

raf

TJ213 .W66 2009.



0000065894

DC motor control using fuzzy logic with PC based
implementation / Wong Hien Lang.



UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FACULTY OF ELECTRICAL ENGINEERING

PROJEK SARJANA MUDA
(PSM 2)
BEKU 4983

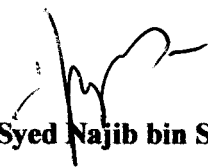
DC MOTOR CONTROL USING FUZZY LOGIC
WITH PC BASED IMPLEMENTATION

WONG HIEN LANG

BEKM

May 2009

**“ I hereby declare that I have read through this report entitle “title of the project”
and found that it has comply the partial fulfillment for awarding the degree of
Bachelor of Electrical Engineering (Mechatronics)”**

Signature : 
Supervisor's Name : **En. Syed Najib bin Syed Salim**
Date : **13 May 2009**

**DC MOTOR CONTROL USING FUZZY LOGIC
WITH PC BASED IMPLEMENTATION**


WONG HIEN LANG

**A report submitted in partial fulfillment of the requirements for the degree
of Bachelor of Electrical Engineering (Mechatronics)**

**Faculty of Electrical Engineering
UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

MAY 2009

I declared that this report entitle “DC Motor Control Using Fuzzy Logic With PC Based Implementation” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : 
Name : WONG HIEN LANG
Date : 13 May 2009

To my beloved father and mother

To my sisters and brothers

To my friends and lecturers

ACKNOWLEDGEMENT

I would like say thank to my supervisor, En. Syed Najib bin Syed Salim for his guidance for the PSM project. Besides this, I would like thank to my entire friend especially Eric Chee and Tang Wee Meng who help me during the project by giving some advice to me.

Besides this, I would like to thank to my parents for moral supporting me. I would say thank you to my mom and dad.

ABSTRACT

This project is to develop a fuzzy logic control system for DC motor. The specified task for the fuzzy logic control system is to control the position of the DC motor. Besides this, a real time fuzzy logic control system will be built to control the speed and position of the DC motor. Matlab simulink is used for DC motor model to get the simulation results. The PIC16F877A microcontroller is used to communicate to computer through serial port RS-232 and Max232 dual driver receiver. Thus, PIC16F877A is used as communication tools for sending and receiving the data from computer. Max232 dual driver receiver is used as a converter signal voltage between the PIC16F877A and serial port RS-232. Flip-flop is used as a decoder for decode the encoder readings from the DC motor. Proteus is used to draw the schematic circuit and printed circuit board layout for the circuit. Thus, the schematic circuit is fabricated. There is a comparison from the simulation results for both single loop control and cascade control between the 3x3, 5x5 and 7x7 membership functions. Microsoft Visual C# is used as the graphical user interface (GUI) communicates with PIC16F877A microcontroller with computer to control the DC motor. Recommendation is included for overcome the schematic circuit current leakage problem.

ABSTRAK

Projek ini adalah untuk membina satu sistem kawalan logik kabur untuk DC motor. Tugas yang dispeksifikasikan untuk sistem kawalan logik kabur adalah untuk mengawal kedudukan DC motor. Selain itu, satu masa nyata sistem kawalan logik kabur akan dibina untuk mengawal kedudukan DC motor. Matlab simulink digunakan terhadap DC motor model untuk mendapat hasil simulasi. PIC16F877A mikropengawal akan digunakan untuk berkomunikasi dengan komputer melalui RS-232 dan Max232 penerima dua hala. Oleh itu, PIC16F877A digunakan sebagai alat perhubungan untuk penghantaran dan penerimaan data daripada komputer. Max232 penerima dua hala digunakan sebagai satu penukar voltan isyarat antara PIC16F877A dan RS-232. Flip-flop digunakan sebagai satu penyahkod untuk menyahkod bacaan pengekod daripada DC motor. "Proteus" digunakan untuk melukis gambar rajah skematik dan "PCB layout" untuk litar. Dengan itu, skematik litar dihasilkan. Terdapat perbandingan di antara 3x3, 5x5 dan 7x7 "membership function" bagi "single loop" dan "cascade loop" daripada hasil simulasi. Microsoft Visual C# digunakan bagi membina antara muka pengguna grafik (GUI) untuk berhubung dengan mikropengawal PIC16F877A dengan computer untuk mengawal DC motor. Cadangan dimasukkan untuk megatasi masalah litar skematik kebocoran arus.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	SUPERVISOR'S VERIFICATION	i
	PROJECT TITLE	ii
	AUTHENTICATION	iii
	DEDICATION	iv
	ACKNOWLEDGEMENT	v
	ABSTRACT	vi
	TABLE OF CONTENTS	viii
	LIST OF TABLES	x
	LIST OF FIGURES	xi
	LIST OF ABBREVIATIONS	xv
	LIST OF APPENDIXES	xvi
1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statement	2
	1.3 Project Objectives	2
	1.4 Project Scope	2
	1.5 Flow Chart	3
2	LITERATURE REVIEWS	4
	2.1 Introduction	4
	2.2 Concept and Theory	5
	2.3 Process for Apply Fuzzy Inference into the system	5
	2.4 Summary the paper about Fuzzy Logic Control of an Industrial Indexing Motion Application	7

