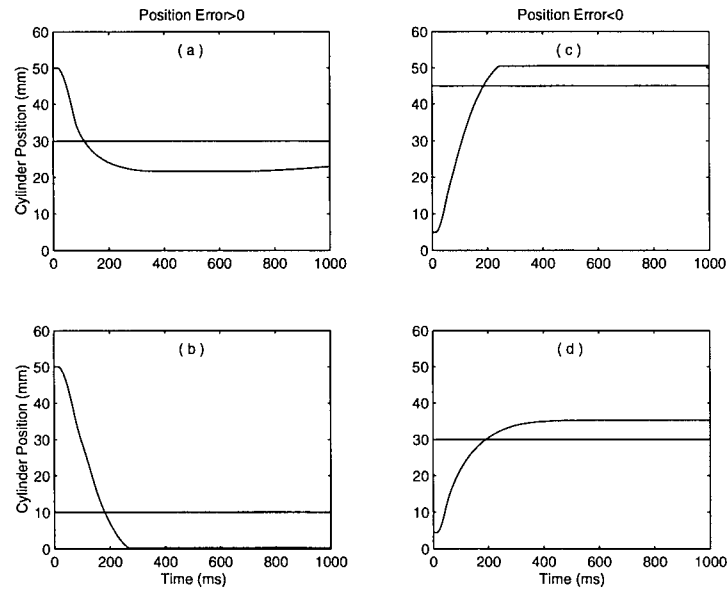


Figure 5.10 Response of the system in the case $N=13$ and $M=9$.
(Four initial error conditions (a)-(d)).

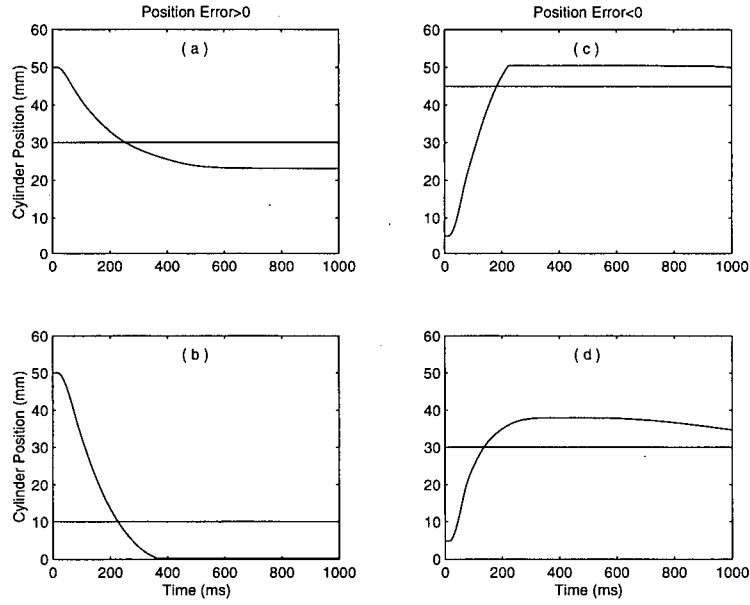


Figure 5.11 Response of the system in the case $N=7$ and $M=17$. (Four initial error conditions (a)-(d)).

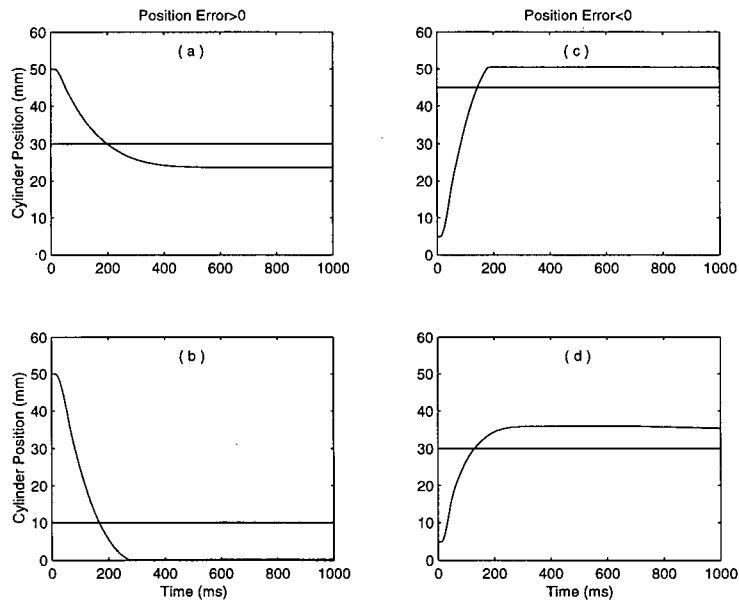


Figure 5.12 Response of the system in the case $N=13$ and $M=17$. (Four initial error conditions (a)-(d)).

5.2 Discussion

The fuzzy controller based on the “compositional rule of inference” operates in two phases, one offline and the other processed at each sampling time. Steps 1 and 2, which are discretization of the membership functions and generation of the rulebase, are processed offline. Depending on the number of discretization points, the dimension of the fuzzy relation can be quite large, requiring a large memory. The effect of the number of discretization points on the performance of the system has not been discussed in the literature. However, in the present experiments, two different discretization levels were used for the input membership functions and two different discretization levels were used for the output membership functions as well. The reason that the effect of the number of discretization points would be of interest is the fact that the size of the created fuzzy relation is influenced by number of these points, which can be shown using Figure 5.5 and Eq (5.20) and (5.22). The higher the resolution of the points on the universe of discourse, the greater the memory requirement for storing the fuzzy relation. Two sets of discretization levels are compared for input and output membership functions. The number of discretization points for the input membership functions is chosen as $N=7$ or 13 the number of discretization points for the output membership functions is chosen as $M=9$ or 17. Figures 5.1 and 5.2 show the rationale for choosing the particular numbers for N and M . The universe of discourse is divided evenly so that at least one point has the membership grade of 1.0 for each state of $(NL, NM, NS, ZR, PS, PM, PL)$. In other words, all the membership functions are designed to be *normal*, meaning the *height* of the set is 1.0 (see Appendix A). As shown in Figure 5.1(a), the universe of discourse is $[-30,30]$ mm. Figure 5.1(b) shows the universe of discourse $[-30,30]$ mm, which has been divided evenly into smaller segments of 5mm. In this case, we have more than one point that has a membership grade greater than zero and has small enough resolution (5mm) to realize the effect of changing the number of discretization points for the input membership function. It is not necessary to divide the universe of discourse into equal segments. However, for the sake of comparison and to study the effect of resolution of the universe of discourse in terms of the number of discretization points, the universe of discourse is divided into equal segments. The resulting membership functions, as shown in figures 5.1 and 5.2 are exactly the same as the membership functions shown in Figure 4.20. In figures 5.1 and

5.2 , each state (NL, NM, NS, ZR, PS, PM, PL) is plotted separately, while in Figure 4.20 all the states are plotted on a common universe of discourse. In particular, in this manner the membership function associated with a particular discretized point on the universe of discourse can be shown with improved clarity.

5.2.1 Effect of number of discretized points of input membership functions

The effect of the input membership function when the number of discretized points of the output is $M=9$, can be examined through the comparison of figures 5.9 and 5.10. The effect of the input membership function when the number of discretized points of the output is $M=17$ can be examined through the comparison of figures 5.11 and 5.12. The membership functions of the input are shown in Figure 5.1. With regard to the speed of response and the offset, the two levels of discretization both provide the same performance. Now compare the following response pairs: Figure 5.9(a) and Figure 5.10(a); Figure 5.9(b) and Figure 5.10(b); Figure 5.9(c) and Figure 5.10(c); and Figure 5.9(d) and Figure 5.10(d). The performance in the two cases shown in Figure 5.9 and Figure 5.10 appears to be quite similar. A comparison of the performance shown in Figures 5.11 and 5.12 indicates that no consistent difference exists between the two cases. In Figure 5.12, all the four cases (a),(b),(c),(d) show a stable response where it reaches a steady state with an offset, but Figure 5.11(c) shows that the system is not stable yet and has not reached a steady state. The foregoing experiments demonstrate that the number of discretization points of the input membership function does not affect the performance of the system. If the higher resolution doesn't significantly improve the performance of the system, then it is not necessary to increase the size of the fuzzy relation.

5.2.2 Effect of number of discretized points of output membership functions

The effect of the output membership function when the number of discretized points of the input is $N=7$ can be examined through the comparison of figures 5.9 and 5.11. The effect of the output membership function when the number of discretized points of the input is $N=13$ can be examined through the comparison of figures 5.10 and 5.12. The membership function of the output is shown in Figure 5.2. Figure 5.11(a) shows a stable response when compared with

Figure 5.9(a). However, the comparison of Figure 5.10(d) and Figure 5.11(d) indicates that increasing the number of discretized points of the output would not necessarily improve the performance of the system. Also, the comparison of figures 5.10 and 5.12 indicates that increasing the number of discretized points of the output does not affect the performance of the system drastically. All four cases (a),(b),(c),(d) in Figure 5.12 seem to have reached the steady state with an offset of 5mm in less than 200ms. However, this stable performance has not occurred in case (d) of Figure 5.11; case (a) of Figure 5.10; and cases (a) and (d) of Figure 5.9. It follows that, the case $N=13$ and $M=17$ in Figure 5.12 has shown the best performance among all cases.

5.2.3 Effect of inference method

The main objective of the present chapter is to compare the practical performance of the two inference methods: (individual rule-based inference and composition-based inference), through experimentation on the prototype industrial hydraulic manipulator. To accomplish this, we compare the performance of the system, while keeping the membership functions and the rulebase the same but using two different inference methods; “individual-rule-based inference” and “composition-based inference”. If these two methods produce practically the same results, then in view of the analytical developments in the area of rulebase decoupling [18,25,32,33,45] and related analytical concepts of “composition based inference”, one may proceed to employ the “individual-rule-based inference” instead, since the development of a fuzzy controller based on the “individual-rules” inference is significantly simpler. For this reason, a fuzzy controller based on the composition-based inference is developed for the P-type fuzzy controller,(with membership functions as given in Figure 4.20 and the rulebase). The performance of the composition-based inference as shown in Figure 5.12 is now compared with that of the individual-rule-based inference as shown in Figure 4.21. Figure 4.17(a) is compared with Figure 5.12(b) and Figure 4.17(b) is compared with Figure 5.12.(c). As can be seen, in all cases of composition-based inference, there is a significant offset of 5 mm; however, in individual rule-based inference, there is no offset and the response has reached the steady state, in less than 600ms.

The present experimental results show that the two inference methods do not produce the same performance. Furthermore, the composition-based inference method requires more memory. It follows that, if the composition-based inference method presents practical limitations in the development of a fuzzy controller, due to the large memory that it occupies and the rather inconvenient form of discretization of the universe of discourse, it would be advantageous to use the individual-rule- based inference method .

Chapter 6

Development of an Intelligent Hierarchical Control System for a Hydraulic Manipulator

Previous chapters investigated two types of low level direct fuzzy logic controllers with respect to such considerations as the membership functions, rulebase, inference technique, and the defuzzified approach, as applied to a hydraulic manipulator of a prototype fish processing machine. This chapter presents the development of a hierarchical intelligent control system for the same prototype hydraulic manipulator. This supervisory control system is designed to be an integral part of the prototype machine. Even though the hierarchical control system possesses some characteristics that are not common with the low-level direct controller, there are other considerations that can benefit from the knowledge of developing the low-level controllers that occupy the lowest level of the overall hierarchical control system.

6.1 Introduction

The concept of *intelligent control* was introduced by Fu [30] as an attempt to formalize in a systematic way, an engineering discipline that deals with knowledge-based control. Within this framework, a *hierarchical intelligent control* architecture was proposed by Saridis [35,68] as a unified theoretic model of cognitive and control systems methodologies.

Since an automated machine may consist of many components (sensors and actuators) that will be required to deal with a large amount of information, an efficient structure for handling information from various sources would be required. In the literature, one finds the concept of “intelligent hierarchical control”, which may be particularly suitable for complex industrial machine such as the prototype fish cutting machine as described in Chapter 1.

Saridis [4,35,68] introduced intelligent Control as a combination of three areas; Artificial Intelligence (AI), Operation Research (OR) and Control theory. He then presented the concept of “Hierarchical Intelligent Control” as a unified theoretic approach of *cognitive* and *control*

systems methodologies. Inspired by Saridis' idea, a hierarchical structure for processing the information from various sensors and actuators, based on principle of "*Decreasing precision with Increasing Intelligence*" [56], has been proposed and developed on the prototype fish cutting machine [21,84]. This hierarchical structure has been designed in such a way that the lowest level of the hierarchy is occupied by the various controllers and sensors that provide feedback information for process control. The information generated at this level is mainly visual data from the CCD cameras, position and speed data from the encoders and cutter load data from the induction motors. In this level all the information is present in large quantities and of high resolution. The second level of the hierarchy is a data processing level where high-resolution crisp data from the sensors are filtered. This filtered data (pre-processed data) provides an abstract representation of the current state of the machine. The filter operations include averaging the signal, peak detection, pattern recognition, and computation of performance attributes such as rise time, damping ratio, and settling time. These parameters are interpreted according to some subjective criteria of performance evaluation and then transcribed into fuzzy sets. The top level of the hierarchy is the knowledgebase and the inference engine which makes decisions based on the current fuzzy information about the process. The knowledgebase can be considered as a collection of expert instructions on how to improve the yield while maintaining the product requirements and keeping the individual components at their optimum performance. The information at this level, as well as the inference, are exclusively represented by fuzzy linguistic variables. The control decisions are made based on various monitored fuzzy variables, in order to improve the performance of the machine by means of adjustments made on the parameters of the process actuators. Some related works is presented in [3,9,14,46].

Based on this hierarchical concept, de Silva and MacFarlane [19,20] developed a hierarchical knowledge-based control system for a two-degree-of-freedom robot. The system consists of a robot simulator, two servo experts for the joints of the robot, and a fuzzy controller to generate the tuning commands for the PID controllers of the joints. Extending this work, Barlev [5] implemented a hierarchical intelligent tuner on a servomotor system in real-time. Wickramarachchi [21,84] developed a knowledge-based hierarchical control structure for process automation and applied it to a prototype fish-processing machine in the Industrial Automation Laboratory, University of British Columbia. This machine has been designed to

automatically cut fish, taking advantage of many sensors and actuators available in this machine. Goulet [34] implemented an intelligent hierarchical controller for a deployable manipulator, which may be used in both space and ground-based applications, where the direct controller is a proportional-integral-derivative (PID) controller for the servomotor. The chapter presents the development of an intelligent hierarchical controller for the prototype industrial fish cutting machine (see Chapter 2) in the Industrial Automation Laboratory. In this system, the lowest level consists of conventional proportional-plus-derivative (PD) controllers for the hydraulic actuators. The middle level monitors the performance of the manipulator, preprocesses the response signal and extracts the performance parameters, based on a step-input response. The top level makes inferences giving appropriate tuning actions for the PD controllers.

6.2 Development of the Hierarchical Intelligent Control System

The general structure of the developed intelligent hierarchical control system is shown in Figure 6.1. Note that at the lowest level is conventional PD controller. The second level acts as an interface between the direct controller at the bottom layer, and the knowledge-based monitoring controller at the top layer. The second layer handles such tasks as performance specification, response processing, and computation of performance indices. This stage involves, for example, averaging or filtering of the data points and computation of the rise time, overshoot, and steady state offset [24].

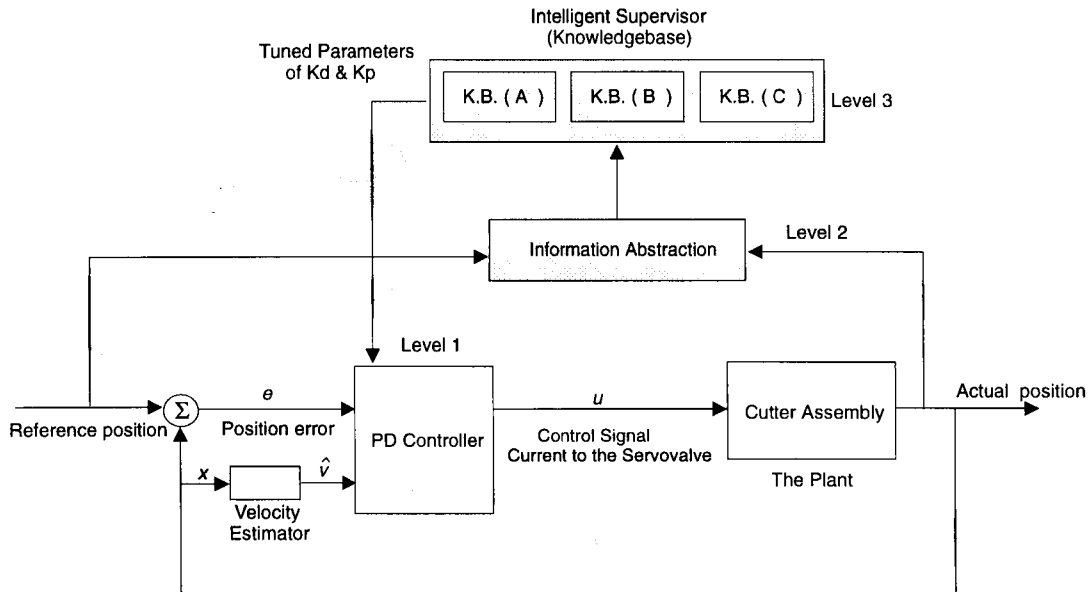


Figure 6.1 Schematic representation of the intelligent hierarchical control system.