2.5	Summary the paper about Performance Comparison of PID based Position Control System with Fuzzy Logic Controller based Position Control System	8
3	METHODOLOGY	10
3.1	Methodology Flow Chart	10
3.2	DC Motor Modeling	11
3.3	H-Bridge Driver LMD18201T	13
3.4	PIC16F877A Microcontroller	14
3.5	Max232 Dual Driver Receiver	14
3.6	Serial Port RS-232	14
3.7	D-type Positive-Edge Triggered Flip-Flop	16
3.8	Microsoft Visual C# 2008 Express Editions	16
3.9	Fuzzy logic controller Mamdani-type	17
4	RESULTS AND DISCUSSIONS	18
4.1	DC Motor Single Loop Control Simulation Results	18
4.1.1	3x3 DC Motor Single Loop Control	19
4.1.2	5x5 DC Motor Single Loop Control	22
4.1.3	7x7 DC Motor Single Loop Control	25
4.1.4	Comparison Performance of the DC Motor Single Loop Control	28
4.2	DC Motor Cascade Control Simulation Results	29
4.2.1	3x3 DC Motor Cascade Control	30
4.2.2	5x5 DC Motor Cascade Control	34
4.2.3	7x7 DC Motor Cascade Control	38
4.2.4	Comparison Performance of the DC Motor Cascade Control	42
5	CONCLUSIONS AND RECOMMENDATIONS	43
5.1	Conclusions	43
5.2	Recommendations	44
	REFERENCES	45
	APPENDIXES	46

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	The time response for the general PID controller and fuzzy logic controller	9
3.1	The typical performance of the DC motor	12
3.2	The logic table shows the Brake, PWM and Direction pins are variables for control the speed and direction of the DC motor	13
3.3	Function of the serial port RS-232 pins	15
3.4	The logic diagram of a D-type positive-edge-triggered flip-flop	16
4.1	The comparison of the 3x3, 5x5 and 7x7 for the DC motor single loop control	28
4.2	The comparison of the 3x3, 5x5 and 7x7 for the DC motor cascade control	42

LIST OF FIGURES

FIGURE	TITLE	PAGE
1.1	Flow Chart	3
2.1	Process apply for the Mamdani-type fuzzy inference system	6
2.2	Process apply for the Sugeno-type fuzzy inference system	7
2.3	Fuzzy Logic Controller Control Configuration	7
2.4	Fuzzy Logic Controller	8
3.1	Methodology flow chart	10
3.2	The DC motor modeling in Matlab simulink	12
3.3	The pin of the H-Bridge driver LMD18201T	13
3.4	Serial Port RS-232	15
3.5	6 nand gate inside flip-flop chip	16
4.1	The DC motor single loop control with fuzzy logic controller	18
4.2	The 3 membership function for the difference of the desired angle and the output angle is “error”	19
4.3	The 3 membership function for the input speed is “delta-error”	19
4.4	The 3 membership function for the output voltage is “voltage”	19
4.5	The graph current (A) versus time (ms) for 3x3 DC motor single loop control	20
4.6	The graph speed (rps) versus time (ms) for 3x3 DC motor single loop control	20
4.7	The response graph of the output angle (degree) versus time (ms) for 3x3 DC motor single loop control	21
4.8	The 5 membership function for the difference of the desired angle and the output angle is “error”	22
4.9	The 5 membership function for the input speed is “delta-error”	22
4.10	The 5 membership function for the output voltage is “voltage”	22
4.11	The graph current (A) versus time (ms) for 5x5 DC motor single loop control	23

4.12	The graph speed (rps) versus time (ms) for 5x5 DC motor single loop control	23
4.13	The response graph of the output angle (degree) versus time (ms) for 5x5 DC motor single loop control	24
4.14	The 7 membership function for the difference of the desired angle and the output angle is “error”	25
4.15	The 7 membership function for the input speed is “delta-error”	25
4.16	The 7 membership function for the output voltage is “voltage”	25
4.17	The graph current (A) versus time (ms) for 7x7 DC motor single loop control	26
4.18	The graph speed (rps) versus time (ms) for 5x5 DC motor single loop control	26
4.19	The response graph of the output angle (degree) versus time (ms) for 5x5 DC motor single loop control	27
4.20	The DC motor cascade control with fuzzy position and speed controller	29
4.21	The 3 membership function for the difference of the desired angle and the output angle is “error” for the position controller	30
4.22	The 3 membership function for the difference of the current error and the previous error is “delta-error” for the position controller	30
4.23	The 3 membership function for the output speed is “speed” for position controller	30
4.24	The 3 membership function for the difference of the desired speed and the output speed is “error” for speed controller	31
4.25	The 3 membership function for the difference of the current error and the previous error is “delta-error” for speed controller	31
4.26	The 3 membership function for the output voltage is “voltage” for speed controller	31
4.27	The graph current (A) versus time (ms) for 3x3 DC motor cascade control	32
4.28	The graph speed (rps) versus time (ms) for 5x5 DC motor cascade control	32
4.29	The response graph of the output angle (degree) versus time (ms) for 3x3 DC motor cascade control	33

4.30	The 5 membership function for the difference of the desired angle and the output angle is “error” for the position controller	34
4.31	The 5 membership function for the difference of the current error and the previous error is “delta-error” for the position controller	34
4.32	The 5 membership function for the output speed is “speed” for position controller	34
4.33	The 5 membership function for the difference of the desired speed and the output speed is “error” for speed controller	35
4.34	The 5 membership function for the difference of the current error and the previous error is “delta-error” for speed controller	35
4.35	The 5 membership function for the output voltage is “voltage” for speed controller	35
4.36	The graph current (A) versus time (ms) for 5x5 DC motor cascade control	36
4.37	The graph speed (rps) versus time (ms) for 5x5 DC motor cascade control	36
4.38	The response graph of the output angle (degree) versus time (ms) for 5x5 DC motor cascade control	37
4.39	The 7 membership function for the difference of the desired angle and the output angle is “error” for the position controller	38
4.40	The 7 membership function for the difference of the current error and the previous error is “delta-error” for the position controller	38
4.41	The 7 membership function for the output speed is “speed” for position controller	38
4.42	The 7 membership function for the difference of the desired speed and the output speed is “error” for speed controller	39
4.43	The 7 membership function for the difference of the current error and the previous error is “delta-error” for speed controller	39
4.44	The 7 membership function for the output voltage is “voltage” for speed controller	39
4.45	The graph current (A) versus time (ms) for 7x7 DC motor cascade control	40
4.46	The graph speed (rps) versus time (ms) for 7x7 DC motor cascade control	40

4.47	The response graph of the output angle (degree) versus time (ms) for 3x3 DC motor cascade control	41
------	--	----

LIST OF SYMBOLS

V_f	-	Field voltage
R_f	-	Field resistance
i_f	-	Field current
L_f	-	Field inductance
V_a	-	Armature voltage
i_a	-	Armature current
L_a	-	Armature inductance
B	-	Friction
ω	-	Angular speed
J	-	Inertia
T_L	-	Torque of the load
T_e	-	Torque produced by motor
E	-	Back electromotive force (emf)
k_e	-	Back emf constant
k_t	-	Torque constant
ψ	-	Field Flux