In the first layer, a conventional PD controller uses as the control signal the current to the servovalve. This is determined using, the error (e) between the actual position (x) and the desired position (x_d) of the cylinder of the hydraulic actuator. The parameters K_p and K_d are the proportional gain and the derivative gain, respectively. The control law may be expressed as

$$i = K_p(x_d - x) + K_d\hat{v} + \Delta i \quad (6.1)$$

with

$$\Delta i = -I_d \operatorname{sgn}(x_d - x) \quad (6.2)$$

Equation (6.2) represents a correction term which is added to the control signal, to compensate for the deadband behavior of the servovalve [75,76]. The variable \hat{v} is the velocity estimated by an observer that has been designed in [75,76] and represented by the transfer function in Figure 6.2.

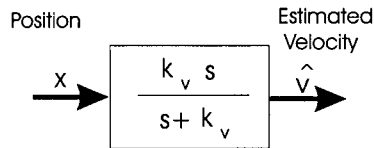


Figure 6.2 The observer for velocity estimation.

In this system, the second layer is designed to monitor the response of the system, which extracts three parameters: Overshoot in the response, oscillation in the response, and speed of response.

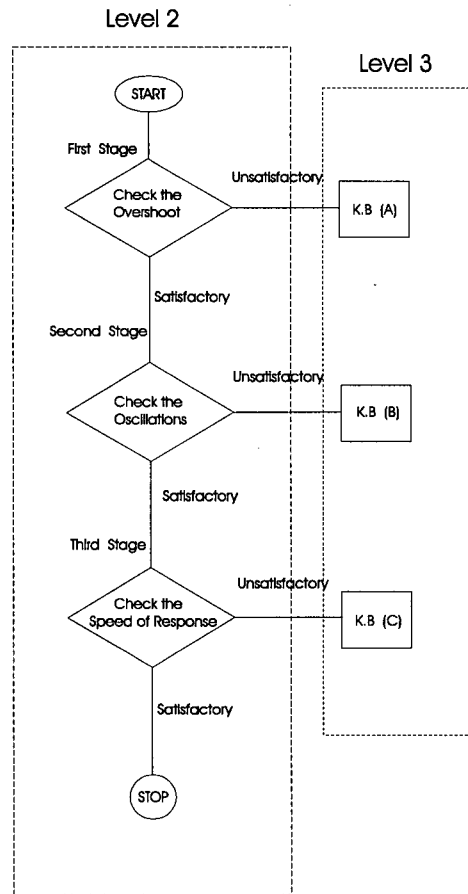


Figure 6.3 The flow chart of the algorithm of information abstraction and decision making.

As shown in the flow chart of Figure 6.3, the second layer monitors the response. The sequence of monitoring the performance has been prioritized based on the degree of importance of the parameters (Overshoot, Oscillations, Speed of Response) for the system. This system has been designed to first check for the overshoot, because the overshoot in the response is the most critical parameter and may be damaging for the system. The overshoot is calculated by determining the first peak after crossing the setpoint. The second stage in the algorithm shown in the flow chart checks for oscillations in the response, while the response does not have an overshoot. Once it detects an overshoot, the system operation immediately goes to the top level

where the knowledgebase (A) deals with unsatisfactory overshoot. The knowledge base at the top level matches with the existing information about the system, arriving from the lower level (second layer). Consequently it makes a decision for changing (tuning) the parameters of the direct controllers. After defuzzification, the resulting crisp values for changing the parameters of the controllers are sent to the corresponding direct controllers. Figure 6.4 shows the response of the system with different overshoots. An example of responses with oscillations is shown in Figure 6.5.

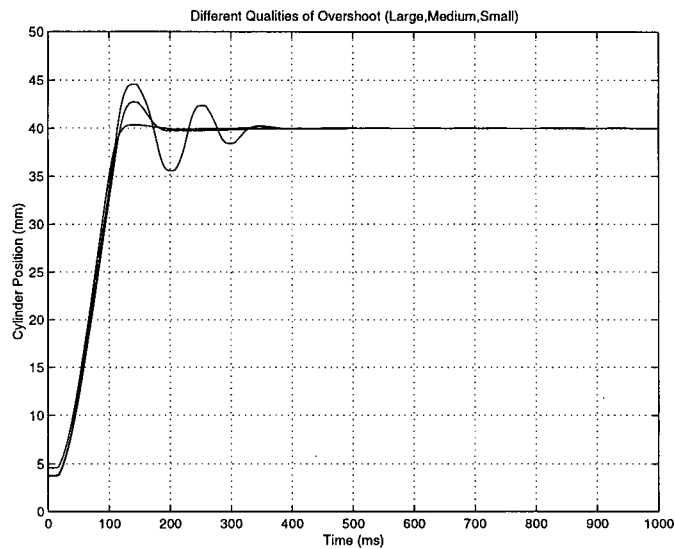


Figure 6.4 The response of system with different overshoots.

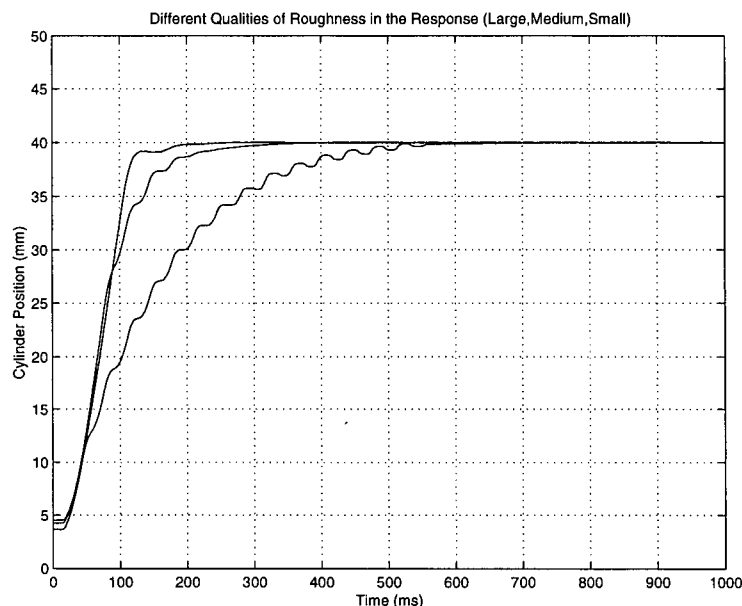


Figure 6.5 System response with different degrees of oscillations.

In the second layer, the algorithm detects the location and the magnitude of the local peaks on the response curve and describes the monitored system in a qualitative form, by the label “roughness of the response”. The algorithm detects a peak and the immediately next minimum point of the response and the separation in position between these two points. This magnitude is considered as an index for the depth of the peaks. If the roughness of the response is unsatisfactory, then the control immediately goes to the top level knowledge base (B). It deals with the oscillations in response, by commanding a change to the parameters of the controller. Table 6.1 shows the rulebase for determining the roughness of the response in the second layer of the structure.

Table 6.1. Rulebase for the roughness of the response.

IF			THEN
Peak	AND	Depth	Roughness
1		Large	Large
1		Small	Small
More than 1		Large	Large
More than 1		Small	Moderate

The third stage in the flow chart in Figure 6.3 is the second layer, which monitors the speed of response. Figure 6.6 shows three responses having different speeds.

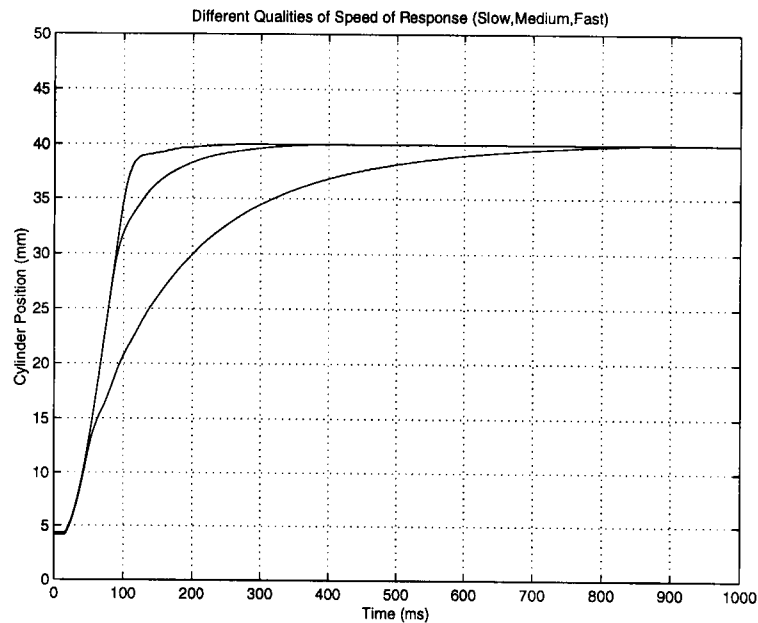


Figure 6.6 Examples of different speeds of response of the system.

The speed of response is determined by the time in which the response first reaches the desired position. The algorithm computes the index for the speed of response using this data. This index is transmitted to the top level of the knowledgebase (C). The knowledgebase in the top layer is shown in Figure 6.7 and the membership functions in the knowledgebase are shown in Figure 6.8.

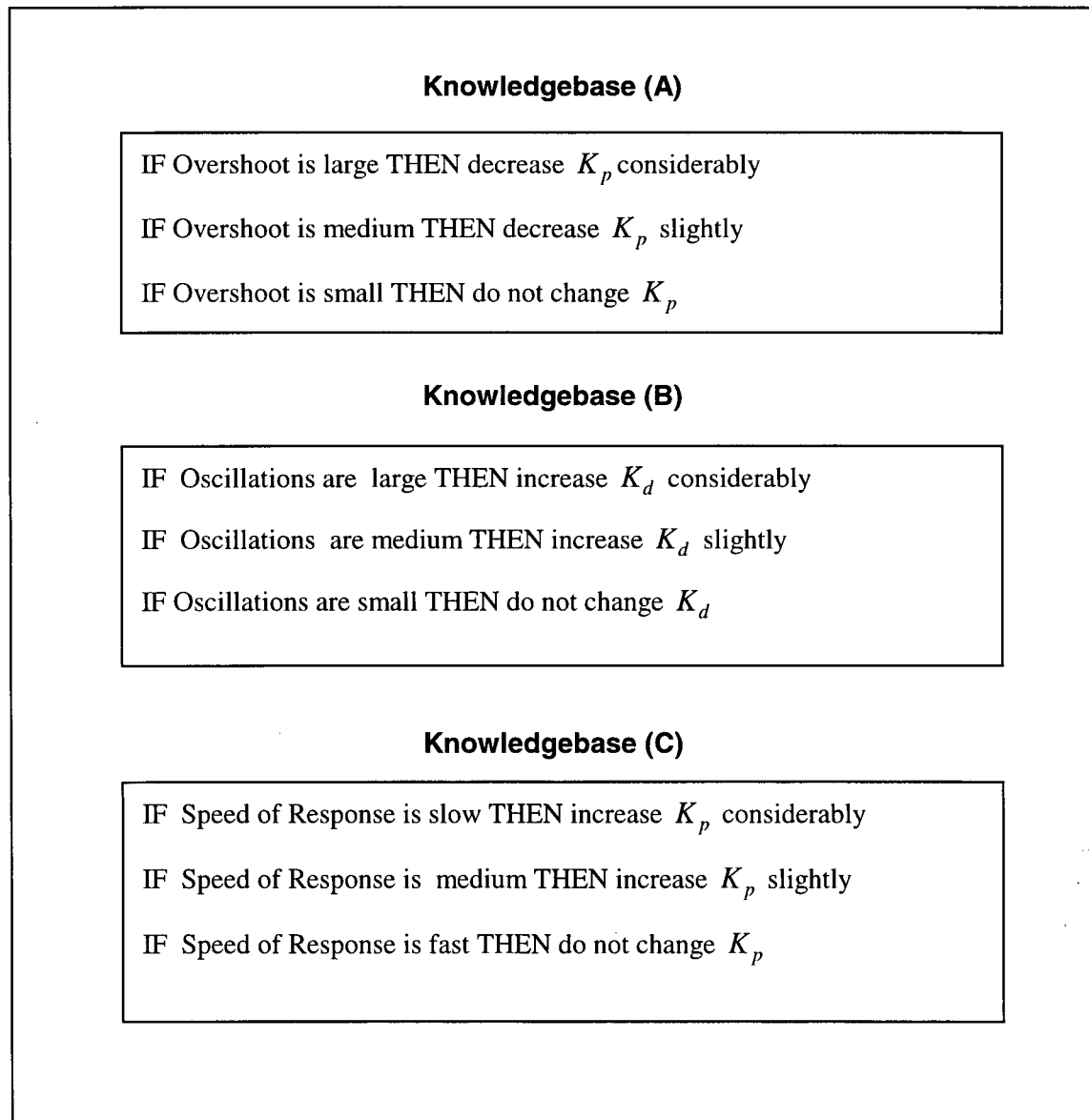


Figure 6.7 Knowledgebase of the top level of the hierarchy.

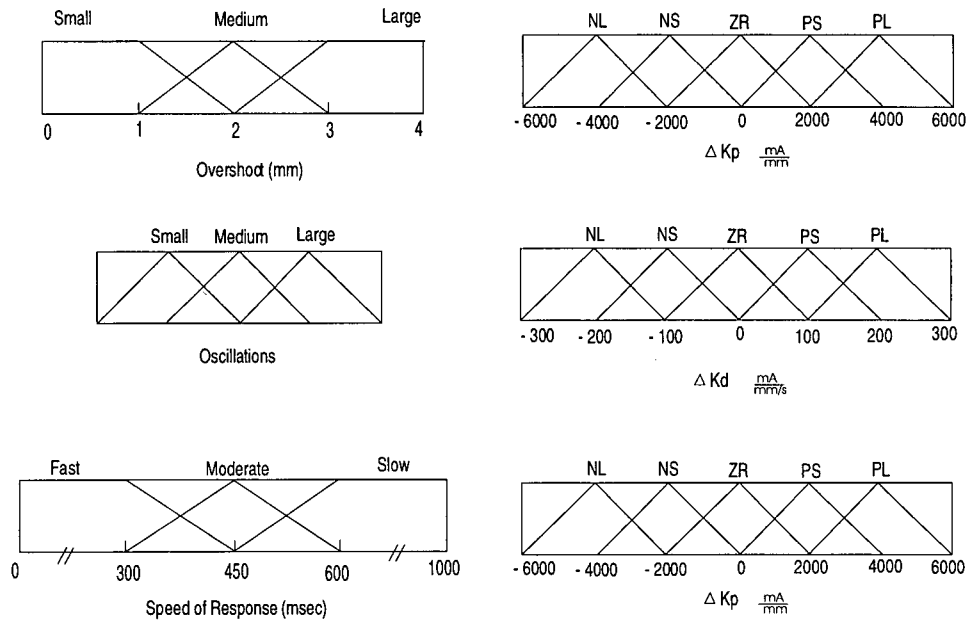


Figure 6.8 Membership functions in the knowledgebase.

6.3 Experimental Results

Figure 6.9(a) shows the response of the system before tuning where the response has a medium size overshoot, and the response after tuning where there is no overshoot. The knowledgebase (A) makes a decision to decrease the proportional gain slightly. Figure 6.9(b) shows the response of the system before tuning where the response is oscillatory. The knowledge base (B) makes a decision to increase the derivative gain slightly which improves the response, as shown in the same figure. Figure 6.9(c) shows the response of the system where the tuning action has improved the speed of response by slightly increasing the proportional gain.

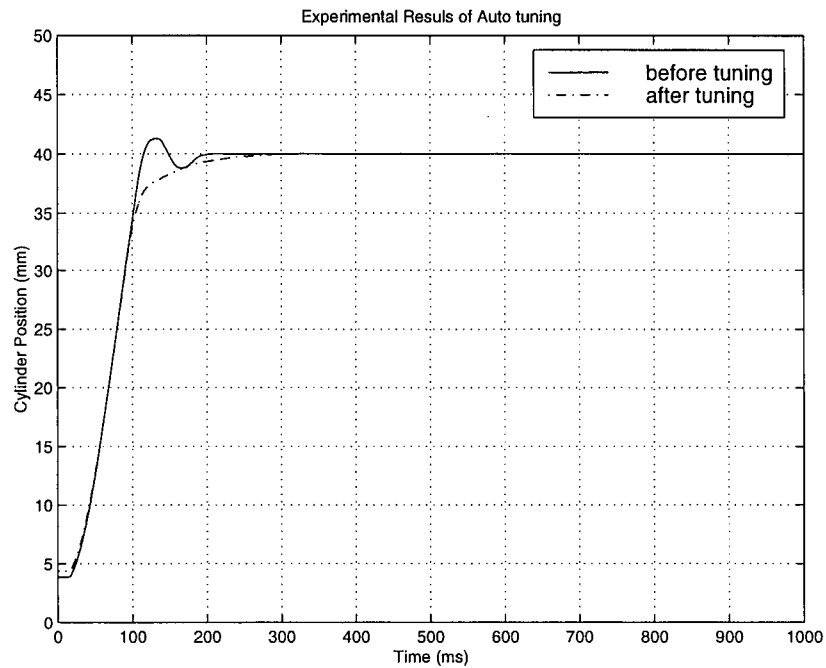


Figure 6.9.a Experimental results (overshoot improvement due to tuning).