LIST OF APPENDIXES

APPENDIX	TITLE	PAGE
A	Schematic design for the H-Bridge driver LMD18201T and PIC16F877A	46
B	Printed circuit board design layout for the schematic diagram	47
C	Printed circuit board with component mounted	48
D	C# program for DC motor single loop control using fuzzy logic	49

CHAPTER 1

INTRODUCTION

1.1 Background

Conventionally, proportional-integral-derivative controller (PID controller) is widely used in industrial control systems. However, PID controller is based on mathematical models in which the control system is described using one or more differential equations that define the system response to its inputs. They are the products of decades of development and theoretical analysis, and are highly effective. In some cases, the mathematical model of the control process for PID controller may not exist or may be too complex.

So, fuzzy logic incorporates a simple, rule-based IF X AND Y THEN Z approach to a solving control problem rather than attempting to model a system mathematically. Thus, fuzzy logic was conceived as a better method for sorting and handling data but has proven to be an excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. This is because it uses an imprecise but very descriptive language to deal with input data more like a human operator. Furthermore, fuzzy logic is well suited to low-cost implementations based on cheap sensors and low-resolution analog-to-digital converters. In many cases, fuzzy control can be used to improve existing traditional controller systems by adding an extra layer of intelligence to the current control method.

Fuzzy controllers are very simple conceptually. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and so on, to the appropriate membership functions and truth values. The processing stage involves each appropriate rule and generates a result for each,

then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value.

1.2 Problem Statement

Conventionally, the brushed DC motor is controlled by using the PID controller. Due to the complexity of the mathematical model of PID system, the fuzzy logic controller offers a simple rule-based to solve the control system. Besides this, fuzzy logic controller is ease to design if a system required a lot of inputs. The fuzzy logic controller also eases for tuning if the input value is changed.

1.3 Project Objectives

The objectives of this project are as follows:

- a) To simulate the DC motor model with fuzzy logic controller using Matlab simulink
- b) To develop a Graphical User Interface (GUI) for control the speed and position of the motor
- c) To integrate the DC motor with the GUI for control the speed and position with computer

1.4 Project Scope

The focus of this project is on the software programming and graphical user interface with the Microsoft Visual C# 2008 Express Editions. Besides this, the Matlab simulink is used for DC motor model simulation. The driver for DC motor is LMD 18201T which is a full bridge driver. The PIC 16F877A microcontroller is used to transmit and receive the data to the computer. The computer is used to control the position of the motor. The flip-flop is used as a decoder for the motor. Besides this, Max232 and serial port RS-232 are used to communicate the computer and the PIC16F877A. Figure 1 shows the process flow of the project.

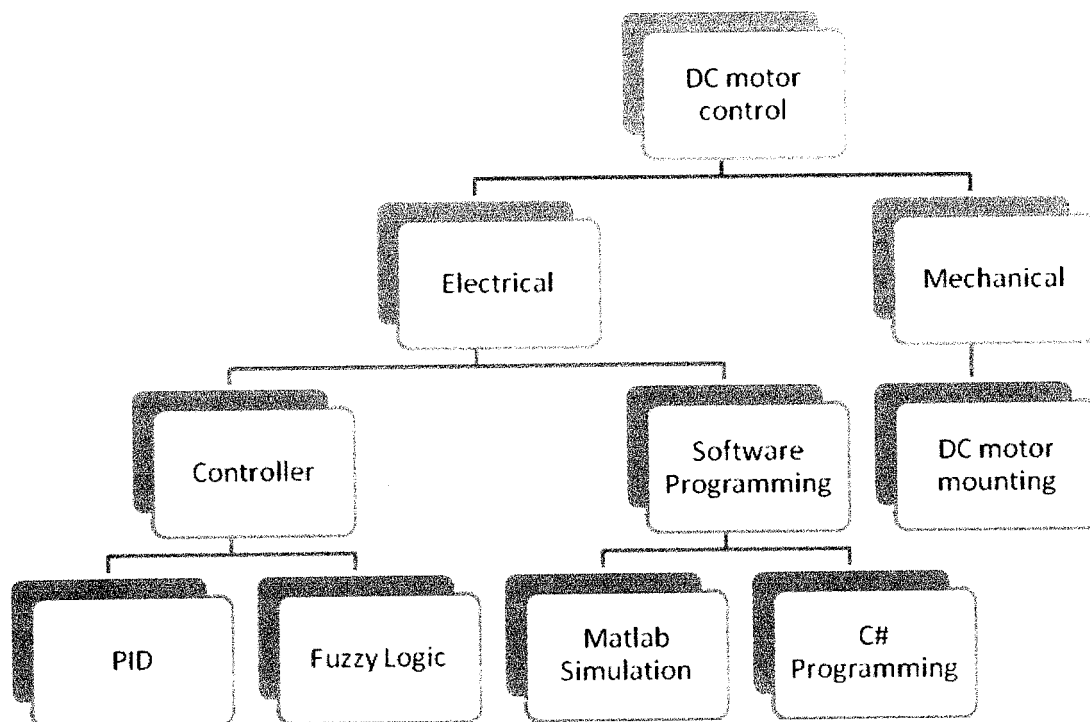


Figure 1: Flow chart

1.5 Outline

The chapter 1 is introduction which is about the project objectives and scope. The chapter 2 is literature reviews which are about the theory and summary research genre from paper. Chapter 3 is methodology which is about the DC motor modeling and component chosen for the schematic circuit. Chapter 4 is results and discussions about the simulation result for DC motor single loop control and cascade control. Chapter 5 is about the conclusions and recommendation solution for the project.

CHAPTER 2

LITERATURE REVIEWS

2.1 Introduction

Fuzzy logic was first proposed by Lotfi A. Zadeh of the University of California at Berkeley in a 1965 paper. He elaborated on his ideas in a 1973 paper that introduced the concept of "linguistic variables", which in this article equates to a variable defined as a fuzzy set. Other research followed, with the first industrial application, a cement kiln built in Denmark, coming on line in 1975.

Fuzzy systems were largely ignored in the U.S. because they were associated with artificial intelligence (AI), a field that periodically oversells itself, especially in the mid-1980s, resulting in a lack of credibility within the commercial domain. The Japanese did not have this prejudice. Interest in fuzzy systems was sparked by Seiji Yasunobu and Soji Miyamoto of Hitachi, who in 1985 provided simulations that demonstrated the superiority of fuzzy control systems for the Sendai railway. Their ideas were adopted, and fuzzy systems were used to control accelerating, braking, and stopping when the line opened in 1987.

Another event in 1987 helped promote interest in fuzzy systems. During an international meeting of fuzzy researchers in Tokyo that year, Takeshi Yamakawa demonstrated the use of fuzzy control, through a set of simple dedicated fuzzy logic chips, in an "inverted pendulum" experiment. This is a classic control problem, in which a vehicle tries to keep a pole mounted on its top by a hinge upright by moving back and forth.

Observers were impressed with this demonstration, as well as later experiments by Yamakawa in which he mounted a wine glass containing water or even a live mouse to the top of the pendulum. The system maintained stability in both cases. Yamakawa eventually

went on to organize his own fuzzy-systems research lab to help exploit his patents in the field.

2.2 Concept and Theory

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves: membership functions, logical operations, and if-then rules. There are two types of fuzzy inference systems which is Mamdani-type and Sugeno-type.

These two types of inference systems vary somewhat in the way outputs are determined. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant. There are 5 steps for fuzzy inference process: fuzzification of the input variables, application of the fuzzy operator (AND or OR) in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules, and defuzzification.

2.3 Process for Apply the Fuzzy Inference into the system

Firstly, the inputs and the degree to which they belong to each of the appropriate fuzzy sets via membership functions are determined. Input is always a crisp numerical value limited to the universe of discourse of the input variable. Output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1).

Secondly, input to the fuzzy operator is two or more membership values from fuzzified input variables. Then, two AND methods: min (minimum) and prod (product) or two OR methods: max (maximum) and the probabilistic OR method probor are applied for the inputs. Finally, the output is a single truth value.