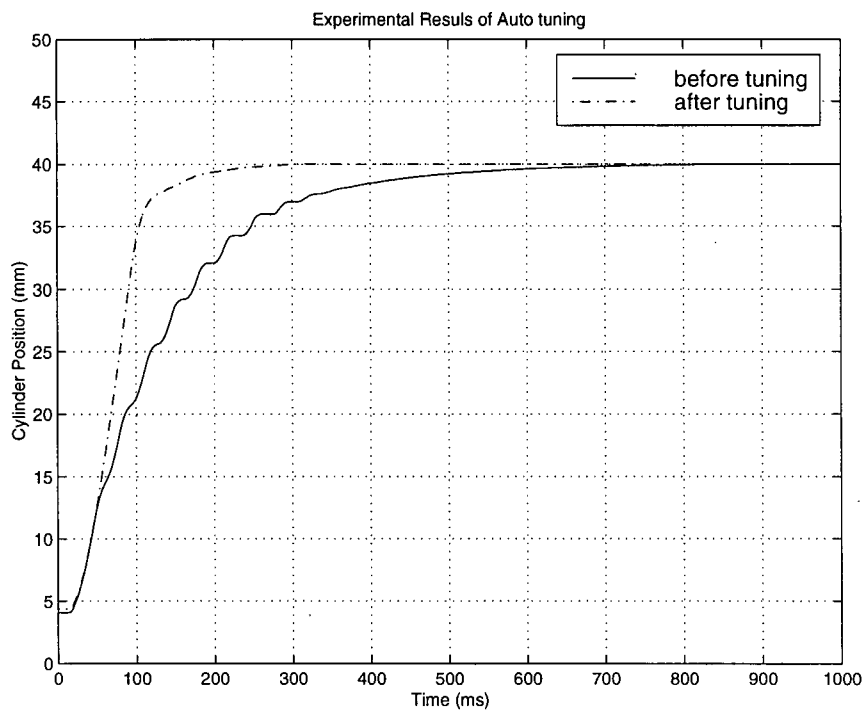


Figure 6.9.b Experimental results (improved oscillations due to tuning).

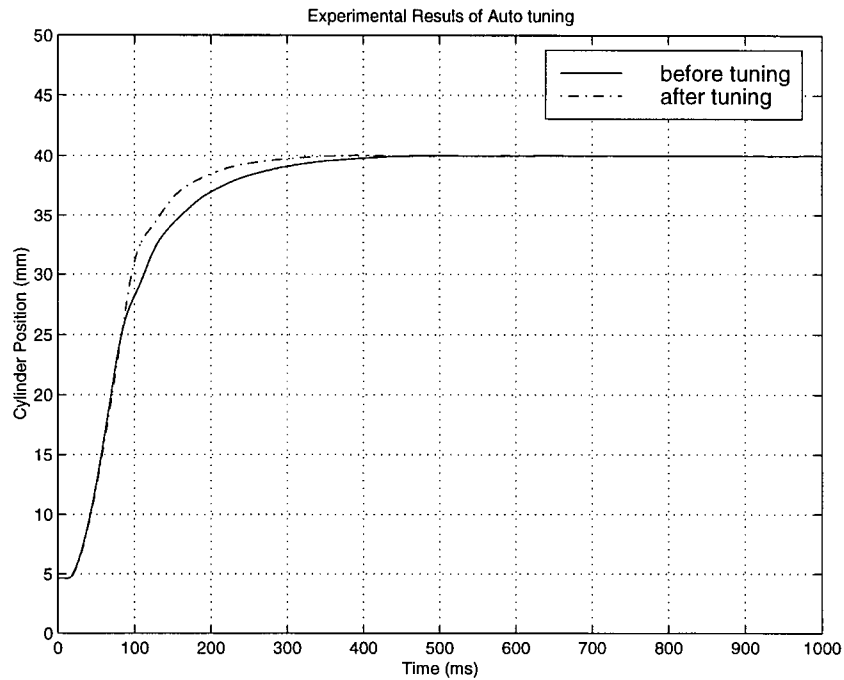


Figure 6.9.c Experimental results (improved speed of response due to tuning).

6.4 Discussion

The period of the cyclic motion of the conveyor is 1.24 seconds. The fish are stationary during half the period (620 ms) and moving during the other half period. While a fish is moving (i.e., being pushed forward by the conveyor), the manipulator is at rest for half a cycle (620 ms). The tuning action at the higher level takes place during this 620ms. In the control system of the manipulator, the performance of the direct controller is monitored after completing each cutting task. Therefore, the computation of the information abstraction in the second level and the decision making in the knowledge base at the top level, have to be completed in 620 ms.

Knowledge base modules at the top level are designed in the form of SISO fuzzy rules where they make tuning decisions regarding one parameter of the controller at a time. This means that in each step of tuning, only one parameter is tuned based on the priority of the performance

indices. Since the membership functions are designed in such a form that each crisp value can represent at most two fuzzy variables, at each time instant a maximum of two rules would be fired and processed. Consequently, the overall computations can be carried out in 620ms or less. The rulebase modules are based on both common knowledge of control and extensive experimentation on the setup. Mamdani's method of fuzzy reasoning is used and the center of gravity method is used for defuzzification.

Chapter 7

A New Inference Method Based on Fuzzy Integral

From the investigation that has been carried out in the previous chapters it is clear that the method of decision making, or inference generation, plays a crucial role in a fuzzy expert system, particularly a fuzzy logic controller. Two common approaches of inference generation—the individual-rule-based method and the composition-based method are investigated and experimentally evaluated in Chapter 2 through 5, giving their respective advantages and shortcomings. Clearly, there is room for improvement in these inference methods. The present chapter develops a new fuzzy inference method, which is particularly useful for multivariable fuzzy systems. This method is developed using the concept of fuzzy integral. In the first section, the purpose of the development, the motivation, and the rationale behind the new inference method are explained, followed by literature review from different view points. Then the required theoretical background for understanding the basis of the new inference method is explained in section 7.2. The detailed development of the proposed new inference method is explained in section 7.3, followed by the experimental results and discussions.

7.1 Introduction

In complex processes, often many variables have to be monitored by the control system and many parameters need to be changed accordingly. In a multi-input-multi-output (MIMO) system, it is difficult to expect a human to keep track of a large number of interacting variables and to express the expertise as a set of coupled rules consisting of all input and output variables. Even if it is possible to build a multi-input-multi-output (MIMO) rulebase system, the issue of speed of processing needs to be addressed. In view of this, it is quite appealing to consider the possibility of decoupling the MIMO rulebase into a set of single-input-single-output (SISO) rules. This chapter develops a novel method of inference which is able to deal with multivariable fuzzy systems.

7.1.1 Purpose of the development

This chapter presents an approach for processing large numbers of fuzzy variables that are present in complex knowledge-based expert systems. Motivated by the practical problems that are present in complex industrial machines, the present chapter focuses on processing linguistic fuzzy information from many different sources, as used in practical expert systems and supervising control systems. The associated problem may be interpreted as both an inference method for MIMO fuzzy rulebases that are employed in knowledge-based systems such as control expert systems, and information fusion in complex systems that use expert systems. The method introduces a structured way to organize the variables and build a manageable rulebase. It is based on the concept of fuzzy measure and fuzzy integral, which has been introduced by Sugeno [72,74], and has been inspired by the work done on information fusion in computer vision using the fuzzy integral [77].

7.1.2 Motivation

The need to develop an effective inference method for multivariable fuzzy systems is motivated by the control needs of an automated fish cutting machine, two prototypes of which have been developed at the Industrial Automation Laboratory of the University of British Columbia (see Chapter 1). The industrial prototype is a complex machine, which consists of many sensors, actuators, controllers, and mechanical devices. The overall performance of the machine depends on improving the meat recovery and the quality of the cut fish, which are subjective measures. Autonomous operation of the machine relies on an intelligent supervisory control system which is based on fuzzy logic. The associated fuzzy systems are multivariable. The realtime operation of such control expert systems will considerably benefit from fast and accurate inference methods.

7.1.3 Rationale

As noted before in this thesis, in order to control a process using a fuzzy knowledge-based technique, the control decisions have to be made based on monitored process variables. For this purpose, a knowledgebase in the form of fuzzy rulebase has to be developed. During the system operation, the monitored process variables are matched with the knowledgebase and the

control decisions are made. There are many variables in a “multi-input-multi- output (MIMO) systems”. Since each fuzzy rule in the knowledgebase is developed based on the expertise of a human expert, expecting a human to keep track of many variables at a time is rather unrealistic, since a human may typically react to a single response variable before turning attention to another response variable. However, in a complex system, the number of process variables (number of information sources) can be quite large. Even when the MIMO system can be built, still the speed of processing would be low. To overcome this problem, one needs to find a way to separate the system into modules of SISO rulebases.

7.1.4 Related literature

The problem of dealing with large fuzzy rule-based systems (MIMO fuzzy rulebases) has been investigated from the point of view of number of rules in [62]. The main issue addressed in that work is to reduce the number of fuzzy rules in a control system. In a complete fuzzy rulebase, the number of rules increases exponentially with the system variables. Specifically, a complete set of rules in a rulebase can have m^n different rules, where m is the number of fuzzy states in a variable and n is the number of system variables in the rulebase. The work in [62] has proposed a method to reduce the number of rules so that it would have a linear relation with number of system variables. Their method is based on a hierarchical structure in which the most influential parameters are chosen as the system variables in the first level, the next most important parameters are chosen as the system variables in the second level, and so on. In the hierarchy, the first level gives an approximate output, which is then modified by the rule set in the second level. The variables in the second level include the approximate output of the first level and system variables. This process is repeated in succeeding levels of hierarchy. The total number of rules is equal to the sum of the rules at each level. Apart from reducing the number of rules, their approach introduces an architecture where the large number of system variables can be organized in a hierarchical manner. In a simulation example, they have shown that this method would be useful when the system is such that it is possible to construct the rulebase in a form where the output of one set in the low level of the hierarchy can be treated as one of the input variables of the set of rules in the next level. They have applied their fuzzy control architecture to control the feedwater flow to a steam generator of a power plant. Their simulation

results show that the hierarchical fuzzy controller has a superior performance over a conventional proportional-integral-derivative (PID) controller.

Large rulebases have also been studied from another point of view--the decomposition of MIMO fuzzy rulebases. There are two primary methods of making inferences using a fuzzy rulebase [28]; namely, the "compositional rule of inference" of Zadeh[102] and "direct fuzzy inference" of Mamdani [52]. One group of researchers have approached the problem of simplifying the MIMO fuzzy rulebase by decomposing it into fuzzy relations [25,26,32,33,36,45,48,67,81,82,88,89,90]. Assuming that the fuzzy relations of the MIMO rulebase can be easily constructed, they have proposed methods to manipulate and simplify the fuzzy relations so that the number of real-time calculations required for inference making can be reduced and as a result the control signal could be computed faster. It has been shown that complex rules can be broken up into simple rules when the rules are represented with any S-implication or any R-implication [16]. Based on the method presented in [16], a method has been introduced for parallel processing of fuzzy rules [53]. Some related works are presented in [39,41,80,91].

A method for decomposing a MIMO fuzzy rulebase has been proposed in [105] where the input space is divided into sub-input spaces so that a general fuzzy system can be decomposed into several simpler sub-fuzzy systems. That paper argues that in practice a fuzzy rulebase is constructed based on the information of human experts. Since it is rather difficult for a human to keep track of many variables; say, more than two variables at a time, forming a large rulebase incorporating many input fuzzy variables may be practically somewhat unrealistic. Decomposition of a MIMO system has been approached in [38] through a structure called "Network of Fuzzy Adaptive Nodes" (NetFAN) which is a structured network of simple, interacting fuzzy or neuro-fuzzy systems. Its main characteristic is that it propagates numerical values as intermediate signals. Another problem with large rulebases is the speed of processing. Methods have been introduced, however to handle this problem [11,15].

7.2 Theoretical Background

Fuzzy measures and fuzzy integrals have emerged in the context of fuzzy sets. These concepts were envisioned by Sugeno [74] in his efforts to compare membership functions of fuzzy sets with probabilities. Since no direct comparison is possible, Sugeno conceived the

generalization of classical measures to fuzzy measures as analogous to the generalization of classical (crisp) sets to fuzzy sets [44,50,58,59,60,79,85,106]. Specifically, fuzzy measures are obtained by replacing the *additivity* requirement of classical measures with weaker requirements of *monotonicity* and *continuity*.

7.2.1 Fuzzy measure

Fuzzy measures are generally considered as subjective scales for fuzziness [27,74]. These are set functions that satisfy monotonicity but do not necessarily possess additivity. Let X be a nonempty set, ℓ be a nonempty class of subsets of X , and $\mu: \ell \rightarrow [0, \infty]$ be a nonnegative, extended real-valued set function defined on ℓ .

Definition 1. (Sugeno [85]): μ is called a fuzzy measure on (X, ℓ) iff (if and only if)

- 1) $\mu(\emptyset) = 0$ when $\emptyset \in \ell$ (vanishing at \emptyset);
- 2) $E \in \ell$, $F \in \ell$, and $E \subset F$ imply $\mu(E) \leq \mu(F)$ (monotonicity);
- 3) $\{E_n\} \subset \ell$, $E_1 \subset E_2 \subset \dots$, and $\bigcup_{n=1}^{\infty} E_n \in \ell$ imply

$$\lim_n \mu(E_n) = \mu\left(\bigcup_{n=1}^{\infty} E_n\right) \text{ (continuity from below);}$$

- 4) $\{E_n\} \subset \ell$, $E_1 \supset E_2 \supset \dots$, $\mu(E_1) < \infty$, and $\bigcap_{n=1}^{\infty} E_n \in \ell$ imply

$$\lim_n \mu(E_n) = \mu\left(\bigcap_{n=1}^{\infty} E_n\right) \text{ (continuity from above);}$$

7.2.2 λ -Fuzzy measure

Definition 2: (Sugeno [85]): μ is called a λ -fuzzy measure on ℓ iff it satisfies the $\sigma - \lambda$ -rule on ℓ and there exists at least one set $E \in \ell$ such that $\mu(E) < \infty$.

Definition 3: (Sugeno [85]):

- (a) μ satisfies the “ λ -rule” (on ℓ) iff there exists

$$\lambda \in \left(-\frac{1}{\sup \mu}, \infty\right) \cup \{0\}$$

where $\sup \mu = \sup_{E \in \ell} \mu(E)$, such that

$$\mu(E \cup F) = \mu(E) + \mu(F) + \lambda \mu(E) \mu(F) \quad (7.1)$$

whenever

$$E \in \ell, F \in \ell, E \cup F \in \ell, \text{ and } E \cap F = \phi.$$

(b) μ satisfies the “finite λ -rule” (on ℓ) iff there exists the above-mentioned λ such that

$$\mu\left(\bigcup_{i=1}^n E_i\right) = \begin{cases} \frac{1}{\lambda} \left\{ \prod_{i=1}^n [1 + \lambda \mu(E_i)] - 1 \right\}, & \text{for } \lambda \neq 0, \\ \sum_{i=1}^n \mu(E_i), & \text{for } \lambda = 0, \end{cases} \quad (7.2)$$

for any finite disjoint class $\{E_1, \dots, E_n\}$ of sets in ℓ whose union is also in ℓ .

(c) μ satisfies the “finite σ - λ -rule” (on ℓ) iff there exists the above-mentioned λ such that

$$\mu\left(\bigcup_{i=1}^{\infty} E_i\right) = \begin{cases} \frac{1}{\lambda} \left\{ \prod_{i=1}^{\infty} [1 + \lambda \mu(E_i)] - 1 \right\}, & \text{for } \lambda \neq 0, \\ \sum_{i=1}^{\infty} \mu(E_i), & \text{for } \lambda = 0, \end{cases} \quad (7.3)$$

for any disjoint sequence $\{E_n\}$ of sets in ℓ whose union is also in ℓ .

Usually the λ -fuzzy measure is denoted by g_λ . When ℓ is a σ -algebra (Appendix B) and $g_\lambda(X) = 1$, the λ -fuzzy measure g_λ is called a Sugeno measure.

Theorem 1: (Sugeno [85]): If g_λ is a λ -fuzzy measure on a class ℓ containing the empty set ϕ , then $g_\lambda(\phi) = 0$, and g_λ satisfies the finite λ -rule.

If $X = \{x_1, \dots, x_n\}$ is a finite set, ℓ consists of X and all singletons of X , and μ is defined on ℓ such that $\mu(\{x_i\}) < \mu(X) < \infty$ for $i=1, 2, \dots, n$, and there are at least two points x_{i_1} and x_{i_2}

satisfying $\mu(\{x_{i_j}\}) > 0, j=1, 2$, then such a set function μ is always a λ -fuzzy measure on ℓ for some parameter λ . When $\mu(X) = \sum_{i=1}^n \mu(\{x_i\})$, the parameter $\lambda=0$; otherwise λ can be determined by the following equation :

$$\mu(X) = \frac{1}{\lambda} \left[\prod_{i=1}^n (1 + \lambda \mu(\{x_i\})) - 1 \right] \quad (7.4)$$

Theorem 2: (Sugeno [85]): Under the condition mentioned above, the equation

$$1 + \lambda \mu(X) = \prod_{i=1}^n [1 + \lambda \mu(\{x_i\})] \quad (7.5)$$

determines the parameter λ uniquely. In particular,

- (1) $\lambda > 0$ when $\sum_{i=1}^n \mu(\{x_i\}) < \mu(X)$.
- (2) $\lambda = 0$ when $\sum_{i=1}^n \mu(\{x_i\}) = \mu(X)$.
- (3) $-\frac{1}{\mu(X)} < \lambda < 0$ when $\sum_{i=1}^n \mu(\{x_i\}) > \mu(X)$.

When $\mu(X) = 1$, we can determine λ by solving the following equation:

$$\lambda + 1 = \prod_{i=1}^n [1 + \lambda \mu(\{x_i\})] \quad (7.6)$$

7.2.3 Fuzzy integral

Definition 4. (Sugeno [85]): Let $A \in F$, $f \in F$. The **fuzzy integral** of f on A with respect to μ , is denoted by :

$$\int_A f d\mu = \sup_{\alpha \in [0, \infty]} [\alpha \wedge \mu(A \cap F_\alpha)]. \quad (7.7)$$

When $A = X$, the fuzzy integral may also be denoted by $\int f d\mu$. The fuzzy integral is also called *Sugeno integral*. It is sufficient to define a fuzzy integral for all $\alpha \in [0,1]$. Suppose that (X, F) is a measurable space, where $X \in F$, $\mu: F \rightarrow [0,\infty]$ is a fuzzy measure, and that F is the class of all finite nonnegative measurable functions defined on (X, F) . For any given $f \in F$, we write $F_\alpha = \{x | f(x) \geq \alpha\}$ where $\alpha \in [0,\infty]$. The set F_α is the well-known α -cut of f .

7.2.4 Information fusion using “fuzzy integral”

Consider the problem of information fusion in computer vision [12,43]. An object can be recognized and classified by integrating information from a variety of sources. This information can be used to increase the confidence in object classification. Many types of features such as shape measures, texture measures, and statistical measures may be determined through sensing and signal processing. Since, usually there is no single feature that can identify an object of interest, it would be useful to utilize several features. There is always uncertainty inherent in the recognition problem. Each feature or *group of features* can be considered as evidence in the identification of an object. Obviously, each of these features or group of features would have a degree of importance in the identification process of the object.

Fuzzy integral has been used in the recognition problem and classification integration. For improved clarity we now present the definition of fuzzy integral using the notation given in [77]:

Definition 5. (Sugeno [37,77]) Let (X, Ω) be a measurable space and let $h: X \rightarrow [0,1]$ be a Ω -measurable function. The fuzzy integral over $A \subseteq X$ of the function h with respect to a fuzzy measure g is defined by

$$\int_A h(x) \circ g(.) = \sup_{\alpha \in [0,1]} [\min(\alpha, g(A \cap F_\alpha))] \quad (7.8)$$

where

$$F_\alpha = \{x : h(x) \geq \alpha\}. \quad (7.9)$$

In [77] X has been interpreted as a set of sources of information (sensors); $h(x) \in [0,1]$ as the decision for the object when source $x \in X$ is considered; and the fuzzy measure $g(\{x\})$ (density function) as the degree of importance of this source (sensor). Corresponding to each x_i , there is a degree of importance g^i , which gives how important x_i is in the recognition of the class t_k , where $t_k \in T = \{t_1, t_2, \dots, t_m\}$, and T is a set of classes of interest. These densities can be subjectively assigned by an expert, or can be generated from training data. The method for “evidence fusion” [77] combines objective evidence (in the form of a fuzzy membership function), with subjective evaluation of the *worth of the sources* with respect to the decision. The following example illustrates the approach.

Example 1: Consider the two classes:

$$T = \{\text{beautiful}, \text{mediocre}\}$$

as applied to the appearance of a person. Keller [77] interprets X as a set of features and $h(x) \in [0,1]$ as the decision for the person’s appearance when feature $x \in X$ is considered. For class one: $t_1 = \{\text{beautiful}\}$ the feature set is $A = \{\text{eyes}, \text{nose}, \text{lips}\}$ with $x_1 = \text{eyes}$, $x_2 = \text{nose}$, $x_3 = \text{lips}$. Then, $h_1(x_1) = 0.8$, $h_1(x_2) = 0.7$, $h_1(x_3) = 0.68$ correspond to the class $t_1 = \{\text{beautiful}\}$. Note that $h_k : X \rightarrow [0,1]$ is a partial evaluation of a person for class t_k . In other words, $h_k(x_i)$ is an indication of how certain one is in the classification of the person (the object) to be in class t_k . Using the feature x_i , where a value of 1 indicates absolute certainty that the object (the person) is in class t_k and a value of 0 implies absolute certainty that the object is not in class t_k , Keller interprets the fuzzy measure $g(\{x_i\})$ as the degree of importance of the feature x_i (density function) for evaluating the person. These densities $g(\{x_i\})$ can be subjectively assigned by an expert, or can be generated from training data. Suppose that we are given;

$$g(\{x_1\}) = 0.9 \quad g(\{x_2\}) = 0.6 \quad g(\{x_3\}) = 0.85$$

In order to make a decision at this level, the class t_k with the largest integral value, is chosen. The fuzzy integral can be realized as follows [77]:

$$e = \max_{i=1}^n [\min(h(x_i), g(A_i))] \quad (7.10)$$

for the given example,

$$\min(h_1(x_1), g(x_1)) = \min(0.8, 0.9) = 0.8$$

$$\min(h_1(x_2), g(x_2)) = \min(0.7, 0.6) = 0.6$$

$$\min(h_1(x_3), g(x_3)) = \min(0.68, 0.85) = 0.68$$

It follows that

$$e = \max(0.8, 0.6, 0.68) = 0.8$$

Accordingly, it may be concluded that the appearance of person is “beautiful” with a fuzzy integral value $e = 0.8$.

Next, suppose that $h_2(x_1) = 0.3$ $h_2(x_2) = 0.35$ $h_2(x_3) = 0.2$ corresponding to class two $t_2 = \text{mediocre}$. Then, with the same density values $g(\{x_i\})$ as before, we have

$$\min(h_2(x_1), g(x_1)) = \min(0.3, 0.9) = 0.3$$

$$\min(h_2(x_2), g(x_2)) = \min(0.35, 0.6) = 0.35$$

$$\min(h_2(x_3), g(x_3)) = \min(0.2, 0.85) = 0.2$$

we get

$$e = \max(0.3, 0.35, 0.2) = 0.35$$

It follows that the appearance of the person is considered mediocre with a fuzzy integral value $e = 0.35$. Note that if the goal is to make a decision at this level, then class t_k with the largest integral value is chosen. In this simple example, the largest value of fuzzy integral is $e = 0.8$. Consequently, the class corresponding to the value “beautiful” is chosen. In the present example, the subsets of $A = \{x_1, x_2, x_3\}$ given by $\{x_1\}$, $\{x_2\}$, $\{x_3\}$ are used. In other words, the effect of each

individual feature alone is considered. The fuzzy measure corresponding to each individual feature is given by:

$$g^1 = g(\{x_1\}) = 0.9$$

$$g^2 = g(\{x_2\}) = 0.6$$

$$g^3 = g(\{x_3\}) = 0.85$$

This is called the *fuzzy measure* g . However, in the case where all the subsets are considered, as given by: $\{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_1, x_2, x_3\}$ then, all combinations of these features have to be used. In this case, the g_λ -fuzzy measure is calculated as follows:

Step 1: Determination of λ

λ can be found according to Theorem 2, Eq(7.5) and (7.6) as given below :

$$\lambda + 1 = \prod_{i=1}^n (1 + \lambda g^i) \quad (7.11)$$

For the present example then,

$$\lambda + 1 = (1 + \lambda g^1)(1 + \lambda g^2)(1 + \lambda g^3)$$

or,

$$\lambda + 1 = (1 + 0.9\lambda)(1 + 0.6\lambda)(1 + 0.85\lambda)$$

we get

$$\lambda = 0, -0.995, -2.955$$

Now, since $\lambda > -1$ and $\lambda \neq 0$, we pick $\lambda = -0.995$

Step 2: Comparison of the density values g^i

This step is carried out as given in Table 7.1.

Table 7.1 Calculation of the values of g^i .

Subset A	$g_{\lambda}(A)$
$\{x_1\}$	$g^1 = 0.9$
$\{x_2\}$	$g^2 = 0.6$
$\{x_3\}$	$g^3 = 0.85$
$\{x_1, x_2\}$	$g^1 + g^2 + \lambda g^1 g^2 = 0.96$
$\{x_2, x_3\}$	$g^2 + g^3 + \lambda g^2 g^3 = 0.94$
$\{x_1, x_3\}$	$g^1 + g^3 + \lambda g^1 g^3 = 0.99$
$\{x_1, x_2, x_3\}$	1.0

Step 3: Computation of the fuzzy integral e for class t_1

This step is carried out as given in Table 7.2.

Table 7.2. Computation of the values of the fuzzy integral for class t_1

Subset A	$g_\lambda(A)$	For the class t_1	
		h_1	$\min(h_1(x_i), g_\lambda(A_i))$
$\{x_1\}$	$g^1 = 0.9$	0.8	0.8
$\{x_2\}$	$g^2 = 0.6$	0.7	0.6
$\{x_3\}$	$g^3 = 0.85$	0.68	0.68
$\{x_1, x_2\}$	$g^1 + g^2 + \lambda g^1 g^2 = 0.96$	0.85	0.85
$\{x_2, x_3\}$	$g^2 + g^3 + \lambda g^2 g^3 = 0.94$	0.9	0.9
$\{x_1, x_3\}$	$g^1 + g^3 + \lambda g^1 g^3 = 0.99$	0.95	0.95
$\{x_1, x_2, x_3\}$	1.0	0.88	0.88
$e = \max_{i=1}^n [\min(h_1(x_i), g_\lambda(A_i))] = 0.95$			

Step 4: Computation of the fuzzy integral e for class t_2

This step is carried out as given in Table 7.3.

Table 7.3. Computation of the values of the fuzzy integral for class t_2 .

Subset A	$g_\lambda(A)$	For the class t_2	
		h_2	$\min(h_2(x_i), g_\lambda(A_i))$
$\{x_1\}$	$g^1 = 0.9$	0.3	0.3
$\{x_2\}$	$g^2 = 0.6$	0.35	0.35
$\{x_3\}$	$g^3 = 0.85$	0.2	0.2
$\{x_1, x_2\}$	$g^1 + g^2 + \lambda g^1 g^2 = 0.96$	0.51	0.51
$\{x_2, x_3\}$	$g^2 + g^3 + \lambda g^2 g^3 = 0.94$	0.35	0.35
$\{x_1, x_3\}$	$g^1 + g^3 + \lambda g^1 g^3 = 0.99$	0.1	0.1
$\{x_1, x_2, x_3\}$	1.0	0.2	0.2
$e = \max_{i=1}^n (\min(h_2(x_i), g_\lambda(A_i))) = 0.51$			

It follows that the class “beautiful” is recognized with $e = 0.95$ and the class “mediocre” is recognized with $e = 0.51$. Note that, if the goal is to make a decision at this level, the class t_k with the largest integral value should be chosen. Keller used this technique for both object classification and multisource integration [77].

7.3 Development of the inference technique

Inspired by Keller's technique for evidence fusion [77], a new inference method is proposed for knowledge-based decision making in multi-input-multi-output (MIMO) systems. The present approach uses the concepts of *fuzzy measure* and *fuzzy integral* as defined in the previous section. As mentioned in the beginning of the chapter, the development of the present technique was originally motivated by the requirements of an industrial fish cutting machine that has been developed in the Industrial Automation Laboratory [17] of the University of British Columbia (see Chapter 1, Section 1.1). A hierarchical structure has been developed for this machine, in order to deal with large amount of information that is generated [21, 84]. In the developed system, the low level information is processed and transformed into a high level linguistic form. At the top level of this hierarchy, a knowledgebase receives the preprocessed (fuzzy) information from the lower levels. In order to improve the performance of the system, the top level knowledgebase should combine the information from various sources and make a control decision, in fuzzy form.

The status of the machine is monitored through fuzzified response measurements from various sensors. The rules in the knowledge base are extracted from human knowledge and expertise. However, expecting a human to be able to relate more than two variables to one control action at a time is not realistic. To explain the basis of the present method in general terms, consider a multi-input-multi-output (MIMO) system with the four input variables: x_1, x_2, x_3, x_4 and the two control actions (output variables): y_1, y_2 . If there are 3 fuzzy states : (small, medium, large) for each input variable, then in a complete rulebase there will be $m^n = 3^4 = 81$ rules, where m is the number of fuzzy states in each input variable and n is the number of fuzzy input variables [24]. Instead of building these 81 rules using domain experts, which may not be practical, the present approach builds the knowledgebase in the form of several single-input-single-output (SISO) fuzzy rulebases and then fuse them using the fuzzy integral.

First, the knowledge base is divided to two (number of outputs) sections, as shown in Figure 7.1. The modules of rules in sections 1 and 2 run independently, since the outputs are different. Section 1 gives all the SISO rules corresponding to one output (y_1) and Section 2 gives all the SISO rules corresponding to the second output (y_2). Each section consists of a number of modules which depends on the number of input fuzzy variables. In our example there are four input variables, and accordingly, there are four modules in each section to relate each of the input variables to the particular output. Each module contains the rules that relate just one input fuzzy variable to only one output. Each module consists of several rules which depend on the number of fuzzy states in the particular input variable. For the case of three fuzzy states (for example: low, medium, high) for each variable, there will be at most three single-input-single-output rules in a module. In most practical cases, these rules are based on heuristic decisions of human experts. Since a human usually finds it difficult to make decisions simultaneous for more than three states, assigning three states for each fuzzy variable is quite reasonable and practical.

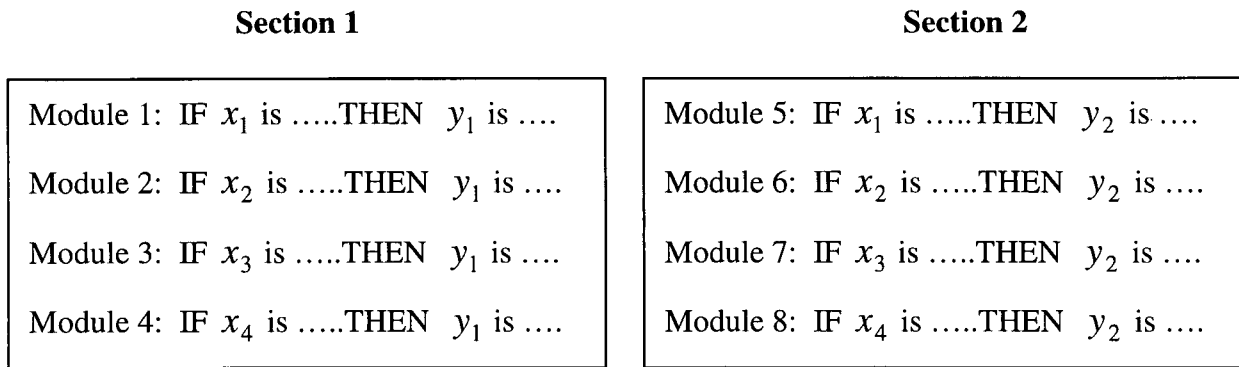


Figure 7.1. The decomposition of the MIMO rulebase into two SISO modules.

The foregoing discussion indicates that, instead of dealing with 81 rules in the form of a MIMO system, it is desirable to deal with the 8 modules shown in Figure 7.1, each module of which containing just 3 rules (for a fuzzy variable having three fuzzy states); giving a total of 24 SISO rules. Note that in the present example, four (number of modules) different decisions are made for each output. Combining these decisions and arriving at the final decision is an important consideration of the new inference method developed in this chapter. In particular, an

approach is needed to decouple the MIMO (4 inputs and 2 outputs in the present example) knowledge base into several sections (which depends on the number of outputs) of SISO modules.

The fuzzy integral is realized through the standard max-min composition of the two series $\{h(x_i)\}$ and $\{g(x_i)\}$ for each $i, i = 1, 2, \dots, n$ [77]; specifically,

$$e = \max_{i=1}^n [\min(h(x_i), g(A_i))] \quad (7.12)$$

where $A_i = \{x_1, x_2, \dots, x_i\}$ and g is a g_λ -fuzzy measure. The values of $g(A_i)$ are determined recursively as:

$$g(A_1) = g(\{x_1\}) = g^1 \quad (7.13)$$

$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}) \quad \text{for } 1 < i \leq n \quad (7.14)$$

According to Keller's interpretation of fuzzy integral (see Example 1), if one considers the fuzzy input variables x_1, x_2, x_3, x_4 as the *features or sources of information* and y_1, y_2 as the required changes for the input signals to the actuators, then density g can be interpreted as the degree of influence of the contribution of each module in achieving the overall goal of the system. Degree of influence is a subjective decision of a human expert. Specifically, a knowledge engineer (KE) would ask a human expert (HE): "if we vary the input signal to the actuator y_j by some amount, what would be its influence in improving the response variable x_i ? In the context of a complex control system (MIMO), the density function g is taken to be the degree of influence of the contribution of each SISO module in achieving the overall goal of the system. It may be possible to build MISO sub-rulebases (say, with two inputs and one output). Then the degree of influence of the contribution of each MISO module in achieving the overall goal of the system, for this module will be a g_λ -fuzzy measure, where λ is computed using Eq (7.11) and g_λ is computed using Eq (7.14), as in Step 3 of Example 1.

Example 2:

Consider a process with three response variables x_1, x_2, x_3 , which are fuzzified. A human expert relates each fuzzy variable to a control action y_1 in a decoupled (SISO) manner with a degree of importance g^i for each modules. Two cases are examined. Case 1 as shown in Figure 7.2(a) is an example of where the human expert is able to relate all the variables of the knowledgebase in a SISO manner. Case 2 as shown in Figure 7.2(c) gives a situation where in addition to the SISO modules, the expert uses a rule of thumb for the two inputs and the single output, which will contribute to the general inference. Figure.7.2 (b) shows the schematic diagram of the inference in Case 1 and Figure.7.2 (d) shows the schematic diagram of the inference in Case 2. Table 7.4 presents the membership of the output for Case 1 and Table 7.5 presents the comparison of the methods of computing inference for this case. Table 7.6 presents the membership functions of the outputs for Case2 and Table 7.7 presents the computation of inference for this case. This example clearly demonstrates the application of the inference method in the two cases. Note that the modules shown in the two cases only show the rules that are fired, not all the rules in the module. For instance, if one of the variables has three fuzzy states, three rules have to be considered for that module; but, depending on the design of the membership functions and the way they overlap different number of rules may be fired. In the present example, each crisp value can be translated to two fuzzy variables; therefore, only two rules would be fired.

Case 1:

Module 1

IF x_1 is x_1^1 THEN y_1 is y_1^2

IF x_1 is x_1^2 THEN y_1 is y_1^1

Module 2

IF x_2 is x_2^2 THEN y_1 is y_1^2

IF x_2 is x_2^3 THEN y_1 is y_1^3

Module 3

IF x_3 is x_3^1 THEN y_1 is y_1^1

IF x_3 is x_3^2 THEN y_1 is y_1^3

Figure 7.2(a) . Rulebase modules for case 1.