Thirdly, the input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Then, two AND method used: min (minimum) which truncates the output fuzzy set and prod (product) which scales the output fuzzy set.

After that, process the fuzzy sets which represent the outputs of each rule are combined into a single fuzzy set. Then, aggregation of the input is the list of truncated output functions returned by the implication process for each rule. Then, the output of the aggregation process is one fuzzy set for each output variable. There are three method used: max (maximum), probor (probabilistic OR) and sum (simply the sum of each rule's output set).

Finally, the input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number.

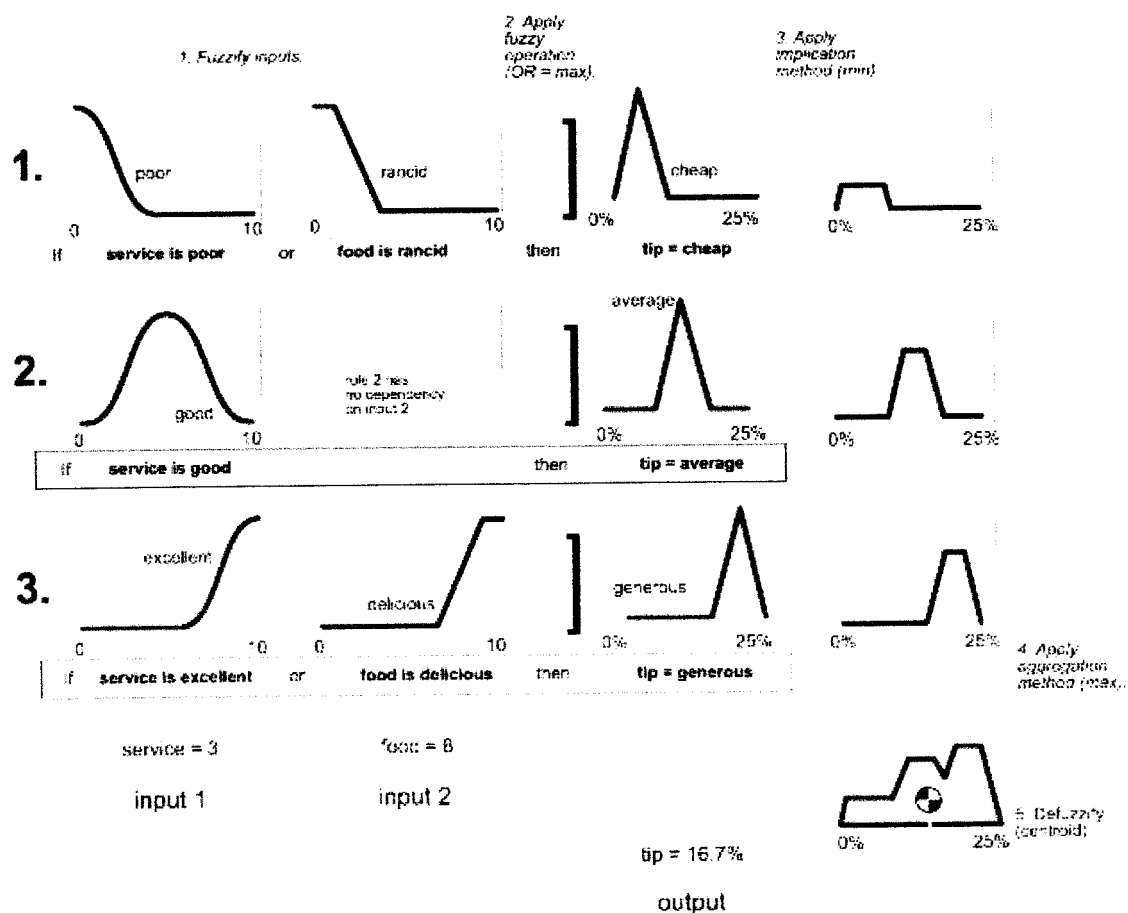


Figure 2.1: Process apply for the Mamdani-type fuzzy inference system

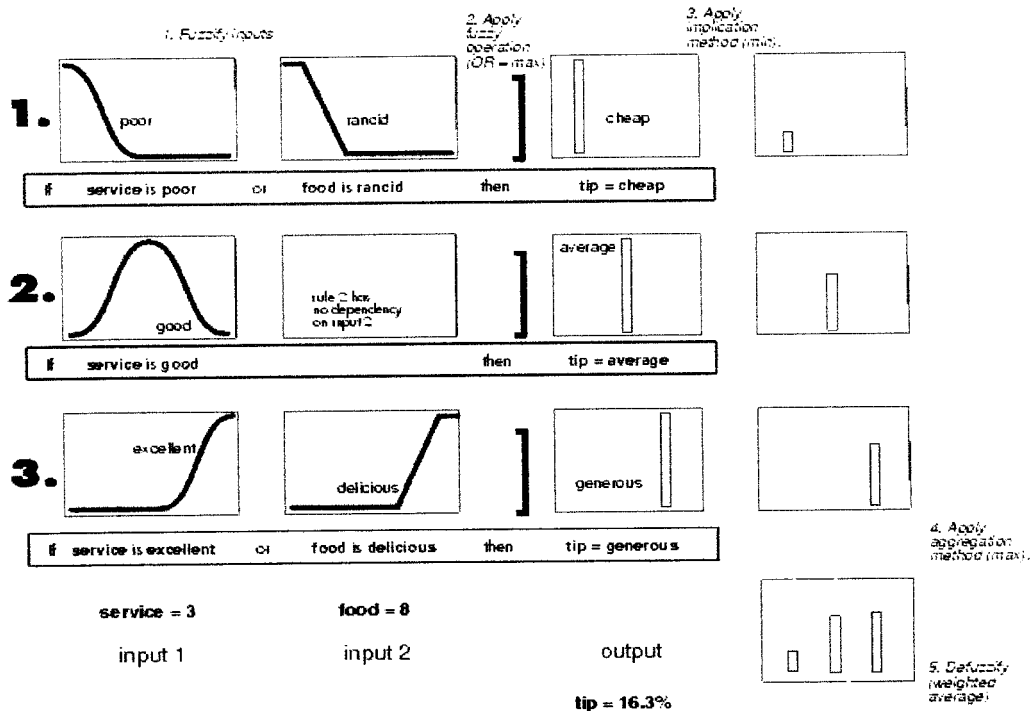


Figure 2.2: Process apply for the Sugeno-type fuzzy inference system

2.4 Summary the paper with the title “Fuzzy Logic Control of an Industrial Indexing Motion Application”

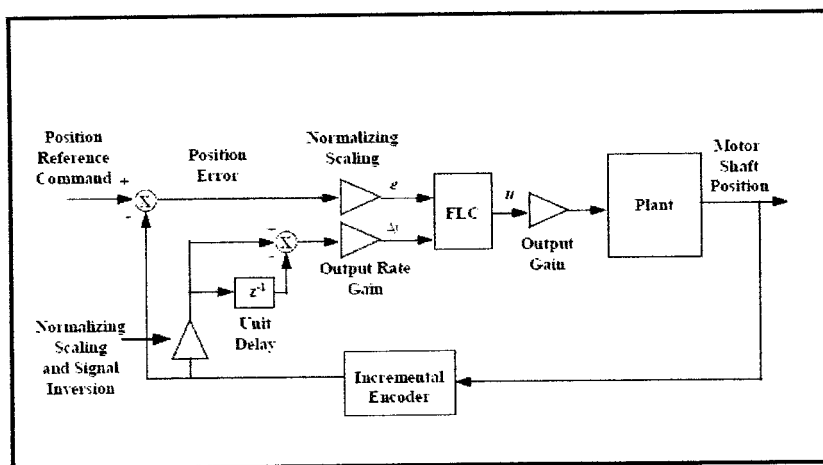


Figure 2.3: Fuzzy Logic Controller Control Configuration

Most PID controllers used in industry are tuned by a trial and error method. This is because the nonlinearities include friction, saturation, backlash and etc affects the system to be controlled. The paper is about the comparison on performance between the fuzzy