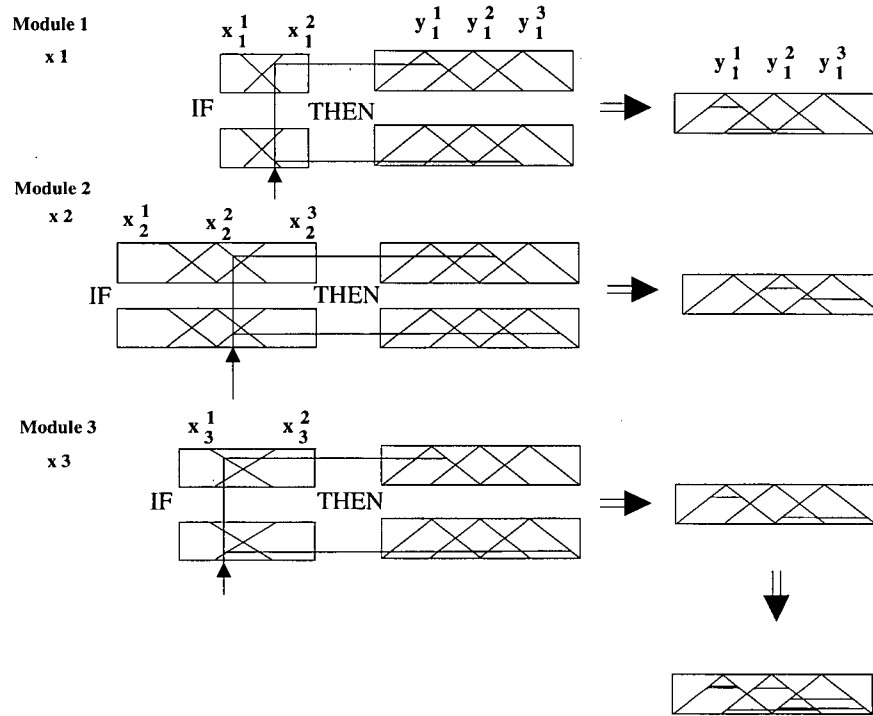


Figure 7.2(b) Schematic representation of inference in Case 1.

Table 7.4. Membership functions of the outputs for Case 1.

		Output State 1 y_1^1	Output State 2 y_1^2	Output State3 y_1^3
Module 1	g^1	$\mu(y_1^1) = \mu(x_1^2)$	$\mu(y_1^2) = \mu(x_1^1)$	
Module 2	g^2		$\mu(y_1^2) = \mu(x_2^2)$	$\mu(y_1^3) = \mu(x_2^3)$
Module 3	g^3	$\mu(y_1^1) = \mu(x_3^1)$		$\mu(y_1^3) = \mu(x_3^2)$
Overall Conclusion \Rightarrow		y_c^1	y_c^2	y_c^3

Table 7.5. Comparison of the methods of computing inference in case 1.

	New Inference Method	Conventional Inference Method
y_c^1	$\max[\min(g^1, \mu(x_1^2)), \min(g^3, \mu(x_3^1))]$	$\max[\mu(x_1^2), \mu(x_3^1)]$
y_c^2	$\max[\min(g^1, \mu(x_1^1)), \min(g^2, \mu(x_2^2))]$	$\max[\mu(x_1^1), \mu(x_2^2)]$
y_c^3	$\max[\min(g^2, \mu(x_2^3)), \min(g^3, \mu(x_3^2))]$	$\max[\mu(x_2^3), \mu(x_3^2)]$

Case 2:

Module 1

IF x_1 is x_1^1 THEN y_1 is y_1^2 IF x_1 is x_1^2 THEN y_1 is y_1^1
--

Module 2

IF x_2 is x_2^2 THEN y_1 is y_1^2 IF x_2 is x_2^3 THEN y_1 is y_1^3
--

Module 3

IF x_3 is x_3^1 THEN y_1 is y_1^1 IF x_3 is x_3^2 THEN y_1 is y_1^3
--

Module 4

IF x_1 is x_1^2 AND x_2 is x_2^2 THEN y_1 is y_1^2 IF x_1 is x_1^1 AND x_2 is x_2^3 THEN y_1 is y_1^3
--

Figure 7.2 (c). Rulebase modules for case 2.

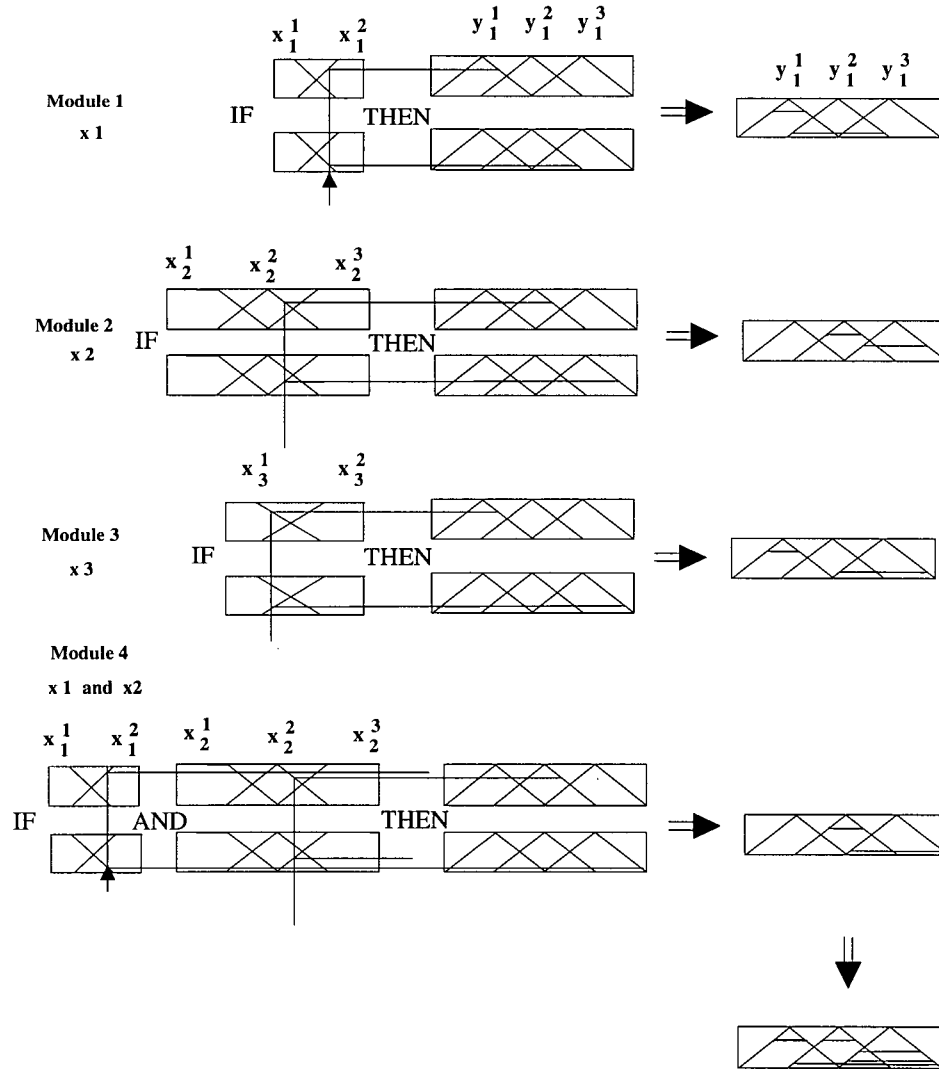


Figure 7.2 (d). Schematic representation of the inference in case 2.

Table 7.6. Membership functions of the outputs for case 2.

		Output State 1 y_1^1	Output State 2 y_1^2	Output State 3 y_1^3
Module 1	g^1	$\mu(y_1^1) = \mu(x_1^2)$	$\mu(y_1^2) = \mu(x_1^1)$	
Module 2	g^2		$\mu(y_1^2) = \mu(x_2^2)$	$\mu(y_1^3) = \mu(x_2^3)$
Module 3	g^3	$\mu(y_1^1) = \mu(x_3^1)$		$\mu(y_1^3) = \mu(x_3^2)$
Module 4	g_λ		$\mu(y_1^2) = \min(\mu(x_1^2), \mu(x_2^2))$	$\mu(y_1^3) = \min(\mu(x_1^1), \mu(x_2^3))$
Overall Conclusion \Rightarrow		y_c^1	y_c^2	y_c^3

Table 7.7. Computation of inference in case 2.

	New Inference Method
y_c^1	$\max[\min(g^1, \mu(x_1^2)), \min(g^3, \mu(x_3^1))]$
y_c^2	$\max[\min(g^1, \mu(x_1^1)), \min(g^2, \mu(x_2^2)), \min(g_\lambda, \min(\mu(x_1^2), \mu(x_2^2)))]$
y_c^3	$\max[\min(g^2, \mu(x_2^3)), \min(g^3, \mu(x_3^2)), \min(g_\lambda, \min(\mu(x_1^1), \mu(x_2^3)))]$

In Table 7.6 , g_λ is calculated using definition 3 and Eq (7.14). Equation(7.11) is used to determine λ .

Table 7.8. Computation of the conventional inference in case 2.

	Conventional Inference Method
y_c^1	$\max[\mu(x_1^2), \mu(x_3^1)]$
y_c^2	$\max[\mu(x_1^1), \mu(x_2^2), \min(\mu(x_1^2), \mu(x_2^2))]$
y_c^3	$\max[\mu(x_2^3), \mu(x_3^2), \min(\mu(x_1^1), \mu(x_2^3))]$

The two inference methods given in tables 7.7 and 7.8 cannot be directly compared. Specifically, the inference method that is developed here uses a structured rulebase method for dealing with many fuzzy variables in the form of SISO modules. In summary, the new inference method may be represented in the general form :

$$\mu_c^i = e_i = \max_{j=1}^m [\min(g^j, \mu_j^i)] \quad (7.15)$$

$$u = \frac{\sum_{i=1}^n u_i \cdot \mu_c^i}{\mu_c^i} \quad (7.16)$$

where m is the number of module, j is a variable which represents various modules, and i is a variable which represents each state of the output. A crisp control action can be computed using the centroid method.

7.3.1 Universal approximation theorem

Suppose that the input universe of discourse U is a compact set in R^n . Then, for any given real continuous function $g(x)$ on U and arbitrary $\varepsilon > 0$, there exists a fuzzy system $f(x)$ such that

$$\sup_{x \in U} |f(x) - g(x)| < \varepsilon \quad (7.17)$$

The proof relies on the Stone-Weirstrass theorem [66] . The *universal approximation property* of fuzzy systems guarantees the existence of a fuzzy system that can approximate any continuous function.

7.3.2 Stone-Weirstrass theorem

Let Z be a set of real continuous functions on a compact set U . If

- (i) Z is an ***algebra***, that is, the set Z is closed under addition, multiplication, and scalar multiplication;
- (ii) Z ***separates points*** on U , that is, for every $x, y \in U, x \neq y$, there exists $f \in Z$ such that $f(x) \neq f(y)$; and
- (iii) Z does not ***vanish at any point*** of U , that is, for each $x \in U$ there exists $f \in Z$ such that $f(x) \neq 0$; then for any real continuous function $g(x)$ on U and arbitrary $\varepsilon > 0$, there exists $f \in Z$ such that

$$\sup_{x \in U} |f(x) - g(x)| < \varepsilon .$$

The universal approximation property for fuzzy systems shows that; *if one works sufficiently hard at tuning, one should be able to make the system perform in a desired manner*[57]. It is seen that the universal approximation theorem guarantees that there exists a way to define a fuzzy system $f(x)$, but it does not show how to find that system. Many fuzzy systems are known to satisfy the universal approximation property. For example, in [86,87] it has been proved that a fuzzy logic control (FLC) system with Gaussian membership functions, the product fuzzy implication and the center of area defuzzification, is a universal approximator. In [7] it has been proved that modified Sugeno type fuzzy controllers are universal approximators. In [8] it has been shown that approximation is possible for all FLC systems, with both fuzzy consequents and nonfuzzy consequents. Essentially, then all FLC systems are universal approximators. The analytical proof in [8], guarantees that a fuzzy system with the inference method developed here is also a universal approximator.

7.4 Experimental Investigation

A fuzzy logic control system with the new inference method developed in this chapter is implemented on the prototype hydraulic manipulator (see Chapter 2, Section 2.3). The knowledge base of this system has two input variables (position error and velocity) and one output variable (current input to the servovalve of the hydraulic actuator). Velocity is calculated according to Eq. (2.1), as shown in Figure 2.2. A block diagram of the fuzzy knowledge-based controller (FKBC) used in the control of the hydraulic manipulator is shown in Figure 7.3.

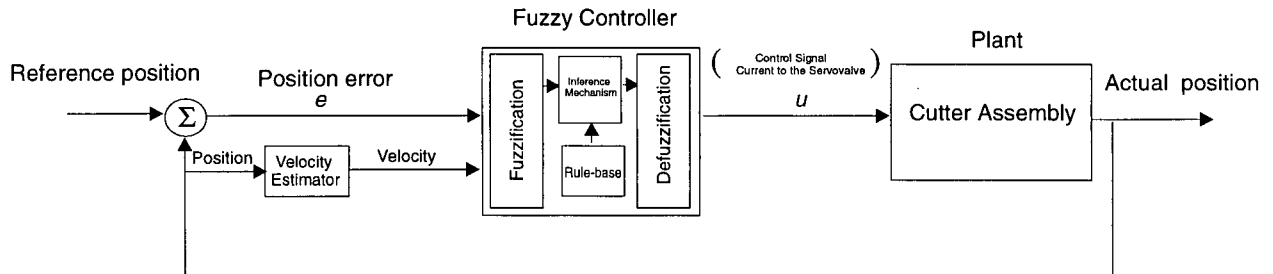


Figure 7.3. Block diagram of the fuzzy logic controller.

The goal of the control system is to control the position of the cylinder in the hydraulic actuator. The membership functions of the input and output variables are shown in Figure 7.4. The centroid method is used to defuzzify the control action. Three cases are investigated where three different rulebases for the variable “velocity” are tested, each case having three different *degrees of importance* g in the contribution of modules. The results are shown in Figure 7.6 through Figure 7.8. For Case 1 we have two modules: Module 1 and Module 2(a). For Case 2, we have two modules: Module 1 and Module 2(b). For Case 3, we have two modules: Module 1 and Module 2(c), as shown in Figure 7.5.

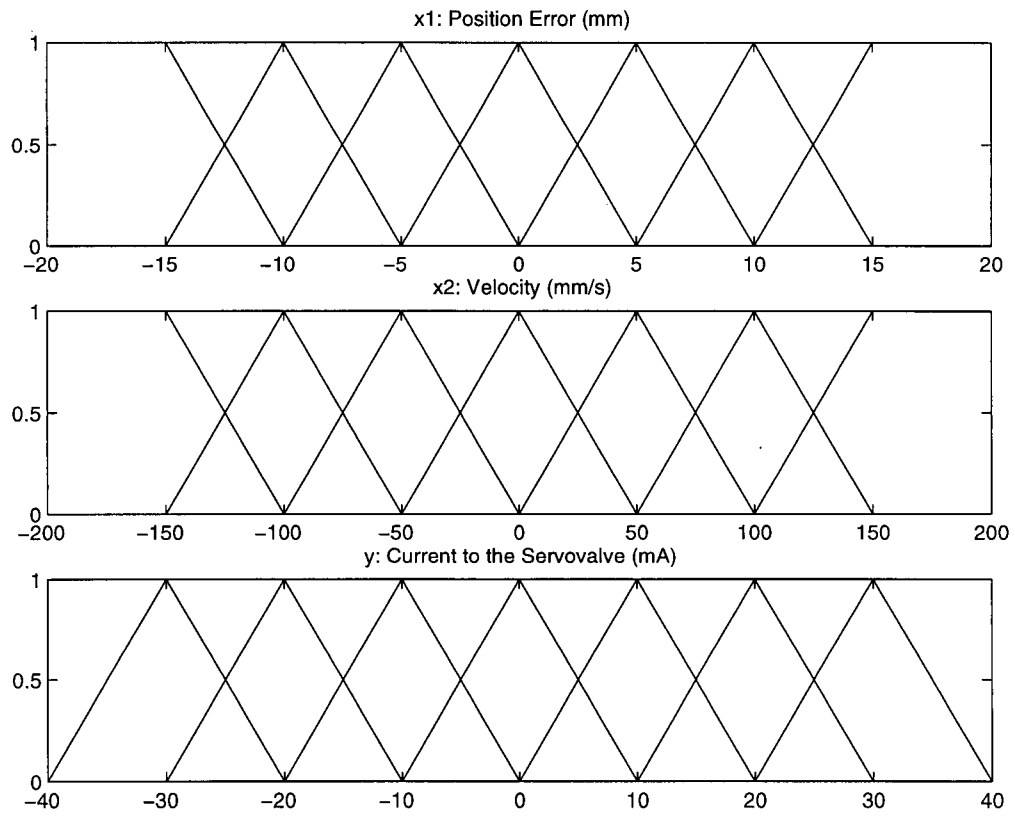


Figure 7.4. Membership functions of input and output variables.

Module 1

IF x_1 is NL	THEN y is PL
IF x_1 is NM	THEN y is PM
IF x_1 is NS	THEN y is PS
IF x_1 is ZR	THEN y is ZR
IF x_1 is PS	THEN y is NS
IF x_1 is PM	THEN y is NM
IF x_1 is PL	THEN y is NL

Module 2(a): Case 1

IF x_2 is NL	THEN y is PL
IF x_2 is NM	THEN y is PM
IF x_2 is NS	THEN y is PS
IF x_2 is ZR	THEN y is ZR
IF x_2 is PS	THEN y is NS
IF x_2 is PM	THEN y is NM
IF x_2 is PL	THEN y is NL

Module 2(b): Case 2

IF x_2 is NL	THEN y is PM
IF x_2 is NM	THEN y is PS
IF x_2 is NS	THEN y is PS
IF x_2 is ZR	THEN y is ZR
IF x_2 is PS	THEN y is NS
IF x_2 is PM	THEN y is NS
IF x_2 is PL	THEN y is NM

Module 2(c): Case 3

IF x_2 is NL	THEN y is PS
IF x_2 is NM	THEN y is PS
IF x_2 is NS	THEN y is PS
IF x_2 is ZR	THEN y is ZR
IF x_2 is PS	THEN y is NS
IF x_2 is PM	THEN y is NS
IF x_2 is PL	THEN y is NS

Figure 7.5. Modules of the rulebase in three cases of investigation.

Table 7.9. Output membership functions.

Module	g	Output Membership Functions (Current to the servovalve) (mA)						
		NL	NM	NS	ZR	PS	PM	PL
1 (x_1)	g_1	μ_1^1	μ_1^2	μ_1^3	μ_1^4	μ_1^5	μ_1^6	μ_1^7
2 (x_2)	g_2	μ_2^1	μ_2^2	μ_2^3	μ_2^4	μ_2^5	μ_2^6	μ_2^7
output		μ^1	μ^2	μ^3	μ^4	μ^5	μ^6	μ^7

The control membership functions are computed using

$$\mu_c^i = e_i = \max_{j=1}^2 [\min(g^j, \mu_j^i)] \quad (7.18)$$

where i is the number of states in the output and j is the number of modules of rules in the knowledgebase. Specifically,

$$\begin{aligned} \mu^1 &= \max[\min(g_1, \mu_1^1), \min(g_2, \mu_2^1)] \\ \mu^2 &= \max[\min(g_1, \mu_1^2), \min(g_2, \mu_2^2)] \\ \mu^3 &= \max[\min(g_1, \mu_1^3), \min(g_2, \mu_2^3)] \\ \mu^4 &= \max[\min(g_1, \mu_1^4), \min(g_2, \mu_2^4)] \\ \mu^5 &= \max[\min(g_1, \mu_1^5), \min(g_2, \mu_2^5)] \\ \mu^6 &= \max[\min(g_1, \mu_1^6), \min(g_2, \mu_2^6)] \\ \mu^7 &= \max[\min(g_1, \mu_1^7), \min(g_2, \mu_2^7)] \end{aligned} \quad (7.19)$$

Next, the centroid method is used to defuzzify the control action, according to

$$u = \frac{\sum_{i=1}^7 u_i \cdot \mu_c^i}{\mu_c^i} \quad (7.20)$$

7.5 Discussion

From the results given in figures 7.6 through 7.8, the effect of changing the rulebase (tuning the rulebase) of the second module on the performance of the system is observed. Note that the inference method has been able to produce a good response. It is assumed that the rulebases of the modules are tuned properly, which is a common situation with fuzzy systems.

In each of the cases, the effect of the degree of importance of the second module g_2 within the contribution of each module is studied. In all the cases, the importance of the first module is considered to be $g_1=1$. This is because the first module is about the rules related to position variable. In Figure 7.9, the case where the effect of the second module has been eliminated ($g_1 = 1, g_2 = 0$) is compared with the other three cases. This result shows the contribution of the second fuzzy input variable, velocity, to the fuzzy controller. The contribution of the second module in Case 1 (Figure 7.6) results in an overshoot in the response. The other two cases have produced a smooth and damped response; however, the case $g_1 = 1, g_2 = 0$ where the effect of the second module has been eliminated, shows a faster response. This simply illustrates the damping effect of the velocity contribution to the inference. However, on relative terms, the degree of influence of the second module as determined by the value of g_2 , is rather small.

The method developed here provides a generalized and organized way of handling the problem of fuzzy decision making in complex multivariable systems. The technique uses a particular way to arrange the variables into single-input-single-output modules. It follows that there is no limitation as to how many variables can be handled by this technique. An important aspect of the present technique is that it provides the potential of using fuzzy logic for controlling multivariable systems, with reduced number of rules and increased speed of processing.

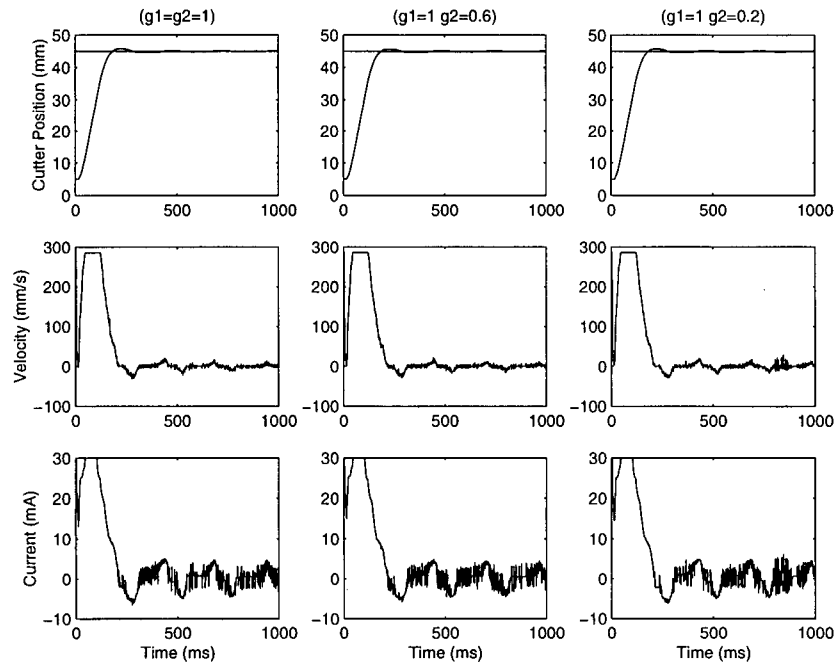


Figure 7.6. Module 1 and module 2(a) (case1).

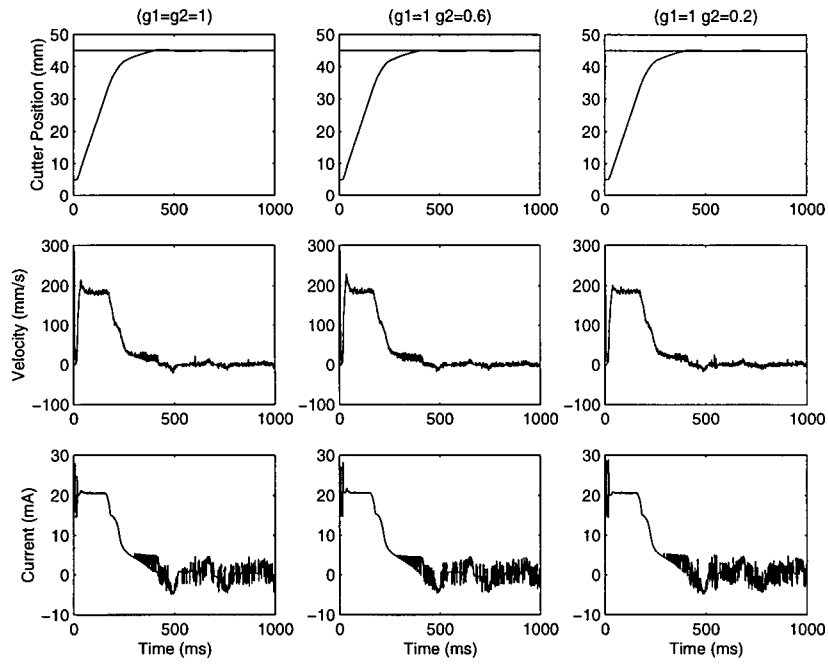


Figure 7.7. Module 1 and module 2(b) (case2).

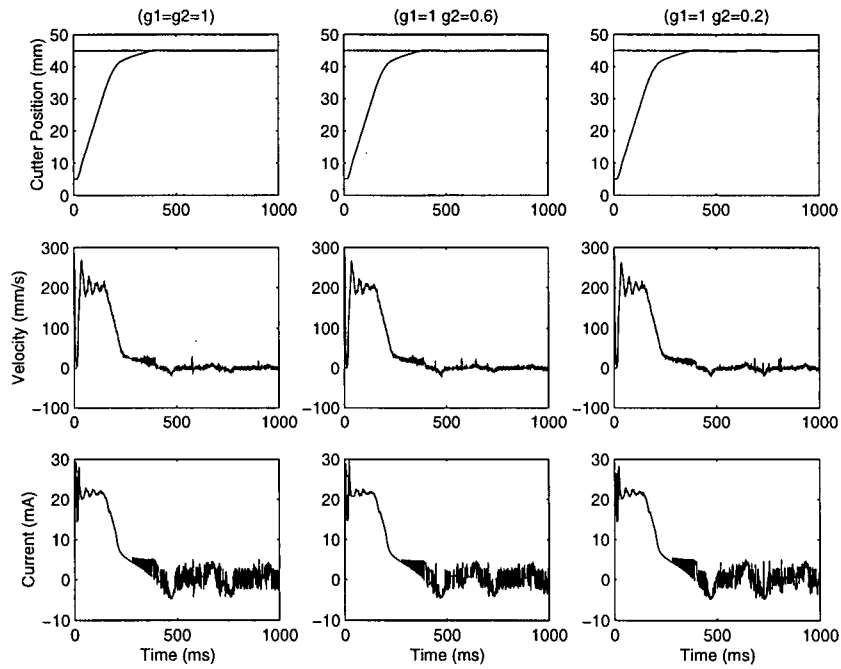


Figure 7.8. Module 1 and module 2(c) (case3).

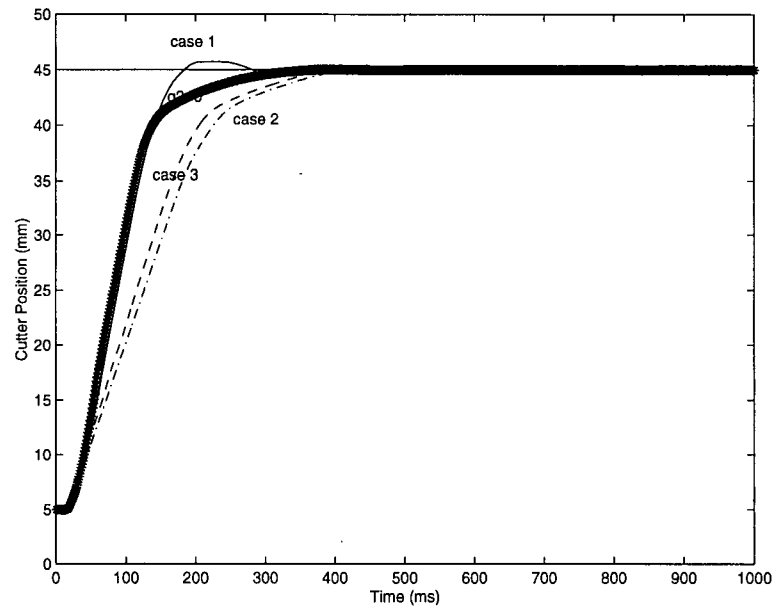


Figure 7.9. Comparison of the responses ($g1=1, g2=0$)

Chapter 8

Conclusion

The theme of the research reported in this thesis is the efficient practical implementation of fuzzy logic in knowledge-based decision making, particularly in industrial process control. Considerations such as the inference scheme, knowledge base, membership functions and the defuzzification approach are all important in these applications, and have been investigated in the thesis. Fuzzy logic is known to be useful for information fusion and control of plants that are complex and the overall goal is subjective. The thesis specifically considered the application of fuzzy logic in the control of an industrial prototype of an automated fish cutting machine that has been developed in the Industrial Automation Laboratory, University of British Columbia. The research was motivated by the complexity of this machine, which consists of many sensors and actuators that has made it hard to model and control. The application of fuzzy logic on a hydraulic manipulator of the prototype fish cutting machine as a direct fuzzy controller was investigated. As an original contribution to the knowledge, a new inference method was developed, implemented, and tested on this machine. This concluding chapter summarizes the primary contributions of the research, outlines the significance of these contributions, and indicates possible directions of further research in the area.

8.1 Major Contributions

The major contributions of this research are summarized below:

1. A novel inference method for multivariable fuzzy systems was developed, implemented on an industrial machine, and evaluated. This inference method is based on the concept of fuzzy integral. The technique can be embedded in any commercial expert system, particularly any direct fuzzy logic controller that has to deal with many fuzzy variables in a coupled rulebase. The feasibility and effectiveness of this technique were demonstrated by implementing it on a hydraulic manipulator of a prototype industrial fish cutting machine.

2. A low level proportional plus derivative (PD)-type fuzzy logic controller was designed and implemented in realtime on the prototype hydraulic manipulator. Choice of membership functions, effect of tuning of the membership functions of inputs and outputs the effect of tuning the rulebase, and the choice among common defuzzification methods, were studied through experimentation and comparative evaluation.
3. A low level proportional (P)-type fuzzy logic controller was designed and implemented in realtime on the prototype hydraulic manipulator. In particular, two common methods of fuzzy inference—individual-rule-based and composition-based, were studied using this controller. The performance of the prototype system was studied in detail under these schemes of control inference.
4. A supervisory fuzzy control system containing an intelligent tuner for low-level controllers was developed and implemented in the prototype hydraulic manipulator. This control system automatically tunes the parameters of the conventional PD controllers which drive the hydraulic actuators of the prototype machine. This control system has three-layer hierarchical structure. In the bottom layer of the hierarchy, a conventional PD controllers are used to position the two-degree-of-freedom cutting module using hydraulic actuators. The middle layer monitors the performance of the position response, preprocesses the response signals, and extracts the performance parameters, based on a step-input response. The top layer infers the tuning actions for the PD servo units of the actuators. The knowledge base for tuning the low level controllers was developed using control experience and represented by fuzzy rulebase modules.

8.2 Significance of the Research

While this research originated from an analytical perspective, its results have significant practical value, which can impact a wide range of applications. In particular, the new inference method which was developed can be embedded in any commercial expert system. Applications include process control, professional consulting (medical,

legal, business, etc.) or any other application of fuzzy logic such as pattern recognition. Such applications deal with multivariable systems with high complexity. The problem of knowledge-based decision making in such a system can be greatly facilitated by the approach developed in the thesis. This research also investigated the performance of two well-known inference methods in fuzzy logic. Observations were made as to which inference method has particular merits in practical applications. The considerations of the importance of proper design related to the membership functions, rulebase, and the defuzzifications are also of practical significance. An intelligent tuner for the hydraulic manipulators of an industrial prototype of an automated fish cutting machine was developed. A system of this type may be appropriate as an intelligent supervisory control system for industrial machines, and can relieve the burden of informed decision making on the product quality and the machine performance from dedicated operators.

8.3 Directions for Further Research

No research performed can be complete. There are several issues and aspects that should be further investigated in the context of the research reported in this thesis. The following are some directions for possible future research:

The new inference method that is developed in the present work can be implemented on a variety of systems with multiple fuzzy variables. In that regard, an analytical study can be conducted to explore the effect of number of variables, number of fuzzy states for each variable in the fuzzy rules, with degree of influence of each variable on the output variable and its effect on the performance of the system. In this research, the min-max implication was used; however, the effect of different implication methods can be studied as well. The proportional plus derivative (PD)-type controller was developed based on the individual rulebased method. This controller may be realized through the compositional rule of inference and fuzzy relations too. Further analytical studies may be conducted and experimental investigations maybe carried out on this issue.

Comparison of two well-known inference methods was carried out in the present work. It was observed that these two methods do not generate the same performance. For further research, the level of equivalence of these two inference methods may be analytically explored.

References

1. Ahmed, M. and Ward, R. (1999) "An Expert System for General Symbol Recognition", *Pattern Recognition Journal*, June.
2. Ahmed, M. (1999), *An Expert System for the Recognition of General Symbols*, PhD Thesis, Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada.
3. Anderson, K. L., Blankenship, G. L., and Lebow, L. G. (1988), "A Rule-Based Adaptive PID Controller", *Proc. IEEE Conference on Decision and Control*, Austin, TX.
4. Antsaklis, P. J. and Passino, K. M. (1993), *An Introduction to Intelligent and Autonomous Control*, Kluwer Academic Publishers, Norwell, Massachusetts, USA.
5. Barlev, B. (1992), *Implementation and Evaluation of an Intelligent Tuner for an Ill-Defined Servo-Motor System*, M.A.Sc Thesis, Department of Mechanical Engineering, The University of British Columbia, Vancouver, Canada.
6. Bellman, R. E. and Zadeh, L. A. (1970), "Decision-Making in a Fuzzy Environment", *Management Science*, Vol.17, No.4, pp141-164.
7. Buckley, J. J. (1992), "Universal Fuzzy Controller", *Automatica*, Vol.28, pp.1245-1248, 1992.
8. Castro, J. L (1995), "Fuzzy Logic Controllers Are Universal Approximators", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No.4, pp. 629-635.
9. Chand, S., (1991), "On-line, Self-Monitoring Tuner for Proportional Integral Derivative Controllers", *Proceedings of the 30th Conference on Decision and Control*, Brighton, England, December, pp.1905-1906.
10. Chang, S. S. L and Zadeh, L.A. (1972), "On Fuzzy Mapping and Control", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-2, No.1, pp.30-34.
11. Choi, S. (1996) "High Performance Fuzzy Logic Processing Method", US patent No.5572629.
12. Clark, J. J and Yuille, A. L. (1990), *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic Publishers, Norwell, Massachusetts, USA.

13. Corbet, T., Sepehri, N., and Lawrence, P. D.,(1996), "Fuzzy Control of a Class of Hydraulically Actuated Industrial Robots", *IEEE Transactions on Control Systems Technology*, Vol.4, No.4, pp.419-426.
14. Daugherty, W. C., Rathakrishnan, B., and Yen, J. (1992), "Performance Evaluation of a Self-Tuning Fuzzy Controller", *Proc. IEEE International Conference on Fuzzy Systems*, pp.389-397, San Diego, CA.
15. Davis, Jr., L. I. Dage. G. A. Gurney, III. E. J. and Bauer, M. B. (1999). "Compact Fuzzy Logic Algorithm", US Patent No.5974350.
16. Demirli, K. and Turksen, I. B., (1992), "Rule Break up with Compositional Rule of Inference", *Proc. IEEE International Conference on Fuzzy Systems*, pp.949-956, San Diego, CA.
17. De Silva, C. W. (1992), "Research Laboratory for Fish Processing Automation" , *Robotics and Computer Integrated Manufacturing* , Vol. 9, No.1 , pp.49-60.
18. De Silva, C. W. (1994), "A Criterion for Knowledge Base Decoupling in Fuzzy-Logic Control Systems" , *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No.10, pp. 1548-1552.
19. De Silva, C. W. and MacFarlane, A. G. J. (1988), "Knowledge-Based Control Approach for Robotic Manipulators", *International Journal of Control* , Vol. 50, No.1 , pp.249-273.
20. De Silva, C. W. and Mac Farlane, A.G. J. (1989), *Knowledge-Based Control with Application to Robots*, Springer-Verlag, Berlin.
21. De Silva, C. W. and Wickramarachchi, N. (1992), "Knowledge-Based Development for an Intelligent Control System of a Process Workcell", *Intelligent Control Systems*, DSC-Vol.45, ASME Press, New York, pp.113-119.
22. De Silva, C. W. and Wickramarachchi, N. (1997), "An Innovative Machine for Automated Cutting of Fish", *IEEE/ASME Transactions on Mechatronics*, Vol. 2, No. 2, pp. 86-98.
23. De Silva, C. W. (1989), *Control Sensors and Actuators*, Prentice-Hall Inc., Englewood Cliffs, NJ.
24. De Silva, C. W. (1995), *INTELLIGENT CONTROL: Fuzzy Logic Applications*, CRC Press, Boca Raton, FL.
25. De Silva, C.W. (2000), *INTELLIGENT MACHINES: Myths and Realities*, CRC Press, Boca Raton, FL.

26. Di Nola, A. and Pedrycz, W. (1985), "When is a fuzzy relation decomposable in two fuzzy sets?", *Fuzzy Sets and Systems*, Vol. 16, pp.87-90.
27. Doob, J. L. (1994), *Measure Theory*, Springer-Verlag, Berlin, Germany.
28. Driankov, D., Hellendoorn, H. and Reinfrank, M. (1993), *An Introduction to Fuzzy Control*, Springer-Verlag, Berlin, Germany.
29. FitzSimons, P. M., and Palazolo, J. J., (1995), "PART I: Modeling of a One Degree-of-Freedom Active Hydraulic Mount", *Proc. ASME International Mechanical Engineering Congress and Exposition, Fluid Power Systems and Technology*, FPST Division, San Francisco, CA, November, Vol.2, pp.1-6.
30. Fu, K. S.(1971), "Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control", *IEEE Transactions on Automatic Control*, Vol. 16, No.1, pp.70-72.
31. Gamage, L. D. K. B. (1993), *A Model-Based Approach to Complex Contour Generation for Process Automation Using Computer Vision*, Ph.D Thesis, Department of Mechanical Engineering, The University of British Columbia, Vancouver, Canada.
32. Gegov, A. and Frank, P. M. (1995), "Hierarchical Fuzzy Control of Multivariable Systems", *Fuzzy Sets and Systems*, Vol.72, pp.299-310.
33. Gegov, A. (1996), *Distributed Fuzzy Control of Multivariable Systems*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
34. Goulet, J. F.(1999), *Intelligent Hierarchical Control of A Deployable Manipulator*, M.A.Sc. Thesis, Department of Mechanical Engineering, The University of British Columbia, Vancouver, Canada.
35. Graham, J. H and Saridis, G. H. (1982), "Linguistic Decision Structures for Hierarchical Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 12, No.3, pp. 325-333.
36. Gupta, M. M., Kiszka, J. B. and Trojan, G. M. (1986), "Multivariable Structure of Fuzzy Control Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 16, No.5, pp. 638- 656.
37. Gupta, M. M., Saridis, G. N. and Gaines, B. R. (1977), *Fuzzy Automata and Decision Processes*, Elsevier North-Holland,Inc. Amsterdam, The Netherlands.

38. Huwendiek, O. and Brockmann, W. (1999), "Function Approximation with Decomposed Fuzzy Systems", *Fuzzy Sets and Systems*, Vol.101, pp.273-286.
39. Janikow, C. Z. (1998), "Fuzzy Decision Trees: Issues and Methods", *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: Vol. 28, No.1, pp. 1-14.
40. Johnston, M. M. (1989), *An Expert System to Predict the Ecological Effects of Prescribed Fire*. M.A.Sc Thesis, Faculty of Commerce and Business administration, The University of British Columbia, Vancouver, Canada.
41. Kim, C. J. (1997), "An Algorithmic Approach for Fuzzy Inference", *IEEE Transactions on Fuzzy Systems*, Vol. 5, No.4, pp.585-598.
42. King, P. J. and Mamdani, E. H. (1977), "The Application of Fuzzy Control Systems to Industrial Processes", *Automatica*, Vol.13, pp 235-242.
43. Klein, L. A. (1993), *Sensors and Data Fusion Concepts and Applications*, Vol. TT14 of Tutorial Texts in Optical Engineering. Bellingham, Washington: SPIE Optical Engineering Press, Bellingham, Washington, USA.
44. Klir, G. J. , St Clair, U.H, and Yuan, B. (1997), *Fuzzy Set Theory*. Prentice Hall, Upper Saddle River, NJ.
45. Kolodziejczyk, W. (1989), "To What Extent Does 'Decomposable' mean 'Transitive' for a Fuzzy Relation?", *Fuzzy Sets and Systems*, Vol. 32, pp.125-128.
46. Krijgsman, A. J., Broeders, H. M. T., Verbruggen, H. B., and Bruijn, P. M., (1988), "Knowledge-based Control", *Proceedings of the 27th Conference on Decision and Control*, Austin, Texas, December.pp.570-574.
47. Kumar, S. (1991), *An Expert System to Diagnose Quality Problems in The Continuous Casting of Steel Billets*. M.A.Sc Thesis, Department of Metals and Materials, The University of British Columbia, Vancouver, Canada.
48. Lee, P. G., Lee. K. K., and Jeon, G. J. (1995), An Index of Applicability for the Decomposition Method of Multivariable Fuzzy Systems. *IEEE Transactions on Fuzzy Systems*, Vol. 3, No.3, pp.364-369.
49. Legzdins C.F. (1996), *An expert system for metal matrix composite selection and design*, M.A.Sc Thesis, Department of Metals and Materials, The University of British Columbia, Vancouver, Canada.

50. Lipschutz S. (1964), *Theory and problems of Set Theory and Related Topics*. McGraw-Hill Inc.
51. Mamdani, E. H. and Assilian, S. (1975), An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller , *Int. J. Man-Machine Studies*, pp 1-13.
52. Mamdani, E. H. (1976), Advances in the linguistic synthesis of fuzzy controllers, *Int. J. Man-Machine Studies*, Vol.8, pp 669-678.
53. Matranga,V., Giacalone, B., and Abruzzese, M. (1998), "Method and Apparatus for Parallel Processing of Fuzzy Rules", US patent 5796917.
54. Meech, J. A. and Kumar, S. (1995), *A Hypermanual on Expert Systems*. version 5.0.
55. Merritt, H. E. (1967), *Hydraulic Control Systems*. John Wiley & Sons Inc., New York.
56. Meystel, A. (1985), Intelligent Control: Issues and Perspectives , *Proc. IEEE Workshop on Intelligent Control*, Rensselaer Polytechnic Institute, Troy, New York, pp1-15.
57. Passino, K. M., and Yurkovich, S. (1998), *Fuzzy Control*, Addison-Wesley Longman Inc. California, USA.
58. Pedrycz, W. and Gomide, F. (1998), *An Introduction to Fuzzy Sets*. The MIT Press, Cambridge, Massachusetts, USA.
59. Pedrycz, W. (1998), *Computational Intelligence, An Introduction*. CRC Press, Boca Raton, FL.
60. Pfeffer, W. F. (1977), *Integrals and Measures*. Marcel Dekker, Inc. New York, USA.
61. Procyk, T. J. and Mamdani, E. H. (1979), A Linguistic Self-Organizing Process Controller. *Automatica*, Vol.15, pp 15-30.
62. Raju, G. V. S., Zhou, J. and Kisner, R. A. (1991), "Hierarchical Fuzzy Control", *International Journal of Control*, Vol. 54, No.5, pp.1201-1216.
63. Rakocovic, V. (1995), *Development of SCADA System to Monitor and Control Continuous Casting of Steel Billets*. M.A.Sc Thesis. Department of Mining and Mineral Processing, The University of British Columbia, Vancouver, Canada.
64. Rahbari, R. Meech, J. A., Desilva, C. W., (1997), "Intelligent Tutoring System: An Expert-System Trainer for Herring Roe Grading". *Proceeding of the American Control Conference*, Albuquerque, New Mexico, pp.3171-3175.
65. Riahi, N. and DeSilva, C. W. (1990), "Fast Image Processing for the Gill Position Measurement in Fish", *Proc.IECON'90*, Pacific Grove, CA, Vol.1, pp.476-481.

66. Rudin, W. (1964), *Principles of Mathematical Analysis* , McGraw-Hill Inc.
67. Sanchez, E. (1976), "Resolution of Composite Fuzzy Relation Equations", *Information and Control* , Vol.30, pp.38-48.
68. Saridis, A. (1985), "Foundations of the Theory of Intelligent Controls", *Proc. IEEE Workshop on Intelligent Control*, Rensselaer Polytechnic Institute, Troy, New York, pp23-28, 1985.
69. Sayed, T. A. (1995), *A Highway Safety Expert System: A New Approach to Safety Programs*. PhD Thesis, Department of Civil Engineering, The University of British Columbia, Vancouver, Canada.
70. Sepehri, N., Corbet, T. and Lawrence, P D. (1995), "Fuzzy Position Control of Hydraulic Robots with Valve Deadbands", *Mechatronics* Vol.5, No.6, pp.623-643.
71. Shawwash. Z.K.E. (2000), *A Decision Support System For Real-Time Hydropower Scheduling in a Competitive Power Market Environment*. PhD Thesis, Department of Civil Engineering, The University of British Columbia, Vancouver, Canada.
72. Sugeno, M. (1974), *Theory of Fuzzy Integrals and its applications*. PhD Thesis, Tokyo Institute of Technology.
73. Smith, M. D., Bonde, Jr., A. R., Catherine, G. J., Napolitano, R. and Wolfe, D. M. (1998), "Fuzzy Logic-based Evidence Fusion Tool for Network Analysis", US Patent No.5787235.
74. Sugeno, M.(1977), "Fuzzy Measures and Fuzzy Integrals: A Survey". M. M. Gupta, G.N. Saridis and B. R.Gaines, *Fuzzy Automata and Decision processes*, pp.89-102, Elsevier North Holland, Inc., Amsterdam, The Netherlands.
75. Tafazoli, S., De Silva, C. W. and Lawrence, P. D. (1998), "Tracking Control of an Electrohydraulic Manipulator in the presence of friction", *IEEE Trans Control Systems*, Vol.6, pp.401-411.
76. Tafazoli. S. (1996), *Identification of Frictional Effects and Structural Dynamics for Improved Control of Hydraulic Manipulators*. PhD Thesis, Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada.
77. Tahani, H. and Keller, J. M. (1990), "Information Fusion in Computer Vision Using the Fuzzy Integral" , *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No.3, pp. 733-741.

78. Tanaka. K. (1996), *An Introduction to Fuzzy Logic for practical Applications*, Springer-Verlag New York, Inc. New York, USA.
79. Taylor,J.C. (1996). *An Introduction to Measure and Probability*, Springer-Verlag New York, Inc. New York, USA.
80. Tunsel, E. W. (1996), *Adaptive Hierarchy of Distributed Fuzzy Control: Application to behavior control of rovers*, Ph.D Thesis, Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque, New Mexico.
81. Tong, R. M. (1978), "Analysis and Control of Fuzzy Systems Using Finite Discrete Relations", *International Journal of Control*, Vol.27, No.3, pp.431-440.
82. Tong, R. M. and Efstathiou, J. (1980), "Rule-Based Decomposition of Fuzzy Relational Models", *Proceeding of Automatic Control Conference*, FA10-E.
83. Van Nauta Lemke, H. R. and De-zhao, W. (1985), "Fuzzy PID Supervisor", *Proceeding of 24th Conference on Decision and Control*, Lauderdale, FL, December.
84. Wickramarachchi, N. K. (1995), *Development of Knowledge-Based Hierarchical Control Structure for process Automation*, Ph.D Thesis, Department of Mechanical Engineering, The University of British Columbia , Vancouver, Canada.
85. Wang. Z. and Klir, G. J. (1992), *Fuzzy Measure Theory*, Plenum Press, New York.
86. Wang, L. X. (1992), "Fuzzy systems are universal approximators", *Proc. of IEEE Int.Conf. on Fuzzy Systems*, San Diego, CA, pp.1163-1170.
87. Wang, L. X. (1997), *A Course in Fuzzy Systems and Control*, Prentice Hall Inc., NJ, 1997.
88. Xu, C. W. (1989),"Decoupling Fuzzy Relational Systems -An Output Feedback Approach", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No.2, pp. 414-418.
89. Xu, C. W., and Lu, Y. Z., (1989), "Decoupling in Fuzzy Systems: A Cascade Composition Approach", *Fuzzy Sets and Systems*, Vol. 29, pp.177-185.
90. Xu, C. W., (1991), "Linguistic Decoupling Control of Fuzzy Multivariable Processes", *Fuzzy Sets and Systems*, Vol. 44, pp.209-217.
91. Yeung, D. S. and Tsang, E. C. C. (1998), "A Multilevel Weighted Fuzzy Reasoning Algorithm for Expert Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, Part A: Vol. 28, No.2, pp. 149-158.
92. Zadeh,L.A. (1965), "Fuzzy Sets", *Information and Control*, Vol.8, pp.338-353.
93. Zadeh,L.A. (1968), "Fuzzy Algorithms", *Information and Control*, Vol.12, pp.94-102.

94. Zadeh, L.A. (1971), "Quantitative Fuzzy Semantics", *Information Sciences*, Vol.3, pp.159-176.
95. Zadeh, L.A. (1971), "Similarity Relations and Fuzzy Orderings", *Information Sciences*, Vol.3, pp.177-200.
96. 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*, Vol. SMC-3, No.1, pp.28-44.
97. Zadeh, L.A. (1975), "The Concept of a Linguistic Variable and its Application to Approximate Reasoning - Part I", *Information Sciences*, Vol. 8, pp.199-249.
98. Zadeh, L.A. (1975), "The Concept of a Linguistic Variable and its Application to Approximate Reasoning - Part I", *Information Sciences*, Vol. 8, pp.301-357.
99. Zadeh, L.A. (1975), "The Concept of a Linguistic Variable and its Application to Approximate Reasoning - Part III", *Information Sciences*, Vol. 9, pp.43-80.
100. Zadeh, L. A. (1975), "Calculus of Fuzzy Restrictions", *Fuzzy Sets and Their Applications*, pp.1-39.
101. Zadeh, L. A. (1978), "Fuzzy Sets as a Basis for a Theory of a Possibility", *Fuzzy Sets and Systems*, Vol.1, pp.3-28.
102. Zadeh, L. A. (1979), "A Theory of Approximate Reasoning", *Machine Intelligence*, pp.149-194.
103. Zadeh, L. A. (1985), "The Role of Fuzzy Logic in The Management of Uncertainty in Expert Systems", *Approximate Reasoning in Expert Systems*, pp.3-31.
104. Zadeh, L. A. (1989), "Knowledge Representation in Fuzzy Logic", *IEEE Transactions on Knowledge and Data Engineering*, Vol.1 No.1, pp.89-100.
105. Zeng, X. J. and Singh, M. G. (1996), "Decomposition Property of Fuzzy Systems and Its Application", Vol. 4, pp.149-165.
106. Zimmermann, H.J. (1996), *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, Norwell, Massachusetts, USA.
107. Zhao, T. and Virvalo, T. (1993), "Fuzzy Control of a Hydraulic Position Servo with Unknown Load" *Proc. IEEE 2nd International Conference on Fuzzy Systems*, San Francisco, CA, pp.785-788.

Appendix A

Theoretical Background

A.1 Set Theory

A *classical set* is a collection of objects of any kind. A *set* is fully specified by the elements it contains. In fuzzy set theory, normal sets are called *crisp sets*, in order to distinguish them from *fuzzy sets*. Let C be a crisp set defined on the universe X . Then for any element x of X , either $x \in C$ or $x \notin C$. In fuzzy set theory this property is generalized. Therefore in a *fuzzy set* F , it is not necessary that either $x \in F$ or $x \notin F$ but it can be the member of that set with a certain degree of membership that is assigned to every $x \in X$ a value from the unit interval $[0,1]$ instead of the two-element set $\{1,2\}$. The set that is defined based on such an extended membership function is called a *fuzzy set*.

Definition A.1: Fuzzy Set. A fuzzy set F in a universe of discourse X is defined by a membership function μ_F which takes values in the interval $[0,1]$. Specifically this may be represented by the mapping, $\mu_F: X \rightarrow [0,1]$. When the universe of discourse X is continuous, a fuzzy set F is written concisely as,

$$F = \int_{x \in X} \mu_F(x) / x \quad (\text{A.1})$$

When U is discrete, a fuzzy set F is represented as,

$$F = \sum_{i=1}^n \mu_F(x_i) / x_i \quad (\text{A.2})$$

Definition A.2: Support set

The support set of a fuzzy set F is the crisp set formed by the collection of all elements $x_i \in X$ such that $\mu_F(x_i) > 0$.

Definition A.3: Fuzzy Singleton

A fuzzy set whose support set is a single point (or element) in X is referred to as a *fuzzy singleton*. Typically but not necessarily, the associated membership grade is unity; $\mu_F = 1.0$.

A.2 Operations on Fuzzy Sets

Definition A.4: Two fuzzy sets are **equal** ($A = B$) if and only if

$$\forall x \in X: \quad \mu_A(x) = \mu_B(x) \quad (\text{A.3})$$

Definition A.5: A is a **subset** of B ($A \subseteq B$) if and only if

$$\forall x \in X: \quad \mu_A(x) \leq \mu_B(x) \quad (\text{A.4})$$

Definition A.6: A' is the **complement** of the fuzzy set A if and only if

$$\forall x \in X: \quad \mu_{A'}(x) = 1 - \mu_A(x) \quad (\text{A.5})$$

Definition A.7: Union (OR) $A \cup B$

Consider two fuzzy sets A and B in the **same** universe X . Their union is a fuzzy set $A \cup B$.

The membership function is given by:

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad \forall x \in X \quad (\text{A.6})$$

Definition A.8: Intersection (AND) $A \cap B$

Consider two fuzzy sets A and B in the **same** universe X . Their intersection is a fuzzy set

$A \cap B$. Its membership function is given by:

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad \forall x \in X \quad (\text{A.7})$$

Definition A.9: A **triangular norm** or *T-norm* denotes a class of binary operations that can represent the generalized intersection operation, which has the following properties :

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

$$1 - \quad xTy = yTx \quad (A.8)$$

$$2 - \quad (xTy)Tz = xT(yTz) \quad (A.9)$$

$$3 - \quad x \leq y \text{ and } w \leq z \text{ implies } xTw \leq yTz \quad (A.10)$$

$$4 - \quad xT1 = x \text{ and } xT0 = 0 \quad (A.11)$$

Definition A.10: A **Triangular Co-norm** or *S-norm* denotes a class of binary operations that can represent the generalized union operation which has the following properties: the first three criteria of the *T-norm* are the same as *S-norm* but $xS0 = x$.

Definition A.11: Cartesian Product of Two Fuzzy Sets

Let A be a fuzzy subset of a universe of discourse U , and let B be a fuzzy subset of a possibly different universe of discourse V . The Cartesian product of A and B as denoted by $A \times B$ is the set of all possible ordered pairs constructed by combining elements of A with elements of B such that the first element in each pair is a member of A and the second element is a member of B . Its membership function is defined by:

$$\mu_{A \times B}(u, v) = \min[\mu_A(u), \mu_B(v)] \quad \forall u \in U, \forall v \in V \quad (A.12)$$

A.3 Classical Relations

A **relation** can be considered as a set of *tuples*, where a tuple is an ordered pair. A binary tuple is denoted as (x, y) ; a ternary tuple is (x, y, z) ; and an n -ary tuple is (x_1, x_2, \dots, x_n) .

Example A.1: Let X be the domain of men {Adam, Bernhard, Charles} and Y the domain of women {Diana, Eva, Francis}, then the relation “*married to*” on $X \times Y$ is :

$$\{(Adam, Eva), (Bernhard, Francis), (Charles, Diana)\}.$$

Consider the domain of natural numbers. Then \leq is a relation on $N \times N$ defined as $\{(m,n) \mid m \leq n\}$. Therefore, this relation contains pairs like, (2,3), and (5,18). But the pair(4,1) is not in this relation .

A ternary relation on N^3 is $\{(p,q,r) \mid p^2 + q^2 = r^2\}$. This relation contains elements like (3,4,5) and (5,12,13).

A.4 Fuzzy Relation

The former section stated that a relation can be considered as a set of tuples. Extending this, a fuzzy relation is a *fuzzy set of tuples*, where each tuple has a membership degree between 0 and 1. Fuzzy Relations are fuzzy sets defined on universal sets which are Cartesian products.

Definition A.12: Let U and V be continuous universes, and $\mu_R: U \times V \rightarrow [0,1]$. Then

$$R = \int_{U \times V} \mu_R(u,v) / (u,v) \quad (\text{A.13})$$

is a **binary relation** on $U \times V$.

If U and V are countable (discrete) universes, then $R = \sum_{U \times V} \mu_R(u,v) / (u,v)$.

Example A.2: Let $U = V = \{1,2,3\}$. Then “approximately equal” is the binary fuzzy relation R

$$\begin{aligned} &1/(1,1) + 1/(2,2) + 1/(3,3) + 0.8/(1,2) + 0.8/(2,3) + 0.8/(2,1) + 0.8/(3,2) \\ &+ 0.3/(1,3) + 0.3/(3,1). \end{aligned} \quad (\text{A.14})$$

The membership function μ_R of this relation can be described by

$$\mu_R(x, y) = \begin{cases} 1 & \text{when } x = y \\ 0.8 & \text{when } |x - y| = 1 \\ 0.3 & \text{when } |x - y| = 2 \end{cases} \quad (\text{A.15})$$

In a matrix notation this can be represented as in Table A.1.

Table A.1: Degree of membership of relation “*R*”

		<i>Y</i>		
		1	2	3
<i>X</i>	1	1	0.8	0.3
	2	0.8	1	0.8
	3	0.3	0.8	1

Example A.3: Consider a fuzzy rule:

if e is PB and \dot{e} is PS, then \dot{u} is NM

where *PB*, *PS* and *NM* are fuzzy sets, defined on the universes of discourse (domains) *E*, ΔE , ΔU , respectively. This rule is often represented by the ternary fuzzy relation (three dimensional relation) *R* given by

$$R = \int_{E \times \Delta E \times \Delta U} \min(\mu_{PB}(e), \mu_{PS}(\dot{e}), \mu_{NM}(\dot{u})) / (e, \dot{e}, \dot{u}) \quad (\text{A.16})$$

Each triple (e, \dot{e}, \dot{u}) has a membership degree equal to the minimum of $\mu_{PB}(e)$, $\mu_{PS}(\dot{e})$ and $\mu_{NM}(\dot{u})$.

A.5 Operations on Fuzzy Relations

Fuzzy relations are very important in fuzzy control because they can describe **interactions** among fuzzy variables. This is quite important in the case of if-then rules.

Let us first define two binary relations R and S (in tables A.2 and A.3), where as:

$R = "x \text{ is considerably larger than } y"$

$S = "y \text{ is very close to } x"$

Table A.2: Degree of membership relation " R "

	y_1	y_2	y_3	y_4
R x_1	0.8	1	0.1	0.7
x_2	0	0.8	0	0
x_3	0.9	1	0.7	0.8

Table A.3: Degree of membership relation " S "

	y_1	y_2	y_3	y_4
S x_1	0.4	0	0.9	0.6
x_2	0.9	0.4	0.5	0.7
x_3	0.3	0	0.8	0.5

Definition A.13: Let R and S be binary relations defined on $X \times Y$. The **intersection** of R and S is defined by :

$$\forall (x, y) \in X \times Y: \mu_{R \cap S}(x, y) = \mu_R(x, y) \ T \ \mu_S(x, y) \quad (\text{A.17})$$

where T is the T -norm operation. Since *minimum* operation is the most common operation in the category of T -norm, the intersection operation may also be defined as:

$$\mu_{R \cap S}(x, y) = \min(\mu_R(x, y), \mu_S(x, y)) \quad (\text{A.18})$$

However, any other operation that satisfies the T -norm conditions, can be used as well.

Definition A.14: Let R and S be binary relations defined on $X \times Y$. The **union** of R and S is defined by :

$$\forall (x, y) \in X \times Y: \mu_{R \cup S}(x, y) = \mu_R(x, y) \ S \ \mu_S(x, y) \quad (\text{A.19})$$

where S is the S -norm operation. Since *maximum* operation is the most common operation in the category of S -norm, the union operation can also be defined as:

$$\mu_{R \cup S}(x, y) = \max(\mu_R(x, y), \mu_S(x, y)) \quad (\text{A.20})$$

However, any other operation that satisfies the S -norm conditions, can be used as well.

Definition A.15: In the binary case, let R be defined on $X \times Y$. The **Projection** of R on Y can be defined as follows:

$$\text{Proj } R \text{ on } Y = \int_Y \sup_x \mu_R(x, y) / y \quad (\text{A.21})$$

Instead of the sup (Supremum) operation which is necessary when X and Y are continuous, it is usual to deal with discrete domain using the maximum operation. In fact the projection operation brings a ternary relation (three dimension) back to a binary relation or a binary relation to a fuzzy set, or a fuzzy set; to a single crisp value.

Example A.4: Consider the relation R as given before (see Table A.4).

Table A.4: Degree of membership “ R ”

	y_1	y_2	y_3	y_4
x_1	0.8	1	0.1	0.7
x_2	0	0.8	0	0
x_3	0.9	1	0.7	0.8

Then the projection on X means that;

x_1 is assigned the highest membership degree from the tuples $(x_1, y_1), (x_1, y_2), (x_1, y_3)$ and (x_1, y_4) ; i.e., 1 is the maximum of the first row.

x_2 is assigned the highest membership degree from the tuples

$(x_2, y_1), (x_2, y_2), (x_2, y_3)$ and (x_2, y_4) ; i.e., 0.8 is the maximum of the second row.

x_3 is assigned the highest membership degree from the tuples

$(x_3, y_1), (x_3, y_2), (x_3, y_3)$ and (x_3, y_4) ; i.e., 1 is the maximum of the third row.

So we can obtain the fuzzy set

$$\text{Proj } R \text{ on } X = 1/x_1 + 0.8/x_2 + 1/x_3 \quad (\text{A.22})$$

In the same way, the projection on Y can be taken by searching for the maximum of the four columns. We have

$$\text{Proj } R \text{ on } Y = 0.9/y_1 + 0.8/y_2 + 0.7/y_3 + 0.8/y_4 \quad (\text{A.23})$$

Definition A.16: In the binary case, let F be a fuzzy set defined on Y . The **cylindrical extension** of F on $X \times Y$ is the set of all tuples $(x, y) \in X \times Y$ with membership degree equal to $\mu_F(y)$:

$$ce(F) = \int_{X \times Y} \mu_F(y) / (x, y) \quad (\text{A.24})$$

In fact, cylindrical extension extends fuzzy sets to fuzzy binary relations; fuzzy binary relations to fuzzy ternary relations; etc. If we consider A as a fuzzy set defined on X , and R as a fuzzy relation defined on $X \times Y$, it is obviously not possible to take the intersection of A and R . But, when A is extended to $X \times Y$, it will make the cylindrical extension possible.

Example A.5: Consider the fuzzy set

$$A = \text{Proj of } R \text{ on } X = 1/x_1 + 0.8/x_2 + 1/x_3 \quad (\text{A.25})$$

Its cylindrical extension on the domain $X \times Y$ is given in Table A.5.

Table A.5: Cylindrical extension of fuzzy set “A”, $ce(A)$

	y_1	y_2	y_3	y_4
x_1	1	1	1	1
x_2	0.8	0.8	0.8	0.8
x_3	1	1	1	1

Consider the fuzzy set

$$B = \text{Proj } R \text{ on } Y = 0.9/y_1 + 0.8/y_2 + 0.7/y_3 + 0.8/y_4 \quad (\text{A.26})$$

Its cylindrical extension on $X \times Y$ is given in Table A.6.

Table A.6: Cylindrical extension of fuzzy set “B”, $ce(B)$

	y_1	y_2	y_3	y_4
x_1	0.9	1	0.7	0.8
x_2	0.9	1	0.7	0.8
x_3	0.9	1	0.7	0.8

Example A.6: Consider the relation R expressing “ x is considerably larger than y ” and suppose it is known that “ x is small” by the fuzzy set:

$$A = 0.3/x_1 + 1/x_2 + 0.8/x_3 \quad (\text{A.27})$$

The combination of the fuzzy relation R and the fuzzy set A , expressed by “ x is considerably larger than y AND x is small” can be given by the intersection of the relation and the extension of A . The extension of A into $X \times Y$ is given in Table A.7.

Table A.7: Cylindrical extension of fuzzy set "A", $ce(A)$

	y_1	y_2	y_3	y_4
x_1	0.3	0.3	0.3	0.3
x_2	1	1	1	1
x_3	0.8	0.8	0.8	0.8

The intersection of R and $ce(A)$ is given in Table A.8.

Table A.8: Intersection of relation "R" and the cylindrical extension of fuzzy set "A",

$R \cap ce(A)$				
	y_1	y_2	y_3	y_4
x_1	0.3	0.3	0.1	0.3
x_2	0	0.8	0	0
x_3	0.8	0.8	0.7	0.8

The combination of fuzzy sets and fuzzy relation with the aid of the cylindrical extension and the projection is called "*Composition*" and it is denoted by " \circ ".

Definition A.17: Let A be a fuzzy set defined on X and R be a fuzzy relation defined on $X \times Y$. Then the **composition** of A and R resulting in a fuzzy set B defined on Y is given by :

$$B = A \circ R = \text{Proj}(ce(A) \cap R) \text{ on } Y \quad (\text{A.28})$$

or, if intersection is performed with the minimum operation and projection with the maximum, then

$$\mu_B(y) = \max_x \min(\mu_A(x), \mu_R(x, y)) \quad (\text{A.29})$$

This is called the *max-min composition*.

A.6 Implication (IF-THEN): $A \rightarrow B$

Consider a fuzzy set A in a universe X and a second fuzzy set B in another universe Y . The **fuzzy implication** $A \rightarrow B$ is a fuzzy relation in the Cartesian product space $X \times Y$. Different methods for determining the implication operation have been proposed in literature. Some examples are given below:

1. Mamdani's method (Logical Product), $A \rightarrow B \equiv A \wedge B$
2. Algebraic Product $A \rightarrow B \equiv A \cdot B$
3. Bounded Product $A \rightarrow B \equiv 0 \vee (A + B - 1)$
4. Drastic Product $A \rightarrow B \equiv \begin{cases} A & \text{as } B = 1 \\ B & \text{as } A = 1 \\ 0 & \text{otherwise} \end{cases}$
5. Zadeh's Method (Lukasiewicz's implication method), $A \rightarrow B \equiv 1 \wedge (1 - A + B)$
6. Boolean Logic Implication $A \rightarrow B \equiv (1 - A) \vee B$
7. Goedel Logic Implication $A \rightarrow B \equiv \begin{cases} 1 & A \leq B \\ B & A > B \end{cases}$
8. Goguen's Implication $A \rightarrow B \equiv \begin{cases} 1 & A \leq B \\ B / A & A > B \end{cases}$

The first four methods use “ \vee ” operation to aggregate the individual conclusions, however the second four methods use “ \wedge ” operation to aggregate the individual conclusions. The first method is the most common one used to express the implication in fuzzy relations.

A.7 The Compositional Rule of Inference

If R is a fuzzy relation from U to V , and A is a fuzzy subset of U , then the fuzzy subset B of V which is induced by A is given by the *composition of R and A* ; that is

$$B = A \circ R \quad (\text{A.30})$$

Therefore, we should take the cylindrical extension of A , take the intersection with R , and project the result onto the Y-axis.

A.8 Modus Ponens (Deduction)

If statement “IF A THEN B ” is true and we know that A is true too, then we can infer that B is true as well.

$$\frac{A \rightarrow B}{A} \quad B \quad (A.31)$$

A.9 Approximate Reasoning

The inferencing technique of fuzzy logic could be considered as a generalization of the classical rule of implication of **modus ponens**. A major difference from the modus ponens based on binary logic is that in the generalized modus ponens, we can use a different fuzzy set for A in premise 1 and A' (NOT A) in premise 2. In binary logic, crisp set A in premise 1 and premise 2 must be completely identical. In other words, in the generalized modus ponens based on fuzzy logic, fuzzy sets A and A' in premise 1 and premise 2 do not have to be precisely the same. We can still infer the conclusion that B' is true. For this reason, fuzzy reasoning is called “**approximate reasoning**”.

Appendix B

Definition.B1. A family Q of complex functions defined on a set E is said to be an **algebra** if :

- (i) $f + g \in Q$
- (ii) $fg \in Q$
- (iii) $cf \in Q$

for all $f \in Q$, $g \in Q$ and for all complex constants c ; that is, if Q is closed under addition, multiplication, and scalar multiplication.

Definition.B2. For each positive integer k , let R^k be the set of all ordered k -tuples $\mathbf{x} = (x_1, x_2, \dots, x_k)$,

where x_1, \dots, x_k are real numbers, called the *coordinates* of \mathbf{x} . The elements of R^k are called points, or vectors, especially when $k > 1$. We denote vectors by boldfaced letters. If $\mathbf{y} = (y_1, \dots, y_k)$ and if α is a real number, put

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, \dots, x_k + y_k)$$

$$\alpha \mathbf{x} = (\alpha x_1, \dots, \alpha x_k)$$

so that $\mathbf{x} + \mathbf{y} \in R^k$ and $\alpha \mathbf{x} \in R^k$. This defines the addition of vector, as well as multiplication of a vector by a real number (a scalar). These two operations satisfy the commutative, associative, and distributive laws and make R^k into a vector space over the real field. The “inner product” (or scalar product) of \mathbf{x} and \mathbf{y} is given by

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^k x_i y_i$$

and the *norm* of \mathbf{x} by

$$|\mathbf{x}| = (\mathbf{x} \cdot \mathbf{x})^{1/2} = \left(\sum_{i=1}^k x_i^2 \right)^{1/2}.$$

The structure defined in this manner (the vector space R^k with the above inner product and norm) is called **Euclidean** k -space.

Definition.B3. Let X be a metric space and E is a subset of X .

E is **closed** if every limit point of E is a point of E .

E is **open** if every point of E is an interior point of E .

Definition B4. A set X , whose elements we shall call *points*, is said to be a **metric space** if with any two points p and q of X there is associated a real number $d(p,q)$ called the distance from p to q , such that

- (a) $d(p,q) > 0$ if $p \neq q$; $d(p,p) = 0$;
- (b) $d(p,q) = d(q,p)$;
- (c) $d(p,q) \leq d(p,r) + d(r,q)$, for any $r \in X$.

Definition B5. of a set E in a metric space X we mean a collection $\{ G_\alpha \}$ of open subsets of X such that $E \subset \bigcup_\alpha G_\alpha$.

Definition B6. A subset K of a metric space X is said to be **compact** if every open cover of K contains a *finite* subcover.